Springer Series in Supply Chain Management

Xi Chen
Stefanus Jasin
Cong Shi *Editors*

# The Elements of Joint Learning and Optimization in Operations Management

Springer

# Springer Series in Supply Chain Management

Volume 18

**Series Editor**
Christopher S. Tang, University of California, Los Angeles, CA, USA

Supply Chain Management (SCM), long an integral part of Operations Management, focuses on all elements of creating a product or service, and delivering that product or service, at the optimal cost and within an optimal timeframe. It spans the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption. To facilitate physical flows in a time-efficient and cost-effective manner, the scope of SCM includes technology-enabled information flows and financial flows.

The Springer Series in Supply Chain Management, under the guidance of founding Series Editor Christopher S. Tang, covers research of either theoretical or empirical nature, in both authored and edited volumes from leading scholars and practitioners in the field – with a specific focus on topics within the scope of SCM.

This series has been accepted by Scopus.

Springer and the Series Editor welcome book ideas from authors. Potential authors who wish to submit a book proposal should contact Ms. Jialin Yan, Associate Editor, Springer (Germany), e-mail: jialin.yan@springernature.com

Xi Chen • Stefanus Jasin • Cong Shi
Editors

# The Elements of Joint Learning and Optimization in Operations Management

Springer

*Editors*
Xi Chen
New York University
New York, NY, USA

Stefanus Jasin
University of Michigan–Ann Arbor
Ann Arbor, MI, USA

Cong Shi
University of Michigan–Ann Arbor
Ann Arbor, MI, USA

ISSN 2365-6395 ISSN 2365-6409 (electronic)
Springer Series in Supply Chain Management
ISBN 978-3-031-01925-8 ISBN 978-3-031-01926-5 (eBook)
https://doi.org/10.1007/978-3-031-01926-5



This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To our parents:*

*Jianming Chen and Xiaohong Yu*
*Andi Wirawan Jasin and Sandra Widjaja*
*Xiping Shi and Qiong Yao*

*and to our families:*

*Yingze Wang and Andrew Chen*
*Yan Huang*
*Miao Ni and Janie Shi and Anna Shi*

# Preface

The last decade has seen an explosion of research at the intersection of operations research and machine learning. While the classical operations research has focused largely on optimizing the system under the assumption of known dynamics and known parameters, in reality, the "known" are typically unknown and need to be estimated from the continuously generated data. The later gives rise to the problem of joint learning and optimization, which is one of the core research topics in the machine learning community. However, while the machine learning community has largely focused on solving problems that are directly relevant for computer science applications, the operations research community has its own long list of problems that are not typically considered in the context of joint learning and optimization. This presents a wonderful opportunity for combining operations research and machine learning techniques to solve some of the most fundamental analytic problems.

This book consists of 15 chapters written by some of the world's leading experts on the subject, covering a wide range of topics such as price optimization, assortment optimization, inventory optimization, and healthcare operations. As noted above, the field has grown very quickly within the last decade, and it is not our intention to provide a comprehensive overview of the field. Rather, we have a more modest aim to introduce interested readers to some fundamental results that have been developed in the field within the last decade. This book is a suitable reading for graduate students (either PhD or advanced master's) in operations research and/or machine learning. It is also suitable for researchers in other fields who are interested in the topic of joint learning and optimization.

For a better organization, we cluster the 15 chapters into five different parts:

**Part I. Generic Tools** The first part of the book consists of Chaps. 1–3 and covers standard tools and concepts that are commonly used in the learning literature. Many of the topics discussed in this part are also covered in more details in other more specialized books. Our objective here is to quickly introduce readers to some of the key tools and concepts. Chapter 1 discusses fundamental algorithms for multi-armed bandit; Chap. 2 discusses fundamental algorithms for reinforcement learning;

and Chap. 3 discusses optimal learning from the perspective of statistical design of experiments.

**Part II. Price Optimization** The second part of the book consists of Chaps. 4–7 and covers a variety of topics on joint learning and price optimization. Chapter 4 discusses state-of-the-art parametric and non-parametric learning algorithms for single-product and multiple-product settings; Chap. 5 discusses learning algorithms in the presence of inventory constraints; Chap. 6 provides literature review on joint learning and pricing in non-stationary environments; and Chap. 7 discusses learning algorithms for high dimensional setting.

**Part III. Assortment Optimization** The third part of the book consists of Chaps. 8–10 and covers a variety of topics on joint learning and assortment optimization. Chapter 8 discusses recent advances in non-parametric estimation of choice models; Chap. 9 discusses learning algorithms for assortment optimization under the popular multinomial logit (MNL) choice model; and Chap. 10 discusses learning algorithms for assortment optimization under non-MNL choice model.

**Part IV. Inventory Optimization** The fourth part of the book consists of Chaps. 11–13 and covers a variety of topics on joint learning and inventory optimization. Chapter 11 discusses state-of-the-art algorithms on inventory optimization with censored demand; Chap. 12 discusses learning algorithms for the joint inventory and price optimization problem where both the price and inventory decisions need to be simultaneously optimized; and Chap. 13 discusses optimization in the "small data, large scale" regime.

**Part V. Healthcare Operations** The fifth part of the book consists of Chaps. – 15 and covers topics related to healthcare operations. Chapter  discusses bandit algorithms/procedures for clinical trials and Chap. 15 provides an in-depth overview of dynamic treatment regime.

This book would not have been possible without the excellent contribution of all authors and the help of the team at Springer, for which we are forever grateful.

New York, NY, USA — Xi Chen
Ann Arbor, MI, USA — Stefanus Jasin
Ann Arbor, MI, USA — Cong Shi

# Contents

**Part I Generic Tools**

**1 The Stochastic Multi-Armed Bandit Problem** ..... 3
Shipra Agrawal

**2 Reinforcement Learning** ..... 15
Zheng Wen

**3 Optimal Learning and Optimal Design** ..... 49
Ilya O. Ryzhov

**Part II Price Optimization**

**4 Dynamic Pricing with Demand Learning: Emerging Topics and State of the Art** ..... 79
Arnoud V. den Boer and Nuri Bora Keskin

**5 Learning and Pricing with Inventory Constraints** ..... 103
Qi (George) Chen, He Wang, and Zizhuo Wang

**6 Dynamic Pricing and Demand Learning in Nonstationary Environments** ..... 137
Arnoud V. den Boer and Nuri Bora Keskin

**7 Pricing with High-Dimensional Data** ..... 151
Gah-Yi Ban

**Part III Assortment Optimization**

**8 Nonparametric Estimation of Choice Models** ..... 177
Srikanth Jagabathula and Ashwin Venkataraman

**9 The MNL-Bandit Problem** ..... 211
Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi

**10 Dynamic Assortment Optimization: Beyond MNL Model** ..... 241
Yining Wang and Yuan Zhou

**Part IV Inventory Optimization**

**11 Inventory Control with Censored Demand** ..... 273
Xiangyu Gao and Huanan Zhang

**12 Joint Pricing and Inventory Control with Demand Learning** ..... 305
Boxiao Chen

**13 Optimization in the Small-Data, Large-Scale Regime** ..... 337
Vishal Gupta

**Part V Healthcare Operations**

**14 Bandit Procedures for Designing Patient-Centric Clinical Trials** ..... 365
Sofia S. Villar and Peter Jacko

**15 Dynamic Treatment Regimes for Optimizing Healthcare** ..... 391
Nina Deliu and Bibhas Chakraborty

# Editors and Contributors

## About the Editors

**Xi Chen** is a professor in the Department of Technology, Operations, and Statistics at Stern School of Business, New York University. He is also Professor of Computer Science at the Center for Data Science at New York University. His research and teaching have been recognized by numerous awards, including The World's Best 40 under 40 MBA Professors by Poets & Quants, NSF CAREER Award, Forbes 30 Under 30, the Inaugural International Chinese Statistical Association Outstanding Young Researcher Award, and Faculty Research Awards, and by a number of leading technology and financial giants, such as Google, Facebook, Adobe, JPMorgan, and Bloomberg. In addition, he is an elected member of the International Statistical Institute (ISI) and an associate editor of Management Science, Operations Research, and Annals of Statistics.

**Stefanus Jasin** is a professor in the Department of Technology and Operations at the Ross School of Business, University of Michigan, Ann Arbor. His research focuses on algorithmic and/or prescriptive business analytics and has been recognized by numerous awards, including INFORMS Revenue Management and Pricing Section Prize Award, and INFORMS eBusiness Section Best Paper Award. He is a department editor of Production and Operations Management. In addition, he is also an associate editor of Management Science, Operations Research, Manufacturing and Service Operations Management, Production and Operations Management, and Naval Research Logistics.

**Cong Shi** is a professor in the Department of Industrial and Operations Engineering at the University of Michigan at Ann Arbor. His research and teaching have been recognized by numerous awards, including INFORMS George Nicholson Paper Competition, INFORMS JFIG Paper Competition, Amazon Research Award, UM

IOE Professor of the Year, and UM CoE Vulcans Education Excellence Award. He is an associate editor of Management Science, Production and Operations Management, IISE Transactions, and Operations Research Letters.

## Contributors

**Shipra Agrawal** Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA

**Vashist Avadhanula** Facebook, Menlo Park, CA, USA

**Gah-Yi Ban** Department of Decision, Operations & Information Technologies, Robert H. Smith Business School, University of Maryland, College Park, MD, USA

**Bibhas Chakraborty** Center for Quantitative Medicine, Duke-NUS Medical School, National University of Singapore, Singapore, Singapore

**Boxiao Chen** College of Business Administration, University of Illinois Chicago, Chicago, IL, USA

**Qi (George) Chen** Department of Management Science and Operations, London Business School, London, UK

**Nina Deliu** MRC Biostatistics Unit, School of Clinical Medicine, University of Cambridge, Cambridge, UK

**Arnoud V. den Boer** Department of Mathematics, University of Amsterdam, Amsterdam, GE, Netherlands

**Xiangyu Gao** Department of Decision Sciences and Managerial Economics, The Chinese University of Hong Kong, Hong Kong, China

**Vineet Goyal** Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA

**Vishal Gupta** Department of Data Science and Operations, Marshall School of Business, University of Souther California, Los Angeles, CA, USA

**Peter Jacko** Department of Management Science, Lancaster University, Lancaster, UK

Berry Consultants, Abingdon, UK

**Srikanth Jagabathula** Department of Information, Operations & Management Sciences, Leonard N. Stern School of Business, New York University, New York, NY, USA

**Nuri Bora Keskin** Department of Operations Management, Fuqua School of Business, Duke University, Durham, NC, USA

**Ilya O. Ryzhov** Department of Decision, Operations, and Information Technologies, Robert H. Smith School of Business, University of Maryland, College Park, MD, USA

**Ashwin Venkataraman** Department of Operations Management, Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA

**Sofia S. Villar** MRC Biostatistics Unit, School of Clinical Medicine, University of Cambridge, Cambridge, UK

**He Wang** H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

**Yining Wang** Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA

**Zizhuo Wang** School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

**Zheng Wen** Google DeepMind, Mountain View, CA, USA

**Assaf Zeevi** Department of Decision, Risk, and Operations, Columbia Business School, Columbia University, New York, NY, USA

**Huanan Zhang** Department of Strategy, Entrepreneurship, and Operations, Leeds School of Business, University of Colorado Boulder, Boulder, CO, USA

**Yuan Zhou** Mathematical Sciences Center, Tsinghua University, Beijing, China

# Part I
# Generic Tools

# Chapter 1
# The Stochastic Multi-Armed Bandit Problem


**Shipra Agrawal**

## 1.1 Introduction

Consider a decision maker picking one out of $N$ available options repeatedly over sequential rounds. The reward of each option is uncertain, and the (stochastic) reward model is a priori unknown. Given the sequential nature of this problem, the decision maker could benefit from using the observed rewards from the previous rounds to learn the reward models and use those model predictions to improve the decisions over time. However, in doing so, the decision maker faces a tradeoff between learning and optimization: that is, whether to pick one of the less-explored options in order to improve their reward predictions which could benefit future decisions or exploit the option that is currently predicted to have the maximum reward. This tradeoff, referred to as the exploration-exploitation tradeoff, lies at the heart of the Multi-Armed Bandit (MAB) problem (e.g., Agrawal, 2019).

The basic formulation of the stochastic MAB problem considers the setting where in every round the decision maker must pick a *single* option out of $N$ discrete options, referred to as the $N$ arms. The rewards for each arm are independent across time and are generated from an (a priori unknown) stationary distribution. Importantly, observing the reward from one arm reveals no information about the reward distribution of other arm(s). The goal is to maximize total reward over $T$ sequential rounds. More general versions of the stochastic MAB problem relax several of these restrictions and allow for applications that are beyond the purview of the classic N-armed bandit setting. This includes continuous space of arms with parametric reward models (linear bandits), non-stationary and context-dependent

S. Agrawal (✉)
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA
e-mail: sa3305@columbia.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_1

reward models (contextual bandits), and choice over a combinatorial set of arms in each round (combinatorial bandits).

However, even the seemingly restrictive setting of the N-armed bandit problem has played an important role in developing the algorithmic techniques for the more general versions. The basic N-armed setting in fact captures the fundamental challenge of handling exploration-exploitation tradeoff, and the algorithmic techniques developed for this setting have formed the basis for many efficient algorithms for the more advanced settings. In this chapter, we, therefore, first discuss the two main algorithmic techniques developed for the $N$-armed bandit problem and then briefly survey the extensions to the more complex settings of contextual bandits and combinatorial bandits.

**Notation** Throughout this chapter, we use $O(\cdot)$ and $\tilde{O}(\cdot)$ notation for brevity when discussing regret bounds. The big-Oh notation $O(\cdot)$ hides only the absolute constants, whereas the tilde-Oh notation $\tilde{O}(\cdot)$ hides absolute constants and logarithmic factors.

## 1.2 The $N$-Armed Bandit Problem

The stochastic $N$-armed bandit problem proceeds in discrete sequential rounds. In each round $t = 1, 2, 3, \ldots$, one of $N$ arms (or actions) must be chosen to be pulled (or played). Let $I_t \in \{1, \ldots, N\}$ denote the arm pulled at the $t^{th}$ time step. On pulling arm $I_t = i$ at time $t$, a random real-valued reward $r_t \in \mathbb{R}$ is observed, generated according to a fixed but unknown distribution associated with arm $i$, and mean $\mathbb{E}[r_t | I_t = i] = \mu_i$. The random rewards obtained from playing an arm repeatedly are independent and identically distributed over time and independent of the plays of the other arms. The reward is observed immediately after playing an arm. An algorithm for the stochastic MAB problem must decide which arm to play at each discrete time step (or round) $t$, based on the outcomes of the previous $t - 1$ plays. The goal is to maximize the expected total reward at time $T$, i.e., $\mathbb{E}[\sum_{t=1}^{T} \mu_{I_t}]$, where $I_t$ is the arm played in step $t$. Here, the expectation is over the random choices of $I_t$ made by the algorithm, where the randomization can result from any randomization in the algorithm as well as the randomness in the outcomes of arm pulls, which may affect the algorithm's sequential decisions.

To measure the performance of an algorithm for the MAB problem, it is common to work with the measure of expected total regret, i.e., the amount lost because of not playing the optimal arm in each step.

To formally define regret, let us introduce some notation. Let $\mu^* := \max_i \mu_i$, and $\Delta_i := \mu^* - \mu_i$. Let $n_{i,t}$ denote the total number of times arm $i$ is played in rounds 1 to $t$; thus $n_{i,t}$ is a random variable. Then the expected total regret in $T$ rounds is defined as

$$\mathcal{R}(T) := \mathbb{E}\left[\sum_{t=1}^{T}(\mu^* - \mu_{I_t})\right] = \mathbb{E}\left[\sum_{i=1}^{N} n_{i,T}\,\Delta_i\right],$$

where expectation is taken with respect to both randomness in outcomes, which may affect the sequential decisions made by the algorithm, and any randomization in the algorithm.

Two kinds of regret bounds appear in the literature for the stochastic MAB problem:

1. logarithmic problem-dependent (or instance-dependent) bounds that may have dependence on problem parameters like $\mu_i$ or $\Delta_i$, and
2. sublinear problem-independent (or worst-case) bounds that provide uniform bounds for all instances with $N$ arms.

The above definition of regret is also referred to as the "frequentist" regret of the algorithm, as opposed to the "Bayesian regret" which may be more useful if there are good priors available on the distribution of instances of an MAB problem. To differentiate between these different types of regret measures, let us use a more detailed notation $\mathcal{R}(T, \Theta)$ to denote regret for problem instance $\Theta$. Then, given a prior $P(\Theta)$ over instances $\Theta$ of the stochastic MAB problem, Bayesian regret is defined as the expected regret over instances sampled from this prior. That is,

$$\text{Bayesian regret in time } T = \mathbb{E}_{\Theta\sim P}[\mathcal{R}(T, \Theta)]$$

Note that in comparison:

$$\text{Frequentist problem-dependent regret in time } T \text{ for instance } \Theta = \mathcal{R}(T, \Theta)$$

$$\text{Frequentist problem-independent regret in time } T = \max_{\Theta} \mathcal{R}(T, \Theta)$$

In this chapter, we focus on the frequentist regret bounds (problem-dependent and problem independent); however, some references to Bayesian regret bounds are provided at relevant places.

Next, we briefly discuss two widely used algorithmic techniques for the multi-armed bandit problems: (1) Optimism under uncertainty, or more specifically, the Upper Confidence Bound (UCB) algorithm (Auer, 2002; Auer et al., 2002a), and (2) Posterior sampling, or more specifically, the Thompson Sampling (TS) algorithm (Thompson, 1933; Agrawal & Goyal, 2012a, 2017; Russo & Van Roy, 2014; Russo et al., 2018). Some other prominent techniques include inverse propensity scoring and multiplicative weight update algorithms, e.g., the EXP3 algorithm (Auer et al., 2002b), epsilon greedy algorithm, and the successive elimination algorithm (see the survey in Bubeck & Cesa-Bianchi, 2012).

### 1.2.1 Upper Confidence Bound (UCB) Algorithm

The UCB algorithm is based on the "optimism under uncertainty" principle. Abstractly, the idea is to maintain an "optimistic" bound on the mean reward for each arm, i.e., a quantity that is above the mean with high probability and converges to the mean as more observations are made. In each round, the algorithm pulls the arm with the largest UCB. Observations made on pulling the arm is used to update its UCB.

The precise mechanics of the algorithm are as follows. As before, let $n_{i,t}$ denote the number of times arm $i$ was played until (and including) round $t$, $I_t \in \{1, \ldots, N\}$ denote the arm pulled at time $t$, and $r_t \in [0, 1]$ denote the reward observed at time $t$. Then, an empirical reward estimate of arm $i$ at time $t$ is defined as:

$$\hat{\mu}_{i,t} = \frac{\sum_{s=1:\, I_s=i}^{t} r_s}{n_{i,t}} \tag{1.1}$$

The UCB algorithm computes the following quantity for each arm $i$ at the end of each round $t$:

$$\text{UCB}_{i,t} := \hat{\mu}_{i,t} + 2\sqrt{\frac{\ln t}{n_{i,t}}} \tag{1.2}$$

Then, the algorithm pulls the arm $i$ that has the highest $UCB_{i,t}$ at time $t$. The algorithm is summarized as Algorithm 1.

Here, for simplicity, it was assumed that $T \geq N$, and the algorithm started by playing every arm once. This algorithm enjoys a logarithmic problem-dependent regret bound of $O(\sum_{i:\mu_i \neq \mu^*} \frac{\ln(T)}{\Delta_i})$ and a sublinear problem-independent regret bound of $O(NT \ln(T))$. Other variations of this algorithm along with detailed proofs regret bounds can be found in Auer (2002), Bubeck and Cesa-Bianchi (2012).

### 1.2.2 Thompson Sampling (TS)

Thompson Sampling aka *Bayesian posterior sampling* is one of the oldest heuristic for the multi-armed bandit problem. It first appeared in a 1933 article by W.

**Algorithm 1** UCB algorithm for the stochastic N-armed bandit problem

1: **for** $t = 1, \ldots, N$ **do**
2: Play arm $t$
3: **end for**
4: **for** $t = N + 1, N + 2 \ldots, T$ **do**
5: Play arm $I_t = \arg\max_{i \in \{1,\ldots,N\}} \text{UCB}_{i,t-1}$.
6: Observe $r_t$, compute $\text{UCB}_{i,t}$
7: **end for**

R. Thompson (1933). In the recent years, there have been significant advances in theoretical regret based analysis of this algorithm for the $N$-armed stochastic MAB problem, including worst-case near-optimal problem-dependent and problem-independent bounds (Agrawal & Goyal, 2012a, 2013a; Kaufmann et al., 2012; Agrawal & Goyal, 2017) and Bayesian regret bounds (Russo & Van Roy, 2014, 2016). The algorithm is based on a Bayesian philosophy of learning.

Consider the problem of learning from observations generated from a parametric distribution. A frequentist approach assumes the parameters to be fixed, and uses sample observations to learn point estimates and confidence intervals for those parameters. On the other hand, a Bayesian learner maintains a probability distribution (aka belief) to capture the uncertainty about the unknown parameter. At the beginning (before seeing the samples), the *prior* distribution encodes the initial belief of the learner about the value of the parameters. Upon seeing the data, the learner updates the belief using Bayes rule. This updated distribution is called the *posterior* distribution.

Thompson Sampling is a an algorithm for the multi-armed bandit problem based on this Bayesian philosophy of learning. (In comparison, the UCB algorithm may be viewed as an algorithm based on a frequentist approach to learning). Going back to the $N$-armed bandit problem, suppose that for each arm $i$, the reward is generated from some parametric distribution $\nu_i$. Then, the overall structure of the Thompson Sampling algorithm, as described in Thompson (1933), is as follows:

- For every arm $i$, start with a prior belief on the parameters of its reward distribution.
- In every round $t$,
  - pull an arm with its probability of being the best arm according to the current belief.
  - use the observed reward to update the posterior belief distribution for the pulled arm.

Given the prior distribution and the likelihood function, in some cases the posterior distribution has a closed analytical form. In particular, given Bernoulli i.i.d. samples, if the prior is a Beta distribution,[1] then the posterior distribution is also given by a Beta distribution. Also, given Gaussian i.i.d. samples, if the prior is a Gaussian distribution, then the posterior is also given by a Gaussian distribution. This property makes these distributions a convenient choice for implementation of Thompson Sampling. Below, (in Algorithms 2 and 3) we give precise details of the TS

[1] A Beta distribution has support $(0, 1)$ with two parameters, $(\alpha, \beta)$ with probability density function

$$f(x : \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

Here, $\Gamma(x)$ is called the Gamma function. For integers $x \geq 1$, $\Gamma(x) = (x-1)!$.

**Algorithm 2** Thompson sampling for Bernoulli MAB using Beta priors

**for** $t = 1, 2, \ldots, T$ **do**
  For each arm $i = 1, \ldots, N$, independently sample $\theta_{i,t} \sim \text{Beta}(S_{i,t-1} + 1, F_{i,t-1} + 1)$.
  Play arm $I_t := \arg\max_i \theta_{i,t}$
  Observe $r_t$.
**end for**

**Algorithm 3** Thompson sampling using Gaussian priors

**for** $t = 1, 2, \ldots,$ **do**
  Independently for each arm $i = 1, \ldots, N$, sample $\theta_{i,t}$ from $\mathcal{N}(\hat{\mu}_{i,t-1}, \frac{1}{n_{i,t-1}+1})$.
  Play arm $I_t := \arg\max_i \theta_{i,t}$
  Observe reward $r_t$.
**end for**

algorithm for the special cases of (a) Bernoulli reward distribution, and (b) Gaussian reward distribution.

In the case of Bernoulli rewards, let $S_{i,t-1}$, $F_{i,t-1}$ be the number of 1s and 0s, respectively, seen for arm $i$ over its plays in rounds $\{1, \ldots, t-1\}$. Then, using Bayes rule, on starting from prior Beta(1, 1), the Beta posterior distribution in round $t$ is $\text{Beta}(S_{i,t-1} + 1, F_{i,t-1} + 1)$. The posterior mean is $\frac{S_{i,t-1}+1}{S_{i,t-1}+F_{i,t-1}+1}$ which is close to the empirical mean $\hat{\mu}_{i,t-1}$. And, the posterior variance is inversely proportional to $S_{i,t-1} + F_{i,t-1} + 2 = n_{i,t-1} + 2$. Therefore, as the number of plays $n_{i,t}$ of an arm increases, the variance of the posterior distribution decreases and the empirical mean $\hat{\mu}_{i,t}$ converges to the true mean $\mu$ of the Bernoulli distribution. For arms with small $n_{i,t}$, the variance is high, which enables exploration of arms that have been played less often and, therefore, have more uncertainty in their estimates.

These observations were utilized to derive optimal problem-dependent regret bounds for the Bernoulli MAB in Kaufmann et al. (2012), Agrawal and Goyal (2013a), Agrawal and Goyal (2017) that match the lower bound given by Lai and Robbins (1985) for this problem. For Thompson Sampling with standard Gaussian prior and Gaussian posteriors, Agrawal and Goyal (2013a), Agrawal and Goyal (2017) also show near-optimal problem-dependent bounds of $O(\sum_{\mu_i \neq \mu^*} \frac{\ln(T)}{\Delta_i})$ and problem-independent bounds of $O(\sqrt{NT})$, assuming arbitrary bounded reward distributions. Note that even though the Thompson Sampling algorithm is Bayesian in nature, all the above-mentioned works derive frequentist regret bounds for this algorithm. Furthermore, the algorithm does not assume the knowledge of true prior distribution and uses a uniform distribution or standard normal prior instead. When the true prior is known, Bayesian regret bounds have also been derived, interested readers may refer to Russo and Van Roy (2014), Russo and Van Roy (2016) and the related literature.

## 1.3 Contextual Bandits

In many sequential decision making applications, including online recommendation systems (Li et al., 2010a), online advertising (Tang et al., 2013), online retail (Cohen et al., 2016), healthcare (Bastani & Bayati, 2015; Tewari & Murphy, 2017; Durand et al., 2018), the decision in every round needs to be customized to the time-varying features of the users being served and/or seasonal factors. The contextual bandit problem (Langford & Zhang, 2007) extends the $N$-armed bandit problem to incorporate these factors and features as the context or "side information" that the algorithm can take into account before making the decision in every round.

The precise definition of this problem is as follows. In every round $t$, first the context $\mathbf{x}_{i,t}$ for every arm $i = 1, \ldots, N$ is observed and then the algorithm needs to pick an arm $I_t \in A_t \subseteq \{1, \ldots, N\}$ to be pulled. The outcome of pulling an arm depends on the context $\mathbf{x}_{I_t,t}$ of the arm pulled.

A special case of this problem is the *linear contextual bandit problem* (Auer, 2002; Chu et al., 2011; Abbasi-yadkori et al., 2011), where the expected reward on pulling an arm is a linear function of the context. Specifically, an instance of the linear contextual bandit problem is defined by a $d$-dimensional parameter $\boldsymbol{\mu} \in \mathbb{R}^d$ a priori unknown to the algorithm. The expected value of the observed reward $r_t$ on pulling an arm $i \in A_t$ with context vector $\mathbf{x}_{i,t}$ is given by $\mathbb{E}[r_t | I_t = i] = \boldsymbol{\mu}^\top \mathbf{x}_{i,t}$. The regret definition compares the performance of an algorithm to a clairvoyant policy that picks the arm with highest expected reward *in every round*:

$$\mathcal{R}(T) := \sum_{t=1}^{T} \left( \max_{i \in A_t} \boldsymbol{\mu}^\top \mathbf{x}_{i,t} \right) - \mathbb{E}\left[ \sum_{t=1}^{T} r_t \right]$$

More generally, the contextual bandit problem is defined via a linear or nonlinear, parametric or non-parametric contextual response function $f(\cdot)$, so that the expected value of the observed reward $r_t$ on pulling an arm $i$ with context vector $\mathbf{x}_{i,t}$ is given by $\mathbb{E}[r_t | I_t = i] = f(\mathbf{x}_{i,t})$. The function $f$ is unknown to the decision maker and may be learned using observations $r_t$. For the special case of the linear contextual bandit problem defined above $f(\mathbf{x}_{i,t}) = \boldsymbol{\mu}^\top \mathbf{x}_{i,t}$. A slight generalization is obtained by using a Generalized Linear Model (GLM) (Filippi et al., 2010), where $f(\mathbf{x}_{i,t}) = g(\boldsymbol{\mu}^\top \mathbf{x}_{i,t})$ for some $g(\cdot) : \mathbb{R} \to \mathbb{R}$. A significant generalization to *Lipschitz bandits* was provided in Slivkins (2011), where the only assumption on $f$ is that it satisfies a Lipschitz condition with respect to a metric.

Both UCB and Thompson Sampling algorithms have been extended to the linear contextual bandit problem. The LinUCB algorithm (Auer, 2002; Li et al., 2010b; Abbasi-yadkori et al., 2011) has been shown to achieve an $\tilde{O}(\sqrt{dT \log N})$ regret bound.[2] In case the number of arms is very large, a modified version of this algorithm can also achieve am $\tilde{O}(d\sqrt{T})$ regret bound independent of the number

[2] The $\tilde{O}(\cdot)$ notation hides logarithmic factors in $T$ and $d$, in addition to the absolute constants.

of arms. These bounds match the available lower bound for this problem within logarithmic factors in $T$ and $d$ (Bubeck & Cesa-Bianchi, 2012); however, the LinUCB algorithm is not efficiently implementable when the number of arms is large. Dani et al. (2008) show a modification to get an efficiently implementable algorithm with regret bound of $\tilde{O}(d^{3/2}\sqrt{T})$.

An extension of Thompson Sampling for linear contextual bandits was introduced in Agrawal and Goyal (2013b), Agrawal and Goyal (2012b). The algorithm is derived using a Gaussian likelihood function and a Gaussian prior on the unknown parameter $\mu$. In every round $t$, it generates a sample parameter $\tilde{\boldsymbol{\mu}}_t$ from the current posterior (Gaussian) distribution and pulls the arm $i$ that maximizes $\mathbf{x}_{i,t}^T \tilde{\boldsymbol{\mu}}_t$. A high probability regret bound of $\tilde{O}(d^{3/2}\sqrt{T})$ (or $\tilde{O}(d\sqrt{T\log(N)})$ for finite number of arms) was derived for this algorithm in Agrawal and Goyal (2012b). Note that the best known regret bound for the Thompson Sampling algorithm has a slightly worse dependence on $d$ compared to the corresponding bounds for the LinUCB algorithm. However, these bounds match the best available bounds for any efficiently implementable algorithm for this problem, e.g., those given by Dani et al. (2008).

## 1.4 Combinatorial Bandits

In many applications of sequential decision making, the decision in every round can be best described as pulling of a set or "assortment" of multiple arms. For example, consider the problem of choosing a set of ads to display on a page in online advertising, or the assortment of products to recommend to a customer in online retail. The decision maker needs to select a subset of items from a universe of items. The objective may be of maximizing expected number of clicks or sales revenue. Importantly, the customer response to the recommended assortment may depend on the *combination* of items and not just the marginal utility of each item, in the assortment. For example, two *complementary* items like bread and milk may generate more purchase when presented together. On the other hand, an item's purchase probability may decrease when presented with a substitutable item like another product with similar functionality but different brand/color/price; also referred to as a *substitution* effect. Thus, pulling an arm (i.e., offering an item as part of an assortment) no longer generates a reward from its marginal distribution independent of other arms.

A general combinatorial bandit problem can be stated as the problem of selecting a subset $S_t \subseteq [N]$ in each of the sequential rounds $t = 1, \ldots, T$. On selecting a subset $S_t$, reward $r_t$ is observed with expected value $\mathbb{E}[r_t|S_t] = f(S_t)$ where the function $f : \mathbb{R}^N \to [0, 1]$ is unknown. The goal is to minimize regret against the subset with maximum expected value:

$$\mathcal{R}(T) := Tf(S^*) - \mathbb{E}\left[\sum_t r_t\right] = \sum_{t=1}^{T}(f(S^*) - f(S_t)) \tag{1.3}$$

where $S^* = \max_{S\subseteq[N]} f(S)$. Unfortunately, it is easy to construct instances of function $f(\cdot)$ such that the lower bounds for the MAB problem would imply a regret at least exponential in $N$. Further, even if the expected reward $f(S)$ is known for all $S$, finding $S^*$ may still be computationally intractable. For this problem to be tractable, some structural assumptions on $f(\cdot)$ must be utilized. Examples of such structural assumptions include the linear model $f(S) = \boldsymbol{\mu}^T \mathbf{1}_S$ or Lipschitz functions (metric bandits) discussed in the previous section. Another example is the assumption of submodularity of function $f$, also known as the submodular bandit problem. The algorithm for online submodular minimization in Hazan and Kale (2012) can achieve a regret that is bounded by $O(NT^{2/3}\sqrt{\log(1/\delta)}$ with probability $1-\delta$, for the submodular bandit problem. Their results are in fact applicable to the adversarial bandit problem, i.e., when $r_t = f_t(S_t)$ for an arbitrary unknown sequence of submodular functions $f_1, \ldots, f_T$.

An important application of combinatorial bandits in revenue management is for dynamic assortment optimization with learning. In assortment optimization, the reward (revenue) $f(S)$ on offering a set of items $S$ is modeled using a consumer choice model. Choice models capture substitution effects among products by specifying the probability that a consumer selects a product from the offered set. The multinomial logit (MNL) model is a natural and convenient way to specify these distributions, giving one of the most widely used choice model for assortment selection problems in retail settings. This model was introduced independently by Luce (1959) and Plackett (1975); see also Train (2009), McFadden (1978), Ben-Akiva and Lerman (1985) for further discussion and survey of other commonly used choice models. Agrawal et al. (2016, 2017) formulate and study the MNL-bandit problem: a combinatorial bandit setting based on the MNL-choice model. They provide UCB and Thompson Sampling based algorithms, along with near-optimal $\tilde{O}(\sqrt{NT})$ regret bounds for this problem. More discussion on the online learning and multi-armed bandit problems resulting from different choice models in assortment optimization appear in the subsequent chapters.

## References

Abbasi-yadkori, Y., Pál, D., & Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In*Advances in Neural Information Processing Systems 24* (pp. 2312–2320)

Agrawal, S. (2019). Recent advances in multiarmed bandits for sequential decision making. *INFORMS TutORials in Operations Research*, 167–168

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2016). A near-optimal exploration-exploitation approach for assortment selection. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC)*.

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2017). Thompson sampling for the MNL-Bandit. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*.

Agrawal, S., & Goyal, N. (2012a). Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*.

Agrawal, S., & Goyal, N. (2012b). Thompson sampling for contextual bandits with linear payoffs. *CoRR abs/1209.3352*. http://arxiv.org/abs/1209.3352

Agrawal, S., & Goyal, N. (2013a). Further optimal regret bounds for Thompson Sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, (AISTATS)*.

Agrawal, S., & Goyal, N. (2013b). Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.

Agrawal, S., & Goyal, N. (2017). Near-optimal regret bounds for Thompson sampling. *Journal of ACM, 64*(5), 1–30. https://doi.org/10.1145/3088510. http://doi.acm.org/10.1145/3088510

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research, 3*, 397–422.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning, 47*(2–3), 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing, 32*(1), 48–77.

Bastani, H., & Bayati, M. (2015). Online decision-making with high-dimensional covariates. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2661896

Ben-Akiva, M., & Lerman, S. (1985). *Discrete choice analysis: Theory and application to travel demand* (Vol. 9). MIT Press.

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning, 5*(1), 1–122.

Chu, W., Li, L., Reyzin, L., & Schapire, R. E. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, (AISTATS)*.

Cohen, M. C., Lobel, I., & Paes Leme, R. (2016). Feature-based dynamic pricing. In *Proceedings of the 2016 ACM Conference on Economics and, Computation., EC '16* (pp. 817–817).

Dani, V., Hayes, T. P., & Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of The 21st Conference on Learning Theory (COLT)* (pp. 355–366).

Durand, A., Achilleos, C., Iacovides, D., Strati, K., Mitsis, G. D., & Pineau, J. (2018). Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Proceedings of the 3rd Machine Learning for Healthcare Conference* (Vol. 85, pp. 67–82).

Filippi, S., Cappe, O., Garivier, A., & Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems* (Vol. 23), Curran Associates. https://proceedings.neurips.cc/paper/2010/file/c2626d850c80ea07e7511bbae4c76f4b-Paper.pdf

Hazan, E., & Kale, S. (2012). Online submodular minimization. *Journal of Machine Learning Research, 13*(1), 2903–2922. http://dl.acm.org/citation.cfm?id=2503308.2503334

Kaufmann, E., Korda, N., & Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory - 23rd International Conference, ALT* (pp. 199–213).

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6*, 4–22.

Langford, J., & Zhang, T. (2007). The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems (NIPS) 20* (pp. 817–824). http://dl.acm.org/citation.cfm?id=2981562.2981665

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010a). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10* (pp. 661–670).

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010b). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the Nineteenth International Conference on World Wide Web (WWW-10)* (pp. 661–670).

Luce, R. (1959). *Individual choice behavior: A theoretical analysis*. Wiley.

McFadden, D. (1978). Modeling the choice of residential location. *Transportation Research Record* (673), 72–77.

Plackett, R. L. (1975). The analysis of permutations. *Applied Statistics, 24*(2), 193–202.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243.

Russo, D., & Van Roy, B. (2016). An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research, 17*, 68:1–68:30.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson sampling. *Found Trends Mach Learn, 11*(1), 1–96. https://doi.org/10.1561/2200000070

Slivkins, A. (2011). Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems 24* (pp. 1602–1610). http://papers.nips.cc/paper/4332-multi-armed-bandits-on-implicit-metric-spaces.pdf

Tang, L., Rosales, R., Singh, A., & Agarwal, D. (2013). Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM international Conference on Information and Knowledge Management (CIKM)* (pp. 1587–1594).

Tewari, A., & Murphy, S. A. (2017). From ads to interventions: Contextual bandits in mobile health. In *Mobile health - Sensors, analytic methods, and applications* (pp 495–517).

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3–4), 285–294.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge University Press.

# Chapter 2
# Reinforcement Learning


**Zheng Wen**

## 2.1 Introduction

Reinforcement learning (RL) (Sutton & Barto, 2018) is a subfield of machine learning concerned with how an *agent* (or decision-maker) should learn to take actions to maximize some notion of cumulative reward while interacting with an *environment*, as is illustrated in Fig. 2.1. Specifically, at each time step, the agent first adaptively chooses an action based on its prior knowledge, past observations, and past rewards; then, it will receive a new observation and a new reward from the environment. In general, the agent's observations and rewards are stochastic and statistically dependent on its chosen action and its *state* in the environment. In most RL problems, the environment is only partially known and the agent cannot compute an optimal or near-optimal *policy* based on its prior knowledge. Consequently, it needs to learn to take optimal or near-optimal actions while interacting with the environment.

RL is one of the three basic machine learning (Friedman et al., 2001; Bishop, 2006) paradigms, alongside *supervised learning* and *unsupervised learning*. While supervised learning and unsupervised learning algorithms aim to learn from labeled or unlabeled datasets, in RL problems, the agent aims to learn to take good actions from its interactions with a usually partially known environment. Due to its generality, RL has also been studied in many other fields, such as operations research, control theory, game theory, multi-agent systems, information theory, and statistics. From the perspective of operations research and control theory, RL is closely related to dynamic programming (DP), approximate dynamic programming (ADP), and optimal control (Bertsekas, 2000, 2011; Powell, 2007). Specifically,

Z. Wen (✉)
DeepMind, Mountain View, CA, USA



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_2

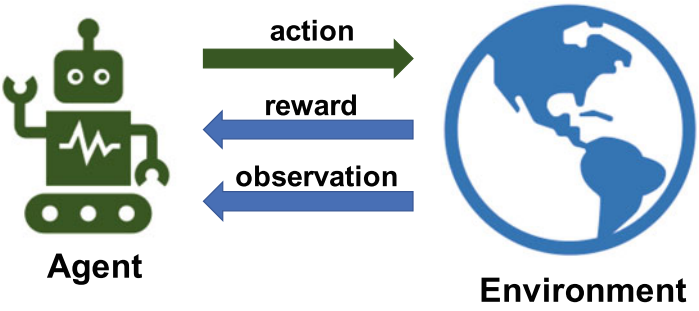


**Fig. 2.1** Illustration of reinforcement learning (RL) problems, in which the agent chooses actions and receives the observations and rewards from the environment. In most RL problems, the agent's observation includes the next state it will transit to

similar to classical DP problems that aim to compute an optimal policy in Markov decision processes (MDPs), basic RL problems are usually formulated as problems that aim to learn an optimal or near-optimal policy in MDPs. The main difference is that, in DP problems the agent is assumed to know the model of the MDP and hence can compute an optimal policy based on that model; however, in most RL problems the agent does not fully know the model and has to learn to take optimal or near-optimal actions.

One key challenge that arises in RL, but not in supervised and unsupervised learning, is the *exploration-exploitation* trade-off. Specifically, in RL, to obtain more reward, an agent should prefer actions that it *has found* effective in producing reward (exploitation). However, to discover such actions, the agent needs to try actions that *might be* effective in producing reward, or actions that might provide *useful information* about an optimal or near-optimal policy (exploration). In other words, the agent needs to exploit what it has already learned to obtain reward, but it also has to explore to make better action selections in the future. If an agent exclusively pursues exploration or exploitation, then it can easily fail or lose a lot of reward in some problems. A successful RL agent should carefully balance the exploration-exploitation trade-off by designing an appropriate exploration scheme.

In addition to the exploration-exploitation trade-off, another challenge for RL is that modern RL problems tend to have intractably large state space and/or action space. For example, in an online recommendation system, the state might include the inventory levels of all items, and the action might be an ordered list of items chosen to display. Hence, both the cardinalities of the state space and the action space can be enormous. For such large-scale RL problems, we cannot expect to learn an optimal policy with limited time, data, and computational resources. Instead, our goal is to learn a good approximate solution within limited time and using limited data and computational resources. Many such agents have been built for large-scale RL problems. In particular, deep reinforcement learning (DRL) is a subfield of RL

aiming to build agents based on (deep) neural networks that can learn approximate solutions for large-scale RL problems.

RL has extensive applications in many fields, such as online recommendation systems (Chen et al., 2019; Kveton et al., 2015), robotics (Kober et al., 2013), information retrieval (Zhang et al., 2020), energy management systems (Kuznetsova et al., 2013; Wen et al., 2015), revenue management (Gosavii et al., 2002), and financial engineering (Fischer, 2018). In the past decade, several high-performance DRL agents have been built for games like Go, Chess, and Atari games (Silver et al., 2016, 2017b, 2017a; Schrittwieser et al., 2020). Many of them have achieved a performance comparable to or even better than that of a professional human player. In particular, the AlphaGo agent (Silver et al., 2016) beat a world champion in the game of Go. Many researchers are working on extending these agents built for games to other exciting application areas.

The remainder of this chapter is organized as follows: in Sect. 2.2, we briefly review Markov decision processes (MDPs) and dynamic programming (DP) solutions. In Sect. 2.3, we provide a high-level review of some classical RL algorithms. We also discuss two key issues for RL algorithm design: exploration scheme design and approximate solution methods for large-scale RL problems, in that section. Finally, we conclude this chapter and provide pointers for further reading in Sect. 2.4.

## 2.2 Markov Decision Process and Dynamic Programming

Markov decision processes (MDPs) are stochastic control processes used in a variety of optimization and machine learning problems where the outcomes (e.g., rewards, next states) are partly random and partly controlled by the agent. They provide a framework for modeling decision making in dynamic systems. As we have mentioned in Sect. 2.1, basic RL problems can be formulated as problems in which an agent aims to learn an optimal or near-optimal policy in partially known MDPs. Several classes of MDPs, such as finite-horizon MDPs, infinite-horizon discounted MDPs, and infinite-horizon average-reward MDPs, have been widely studied in the literature. In this section, we will briefly review two classical classes of MDPs: the finite-horizon MDPs in Sect. 2.2.1 and the infinite-horizon discounted MDPs in Sect. 2.2.2. Interested readers might refer to (Bertsekas, 2000, 2011) for further reading.

When the model of an MDP is completely known, its optimal policy can be computed by dynamic programming (DP) algorithms. Though classical DP algorithms are of limited utility in RL due to the assumption that the MDP model is completely known, it does provide a foundation for understanding RL algorithms described later in this chapter. In this section, we will also briefly review the DP algorithms for the finite-horizon MDPs and the infinite-horizon discounted MDPs.

### 2.2.1 Finite-Horizon Markov Decision Process

A finite-horizon MDP is characterized by a tuple $\mathcal{M}_{\mathrm{F}} = (\mathcal{S}, \mathcal{A}, P, r, H, \rho)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $P$ and $r$, respectively, encode the transition model and the reward model, $H$ is the finite time horizon, and $\rho$ is a probability distribution over the state space $\mathcal{S}$. At the first period $h = 1$, the initial state $s_1$ is independently drawn from the distribution $\rho$. Then, at each period $h = 1, 2, \ldots, H$, if the agent takes action $a_h \in \mathcal{A}$ at state $s_h \in \mathcal{S}$, then it will receive a random reward $r_h \in \Re$ conditionally independently drawn from the reward distribution $r(\cdot|s_h, a_h)$. Moreover, for period $h < H$, the agent will transit to state $s' \in \mathcal{S}$ in the next period $h + 1$ with probability $P\left(s'\middle|s_h, a_h\right)$. The finite-horizon MDP will terminate after the agent receives reward $r_H$ at period $H$. To simplify the exposition, we use $\bar{r}(s, a)$ to denote the mean of the reward distribution $r(\cdot|s, a)$ for all state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. We also define $\mathcal{H} = \{1, 2, \ldots, H\}$ to denote the set of time periods.

In a finite-horizon MDP, the agent's goal is to maximize its expected total reward

$$\mathbb{E}\left[\sum_{h=1}^{H} r_h\right] \tag{2.1}$$

by *adaptively* choosing action $a_h$ for each period $h = 1, \ldots, H$ based on its observations so far, which can be represented as $(s_1, a_1, r_1, s_2, \ldots, s_{h-1}, a_{h-1}, r_{h-1}, s_h)$. Furthermore, since $(s_1, a_1, r_1, s_2, \ldots, s_{h-1}, a_{h-1}, r_{h-1})$ is conditionally independent of future rewards and transitions given the current state $s_h$ and the period $h$ (the Markov property), the agent only needs to choose action $a_h$ based on the state-period pair $(s_h, h)$. This motivates the notion of policy for a finite-horizon MDP. Specifically, a (randomized) policy $\pi : \mathcal{S} \times \mathcal{H} \to \Delta^{\mathcal{A}}$ is defined as a mapping from the state-period pairs to probability distributions over the action space $\mathcal{A}$. Note that $\Delta^{\mathcal{A}}$ denotes the set of probability distributions (i.e., the probability simplex) over the action space $\mathcal{A}$. Under a policy $\pi$, if the agent is at state $s_h$ at period $h$, then it will choose action $a_h = a$ with probability $\pi(a|s_h, h)$. We say a policy $\pi$ is deterministic if $\pi(a|s, h) \in \{0, 1\}$ for all action $a \in \mathcal{A}$ and all state-period pair $(s, h) \in \mathcal{S} \times \mathcal{H}$. That is, at all state-period pair $(s, h)$, the agent will choose one action with probability 1 under policy $\pi$. With a little bit abuse of notation, for a deterministic policy $\pi$, sometimes we use $\pi(s, h)$ to denote the action it chooses with probability 1 at $(s, h)$.

For each policy $\pi$, we define its *state value function* $V^{\pi} : \mathcal{S} \times \mathcal{H} \to \Re$ as

$$V^{\pi}(s, h) = \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h'}\middle|s_h = s\right], \quad \forall (s, h) \in \mathcal{S} \times \mathcal{H}, \tag{2.2}$$

where the subscript $\pi$ in notation $\mathbb{E}_{\pi}$ indicates the expectation is taken under the stochastic process defined by policy $\pi$. Notice that each policy $\pi$ defines a stochastic process evolving as follows: at each period $h \in \mathcal{H}$ with state $s_h$, the agent first chooses action $a_h \sim \pi(\cdot|s_h, h)$, then it will receive a reward $r_h \sim r(\cdot|s_h, a_h)$, and if $h < H$, it will transit to a new state $s_{h+1} \sim P(\cdot|s_h, a_h)$ in the next period $h + 1$.

$V^\pi(s,h)$ is the expected total future reward if the agent starts at state $s$ at period $h$ and chooses actions according to policy $\pi$.

Similarly, we define the *state-action value function* $Q^\pi : \mathcal{S} \times \mathcal{H} \times \mathcal{A}$ for policy $\pi$ as

$$Q^\pi(s,h,a) = \mathbb{E}_\pi \left[ \sum_{h'=h}^{H} r_{h'} \middle| s_h = s, a_h = a \right], \quad \forall (s,h,a) \in \mathcal{S} \times \mathcal{H} \times \mathcal{A} \tag{2.3}$$

that is, $Q^\pi(s,h,a)$ is the expected total future reward if the agent starts at state $s$ at period $h$, chooses action $a$ at period $h$ and chooses actions according to policy $\pi$ for all period $h' \geq h+1$. By definition of $V^\pi$ and $Q^\pi$, we have the following equation for any $(s,h) \in \mathcal{S} \times \mathcal{H}$ and $(s,h,a) \in \mathcal{S} \times \mathcal{H} \times \mathcal{A}$:

$$\begin{aligned} V^\pi(s,h) &= \sum_{a \in \mathcal{A}} \pi(a|s,h) Q^\pi(s,h,a) \\ Q^\pi(s,h,a) &= \begin{cases} \bar{r}(s,a) + \sum_{s' \in \mathcal{S}} P\left(s' \middle| s,a\right) V^\pi(s', h+1) & \text{if } h < H \\ \bar{r}(s,a) & \text{if } h = H \end{cases}. \end{aligned} \tag{2.4}$$

Note that Eq. (2.4) is referred to as the *Bellman equation* under policy $\pi$. We can rewrite the Bellman equation just in $V^\pi$ or $Q^\pi$, e.g.,

$$V^\pi(s,h) = \begin{cases} \sum_{a \in \mathcal{A}} \pi(a|s,h) \left[ \bar{r}(s,a) + \sum_{s' \in \mathcal{S}} P\left(s' \middle| s,a\right) V^\pi(s', h+1) \right] & \text{if } h < H \\ \sum_{a \in \mathcal{A}} \pi(a|s,h) \bar{r}(s,a) & \text{if } h = H \end{cases}. \tag{2.5}$$

We also define the *optimal state value function* $V^* : \mathcal{S} \times \mathcal{H} \to \Re$ as

$$V^*(s,h) = \max_\pi V^\pi(s,h), \quad \forall (s,h) \in \mathcal{S} \times \mathcal{H}, \tag{2.6}$$

which is the maximum[1] (optimal) expected total future reward if the agent starts at state $s$ at period $h$. Similarly, we define the *optimal state-action value function* $Q^* : \mathcal{S} \times \mathcal{H} \times \mathcal{A} \to \Re$ as

$$Q^*(s,h,a) = \max_\pi Q^\pi(s,h,a), \quad \forall (s,h,a) \in \mathcal{S} \times \mathcal{H} \times \mathcal{A}, \tag{2.7}$$

which is the maximum (optimal) expected total future reward if the agent starts at state $s$ at period $h$ and chooses action $a$ at period $h$. Similarly, we have the following Bellman equation for the optimal value function $V^*$ and $Q^*$:

$$V^*(s,h) = \max_{a \in \mathcal{A}} Q^*(s,h,a)$$

[1] Since we assume the time horizon and the cardinalities of $\mathcal{S}$ and $\mathcal{A}$ are all finite, the maximum is always achieved.

$$Q^*(s,h,a) = \begin{cases} \bar{r}(s,a) + \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s', h+1) & \text{if } h < H \\ \bar{r}(s,a) & \text{if } h = H \end{cases}. \tag{2.8}$$

One can prove the above Bellman equation by backward induction, similar to Proposition 1.3.1 in Bertsekas (2000).

We say a policy $\pi'$ is optimal at a state-period pair $(s,h)$ if

$$V^{\pi'}(s,h) = V^*(s,h).$$

It turns out that there exist policies that are simultaneously optimal for all state-period pairs. Specifically, one such policy is a deterministic policy $\pi^*$ satisfying[2]

$$\pi^*(s,h) \in \arg\max_{a \in \mathcal{A}} Q^*(s,h,a), \quad \forall (s,h) \in \mathcal{S} \times \mathcal{H},$$

recall that under a deterministic policy $\pi^*$, $\pi^*(s,h)$ is the action chosen at state-period pair $(s,h)$. Note that by definition, we have

$$Q^*(s,h,\pi^*(s,h)) = \max_{a \in \mathcal{A}} Q^*(s,h,a) = V^*(s,h), \quad \forall (s,h) \in \mathcal{S} \times \mathcal{H}.$$

To prove that $\pi^*$ is simultaneously optimal for all state-period pairs, we prove that $V^{\pi^*}(s,h) = V^*(s,h)$ for all $(s,h) \in \mathcal{S} \times \mathcal{H}$ by backward induction in $h$:

- For $h = H$, we have $Q^*(s,h,a) = \bar{r}(s,a)$. Consequently, we have $\pi^*(s,h) \in \arg\max_{a \in \mathcal{A}} \bar{r}(s,a)$, so we have

$$V^{\pi^*}(s,h) = \bar{r}(s,\pi^*(s,h)) = Q^*(s,h,\pi^*(s,h)) = V^*(s,h).$$

- For any $h < H$, assume that $V^{\pi^*}(s,h+1) = V^*(s,h+1)$ for all $s \in \mathcal{S}$, we now prove that $V^{\pi^*}(s,h) = V^*(s,h)$ for all $s \in \mathcal{S}$. Note that

$$\begin{aligned} V^{\pi^*}(s,h) &= Q^{\pi^*}(s,h,\pi^*(s,h)) \\ &= \bar{r}(s,\pi^*(s,h)) + \sum_{s' \in \mathcal{S}} P(s'|s,\pi^*(s,h)) V^{\pi^*}(s', h+1) \\ &= \bar{r}(s,\pi^*(s,h)) + \sum_{s' \in \mathcal{S}} P(s'|s,\pi^*(s,h)) V^*(s', h+1) \\ &= Q^*(s,h,\pi^*(s,h)) = V^*(s,h), \end{aligned} \tag{2.9}$$

[2] In general, a randomized policy $\tilde{\pi}$ is optimal if $\mathtt{supp}\tilde{\pi}(\cdot|s,h) \subseteq \arg\max_{a \in \mathcal{A}} Q^*(s,h,a)$ for all $(s,h)$, where $\mathtt{supp}\tilde{\pi}(\cdot|s,h)$ is the support of the distribution $\tilde{\pi}(\cdot|s,h)$.

where the first two equalities follow from the Bellman equation under $\pi^*$, the third equality follows from the induction hypothesis, the fourth equality follows from the Bellman equation for the optimal value function, and the last equality follows from the definition of $\pi^*$, as discussed above.

#### 2.2.1.1 Dynamic Programming Solution

Based on our discussion above, for a finite-horizon MDP $\mathcal{M}_{\mathrm{F}}$, we can compute a deterministic optimal policy $\pi^*$ based on the dynamic programming (DP) algorithm below:

**DP algorithm for finite-horizon MDP**

**Initialization:** set $V^*(s, H+1) = 0$ for all $s \in \mathcal{S}$
**Step 1:** for each $h = H, H-1, \ldots, 1$:
compute

$$Q^*(s, h, a) = \bar{r}(s, a) + \sum_{s' \in \mathcal{S}} P(s'|s, a) V^*(s', h+1) \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}$$

and

$$V^*(s, h) = \max_{a \in \mathcal{A}} Q^*(s, h, a) \quad \forall s \in \mathcal{S}$$

**Step 2:** choose a deterministic policy $\pi^*$ s.t.

$$\pi^*(s, h) \in \arg\max_{a \in \mathcal{A}} Q^*(s, h, a) \quad \forall (s, h) \in \mathcal{S} \times \mathcal{H}$$

**Return** $\pi^*$

### 2.2.2 *Discounted Markov Decision Process*

An infinite-horizon discounted Markov decision process (MDP) is characterized by a tuple $\mathcal{M}_{\mathrm{D}} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $P$ and $r$, respectively, encode the transition model and the reward model, $\gamma \in (0, 1)$ is a discrete-time discount factor, and $\rho$ is a probability distribution over the state space $\mathcal{S}$. At the first period $t = 1$, the initial state $s_1$ is independently drawn from the distribution $\rho$. Then, at each period $t = 1, 2, \ldots$, if the agent takes action $a_t \in \mathcal{A}$ at state $s_t \in \mathcal{S}$, then it will receive a random reward $r_t \in [0, 1]$ conditionally independently drawn from the reward distribution $r(\cdot|s_t, a_t)$ and will

transit to state $s' \in \mathcal{S}$ in the next period $t+1$ with probability $P\left(s'|s_t, a_t\right)$. To simplify the exposition, we use $\bar{r}(s,a)$ to denote the mean of the reward distribution $r(\cdot|s,a)$ for all state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. Notice that we assume the random reward $r_t \in [0,1]$ to simplify the exposition.

In an infinite-horizon discounted MDP, the agent's goal is to maximize its expected total discounted reward[3]

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right] \tag{2.10}$$

by adaptively choosing action $a_t$ for period $t = 1, 2, \ldots$ based on its past observations. Similarly as the finite-horizon MDPs, the past observations are conditionally independent of future rewards and transitions given the current state $s_t$ (the Markov property). Moreover, the discounted MDPs are also *time-invariant* in the sense that for any $\tau \geq 1$ and any state $s \in \mathcal{S}$,

$$\max \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_1 = s\right] \quad \text{and} \quad \max \mathbb{E}\left[\sum_{t=\tau}^{\infty} \gamma^{t-\tau} r_t \middle| s_\tau = s\right]$$

are two equivalent problems. Thus, the agent only needs to choose action $a_t$ based on the current state $s_t$. This motivates the notion of policy for a discounted MDP. Specifically, a (randomized) policy $\pi : \mathcal{S} \to \Delta^{\mathcal{A}}$ is defined as a mapping from the state space to probability distributions over the action space $\mathcal{A}$. Under a policy $\pi$, if the agent is at state $s_t$, then it will choose action $a_t = a$ with probability $\pi(a|s_t)$. Similarly, if $\pi$ is a deterministic policy, we use $\pi(s)$ to denote the action it chooses with probability 1 at state $s$.

For each policy $\pi$, we define its *state value function* $V^\pi : \mathcal{S} \to \Re$ as

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_1 = s\right], \quad \forall s \in \mathcal{S}, \tag{2.11}$$

where the subscript $\pi$ in notation $\mathbb{E}_\pi$ indicates the expectation is taken under the stochastic process defined by policy $\pi$. Specifically, note that each policy $\pi$ defines a stochastic process evolving as follows: at each period $t \in 1, 2, \ldots$ with state $s_t$, the agent first chooses action $a_t \sim \pi(\cdot|s_t)$, then it will receive a reward $r_t \sim r(\cdot|s_t, a_t)$ and transit to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$ in the next time $t+1$. $V^\pi(s)$ is the expected total discounted reward if the agent starts at state $s$ and chooses actions according to policy $\pi$.

Similarly, we define the *state-action value function* $Q^\pi : \mathcal{S} \times \mathcal{A}$ for policy $\pi$ as

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_1 = s, a_1 = a\right], \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A} \tag{2.12}$$

[3] Notice that we choose the convention that $t$ starts from 1, thus, the discount at time $t$ is $\gamma^{t-1}$. If $t$ starts from 0, then the discount at time $t$ should be $\gamma^t$. We choose the convention that $t$ starts from 1 to be consistent with the finite-horizon MDPs.

that is, $Q^{\pi}(s,a)$ is the expected total discounted reward if the agent starts at state $s$, chooses action $a$ at the first time period, and chooses actions according to policy $\pi$ at subsequent time periods. By definition of $V^{\pi}$ and $Q^{\pi}$, we have the following equation for any $s \in \mathcal{S}$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned} V^{\pi}(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) Q^{\pi}(s,a) \\ Q^{\pi}(s,a) &= \bar{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s'\middle|s,a\right) V^{\pi}(s'). \end{aligned} \tag{2.13}$$

Note that Eq. (2.13) is referred to as the *Bellman equation* under policy $\pi$ for discounted MDPs. We can rewrite the Bellman equation just in $V^{\pi}$ or $Q^{\pi}$, e.g.,

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left[\bar{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s'\middle|s,a\right) V^{\pi}(s')\right]. \tag{2.14}$$

To simplify the exposition, we define the dynamic programming (DP) operator under policy $\pi$, $\mathbb{T}_{\pi}$, as

$$(\mathbb{T}_{\pi} V)(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left[\bar{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s'\middle|s,a\right) V(s')\right], \tag{2.15}$$

where $V : \mathcal{S} \to \Re$ is a real-valued function with domain $\mathcal{S}$. Notice that by definition, $\mathbb{T}_{\pi} V : \mathcal{S} \to \Re$ is also a real-valued function with domain $\mathcal{S}$. With the shorthand notation $\mathbb{T}_{\pi}$, we can rewrite the Bellman equation 2.14 as $V^{\pi} = \mathbb{T}_{\pi} V^{\pi}$.

We also define the *optimal state value function* $V^* : \mathcal{S} \to \Re$ as

$$V^*(s) = \max_{\pi} V^{\pi}(s), \quad \forall s \in \mathcal{S}, \tag{2.16}$$

which is the maximum (optimal) expected total discounted reward if the agent starts at state $s$. Similarly, we define the *optimal state-action value function* $Q^* : \mathcal{S} \times \mathcal{A} \to \Re$ as

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{2.17}$$

which is the maximum (optimal) expected total discounted reward if the agent starts at state $s$ and chooses action $a$ at the first period. Similarly, we have the following Bellman equation for the optimal value function $V^*$ and $Q^*$:

$$\begin{aligned} V^*(s) &= \max_{a \in \mathcal{A}} Q^*(s,a) \\ Q^*(s,a) &= \bar{r}(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s'). \end{aligned} \tag{2.18}$$

Similarly, we can rewrite the above Bellman equation just in $V^*$ or $Q^*$, e.g.,

$$V^*(s) = \max_{a\in\mathcal{A}} \left[\bar{r}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V^*(s')\right]. \tag{2.19}$$

We define the DP operator $\mathbb{T}$ as

$$(\mathbb{T}V)(s) = \max_{a\in\mathcal{A}} \left[\bar{r}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V(s')\right], \tag{2.20}$$

where $V$ is a real-valued function with domain $\mathcal{S}$. With this shorthand notation, we can rewrite the Bellman equation (2.19) as $V^* = \mathbb{T}V^*$.

We have shown that $V^* = \mathbb{T}V^*$, in other words, $V^*$ is one solution of the equation $V = \mathbb{T}V$. We are also interested in if $V^*$ is the *unique* solution of that equation. It turns out that for the setting considered in this subsection, $V^*$ is the unique bounded function satisfying the equation $V = \mathbb{T}V$. Interested readers might refer to Proposition 1.2.3 in Bertsekas (2011) for a proof.[4] Similarly, we can prove that $V^\pi$ is the unique bounded function satisfying the equation $V = \mathbb{T}_\pi V$.

We say a policy $\pi'$ is optimal at a state $s \in \mathcal{S}$ if

$$V^{\pi'}(s) = V^*(s).$$

It turns out that there exist policies that are simultaneously optimal for all states. Specifically, one such policy is a deterministic policy $\pi^*$ satisfying

$$\pi^*(s) \in \arg\max_{a\in\mathcal{A}} Q^*(s,a), \quad \forall s \in \mathcal{S}$$

recall that under a deterministic policy $\pi^*$, $\pi^*(s)$ is the action chosen at state $s$. Interested readers might refer to Proposition 1.2.5 in Bertsekas (2011) for a proof.

In the remainder of this subsection, we briefly discuss two dynamic programming algorithms for discounted MDPs: *value iteration* and *policy iteration*. Specifically, value iteration can compute a good approximation of $V^*$ in finite steps; and policy iteration can compute an optimal policy $\pi^*$ in finite steps.

#### 2.2.2.1 Value Iteration

Value iteration is one algorithm that asymptotically computes $V^*$ and can compute a good approximation of $V^*$ in finite steps. It is based on the following two observations: first, $V^*$ is a *fixed point* of the DP operator $\mathbb{T}$, since $V^* = \mathbb{T}V^*$. Also, based on the discussion above, we know that it is the unique bounded fixed point. Second, $\mathbb{T}$ is a *contraction mapping* with respect to the $L_\infty$ norm. Specifically, we have that

[4] Chapter 1 in Bertsekas (2011) considers a cost minimization setting, which is equivalent to the reward maximization setting considered in this chapter if we define the cost as one minus the reward.

$$\|\mathbb{T}V_1 - \mathbb{T}V_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty , \tag{2.21}$$

for any $V_1, V_2 : \mathcal{S} \to \Re$. Note that for any $V : \mathcal{S} \to \Re$, $\|V\|_\infty = \max_{s\in\mathcal{S}} |V(s)|$. To see why Eq. (2.21) holds, notice that for any $s \in \mathcal{S}$, we have

$$\begin{aligned}
|(\mathbb{T}V_1)(s) - (\mathbb{T}V_2)(s)| &= \left|\max_{a\in\mathcal{A}} \left[\bar{r}(s,a) + \gamma \textstyle\sum_{s'\in\mathcal{S}} P(s'|s,a)V_1(s')\right]\right. \\
&\quad \left. - \max_{a\in\mathcal{A}} \left[\bar{r}(s,a) + \gamma \textstyle\sum_{s'\in\mathcal{S}} P(s'|s,a)V_2(s')\right]\right| \\
&\leq \gamma \max_{a\in\mathcal{A}} \left|\textstyle\sum_{s'\in\mathcal{S}} P(s'|s,a)\left(V_1(s') - V_2(s')\right)\right| \\
&\leq \gamma \max_{a\in\mathcal{A}} \textstyle\sum_{s'\in\mathcal{S}} P(s'|s,a)\left|V_1(s') - V_2(s')\right| \\
&\leq \gamma \max_{a\in\mathcal{A}} \textstyle\sum_{s'\in\mathcal{S}} P(s'|s,a)\|V_1 - V_2\|_\infty \\
&= \gamma \|V_1 - V_2\|_\infty .
\end{aligned}$$

Consequently, we have

$$\|\mathbb{T}V_1 - \mathbb{T}V_2\|_\infty = \max_{s\in\mathcal{S}} |(\mathbb{T}V_1)(s) - (\mathbb{T}V_2)(s)| \leq \gamma \|V_1 - V_2\|_\infty .$$

Similarly, we can prove that for any policy $\pi$, we have

$$\|\mathbb{T}_\pi V_1 - \mathbb{T}_\pi V_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$$

for any $V_1, V_2 : \mathcal{S} \to \Re$.

Moreover, notice that for any $V : \mathcal{S} \to \Re$, by definition, $\mathbb{T}V$ is also a real-valued function with domain $\mathcal{S}$. Thus, for any integer $k \geq 1$, we can recursively define $\mathbb{T}^{k+1}V = \mathbb{T}\left(\mathbb{T}^k V\right)$. Since $\mathbb{T}$ is a contraction mapping with respect to the $L_\infty$ norm, and $V^*$ is the unique bounded fixed point of $\mathbb{T}$, we have the following result:

**Proposition 2.1** *For any bounded function $V : \mathcal{S} \to \Re$, we have $\lim_{k\to\infty} \mathbb{T}^k V = V^*$. Moreover, for any integer $k = 1, 2, \ldots$, we have $\left\|\mathbb{T}^k V - V^*\right\|_\infty \leq \gamma^k \|V - V^*\|_\infty$.*

***Proof*** Since $\mathbb{T}$ is a contraction mapping with respect to $L_\infty$ norm, for any integer $k = 1, 2, \ldots$, we have

$$\left\|\mathbb{T}^k V - V^*\right\|_\infty = \left\|\mathbb{T}(\mathbb{T}^{k-1}V) - \mathbb{T}V^*\right\|_\infty \leq \gamma \left\|\mathbb{T}^{k-1}V - V^*\right\|_\infty .$$

Thus, by induction, we have $\left\|\mathbb{T}^k V - V^*\right\|_\infty \leq \gamma^k \|V - V^*\|_\infty$. This implies that $\lim_{k\to\infty} \left\|\mathbb{T}^k V - V^*\right\|_\infty = 0$ and hence $\lim_{k\to\infty} \mathbb{T}^k V = V^*$. □

The above proposition implies the following value iteration algorithm:

**Value iteration algorithm**

**Input:** number of iterations $K$
**Initialization:** choose $V_0 : \mathcal{S} \to \Re$ s.t. $V_0(s) = 0$ for all $s \in \mathcal{S}$
**Value Iteration:** for each $k = 1, 2, \ldots, K$, compute $V_k \leftarrow \mathbb{T}V_{k-1}$
**Return** $V_K$

As we have discussed above, as $K \to \infty$, $V_K$ returned by the value iteration algorithm converges to $V^*$. For a finite $K$, $V_K$ is an approximation of $V^*$. Based on Proposition 2.1, we have

$$\|V_K - V^*\|_\infty \overset{(a)}{=} \|\mathbb{T}^K V_0 - V^*\|_\infty \overset{(b)}{\leq} \gamma^K \|V_0 - V^*\|_\infty \overset{(c)}{=} \gamma^K \|V^*\|_\infty \overset{(d)}{\leq} \frac{\gamma^K}{1-\gamma},$$

where (a) follows from the definition of $V_K$, (b) follows from Proposition 2.1, (c) follows from the fact that $V_0(s) = 0$ for all $s \in \mathcal{S}$, and (d) follows from the fact that $r_t \in [0, 1]$ and hence $0 \leq V^*(s) \leq \frac{1}{1-\gamma}$ for all $s \in \mathcal{S}$. Consequently, if we choose a sufficiently large $K$, the value iteration algorithm will compute a good approximation of $V^*$.

Finally, we show that when $K$ is sufficiently large, then $V_K$ induces a near-optimal policy. Specifically, consider a policy $\pi_K$ satisfying[5] $\mathbb{T}_{\pi_K} V_K = \mathbb{T}V_K$, then we have

$$\begin{aligned}
\|V^{\pi_K} - V^*\|_\infty &= \|V^{\pi_K} - \mathbb{T}_{\pi_K} V_K + \mathbb{T}V_K - V^*\|_\infty \\
&\overset{(a)}{\leq} \|V^{\pi_K} - \mathbb{T}_{\pi_K} V_K\|_\infty + \|\mathbb{T}V_K - V^*\|_\infty \\
&= \|\mathbb{T}_{\pi_K} V^{\pi_K} - \mathbb{T}_{\pi_K} V_K\|_\infty + \|\mathbb{T}V_K - \mathbb{T}V^*\|_\infty \\
&\overset{(b)}{\leq} \gamma \|V^{\pi_K} - V_K\|_\infty + \gamma \|V_K - V^*\|_\infty \\
&\overset{(c)}{\leq} \gamma \|V^{\pi_K} - V^*\|_\infty + 2\gamma \|V_K - V^*\|_\infty,
\end{aligned}$$

where (a) and (c) follow from the triangular inequality, and (b) follows from the contraction mapping. Consequently, we have

$$\|V^{\pi_K} - V^*\|_\infty \leq \frac{2\gamma}{1-\gamma} \|V_K - V^*\|_\infty \leq \frac{2\gamma^{K+1}}{1-\gamma} \|V^*\|_\infty \leq \frac{2\gamma^{K+1}}{(1-\gamma)^2}.$$

[5] Note that one choice of such policies is a deterministic policy $\pi'$ satisfying

$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} \bar{r}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V_K(s').$$

Note that $\|V^{\pi_K} - V^*\|_\infty = \max_{s\in\mathcal{S}}\left[V^*(s) - V^{\pi_K}(s)\right]$, thus, for sufficiently large $K$, $\pi_K$ is near-optimal.

#### 2.2.2.2 Policy Iteration

Policy iteration is one algorithm that computes an optimal policy $\pi^*$, which is described as follows:

**Policy iteration algorithm**

**Initialization:** choose an arbitrary initial deterministic policy $\pi_0$
**Policy Iteration:** for each $k = 0, 1, 2, \ldots$
**step 1: (policy evaluation)** compute the state value function $V^{\pi_k}$ for policy $\pi_k$ by solving the system of linear equations

$$V = \mathbb{T}_{\pi_k} V.$$

**step 2: (policy improvement)** compute a deterministic policy $\pi_{k+1}$ satisfying

$$\mathbb{T}_{\pi_{k+1}} V^{\pi_k} = \mathbb{T} V^{\pi_k}$$

**step 3:** if $V^{\pi_k} = \mathbb{T} V^{\pi_k}$, terminate and return $\pi_k$

---

Recall that $V^{\pi_k}$ is the unique bounded solution for the Bellman equation $V = \mathbb{T}_{\pi_k} V$. We also note that by definition, this Bellman equation is a system of linear equations with $|\mathcal{S}|$ variables and $|\mathcal{S}|$ equations, where $|\mathcal{S}|$ is the cardinality of the state space $\mathcal{S}$. Thus, in the policy evaluation step, $V^{\pi_k}$ can be computed by solving linear equations. For the policy improvement step, we can choose a deterministic policy $\pi_{k+1}$ satisfying

$$\pi_{k+1}(s) \in \arg\max_{a\in\mathcal{A}}\left[\bar{r}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) V^{\pi_k}(s')\right], \quad \forall s \in \mathcal{S}.$$

Notice that the policy iteration algorithm terminates if and only if $V^{\pi_k} = V^*$, that is, if and only if $\pi_k$ is optimal.

One can prove that the policy iteration algorithm will find an optimal policy and terminate in a finite number of steps. Interested readers might refer to Proposition 2.3.1 in Bertsekas (2011) for the proof. This is the main advantage of policy iteration over value iteration. On the other hand, the policy evaluation step in policy iteration requires solving a system of linear equations. This step can be computationally expensive if the number of states $|\mathcal{S}|$ is large.

## 2.3 Reinforcement Learning Algorithm Design

Based on the Markov decision process (MDP) frameworks discussed in Sect. 2.2, in this section, we provide a high-level review of some core algorithm design issues for reinforcement learning (RL), such as the choice of learning target, how to design exploration schemes, and approximate solutions for large-scale RL problems. Specifically, this section proceeds as follows: first, we formulate two standard RL problems in Sect. 2.3.1 based on the finite-horizon MDP and the discounted MDP discussed in the previous section. Then, in Sect. 2.3.2, we discuss the differences between model-based RL and model-free RL, which correspond to different choices of learning targets. We also review some classical RL algorithms, such as Q-learning, Sarsa, and REINFORCE, in Sect. 2.3.2. Third, in Sect. 2.3.3, we review some commonly used exploration schemes and discuss why data efficient RL agents need to be able to accomplish "deep exploration". Finally, in Sect. 2.3.4, we briefly review approximate learning algorithms for large-scale RL problems, such as some state-of-the-art deep reinforcement learning (DRL) algorithms (Silver et al., 2016, 2017b).

It is worth mentioning that RL has been an active research field in the past few decades, and many different problem formulations and algorithms have been developed. Due to the space limit, we can only discuss some core algorithm design issues mentioned above and review a few classical algorithms. Interested readers might refer to Sect. 2.4 for pointers to further reading.

### *2.3.1 Reinforcement Learning Problem Formulation*

In this subsection, we formulate two RL problems based on the MDPs discussed in Sect. 2.2: episodic RL in a finite-horizon MDP, and RL in a discounted MDP.

#### 2.3.1.1 Episodic Reinforcement Learning in Finite-Horizon MDP

The first RL problem we consider is an episodic RL problem in a finite-horizon MDP described in Sect. 2.2.1. Recall that a finite-horizon MDP $\mathcal{M}_{\mathrm{F}}$ is characterized by a tuple $\mathcal{M}_{\mathrm{F}} = (\mathcal{S}, \mathcal{A}, P, r, H, \rho)$. In this episodic RL problem, we assume that the agent knows the state space $\mathcal{S}$, the action space $\mathcal{A}$, and the time horizon $H$; but does not fully know the initial state distribution $\rho$, the transition model $P$, or the reward model $r$. We also assume that the agent will repeatedly interact with $\mathcal{M}_{\mathrm{F}}$ for $T$ episodes. For any episode $t = 1, \ldots, T$, and any period $h = 1, \ldots, H$, we use $s_{th}$, $a_{th}$, and $r_{th}$ to, respectively, denote the state, action, and reward at period $h$ in episode $t$.

Each episode $t = 1, 2, \ldots, T$ proceeds as follows: at the beginning of this episode, the agent first observes an initial state $s_{t1}$, which is independently drawn

from the initial state distribution $\rho$. Then, at each period $h = 1, \ldots, H$, the agent adaptively chooses an action $a_{th} \in \mathcal{A}$ based on its prior knowledge and past observations and observes and receives a reward $r_{th} \sim r(\cdot|\, s_{th}, a_{th})$. If $h < H$, the agent will also observe the next state $s_{t,h+1} \sim P(\cdot|\, s_{th}, a_{th})$. Episode $t$ terminates once the agent receives the reward $r_{tH}$ at period $H$. The agent's goal is to maximize its expected cumulative reward in the first $T$ episodes:

$$\max \mathbb{E}\left[\sum_{t=1}^{T} \sum_{h=1}^{H} r_{th}\right].$$

Many canonical or real-world RL problems can be formulated as either special cases or extensions of the episodic RL problems described above. For example, the classical multi-armed bandit problem (Lattimore & Szepesvári, 2020) can be formulated as an episodic RL problem with one state and time horizon $H = 1$. On the other hand, agents aiming to learn good strategies in computer games usually need to interact with the games repeatedly, and each interaction can be viewed as an episode. The computer game setting can be viewed as an extension of the episodic RL problem described above, and the main difference is that the time horizon $H$ in computer games are usually random.[6] Many research works have been dedicated to episodic RL problems in the past decade (Dann et al. 2017; Wen & Van Roy, 2017; Osband et al. 2013, 2019).

#### 2.3.1.2 Reinforcement Learning in Discounted MDP

The second RL problem we consider is a RL problem in a discounted MDP $\mathcal{M}_{\mathrm{D}}$, which has been described in Sect. 2.2.2. Recall that a discounted MDP $\mathcal{M}_{\mathrm{D}}$ is characterized by a tuple $\mathcal{M}_{\mathrm{D}} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$. In this RL problem, we assume that the agent knows the state space $\mathcal{S}$, the action space $\mathcal{A}$, and the discrete-time discount factor $\gamma$; but does not fully know the initial state distribution $\rho$, the transition model $P$, or the reward model $r$. For each time step $t = 1, 2, \ldots$, we use $s_t$, $a_t$, and $r_t$ to, respectively, denote the state, action, and reward at time period $t$.

This RL problem proceeds as follows: at the first time period $t = 1$, the agent observes an initial state $s_1$, which is independently drawn from the initial state distribution $\rho$. Then, at each time step $t = 1, 2, 3, \ldots$, the agent first adaptively chooses an action $a_t \in \mathcal{A}$ based on its prior knowledge and past observations, and then observes the reward $r_t \sim r(\cdot|\, s_t, a_t)$ and the next state $s_{t+1} \sim P(\cdot|\, s_t, a_t)$. The agent's goal is to maximize its expected total discounted reward

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right].$$

[6] More precisely, the time horizon $H$ in a computer game is usually a *stopping time*, rather than deterministic.

In other words, the RL problem described in this subsection is the same as the dynamic optimization problem discussed in Sect. 2.2.2, except that the agent does not fully know $P$, $r$, and $\rho$. Consequently, the agent cannot directly compute an optimal or near-optimal policy via the value iteration algorithm or the policy iteration algorithm described in Sect. 2.2.2. Instead, the agent needs to learn to take optimal or near-optimal actions while interacting with $\mathcal{M}_{\mathrm{D}}$. It is worth mentioning that RL in discounted MDPs is one of the most classical RL problems, and many classical RL algorithms, such as Q-learning (Watkins & Dayan, 1992), were first developed under this problem formulation.

### 2.3.2 Model-Based vs. Model-Free Reinforcement Learning

As we have discussed above, in RL problems, the agent usually does not fully know the environment. For instance, in the RL problems described in Sect. 2.3.1, the agent does not know the reward model $r$ and the transition model $P$. The agent may observe the reward and possibly other observations (e.g., the next state) after taking an action at each time period. The agent needs to learn an optimal or near-optimal or even high-performance policy $\pi^\dagger$ while interacting with the environment.

Note that the agent does not have to attempt to learn $\pi^\dagger$ *directly*. Instead, it can choose to learn a *learning target* $\chi$ (Lu et al., 2021) that contains sufficient information[7] to compute $\pi^\dagger$. We can classify the RL algorithms based on their chosen learning target $\chi$. For the RL problems described in Sect. 2.3.1, some commonly chosen learning targets are:

1. the MDP model;
2. the optimal state-action value function $Q^*$;
3. an optimal policy $\pi^*$, or a near-optimal policy, or just a high-performance policy.

If an algorithm chooses the MDP model as its learning target, then we refer to that algorithm as a *model-based* RL algorithm. On the other hand, if an algorithm chooses $Q^*$, $\pi^*$, or a near-optimal policy as its learning target, then we refer to that algorithm as a *model-free* RL algorithm, since it tries to learn an optimal or near-optimal policy without learning the full MDP model. Specifically, if the learning target of an algorithm is the optimal value function $Q^*$, then that algorithm is referred to as a *value learning* algorithm. On the other hand, if the learning target is an optimal policy, a near-optimal policy, or just a high-performance policy, then the algorithm is referred to as a *policy learning* algorithm. As we will discuss below, there are pros and cons between model-based RL algorithms and model-free RL algorithms.

[7] Mathematically, it means that $\pi^\dagger = \psi(\chi)$, where $\psi$ is a function known to the agent.

#### 2.3.2.1 Model-Based Reinforcement Learning

A model-based RL algorithm chooses the MDP model as its learning target. To simplify the exposition, let us consider the episodic RL problem described in Sect. 2.3.1.1, and the model-based RL in discounted MDPs is similar. For the episodic RL problem, a model-based RL algorithm maintains a "knowledge state" about the MDP model $\mathcal{M}_{\mathrm{F}}$, and updates it while interacting with the environment. Depending on the algorithm, this knowledge state could be a point estimate of $\mathcal{M}_{\mathrm{F}}$, a confidence set of $\mathcal{M}_{\mathrm{F}}$, or the posterior distribution over $\mathcal{M}_{\mathrm{F}}$. In each episode, a model-based RL algorithm chooses actions based on its knowledge state about $\mathcal{M}_{\mathrm{F}}$.

One example of model-based RL algorithms is the posterior sampling for reinforcement learning (PSRL) developed in Osband et al. (2013), which can be viewed as a special case of Thompson sampling (Thompson, 1933; Russo et al., 2017) and is described below.

**Posterior sampling for reinforcement learning (PSRL)**

**Initialization:** a prior distribution $\mathbb{P}_0$ over the environment $\mathcal{M}_{\mathrm{F}}$
**for** each episode $t = 1, 2, \ldots$
**Step 1:** sample a finite-horizon MDP model $\tilde{\mathcal{M}}_t \sim \mathbb{P}_{t-1}$
**Step 2:** compute $\pi_t$, one optimal policy under $\tilde{\mathcal{M}}_t$
**Step 3:** apply $\pi_t$ in episode $t$, receive reward $r_{t1}, r_{t2}, \ldots, r_{tH}$, and observe the state-action-reward trajectory $\mathcal{D}_t = (s_{t1}, a_{t1}, r_{t1}, \ldots, s_{tH}, a_{tH}, r_{tH})$
**Step 4:** update the posterior $\mathbb{P}_t$ over $\mathcal{M}_{\mathrm{F}}$ using Bayes' rule, based on $\mathbb{P}_{t-1}$ and observation $\mathcal{D}_t$

Specifically, the PSRL algorithm maintains and updates a posterior distribution $\mathbb{P}_t$ over the environment $\mathcal{M}_{\mathrm{F}}$. At each episode $t$, it first samples an MDP model $\tilde{\mathcal{M}}_t$ from the posterior, then it computes a policy $\pi_t$ that is optimal under the sampled model $\tilde{\mathcal{M}}_t$. Third, it applies the policy $\pi_t$ in the true environment $\mathcal{M}_{\mathrm{F}}$ and observes the state-action-reward trajectory $\mathcal{D}_t$. Finally, it updates the posterior distribution over the environment $\mathcal{M}_{\mathrm{F}}$ based on $\mathcal{D}_t$, using the Bayes' rule.

Compared with the model-free RL algorithms, one major disadvantage of model-based RL algorithms, including PSRL described above, is that they tend to be computationally expensive for large-scale RL problems. Specifically, a model-based RL algorithm aims to learn the MDP model of the environment and needs to maintain and update a knowledge state about the MDP model. Thus, to decide how to choose actions, a model-based RL algorithm usually needs to compute a policy based on its knowledge state about the MDP model. This step often requires solving a dynamic programming problem. If the MDP model (environment) is large-scale, then this step is usually computationally expensive.

Let us use the PSRL algorithm described above to further illustrate this. In PSRL, the knowledge state about the MDP model is the posterior distribution $\mathbb{P}_{t-1}$ over the MDP model. To choose actions in episode $t$, PSRL first samples a model $\tilde{\mathcal{M}}_t \sim \mathbb{P}_{t-1}$, and then computes a policy $\pi_t$ that is optimal under the sampled model

$\tilde{\mathcal{M}}_t$. Note that computing $\pi_t$ based on $\tilde{\mathcal{M}}_t$ requires solving a dynamic programming problem in the finite-horizon MDP $\tilde{\mathcal{M}}_t$. If the environment $\mathcal{M}_{\mathrm{F}}$ is a large-scale problem and PSRL starts with an appropriately chosen prior, then $\tilde{\mathcal{M}}_t$ is also likely to be a large-scale MDP and hence computing $\pi_t$ can be computationally expensive.

On the other hand, for many RL problems, especially the large-scale RL problems that require approximate solutions (see Sect. 2.3.4), it is usually easier to develop provably data efficient model-based algorithms than provably data efficient model-free algorithms. In particular, the PSRL algorithm described above is data efficient under appropriate technical conditions (Osband et al., 2013), and we will discuss this more in Sect. 2.3.3.

#### 2.3.2.2 Q-Learning and SARSA

A widely used model-free RL algorithm is the classical Q-learning algorithm (Watkins & Dayan, 1992). As its name indicates, the Q-learning algorithm chooses the optimal state-action value function $Q^*$ as its learning target, and hence it is a value learning algorithm. To simplify the exposition, let us consider a version of Q-learning algorithm for the episodic RL problem described in Sect. 2.3.1.1, which is detailed below.

**Q-learning with $\epsilon$-greedy exploration**

**Initialization:** learning step size $\alpha \in (0, 1]$, exploration probability $\epsilon \in (0, 1]$, and initialize $Q(s, h, a)$ arbitrarily for all $(s, h, a) \in \mathcal{S} \times \mathcal{H} \times \mathcal{A}$

**for** each episode $t = 1, 2, \ldots$

observe the initial state $s_{t1} \sim \rho$

**for** each period $h = 1, \ldots, H$:

**Step 1 ($\epsilon$-greedy exploration):** with probability $\epsilon$, choose action $a_{th}$ uniformly randomly from $\mathcal{A}$; with probability $1 - \epsilon$, choose

$$a_{th} \sim \text{unif}\left(\arg\max_{a \in \mathcal{A}} Q\left(s_{th}, h, a\right)\right)$$

that is, $a_{th}$ is sampled uniformly randomly from $\arg\max_{a \in \mathcal{A}} Q\left(s_{th}, h, a\right)$

**Step 2:** take action $a_{th}$, observe reward $r_{th}$; if $h < H$, also observe the next state $s_{t,h+1}$

**Step 3:** compute the temporal difference (TD) error

$$\delta_{th} = \begin{cases} r_{th} + \max_{a'} Q(s_{t,h+1}, h+1, a') - Q\left(s_{th}, h, a_{th}\right) & \text{if } h < H \\ r_{th} - Q\left(s_{th}, h, a_{th}\right) & \text{if } h = H \end{cases} \tag{2.22}$$

**Step 4:** update $Q\left(s_{th}, h, a_{th}\right)$ as

$$Q\left(s_{th}, h, a_{th}\right) \leftarrow Q\left(s_{th}, h, a_{th}\right) + \alpha \delta_{th}$$

Roughly speaking, the above Q-learning algorithm maintains and updates an estimate $Q$ of the optimal state-action value function $Q^*$ and proceeds as follows: at each period $h$ in episode $t$, the agent first chooses an action $a_{th}$ based on the $\epsilon$-*greedy* exploration with current estimate $Q$. That is, with probability $\epsilon$, it chooses the action $a_{th}$ uniformly randomly from $\mathcal{A}$; and with probability $1-\epsilon$, it chooses $a_{th}$ *greedy*[8] to the current estimate $Q$ in the sense that $a_{th} \in \arg\max_{a\in\mathcal{A}} Q(s_{th}, h, a)$. Then, it takes action $a_{th}$, observes the reward $r_{th}$, and also observes the next state $s_{t,h+1}$ if $h < H$. Finally, the agent computes the *temporal-difference (TD) error* $\delta_{th}$ as specified in Eq. (2.22) and uses the TD error to update the value estimate $Q(s_{th}, h, a_{th})$.

Note that the Q-learning algorithm above is a temporal-difference (TD) learning algorithm, since it updates its estimate $Q$ based on the TD error $\delta_{th}$ specified in Eq. (2.22). To see why $\delta_{th}$ is referred to as a TD error, let us consider a period $h < H$ in episode $t$. Recall that $r_{th} \sim r(\cdot|s_{th}, a_{th})$ and $s_{t,h+1} \sim P(\cdot|s_{th}, a_{th})$, thus, conditioning on $s_{th}$ and $a_{th}$, $r_{th} + \max_{a'} Q(s_{t,h+1}, h+1, a')$ is an unbiased estimate of

$$\bar{r}(s_{th}, a_{th}) + \sum_{s'\in\mathcal{S}} P(s'|s_{th}, a_{th}) \max_{a'\in\mathcal{A}} Q(s', h+1, a'), \tag{2.23}$$

and hence $\delta_{th}$ is an unbiased estimate of

$$\bar{r}(s_{th}, a_{th}) + \sum_{s'\in\mathcal{S}} P(s'|s_{th}, a_{th}) \max_{a'\in\mathcal{A}} Q(s', h+1, a') - Q(s_{th}, h, a_{th}). \tag{2.24}$$

If we view $Q$ as an estimate of $Q^*$, then $Q(s_{th}, h, a_{th})$ is an estimate of $Q^*(s_{th}, h, a_{th})$. On the other hand, based on Eq. (2.23), $r_{th} + \max_{a'} Q(s_{t,h+1}, h+1, a')$ is an estimate of

$$\bar{r}(s_{th}, a_{th}) + \sum_{s'\in\mathcal{S}} P(s'|s_{th}, a_{th}) \max_{a'\in\mathcal{A}} Q^*(s', h+1, a') = Q^*(s_{th}, h, a_{th}),$$

where the equality follows from the Bellman equation. Thus, $\delta_{th}$ is the difference between two estimates of $Q^*(s_{th}, h, a_{th})$: $Q(s_{th}, h, a_{th})$ and $r_{th} + \max_{a'} Q(s_{t,h+1}, h+1, a')$. Since $r_{th} + \max_{a'} Q(s_{t,h+1}, h+1, a')$ is based on $Q$ in the next period (period $h+1$), while $Q(s_{th}, h, a_{th})$ is based on $Q$ in the current period (period $h$), this difference is referred to as a temporal-difference (TD) error.

Let us briefly discuss why the Q-learning algorithm might be able to learn the optimal state-action value function $Q^*$. Based on the value update equation

$$Q(s_{th}, h, a_{th}) \leftarrow Q(s_{th}, h, a_{th}) + \alpha\delta_{th},$$

[8] The algorithm breaks ties in a uniformly random manner, as specified in the pseudo-code.

with an appropriately chosen learning step size $\alpha$, the Q-learning algorithm updates $Q$ to minimize the absolute value (or square, which is equivalent) of the TD error $\delta_{th}$. As we have discussed above, the TD error $\delta_{th}$ is an unbiased estimate of Eq. (2.24); and the absolute value of Eq. (2.24) is minimized when $Q = Q^*$. Thus, under appropriate conditions, the Q-learning algorithm can learn $Q^*$. Rigorously speaking, one can prove that if all state-period-action triples are visited infinitely often, with a different choice of the learning step sizes that are episode-varying and satisfy some standard *stochastic approximation* (Kushner & Yin, 2003) conditions, $Q$ will converge to $Q^*$ with probability 1. Please refer to Jaakkola et al. (1994) and Tsitsiklis (1994) for the analysis.

The Q-learning algorithm is an *off-policy* learning algorithm, since it aims to learn a policy different from that used to generate data. The policy used to generate data is also known as the *behavior policy*. Specifically, the Q-learning algorithm aims to learn the optimal state-action value function $Q^*$, or equivalently, the optimal policy $\pi^*$. However, the behavior policy can be any policy that performs sufficient exploration to ensure that all state-period-action triples are visited infinitely often. In the algorithm above, the policy used to generate data is the $\epsilon$-greedy policy with respect to the current estimate $Q$. It can also be other policies, such as the Boltzmann (softmax) exploration policy with respect to the current estimate $Q$ (see Sect. 2.3.3, and Cesa-Bianchi et al. (2017) and the references therein).

The following learning algorithm, which is referred to as Sarsa (Rummery & Niranjan, 1994; Sutton 1996), is an *on-policy* variant of the Q-learning algorithm. We say Sarsa is on-policy since it attempts to evaluate and improve the policy that is used to make decisions (i.e., the behavior policy). The main difference between Sarsa and Q-learning is the TD error for period $h < H$: in Sarsa, the TD error is defined based on the state-action-reward-state-action quintuple[9] $\left(s_{th}, a_{th}, r_{th}, s_{t,h+1}, a_{t,h+1}\right)$:

$$\delta_{th} = r_{th} + Q(s_{t,h+1}, h+1, a_{t,h+1}) - Q\left(s_{th}, h, a_{th}\right).$$

Assume that the current behavior policy is $\pi$, and assume that $a_{t,h+1}$ is chosen under $\pi$, i.e., $a_{t,h+1} \sim \pi(\cdot|s_{t,h+1}, h+1)$. Similar to what we have discussed above, for Sarsa, $\delta_{th}$ is an unbiased estimate of

$$\bar{r}(s_{th}, a_{th}) + \sum_{s' \in \mathcal{S}} P(s'|s_{th}, a_{th}) \sum_{a' \in \mathcal{A}} \pi(a'|s', h+1) Q(s', h+1, a') - Q(s_{th}, h, a_{th}),$$

whose absolute value is minimized by $Q = Q^\pi$. Consequently, Sarsa continually aims to estimate $Q^\pi$ for the current behavior policy $\pi$. Note that at the same time Sarsa also updates $\pi$ toward greediness with respect to $Q^\pi$, as detailed below. Interested readers might refer to Singh et al. (2000) for the convergence analysis of Sarsa.

[9] This state-action-reward-state-action quintuple gives rise to the name Sarsa for the algorithm.

**Sarsa with $\epsilon$-greedy exploration**

**Initialization:** learning step size $\alpha \in (0, 1]$, exploration probability $\epsilon \in (0, 1]$, and initialize $Q(s, h, a)$ arbitrarily for all $(s, h, a) \in \mathcal{S} \times \mathcal{H} \times \mathcal{A}$

**for** each episode $t = 1, 2, \ldots$

observe the initial state $s_{t1} \sim \rho$

choose action $a_{t1}$ using $\epsilon$-greedy policy with respect to $Q$

**for** each period $h = 1, \ldots, H$:

**Step 1:** take action $a_{th}$, observe reward $r_{th}$; if $h < H$, also observe the next state $s_{t,h+1}$, and choose action $a_{t,h+1}$ using $\epsilon$-greedy policy with respect to $Q$

**Step 2:** compute the temporal difference (TD) error

$$\delta_{th} = \begin{cases} r_{th} + Q(s_{t,h+1}, h+1, a_{t,h+1}) - Q(s_{th}, h, a_{th}) & \text{if } h < H \\ r_{th} - Q(s_{th}, h, a_{th}) & \text{if } h = H \end{cases} \tag{2.25}$$

**Step 3:** update $Q(s_{th}, h, a_{th})$ as

$$Q(s_{th}, h, a_{th}) \leftarrow Q(s_{th}, h, a_{th}) + \alpha \delta_{th}$$

Finally, it is worth mentioning that there are many variants and extensions of the Q-learning algorithm and the Sarsa algorithm described above, such as the expected Sarsa algorithm (Van Seijen et al., 2009), the double Q-learning algorithm (Hasselt, 2010), the $n$-step TD algorithms (see van Seijen (2016) and Chap. 7 in Sutton and Barto (2018)) and the TD($\lambda$) algorithms (see Sutton (1988), Dayan (1992), Tsitsiklis (1994), and Chap. 12 in Sutton and Barto (2018)). Interested readers might refer to these references for further reading. Also, this subsection has focused on the episodic RL problem; it is straightforward to develop similar Q-learning and Sarsa algorithms for RL in discounted MDPs described in Sect. 2.3.1.2.

#### 2.3.2.3 Policy Gradient

Another class of widely used model-free RL algorithms are the policy gradient methods (see Williams (1992), Marbach and Tsitsiklis (2001), Sutton et al. (2000), and Chap. 13 in Sutton and Barto (2018)). As the name "policy gradient" indicates, these methods choose an optimal policy $\pi^*$ as their learning target and aim to learn a good approximation of $\pi^*$ with a parametric model, and hence they are policy learning algorithms. To simplify the exposition, let us motivate and consider a version of policy gradient method for the episodic RL problem described in Sect. 2.3.1.1; a similar policy gradient method can be derived for RL in discounted MDPs described in Sect. 2.3.1.2.

Consider a policy $\pi_\theta$ parameterized by $\theta \in \Re^d$, where $d$ is the dimension of $\theta$. Note that the policy $\pi_\theta$ can be parameterized in any way, as long as $\pi_\theta(a|s,h)$ is differentiable with respect to $\theta$ for all $(s,h,a)$. One common kind of parameterization is to parameterize the *preference* $\phi_\theta(s,h,a) \in \Re$ for all state-period-action triple $(s,h,a)$, and define $\pi_\theta$ via the softmax function:

$$\pi_\theta(a|s,h) = \frac{\exp\left(\phi_\theta(s,h,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\phi_\theta(s,h,a')\right)}.$$

For each $\theta \in \Re^d$, we define the expected total reward under policy $\pi_\theta$ as

$$J(\theta) = \mathbb{E}\left[V^{\pi_\theta}(s_1, 1)\right], \tag{2.26}$$

where the expectation is over the initial state[10] $s_1$, which is drawn from the initial state distribution $\rho$. Hence, the problem of finding the best policy in the policy class $\Pi = \left\{\pi_\theta : \theta \in \Re^d\right\}$ can be formulated as $\max_{\theta \in \Re^d} J(\theta)$. Of course, one natural method to maximize $J(\theta)$ is the gradient ascent algorithm based on $\nabla_\theta J(\theta)$.

The following theorem is known as the *policy gradient theorem*, which is the mathematical foundation for all policy gradient methods.

**Theorem 2.1 (Policy Gradient Theorem)** *For $J(\theta)$ defined in Eq. 2.26, we have*

$$\nabla_\theta J(\theta) = \sum_{h=1}^{H} \mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h, h, a_h) \nabla_\theta \log \pi_\theta(a_h|s_h,h)\right],$$

*where the subscript $\pi_\theta$ in notation $\mathbb{E}_{\pi_\theta}$ indicates that the expectation is taken under the stochastic process defined by policy $\pi_\theta$.*

***Proof*** Note that $V^{\pi_\theta}(s_h,h) = \sum_{a \in \mathcal{A}} \pi_\theta(a|s_h,h) Q^{\pi_\theta}(s_h,h,a)$, thus

$$\nabla_\theta V^{\pi_\theta}(s_h,h) = \sum_{a \in \mathcal{A}} \left[Q^{\pi_\theta}(s_h,h,a) \nabla_\theta \pi_\theta(a|s_h,h) + \pi_\theta(a|s_h,h) \nabla_\theta Q^{\pi_\theta}(s_h,h,a)\right].$$

From the Bellman equation (2.4), we have $\nabla_\theta Q^{\pi_\theta}(s_h,h,a) = 0$ if $h = H$ and

$$\nabla_\theta Q^{\pi_\theta}(s_h,h,a) = \sum_{s' \in \mathcal{S}} P(s'|s_h,a) \nabla_\theta V^{\pi_\theta}(s', h+1) \quad \text{if } h < H.$$

Since

[10] In Sect. 2.3.2.3, to simplify the notation, we drop the episode subscript $t$ if the discussion/analysis is within one episode.

$$\sum_{a\in\mathcal{A}}\pi_\theta(a|s_h,h)\sum_{s'\in\mathcal{S}}P(s'|s_h,a)\nabla_\theta V^{\pi_\theta}(s',h+1)=\mathbb{E}_{\pi_\theta}\left[\nabla_\theta V^{\pi_\theta}(s_{h+1},h+1)\big|s_h\right]$$

and

$$\begin{aligned}&\sum_{a\in\mathcal{A}}Q^{\pi_\theta}(s_h,h,a)\nabla_\theta\pi_\theta(a|s_h,h)\\&=\sum_{a\in\mathcal{A}}Q^{\pi_\theta}(s_h,h,a)\pi_\theta(a|s_h,h)\nabla_\theta\log\pi_\theta(a|s_h,h)\\&=\mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h,h,a_h)\nabla_\theta\log\pi_\theta(a_h|s_h,h)\big|s_h\right],\end{aligned}$$

we have

$$\begin{aligned}\nabla_\theta V^{\pi_\theta}(s_h,h)=&\mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h,h,a_h)\nabla_\theta\log\pi_\theta(a_h|s_h,h)\big|s_h\right]\\&+\mathbb{E}_{\pi_\theta}\left[\nabla_\theta V^{\pi_\theta}(s_{h+1},h+1)\big|s_h\right]\mathbf{1}(h<H).\end{aligned}$$

Taking the expectation over $s_h$, we have

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[\nabla_\theta V^{\pi_\theta}(s_h,h)\right]=&\mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h,h,a_h)\nabla_\theta\log\pi_\theta(a_h|s_h,h)\right]\\&+\mathbb{E}_{\pi_\theta}\left[\nabla_\theta V^{\pi_\theta}(s_{h+1},h+1)\right]\mathbf{1}(h<H).\end{aligned}$$

Hence we have

$$\nabla_\theta J(\theta)=\mathbb{E}_{\pi_\theta}\left[\nabla_\theta V^{\pi_\theta}(s_1,1)\right]=\sum_{h=1}^{H}\mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h,h,a_h)\nabla_\theta\log\pi_\theta(a_h|s_h,h)\right].$$

This concludes the proof. □

We now motivate and discuss one policy gradient method, referred to as REINFORCE (Williams, 1992), based on Theorem 2.1. First, note that we can compute a stochastic gradient of $J(\theta)$ based on a state-action-reward trajectory $s_1,a_1,r_1,\ldots,s_H,a_H,r_H$ under policy $\pi_\theta$. To see it, let us define $G_h=\sum_{h'=h}^{H}r_{h'}$ for any $h$, which is the total reward from period $h$ to period $H$. We claim that $\sum_{h=1}^{H}G_h\nabla_\theta\log\pi_\theta(a_h|s_h,h)$ is a stochastic gradient of $J(\theta)$. To see it, notice that

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[G_h\nabla_\theta\log\pi_\theta(a_h|s_h,h)\right]=&\mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}\left[G_h|s_h,a_h\right]\nabla_\theta\log\pi_\theta(a_h|s_h,h)\right]\\=&\mathbb{E}_{\pi_\theta}\left[Q^{\pi_\theta}(s_h,h,a_h)\nabla_\theta\log\pi_\theta(a_h|s_h,h)\right],\end{aligned}$$

where the second equality follows from $Q^{\pi_\theta}(s_h,h,a_h)=\mathbb{E}_{\pi_\theta}\left[G_h|s_h,a_h\right]$. The REINFORCE algorithm is described below. As we have discussed above, it is a stochastic gradient ascent algorithm to maximize $J(\theta)$.

**REINFORCE**

**Initialization:** differentiable policy parameterization $\pi_\theta$, initial $\theta$ and learning step size $\alpha \in (0, 1]$

**for** each episode $t = 1, 2, \ldots$

**Step 1:** generate trajectory $s_{t1}, a_{t1}, r_{t1}, \ldots s_{tH}, a_{tH}, r_{tH}$ under policy $\pi_\theta$

**Step 2:** compute $G_{th} = \sum_{h'=h}^{H} r_{th'}$ for all $h = 1, 2, \ldots, H$

**Step 3:** update $\theta \leftarrow \theta + \alpha \sum_{h=1}^{H} G_{th} \nabla_\theta \log \pi_\theta(a_{th}|s_{th}, h)$

It is worth mentioning that there are other policy gradient methods in addition to the REINFORCE algorithm presented above. Such methods include REINFORCE with baseline (Williams, 1992; Greensmith et al., 2004) and actor-critic methods (Sutton, 1984; Degris et al., 2012). Interested readers might refer to the references for further reading.

### *2.3.3 Exploration in Reinforcement Learning*

In this subsection, we briefly review exploration in RL. As we have discussed above, the exploration-exploitation trade-off is a key challenge in RL. Specifically, balancing this trade-off is crucial for a RL algorithm to be data efficient, i.e., to learn an optimal or near-optimal policy within few interactions with the environment. Specifically, if an agent does not explore enough (under-exploration), then it might get stuck in sub-optimal policies and never learn an optimal or near-optimal policy; on the other hand, if an agent explores too much (over-exploration), then it might choose sub-optimal actions in too many time steps and hence incur a huge reward loss.

This subsection is organized as follows: we briefly review some commonly used exploration schemes in Sect. 2.3.3.1; in Sect. 2.3.3.2, we motivate and discuss why data efficient RL algorithms need to be able to accomplish "deep exploration".

#### 2.3.3.1 Exploration Schemes

We now briefly review some commonly used exploration schemes, including $\epsilon$-greedy exploration, Boltzmann exploration, exploration based on *optimism in the face of uncertainty (OFU)*, and Thompson sampling. To simplify the exposition, we discuss these exploration schemes under the episodic RL problem discussed in Sect. 2.3.1.1.

**$\epsilon$-Greedy Exploration** $\epsilon$-greedy exploration is probably the simplest exploration scheme. In Sect. 2.3.2.2, we have presented two algorithms with $\epsilon$-greedy exploration: Q-learning with $\epsilon$-greedy exploration and Sarsa with $\epsilon$-greedy exploration.

Roughly speaking, in value learning algorithms, $\epsilon$-greedy exploration proceeds as follows: assume that $Q$ is a point estimate of the optimal state-action value function $Q^*$, then at each period $h$ in episode $t$, with probability $1-\epsilon$, the agent chooses an action greedy to the current estimate $Q$, i.e., $a_{th} \in \arg\max_{a\in\mathcal{A}} Q(s_{th}, h, a)$ (exploitation); and with probability $\epsilon$, it chooses a random action (exploration). Similarly, in a model-based RL algorithm that maintains and updates a point estimate of the MDP model, at each time step, the $\epsilon$-greedy exploration chooses an action greedy to the current model estimate with probability $1-\epsilon$ and chooses a random action with probability $\epsilon$. Note that the choice of $\epsilon$ trades off the exploration and exploitation.

**Boltzmann (softmax) Exploration** Boltzmann (softmax) exploration (Cesa-Bianchi et al., 2017) is similar to $\epsilon$-greedy exploration. In value learning algorithms, Boltzmann exploration proceeds as follows: assume that $Q$ is a point estimate of $Q^*$, then at each period $h$ in episode $t$, the agent chooses action $a \in \mathcal{A}$ with probability

$$\pi^{\mathrm{B}}(a|s_{th}, h) = \frac{\exp\left(Q(s_{th}, h, a)/\eta\right)}{\sum_{a'\in\mathcal{A}} \exp\left(Q(s_{th}, h, a')/\eta\right)}, \tag{2.27}$$

where $\eta > 0$ is the *temperature* of Boltzmann exploration and trades off exploration and exploitation. Specifically, as $\eta \to \infty$, $\pi^{\mathrm{B}}(\cdot|s_{th}, h)$ converges to the uniform distribution over $\mathcal{A}$ (exploration only); as $\eta \to 0$, Boltzmann exploration will choose an action greedy to $Q$ (exploitation only).

**Optimism in the Face of Uncertainty (OFU)** OFU is a class of exploration schemes that are widely used to design provably data efficient RL algorithms. One version of the OFU exploration scheme proceeds as follows: the agent maintains and updates a *confidence set* over a learning target $\chi$ (e.g., the MDP model or $Q^*$); then at the beginning of each episode, it uses this confidence set to assign each state-period-action triple $(s, h, a)$ an *optimistically biased* estimate $\hat{Q}(s, h, a)$ of $Q^*(s, h, a)$; finally, at each period $h$ in the current episode $t$, it will choose action $a_{th}$ greedy to $\hat{Q}$, i.e., $a_{th} \in \arg\max_{a\in\mathcal{A}} \hat{Q}(s_{th}, h, a)$.

**Thompson Sampling (TS)** Thompson sampling (Thompson, 1933; Russo et al., 2017) is another exploration scheme widely used to design data efficient RL algorithms. It proceeds as follows: the agent maintains and updates a posterior distribution over a learning target $\chi$ (e.g., the MDP model or $Q^*$); then at the beginning of each episode $t$, it samples a target $\tilde{\chi}_t$ from the posterior distribution and computes a policy $\pi_t$ optimal under the sampled target $\tilde{\chi}_t$; finally, it chooses actions in episode $t$ based on $\pi_t$. Note that the PSRL algorithm in Sect. 2.3.2.1 is a TS algorithm whose learning target is the MDP model.

In general, the $\epsilon$-greedy exploration and the Boltzmann exploration are computationally more efficient than OFU and TS, since they only require a *point estimate* of the learning target (e.g., $Q^*$), while OFU requires maintaining and updating a *confidence set* over the learning target and TS requires maintaining and updating

a *posterior distribution* over the learning target. On the other hand, $\epsilon$-greedy and Boltzmann exploration can easily lead to data inefficient learning, while OFU and TS are widely used to design mathematically provably data efficient RL algorithms (Kearns & Singh, 2002; Brafman & Tennenholtz, 2002, Jaksch et al. 2010; Osband et al. 2013; Wen et al. 2020). In the next subsection, we will use a simple example to illustrate this.

There are other exploration schemes in addition to those mentioned above. One of them that is particularly interesting is the *information-directed sampling (IDS)* (Russo & Van Roy, 2014; Lu et al. 2021), which samples actions in a manner that minimizes the ratio between the squared expected performance loss (known as *regret*) and a measure of information gain. Interested readers might refer to the references for further reading.

#### 2.3.3.2 Deep Exploration

In this subsection, we motivate and discuss why data efficient RL algorithms need to be able to accomplish "deep exploration" (Osband et al., 2019). As we have discussed above, in RL, exploration means that the agent needs to try actions that might provide some useful information feedback. In the special case of multi-armed bandits (MABs) (Lattimore & Szepesvári, 2020), since there is only one state, if the agent wants to gather some information by taking an action, it can always do it. However, this might not be the case for general RL problems. Specifically, some crucial information might only be obtained by trying an action at a particular state $s^\dagger$; consequently, to obtain this information, the agent needs to *learn to plan* to visit $s^\dagger$ first.

Consequently, a reliably data efficient RL algorithm needs to be able to accomplish "deep exploration". By this we mean that, the algorithm does not only consider *immediate* information gain of taking an action but also the consequences of an action or a sequence of actions on *future* learning. A deep exploration algorithm could, for instance, choose to incur performance losses over a sequence of actions while only expecting informative observations after multiple time steps. In the remainder of this section, we use a simple example to illustrate the notion of deep exploration and compare the data efficiencies of the PSRL algorithm described in Sect. 2.3.2.1 and the Q-learning with $\epsilon$-greedy exploration described in Sect. 2.3.2.2.

Let us consider an episodic RL problem with deterministic transitions and rewards, which is illustrated in Fig. 2.2 and referred to as the "chain example". Specifically, in this problem, $\mathcal{S} = \{1, 2, \ldots, H\}$ where $H$ is the time horizon, $\mathcal{A} = \{1, 2\}$, and the initial state in each episode is always $s_1 = 1$. When the agent takes action $a \in \mathcal{A}$ in state $s$ at period $h$:

- it will receive a deterministic reward $z$ if $s = H$ and $a = 1$; otherwise, it will receive reward 0.
- it will transition to state $\min\{s + 1, H\}$ if $a = 1$ and $h < H$; it will transition to state $\max\{s - 1, 1\}$ if $a = 2$ and $h < H$.

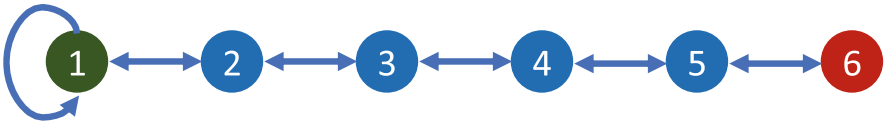


**Fig. 2.2** Illustration of the "chain example" with $H = 6$. The nodes denote the states and the arrows denote the possible state transitions. We use the green node to denote the fixed initial state and use the red node to denote the "informative state"

We assume that the agent knows everything about this environment, except the deterministic reward $z$ at state-action pair $(s = H, a = 1)$. We assume that the agent's prior over $z$ is $\mathbb{P}_0(z = 1) = \mathbb{P}_0(z = -1) = 0.5$. Obviously, the optimal policy $\pi^*$ depends on $z$. For example, if $z = 1$, the only optimal sequence of actions is to always choose $a = 1$. The agent needs to visit state $s = H$ and take action $a = 1$ there to learn the crucial information $z$. If the agent *plans* a sequence of actions to do so, we say it accomplishes the deep exploration in this example.

In this example, the data efficiency of an algorithm can be measured by the expected number of episodes it takes for the algorithm to learn $z$. Let us consider the PSRL algorithm first. Note that for this example, sampling an MDP model $\tilde{\mathcal{M}}_t$ is equivalent to sampling a deterministic reward $\tilde{z}_t \in \{-1, 1\}$ at state-action pair $(s = H, a = 1)$, since other parts of the environment are known. In episode $t = 1$ with prior $\mathbb{P}_0$, the agent will sample $\tilde{z}_1 = \pm 1$ with equal probability 0.5. Note that with $\tilde{z}_1 = 1$, the PSRL algorithm will choose a sequence of actions $a_{t1} = a_{t2} = \ldots = a_{tH} = 1$ in episode $t = 1$ and hence learn $z$; on the other hand, with $\tilde{z}_1 = -1$, the PSRL algorithm will not learn $z$ in this episode. Thus, in episode 1, the PSRL algorithm will learn $z$ with probability 0.5. Since the PSRL algorithm will not update its posterior before learning $z$, the expected number of episodes it takes for PSRL to learn $z$ is 2.

On the other hand, for Q-learning with $\epsilon$-greedy exploration, we assume that $Q$ is initialized as $Q(s, h, a) = 0$ for all $(s, h, a)$. Note that under this algorithm, before the agent observes $z$, $Q(s, h, a) = 0$, $\forall (s, h, a)$ and the algorithm chooses actions uniformly randomly at all state-period pairs. In such episodes, the agent will learn $z$ with probability $2^{-H}$. Hence, the expected number of episodes for this Q-learning algorithm to learn $z$ is $2^H$.

To sum up, in this example, Q-learning with $\epsilon$-greedy exploration is highly data inefficient compared to PSRL. This is because PSRL accomplishes deep exploration: in each episode, it plans based on a sampled MDP model and hence considers the consequences of a sequence of actions. On the other hand, the Q-learning algorithm just chooses random actions before it observes the crucial information $z$.

### 2.3.4 Approximate Solution Methods and Deep Reinforcement Learning

Many modern RL problems tend to have intractably large state space $\mathcal{S}$ and/or action space $\mathcal{A}$. For such large-scale RL problems, an algorithm that aims to learn an optimal policy $\pi^*$ *asymptotically* will require not only an intractably large memory space but also intractably many time steps for learning. Let us still use the episodic RL problem to illustrate the ideas. Consider the Q-learning algorithm with an exploration scheme that performs sufficient exploration (not necessarily the $\epsilon$-greedy exploration). As we have discussed in Sect. 2.3.2.2, under appropriate conditions this algorithm learns $Q^*$ asymptotically. Notice that this algorithm requires an $O(|\mathcal{S}||\mathcal{A}|H)$ memory space to store the point estimate $Q$ of $Q^*$. Moreover, since the algorithm only updates its estimate $Q(s,h,a)$ for state-period-action triple $(s,h,a)$ when it visits that triple, thus, to learn a good estimate of $Q^*$, the algorithm needs to visit each state-period-action triple at least once. This requires $\Omega(|\mathcal{S}||\mathcal{A}|)$ episodes, which is intractably many for large-scale problems.

Thus, for such large-scale RL problems, our goal is to learn a good approximate solution with limited memory space and limited time steps. One such approach, which is commonly used in practice, is to approximate the learning target (e.g., $Q^*$ or $\pi^*$) by a low-dimensional parametric model and learn the parameters of that model. Note that if the parametric model can well approximate the learning target, and the number of parameters to learn is much less than the "size" of the learning target (e.g., the "size" of $Q^*$ is $|\mathcal{S}||\mathcal{A}|H$), then learning with this parametric model can significantly improve the data efficiency.

One such learning algorithm is the REINFORCE algorithm described in Sect. 2.3.2.3. Recall that REINFORCE approximates its learning target $\pi^*$ by a parametric model $\pi_\theta$ and tries to learn a good parameter vector $\theta$ via stochastic gradient ascent.

Similarly, many value learning algorithms for large-scale RL problems aim to learn a good approximation of $Q^*$ via a parametric model $Q_\theta$, where $\theta$ is the parameter vector to be learned. There are many difference choices of the parametric model $Q_\theta$. One classical choice is to choose $Q_\theta$ linear in the parameter vector $\theta$. Specifically, each state-period-action triple $(s,h,a)$ is associated with a known feature vector $\phi(s,h,a) \in \Re^d$, and for any $\theta \in \Re^d$,

$$Q_\theta(s,h,a) = \phi(s,h,a)^T\theta, \tag{2.28}$$

where the superscript $T$ denotes the vector transpose and $d$ is the feature dimension. This parametric model is known as the *linear value function approximation* in the literature (see Chaps. 6 and 7 of Bertsekas (2011) and the references therein).

Another choice of the parametric model, which is widely used in the past decade, is to choose $Q_\theta$ as a (deep) neural network with fixed architecture and parameter vector $\theta$. Note that the parameter vector $\theta$ typically encodes the weights and the biases in all layers of the neural network. Approximate solution methods based

on a (deep) neural network (NN) model are also known as *deep reinforcement learning (DRL)* algorithms (Arulkumaran et al., 2017; Li, 2017). One well-known DRL algorithm is deep Q-learning with *experience replay* (Mnih et al., 2015), which is also known as deep Q-network (DQN) and is described below.

**Deep Q-learning with experience replay (DQN)**

**Initialization:** architecture of NN $Q_\theta$, initial $\theta$, exploration probability $\epsilon$, FIFO replay buffer $\mathcal{D}$ with capacity $N$, minibatch size $B$, and a gradient-based optimization algorithm `optimizer`

**for** each episode $t = 1, 2, \ldots$
set $\theta^- \leftarrow \theta$
observe the initial state $s_{t1} \sim \rho$
**for** each period $h = 1, \ldots, H$:
**Step 1 ($\epsilon$-greedy exploration):** with probability $\epsilon$, choose action $a_{th}$ uniformly randomly from $\mathcal{A}$; with probability $1 - \epsilon$, choose

$$a_{th} \sim \text{unif}\left(\arg\max_{a \in \mathcal{A}} Q_\theta\left(s_{th}, h, a\right)\right)$$

that is, $a_{th}$ is sampled uniformly randomly from $\arg\max_{a \in \mathcal{A}} Q_\theta\left(s_{th}, h, a\right)$
**Step 2:** take action $a_{th}$, observe reward $r_{th}$; if $h < H$, also observe the next state $s_{t,h+1}$
**Step 3:** store transition $(s_{th}, h, a_{th}, r_{th}, s_{t,h+1})$ in the replay buffer $\mathcal{D}$; if $h = H$, set $s_{t,h+1} =$ `null`
**Step 4:** sample a random minibatch of transitions $(s_j, h_j, a_j, r_j, s'_j)$ for $j = 1, 2, \ldots, B$ from $\mathcal{D}$, and set

$$y_j = r_j + \max_{a' \in \mathcal{A}} Q_{\theta_-}(s'_j, h_j + 1, a') \quad \forall j = 1, 2, \ldots, B \tag{2.29}$$

we set $Q_{\theta_-}(s'_j, h_j + 1, a') = 0$ if $s'_j =$ `null`
**Step 5:** define the loss function $\ell(\theta)$ and compute the gradient $g$

$$\ell(\theta) = \tfrac{1}{2} \sum_{j=1}^{B} \left(Q_\theta(s_j, h_j, a_j) - y_j\right)^2, \quad g = \nabla_\theta \ell(\theta),$$

and update $\theta \leftarrow$ `optimizer`$(\theta, g)$ to minimize $\ell(\theta)$

Deep Q-learning with experience replay is similar to the Q-learning algorithm described in Sect. 2.3.2.2. Specifically, its learning target is still the optimal state-action value function $Q^*$, it still uses $\epsilon$-greedy exploration, and it is still an off-policy learning algorithm. However, there are two main differences: the first difference is that the deep Q-learning algorithm approximates $Q^*$ by a neural network $Q_\theta$ and learns the parameter vector $\theta$. The second difference is that it uses

a technique known as experience replay (Lin, 1992) to enhance the data efficiency. Specifically, the transitions are stored in a replay buffer $\mathcal{D}$. At each period, a minibatch of transitions are sampled with replacement from $\mathcal{D}$, and the deep Q-learning algorithm updates $\theta$ using a stochastic gradient computed based on this minibatch. With experience replay, a transition $(s_{th}, h, a_{th}, r_{th}, s_{t,h+1})$ is potentially used in many parameter update steps, which allows for greater data efficiency.

We also would like to clarify some technical issues in the deep Q-learning algorithm described above. First, how to choose the architecture of $Q_\theta$ is highly non-trivial and in general application-dependent. Second, due to the memory space limit, the replay buffer $\mathcal{D}$ has a finite capacity $N$. Hence, when $\mathcal{D}$ is full and the agent would like to store a new transition, it needs to either delete a transition from $\mathcal{D}$ or discard the new transition. There are many ways to do it, and in the algorithm above, the buffer uses a first in, first out (FIFO) buffer replacement strategy. Third, it is worth mentioning that the optimization algorithm `optimizer` can be any gradient-based algorithm (Ruder, 2016), such as the stochastic gradient descent (SGD) algorithm and the Adam algorithm (Kingma & Ba, 2014). Note that some `optimizer` like Adam also needs to update the optimizer state (e.g., the first and second order moments in Adam), which is abstracted away from the pseudo-code above. Finally, note that in Eq. (2.29), $y_j$ is computed based on $\theta_-$ instead of $\theta$. Thus, the gradient $g$ is

$$g = \sum_{j=1}^{B} \left( Q_\theta(s_j, h_j, a_j) - y_j \right) \nabla_\theta Q_\theta(s_j, h_j, a_j).$$

Also notice that though $\theta$ is updated in every period, $\theta_-$ (and hence $Q_{\theta_-}$, the function used to compute the "target values" $y_j$'s) remains fixed within one episode. Keeping $\theta_-$ fixed within one episode might be crucial for the convergence of the deep Q-learning algorithm in some applications.

Deep reinforcement learning (DRL) has been an active research area in the past decade, and the deep Q-learning algorithm described above is one of the first algorithms developed in this area. It is worth mentioning that one agent based on a variant of it has achieved a level comparable to that of a professional human games tester across 49 games of the challenging Atari 2600 games (Mnih et al., 2015). More advanced DRL agents, such as AlphaGo (Silver et al., 2016) and MuZero (Schrittwieser et al., 2020) have also been developed. Interested readers might refer to the references for further reading.

## 2.4 Conclusion and Further Reading

In this chapter, we have briefly reviewed some fundamental concepts, standard problem formulations, and classical algorithms of reinforcement learning (RL). Specifically, in Sect. 2.2, we have reviewed Markov decision processes (MDPs) and dynamic programming (DP), which provide mathematical foundations for both the problem formulation and algorithm design for RL. In Sect. 2.3, we have

classified the RL algorithms based on their learning targets and reviewed some classical algorithms such as PSRL, Q-learning, Sarsa, and REINFORCE. We have also reviewed the standard exploration schemes in RL in Sect. 2.3.3 and reviewed approximate solution methods for large-scale RL problems in Sect. 2.3.4.

Before concluding this chapter, we would like to provide some pointers for further reading. Due to the space limit, we have not covered many exciting topics in RL, such as RL problems based on average-reward MDPs (see Mahadevan (1996) and Chap. 5 of Bertsekas (2011)), hierarchical reinforcement learning (Pateria et al., 2021; Al-Emran, 2015), multi-agent reinforcement learning (Busoniu et al., 2008; Zhang et al., 2021), imitation learning (Hussein et al., 2017), partially observable MDPs (Kaelbling et al., 1998), inverse reinforcement learning (Ng et al. 2000; Arora & Doshi, 2021), and safe reinforcement learning (Garcıa & Fernández, 2015). Interested readers might refer to the references for further reading. There are also several classical textbooks on RL and related topics, such as Sutton and Barto (2018), Bertsekas (2000, 2011, 2019), Szepesvári (2010), and Powell (2007). DRL has been an active research area in the past decade, and there are also some recent and more applied books on DRL (Lapan, 2018; Ravichandiran, 2018). Interested readers might also refer to them for further reading.

## References


Al-Emran, M. (2015). Hierarchical reinforcement learning: A survey. *International Journal of Computing and Digital Systems, 4*(02). https://dx.doi.org/10.12785/IJCDS/040207

Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence, 297*, 103500.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine, 34*(6), 26–38.

Bertsekas, D. (2019). *Reinforcement and optimal control*. Belmont: Athena Scientific

Bertsekas, D. P. (2000). *Dynamic programming and optimal control* (Vol. 1). Belmont: Athena scientific.

Bertsekas, D. P. (2011). *Dynamic programming and optimal control* (Vol. II, 3rd ed.). Belmont: Athena scientific.

Bishop, C. M. (2006). *Pattern recognition and machine learning (Information science and statistics)*. Berlin, Heidelberg: Springer.

Brafman, R. I., & Tennenholtz, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research, 3*(Oct), 213–231.

Busoniu, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(2), 156–172.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., & Neu, G. (2017). *Boltzmann exploration done right*. Preprint. arXiv:170510257.

Chen, X., Li, S., Li, H., Jiang, S., Qi, Y., & Song, L. (2019). Generative adversarial user model for reinforcement learning based recommendation system. In *International Conference on Machine Learning, PMLR* (pp. 1052–1061).

Dann, C., Lattimore, T., & Brunskill, E. (2017). *Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning*. Preprint. arXiv:170307710.

Dayan, P. (1992). The convergence of td (λ) for general λ. *Machine Learning, 8*(3–4), 341–362.

Degris, T., White, M., & Sutton, R. S. (2012). *Off-policy actor-critic*. Preprint. arXiv:12054839.
Fischer, T. G. (2018). *Reinforcement Learning in Financial Markets—A Survey*. Tech. rep., FAU Discussion Papers in Economics.
Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*. *Springer series in statistics*. New York: Springer.
Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research, 16*(1), 1437–1480.
Gosavii, A., Bandla, N., & Das, T. K. (2002). A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. *IIE Transactions, 34*(9), 729–742.
Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research, 5*(9), 1471–1530.
Hasselt, H. (2010). Double q-learning. *Advances in Neural Information Processing Systems, 23*, 2613–2621.
Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR), 50*(2), 1–35.
Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation, 6*(6), 1185–1201.
Jaksch, T., Ortner, R., & Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research, 11*(4), 1563–1600.
Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101*(1–2), 99–134.
Kearns, M., & Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning, 49*(2), 209–232.
Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. Preprint. arXiv:14126980.
Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research, 32*(11), 1238–1274.
Kushner, H., & Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications* (Vol. 35). New York: Springer Science & Business Media.
Kuznetsova, E., Li, Y. F., Ruiz, C., Zio, E., Ault, G., & Bell, K. (2013). Reinforcement learning for microgrid energy management. *Energy, 59*, 133–146.
Kveton, B., Szepesvari, C., Wen, Z., & Ashkan, A. (2015). Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning, PMLR* (pp. 767–776)
Lapan, M. (2018). *Deep reinforcement learning hands-on: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Birmingham: Packt Publishing Ltd.
Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge: Cambridge University Press.
Li, Y. (2017). *Deep reinforcement learning: An overview*. Preprint. arXiv:170107274.
Lin, L. J. (1992). *Reinforcement learning for robots using neural networks*. Pittsburgh: Carnegie Mellon University.
Lu, X., Van Roy, B., Dwaracherla, V., Ibrahimi, M., Osband, I., & Wen, Z. (2021). *Reinforcement learning, bit by bit*. Preprint. arXiv:210304047.
Mahadevan, S. (1996). Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning, 22*(1), 159–195.
Marbach, P., & Tsitsiklis, J. N. (2001). Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control, 46*(2), 191–209.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.
Ng, A. Y., Russell, S. J., et al. (2000). Algorithms for inverse reinforcement learning. In *ICML* (Vol. 1, p. 2).

Osband, I., Russo, D., & Van Roy, B. (2013). *(More) Efficient reinforcement learning via posterior sampling*. Preprint. arXiv:13060940.

Osband, I., Van Roy, B., Russo, D. J., Wen, Z., et al. (2019) Deep exploration via randomized value functions. *Journal of Machine Learning Research, 20*(124), 1–62.

Pateria, S., Subagdja, B., Tan, A. H., & Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR), 54*(5), 1–35.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality* (Vol. 703). New York: Wiley.

Ravichandiran, S. (2018). Hands-on reinforcement learning with Python: Master reinforcement and deep reinforcement learning using OpenAI gym and tensorFlow. Birmingham: Packt Publishing Ltd.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. Preprint. arXiv:160904747.

Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems* (Vol. 37). Citeseer.

Russo, D., & Van Roy, B. (2014). Learning to optimize via information-directed sampling. *Advances in Neural Information Processing Systems, 27*, 1583–1591.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2017). *A tutorial on Thompson sampling*. Preprint. arXiv:170702038.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature, 588*(7839), 604–609.

van Seijen, H. (2016). *Effective multi-step temporal-difference learning for non-linear function approximation*. Preprint. arXiv:160805151.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature, 529*(7587), 484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017a). *Mastering chess and shogi by self-play with a general reinforcement learning algorithm*. Preprint. arXiv:171201815.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017b). Mastering the game of go without human knowledge. *Nature, 550*(7676), 354–359.

Singh, S., Jaakkola, T., Littman, M. L., & Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning, 38*(3), 287–308.

Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning, 3*(1), 9–44.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in neural information processing systems* (pp. 1038–1044). Cambridge: MIT Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge: MIT Press.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 4*(1), 1–103.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3/4), 285–294.

Tsitsiklis, J. N. (1994). Asynchronous stochastic approximation and q-learning. *Machine Learning, 16*(3):185–202.

Van Seijen, H., Van Hasselt, H., Whiteson, S., & Wiering, M. (2009). A theoretical and empirical analysis of expected Sarsa. In *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (pp. 177–184). New York: IEEE.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3–4), 279–292.

Wen, Z., & Van Roy, B. (2017). Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research, 42*(3), 762–782.

Wen, Z., O'Neill, D., & Maei, H. (2015). Optimal demand response using device-based reinforcement learning. *IEEE Transactions on Smart Grid, 6*(5), 2312–2324.

Wen, Z., Precup, D., Ibrahimi, M., Barreto, A., Van Roy, B., & Singh, S. (2020). On efficiency in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems* (Vol. 33)

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3), 229–256.

Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of reinforcement learning and control* (pp. 321–384).

Zhang, W., Zhao, X., Zhao, L., Yin, D., Yang, G. H., & Beutel, A. (2020). Deep reinforcement learning for information retrieval: Fundamentals and advances. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2468–2471)

# Chapter 3
# Optimal Learning and Optimal Design


**Ilya O. Ryzhov**

## 3.1 Introduction

Suppose that $\mu_1$, $\mu_2$ are two population means—perhaps the average clickthrough rates or average session durations for two different designs of an e-commerce website. The firm's online marketing team wishes to know if one design is more effective than the other; to that end, $N$ customers have been randomly selected for an A/B test. The two respective designs are shown to $N_1$ and $N_2$ randomly chosen customers, with $N_1 + N_2 = N$, and sample means $\theta_1, \theta_2$ (empirical average clickthrough rates or session durations) are obtained.

Under the usual normality assumptions, we calculate the two-sample test statistic

$$z^N = \frac{\theta_1 - \theta_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}}, \tag{3.1}$$

where $\sigma_1$, $\sigma_2$ are the population standard deviations for the two designs. (For simplicity, let us suppose that these are known.) As is taught in every statistics course, the statistic (3.1) is used to test the null hypothesis that $\mu_1 = \mu_2$.

The expression in (3.1) also appears in another context, however. Suppose that, in reality, $\mu_1 > \mu_2$. The results of the A/B test will be used to select one design for adoption. The selection decision will be incorrect if $\theta_2 > \theta_1$, that is, the second design seems to be better than the first. As $N$ increases, the probability of incorrect selection will be reduced, and it is possible to characterize the rate at which it vanishes to zero very precisely. Suppose that $N \to \infty$, but $\frac{N_1}{N} \to p_1$, and $\frac{N_2}{N} \to p_2$,

I. O. Ryzhov (✉)
Robert H. Smith School of Business, University of Maryland, College Park, MD, USA
e-mail: iryzhov@umd.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_3

with $p_1 + p_2 = 1$ and $p_1, p_2 > 0$. In other words, the design is tested on more and more customers, but a fixed proportion of the total number $N$ is assigned to each design. Then, one can obtain the so-called *large deviations law*

$$\lim_{N\to\infty} \frac{1}{N} \log P\left(\theta_2 > \theta_1\right) = -\Gamma,$$

where

$$\Gamma = \frac{(\mu_1 - \mu_2)^2}{2\left(\frac{\sigma_1^2}{p_1} + \frac{\sigma_2^2}{p_2}\right)}. \tag{3.2}$$

That is, the probability of incorrect selection behaves like $e^{-\Gamma \cdot N}$, where $\Gamma > 0$ is a fixed constant determined by the various population parameters, as well as by the proportions of customers assigned to each design. It is easy to see that (3.2) is none other than $\lim_{N\to\infty} \frac{1}{2N}\left(z^N\right)^2$, where $z^N$ is the test statistic from (3.1). A quantity that we are used to seeing in the context of testing for differences between populations also plays a second role in the evaluation of *decisions* made as a result of the test (in this case, the decision to select the design with the highest sample mean).

This interpretation opens the door to the study of *optimal designs*. The probability of incorrect selection will vanish more quickly as $\Gamma$ increases, but we have the ability to influence $\Gamma$ by varying $p_1$ and $p_2$. Since the numerator of (3.2) is constant, we can maximize $\Gamma$ by minimizing the denominator, giving rise to the problem

$$\min \frac{\sigma_1^2}{p_1} + \frac{\sigma_2^2}{p_2} \tag{3.3}$$

subject to $p_1 + p_2 = 1$ and $p_1, p_2 \geq 0$. It is easy to see that the solution to (3.3) is characterized by

$$\frac{p_1}{p_2} = \frac{\sigma_1}{\sigma_2}, \tag{3.4}$$

so the population with higher variance should receive a larger proportion of the customers. The ratio (3.4) is well-known in the statistics literature (see, e.g., Dannenberg et al., 1994) in the context of the original two-sample test.

This chapter will explore connections between statistics and optimization that arise in optimal learning. A major focus will be on statistical design of experiments, a classical area that is only now being connected to decision problems. As we saw in the preceding example, the statistical problem of choosing sample sizes for a two-sample test (or, in different words, the problem of *allocating a learning budget* between two alternatives) has direct implications for the probability of making a suboptimal decision based on the results of the sample. We will show a more general

form of this problem, known under the name of *ranking and selection*, and discuss the connections between optimal budget allocations and state-of-the-art learning algorithms. We will also discuss some more general challenges in the design of statistical models for learning problems.

The types of applications for which these mathematical and algorithmic developments are most useful can be seen as generalizations of the A/B testing problem described above. Often, the goal is to choose the best among a finite set of alternatives, with the total number of possible choices being too large for exhaustive experimentation. For example, suppose that, instead of just two competing website designs, we plan to offer personalized content to each user depending on the user's past purchases or preferences. Perhaps an "alternative" could be a set of products or search results, in which case the number of alternatives becomes combinatorially large; such assortment planning problems will be discussed in more detail in Chaps. 8–10. Alternately, an alternative could represent a multi-attribute decision, e.g., in a medical context, where a doctor chooses not only a drug but a dosage level, perhaps customized to patient attributes (see, e.g., Nasrollahzadeh & Khademi, 2020).

It is also possible to have continuous-valued alternatives. Such problems often arise in simulation optimization, where the performance of a complex engineering system may be represented by an expensive simulator that requires days or weeks of machine time in order to evaluate a single scenario. System performance depends on multiple continuous-valued design parameters. For example, as discussed in Qu et al. (2015), the energy output of a wind farm depends on the locations of the individual wind turbines, the lengths of the turbine blades, the wind speed and altitude, and other such factors, and one would like to identify the most (or least) favorable scenario before the wind farm is built. Similar issues also commonly arise in hyperparameter tuning for machine learning models (Eitrich & Lang, 2006), where the performance (predictive power) of the model is a function of a high-dimensional hyperparameter vector.

Our discussion here is mainly motivated by applications where the goal is to identify the best alternative as efficiently as possible under a limited budget. We are not necessarily interested in the outcomes of the individual experiments themselves as long as they lead us to a good selection at the end. In this way, the problems we consider are different from multi-armed bandit problems, which almost always focus on maximizing cumulative reward. Within the bandit literature, the substream that studies "best-arm identification" problems (Garivier & Kaufmann, 2016) is closest to our focus here.

This chapter is organized as follows. Section 3.2 gives a brief overview of key concepts from the classical literature on statistical design of experiments. Section 3.3 then develops a bridge between optimal design and learning using the ranking and selection model for illustration. We explain the meaning of an optimal design in a context where the goal is to select the best alternative and contrast it with the classical meaning. Section 3.4 then shows how this concept of optimal design arises in two popular methodologies for sequential learning, sometimes in unexpected ways. We argue that optimal designs can be directly used to develop

such methods. Section 3.5 describes two instances of very recent research where this concept of optimality was developed in settings that fall outside the scope of ranking and selection. Specifically, we discuss linear regression, the classical setting of the design of experiments literature, and contrast the value-based design with traditional ones; we also briefly touch on an interesting application of these ideas to approximate dynamic programming. Section 3.6 concludes. At the end of each section, we provide additional references on related topics for interested readers.

## 3.2 Statistical Design of Experiments

The experimental design problem has a long history in statistics (see, e.g., Kiefer, 1971). Typically, one begins with a least squares regression model

$$y = \beta^\top x + \varepsilon,$$

where $\beta \in \mathbb{R}^d$ is a vector of unknown regression coefficients, $x \in \mathbb{R}^d$ is a vector of features obtained from historical data, and $\varepsilon$ is an independent zero-mean residual noise. One obtains the dataset

$$\boldsymbol{X}^N = \begin{bmatrix} x_1^1 & \dots & x_d^1 \\ \vdots & \ddots & \vdots \\ x_1^N & \dots & x_d^N \end{bmatrix}, \qquad \boldsymbol{Y}^N = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix}$$

and fits the ordinary least squares (OLS) estimator $\theta^N = \left(\left(\boldsymbol{X}^N\right)^\top \boldsymbol{X}^N\right)^{-1} \boldsymbol{Y}^N$, where it is worth noting that

$$\left(\boldsymbol{X}^N\right)^\top \boldsymbol{X}^N = \sum_{n=1}^N \left(x^n\right)\left(x^n\right)^\top. \tag{3.5}$$

For arbitrary $x$, the quantity $x^\top \theta^N$ is the predicted value of the expected response with $x$ as the features. One can then use the prediction to make decisions; for example, $\arg\max_{x \in \mathcal{X}} x^\top \theta^N$ will give us the set of features, among all elements of some finite or infinite set $\mathcal{X}$, that is predicted to have the highest value. In other words, a single $x$ vector represents a certain decision, with the components of $x$ describing its attributes, and we wish to identify the best decision.

However, the classical literature on this subject generally does *not* associate feature vectors with decisions and thus does not have a notion of the "best" decision or the "value" of a feature vector. Instead, the following approach is adopted. One observes that the covariance matrix of the least squares estimator $\theta^N$ is proportional to $\left(\boldsymbol{A}^N\right)^{-1}$, where

$$\boldsymbol{A}^N = \frac{1}{N}\left(\boldsymbol{X}^N\right)^\top \boldsymbol{X}^N \tag{3.6}$$

is the average information matrix. In a broad sense, this matrix quantifies our uncertainty about every possible $x$ value. That is, for any $x$, $Var\left(x^\top \theta^N\right)$ depends on $\boldsymbol{A}^N$ (and also on $x$, but the statistician does not have any preference regarding which $x$ are more important). The statistician then *designs* the data $\boldsymbol{X}^N$ in a way that makes $\boldsymbol{A}^N$ "large" or, alternately, makes $\left(\boldsymbol{A}^N\right)^{-1}$ "small," thus reducing the overall uncertainty of the predictions (but not necessarily the uncertainty at a certain fixed $x$ value). There are many possible ways to formalize what it means for $\boldsymbol{A}^N$ to be large, leading to such "alphabet-optimal" criteria (Dette, 1997) as:

- *A-optimal*: maximize $\text{tr}\left(\boldsymbol{A}^N\right)$.
- *D-optimal*: maximize $\det\left(\boldsymbol{A}^N\right)$.
- *G-optimal*: minimize $\max_{x\in\mathcal{X}} x^\top \left(\boldsymbol{A}^N\right)^{-1} x$.
- *M-optimal*: maximize $\min_j \boldsymbol{A}^N_{jj}$.

This is not an exhaustive list; for example, Goos et al. (2016) argue in favor of a different criterion called "I-optimal," which minimizes the average (rather than the maximum) variance of the prediction across the space of possible $x$. In any case, all of these criteria are only meaningful if some restrictions are imposed on $\boldsymbol{A}^N$ so that the above optimization problems are not unbounded. Often, one assumes that samples can only be collected from some finite set $y^1, \ldots, y^M \in \mathbb{R}^d$. In other words, every $x^n$ in (3.5) must correspond to one of these $M$ pre-specified feature vectors. In this case, one can rewrite (3.6) as

$$\boldsymbol{A}^N = \sum_{m=1}^{M} p_m \left(y^m\right)\left(y^m\right)^\top,$$

where $p_m = \frac{1}{N}\sum_{n=1}^{N} 1_{\{x^n = y^m\}}$ is the proportion of the total number $N$ of data points that we have assigned to sampling the $m$th data vector. The optimal design problem can now be solved using convex optimization methods. For example, the D-optimal problem can be formulated (Lu et al., 2018) using the concave objective function

$$\max \log\det\left(\sum_{m=1}^{M} p_m \left(y^m\right)\left(y^m\right)^\top\right) \tag{3.7}$$

subject to the linear constraints $\sum_{m=1}^{M} p_m = 1$ and $p \geq 0$ on the decision variable $p \in \mathbb{R}^M$. Similar approaches can be designed for the A-optimal problem (Ahipaşaoğlu, 2015) and others.

There is also a family of optimal designs based purely on the geometry of the design space (set of allowable $x$), with no explicit connection to the prediction error.

These are often called "space-filling" designs, since they seek to space out the rows of $\boldsymbol{X}^N$ uniformly in the design space. For example, Johnson et al. (1990) proposed the maximin design, obtained by solving

$$\max_{x^1,\ldots,x^N \in \mathcal{X}} \left( \min_{n \neq n'} \|x^n - x^{n'}\| \right),$$

which places each data point as far away from the others as possible. Latin hypercube designs (see, e.g., Morris & Mitchell, 1995) also fall into this category. A statistician who adopts such a design may be completely agnostic with regard to the structure of the response variable (i.e., whether it is generated by a linear model or something else) and may opt to use an interpolation model such as Gaussian process regression (Rasmussen & Williams, 2006) to construct the prediction. In this case a space-filling design will also have the effect of reducing one's overall uncertainty about the response as a function of $x$.

All of these streams of research continue to be active, and there is still no consensus on which design criterion is the "best." One can find very recent papers arguing, e.g., that A-optimal is better than D-optimal for certain problem classes (Jones et al., 2020). In some settings, some criteria may be equivalent (O'Brien & Funk, 2003). Others have turned out to be connected to learning theory: for example, the G-optimal criterion is studied by Soare et al. (2014) in the context of best-arm identification in linear bandits. Computation is also an area of active interest; see, e.g., Sagnol and Harman (2015) or Rodriguez et al. (2010) on exact computation of D-optimal and G-optimal designs, respectively.

In general, however, experimental design is solving a different problem from optimal learning. The statistician wants to estimate the regression coefficients accurately and thus focuses on reducing the variance of the OLS estimator in different ways. For us, however, it matters which value we are estimating: we do not necessarily need to reduce the variance of $x^\top \theta^N$ if $x$ itself is unimportant. We are primarily concerned with accurately distinguishing between higher- and lower-valued decisions.

At the same time, the underlying philosophy of experimental design will turn out to be quite useful to us. A problem such as (3.7) is similar, in principle, to the problem we saw in Sect. 3.1 of dividing a sample between two populations: we are pre-allocating the budget ahead of time, and the optimal allocation may turn out to be simple to implement and insightful, as in (3.4). This is different from how most learning algorithms work—they are typically implemented sequentially, so that each new decision is based on updated and more accurate information—but we will soon see that there are deep connections between static optimal designs and dynamic sequential learning methods, and that the former can provide valuable guidance for the latter.

## 3.3 The Ranking and Selection Problem

This section focuses on the *ranking and selection* (R&S) problem, a fundamental model in the study of information collection. R&S has a long history, especially in the simulation literature; many introductory tutorials can be found in the *Proceedings of the Winter Simulation Conference*, with two examples being Hong and Nelson (2009) and Chau et al. (2014). Chen et al. (2015) also provide a good overview of this research area.

Section 3.3.1 briefly describes the basic formalism of R&S. Section 3.3.2 provides a short overview of key results from large deviations theory, which are used to develop an experimental design-like approach (essentially a new optimality criterion) to the R&S problem. Section 3.3.3 illustrates these ideas using a simple example with normal distributions. Section 3.3.4 then shows how this approach can be leveraged to characterize optimal allocations of a learning budget.

### *3.3.1 Model*

Suppose that there are $m$ alternatives with unknown values $\mu_1, \ldots \mu_M$, and we wish to find $\arg\max_m \mu_m$. We can collect independent observations of the form $W_m \sim F_m$, where the distribution $F_m$ satisfies $\mathbb{E}(W_m) = \mu_m$.

As in the very first example in Sect. 3.1, we will divide $N$ samples between $M$ alternatives. Only one alternative can be sampled at a time—the main tradeoff in this problem is that allocating more samples to learn about any particular $m$ leaves fewer samples to learn about other choices. Since we will take $n \to \infty$ in our analysis, the allocation will be represented by a vector $p$ of proportions, much like in the experimental design problem from (3.7). Thus, the number of samples allocated to $m$ is approximately $N_m \approx p_m \cdot N$. For finite $N$, this number may not be integer-valued, but since we will be passing to an asymptotic regime shortly, this is not a major issue.

Given a fixed allocation $p$, we obtain $N_m$ i.i.d. draws from each distribution $F_m$ and calculate sample averages $\theta_m^N$, which are indexed by $N$ to indicate the total number of samples that have been used. Once the learning budget has been used up, our *selection* decision will be $m^{*,N} = \arg\max_m \theta_m^N$. Letting $m^* = \arg\max_m \mu_m$ denote the index of the true best alternative (which we assume to be unique), we say that an incorrect selection occurs if $m^{*,N} \neq m^*$. Just as in Sect. 3.1, we can minimize (in a certain asymptotic sense) the error probability $P\left(m^{*,N} \neq m^*\right)$ through the allocation $p$.

### 3.3.2 Large Deviations Analysis

Much of the following discussion is taken from the seminal paper by Glynn and Juneja (2004), which first formalized this approach to the R&S problem. Let $E = \left\{m^{*,N} \neq m^*\right\}$ denote the "error event," with $P(E)$ being the error probability. Observe that

$$E = \left\{\exists m \neq m^* : \theta_m^N \geq \theta_{m^*}^N\right\}.$$

That is, an incorrect selection is made if and only if there exists some suboptimal alternative $m \neq m^*$ whose sample mean is higher than that of $m^*$. It is clear that $P(E) \to 0$ as $N \to \infty$ as long as the allocation satisfies $p_m > 0$ for any $m$. The question is how quickly this convergence happens. It is fairly intuitive (and also can be proved) that, asymptotically, $P(E) \sim \max_m P(E_m)$, where

$$E_m = \left\{\theta_m^N > \theta_{m^*}^N\right\}, \qquad m = 1, \ldots, M.$$

In order to characterize the probability of falsely selecting *any* suboptimal alternative, we should examine each individual pairwise comparison between $m^*$ and some specific $m$. The probabilities $P(E_m)$ decay at different rates, and the slowest of these is the one that governs the asymptotic behavior of $P(E)$. Thus, if we can show that

$$\lim_{N\to\infty} \frac{1}{N} \log P(E_m) = -\Gamma_m, \qquad m = 1, \ldots, M, \tag{3.8}$$

with $\Gamma_m > 0$, it will automatically follow that

$$\lim_{N\to\infty} \frac{1}{N} \log P(E) = -\min_m \Gamma_m.$$

Results of the form (3.8) are known as large deviations laws and can be derived using the Gärtner-Ellis theorem (Dembo & Zeitouni, 2009). Omitting some technical nuances that in any case will not be important for the present setting, we briefly sketch out the general outline of this analysis.

Let $\{Y^n\}_{n=1}^{\infty}$ be a sequence of random vectors (not necessarily independent or identically distributed) taking values in $\mathbb{R}^d$. Denote by $\Psi^n(\gamma) = \log \mathbb{E}\left(e^{\gamma^\top Y^n}\right)$ the log of the moment-generating function of $Y^n$. Now suppose that the limit

$$\Psi(\gamma) = \lim_{n\to\infty} \frac{1}{n} \Psi^n(\gamma n) \tag{3.9}$$

of a certain scaling of $\{\Psi^n\}$ exists. Then, let

$$I(u) = \sup_{\gamma} \gamma^\top u - \Psi(\gamma)$$

be the Fenchel-Legendre transform of $\Psi$. For certain choices of $\mathcal{E} \subseteq \mathbb{R}^d$, one can then obtain rates of the form (3.8) through the result

$$\lim_{N\to\infty} \frac{1}{N} \log P\left(Y^N \in \mathcal{E}\right) = -\inf_{u\in\mathcal{E}} I(u). \tag{3.10}$$

These derivations are greatly simplified in the special case where each $Y^n$ is a sample average of $n$ i.i.d. observations from the distribution $F$. In this case, (3.9) reduces to

$$\Psi(\gamma) = \log \mathbb{E}\left(e^{\gamma W}\right), \tag{3.11}$$

where $W$ is a single sample from the distribution $F$. The rate function $I$ can then be computed directly from (3.11) without explicitly considering the scaling. If one is seeing large deviations theory for the first time, (3.11) is actually very counterintuitive, because one is used to thinking of sample averages in light of the central limit theorem—that is, one expects that they will behave like normally distributed random variables. Equation (3.11) shows that this is *not* true for error probabilities. Asymptotically, the behavior of $P\left(Y^N \in \mathcal{E}\right)$ is governed by the distribution of a single observation, as long as $\mathbb{E}(W) \notin \mathcal{E}$. To put it another way, the central limit theorem describes the rate at which the sample average converges to the population mean, but not the convergence rate of the "tail probability" of the sample average being outside a neighborhood of the population mean. The scaling (3.9) cancels out the effects of sample averaging.

In the context of R&S, $I$ can be computed in closed form for virtually any commonly used distributional family. In our context, $Y^N = \left(\theta_m^N, \theta_{m^*}^N\right)$ for some fixed $m \neq m^*$, and $\mathcal{E} = \{(u_m, u_{m^*}) : u_m \geq u_{m^*}\}$. Another substantial simplification is possible because $\theta_m^N$ and $\theta_{m^*}^N$ are independent—this is because the allocation $p$ is chosen ahead of time, before any samples are observed. Then, letting $\gamma = (\gamma_m, \gamma_{m^*})$, we have

$$\log \mathbb{E}\left(e^{\gamma^\top Y^N}\right) = \log \mathbb{E}\left(e^{\gamma_m \theta_m^N}\right) + \log \mathbb{E}\left(e^{\gamma_{m^*} \theta_{m^*}^N}\right),$$

so the logs of the moment-generating functions of the two alternatives can be scaled separately. But since both $\theta_m^N$ and $\theta_{m^*}^N$ are sample averages, one can also benefit from the simplification of (3.11). The only nuance is that, in R&S, $\theta_m^N$ is not a sample average of $N$ observations, but rather a sample average of $p_m \cdot N$ observations. This results in an extra factor $p_m$ appearing in the scaling, i.e.,

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{E}\left(e^{\gamma \cdot N \cdot \theta_m^N}\right) = p_m \log \mathbb{E}\left(e^{\frac{\gamma}{p_m} W_m}\right). \tag{3.12}$$

Thus, the shape of the rate function is still determined by the distribution $F$ of a single observation, but it is also affected by the allocation $p$. The same allocation will produce completely different convergence rates when the sampling distribution is, say, exponential as opposed to normal. But, by the same token, the same sampling distribution will produce different rates under different allocations.

### 3.3.3 Example: Normal Sampling Distributions

With these facts, let us briefly go over the case where $F$ is a $\mathcal{N}(\mu, \sigma^2)$ distribution (Example 1 of Glynn & Juneja, 2004). Then, it is easy to see that

$$\log \mathbb{E}\left(e^{\gamma W}\right) = \gamma\mu + \frac{1}{2}\gamma^2\sigma^2,$$

leading to the rate function

$$\sup_{\gamma} \gamma u - \log \mathbb{E}\left(e^{\gamma W}\right) = \frac{(u-\mu)^2}{2\sigma^2}. \tag{3.13}$$

Let us apply this result to an R&S problem with $F_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$. Using the independence of $\theta_m^N$ and $\theta_{m^*}^N$, and recalling (3.12), we obtain

$$\begin{aligned}\Psi(\gamma) &= p_m\left(\frac{\gamma_m}{p_m}\mu_m + \frac{1}{2}\frac{\gamma_m^2}{p_m^2}\sigma_m^2\right) + p_{m^*}\left(\frac{\gamma_{m^*}}{p_{m^*}}\mu_{m^*} + \frac{1}{2}\frac{\gamma_{m^*}^2}{p_{m^*}^2}\sigma_{m^*}^2\right) \\ &= \left(\gamma_m\mu_m + \frac{1}{2}\frac{\gamma_m^2}{p_m}\sigma_m^2\right) + \left(\gamma_{m^*}\mu_{m^*} + \frac{1}{2}\frac{\gamma_{m^*}^2}{p_{m^*}}\sigma_{m^*}^2\right). \end{aligned} \tag{3.14}$$

The expression in (3.14) is separable, so we can apply (3.13) to each term, whence

$$I(u_m, u_{m^*}) = \frac{1}{2}\left(p_m\frac{(u_m-\mu_m)^2}{\sigma_m^2} + p_{m^*}\frac{(u_{m^*}-\mu_{m^*})^2}{\sigma_{m^*}^2}\right). \tag{3.15}$$

We wish to study the error probability $P\left(\theta_m^N \geq \theta_{m^*}^N\right)$, so by (3.10), we must compute $\min I(u_m, u_{m^*})$ subject to the linear constraint $u_m \geq u_{m^*}$. Since $\mu_{m^*} > \mu_m$, the first term on the right-hand side of (3.15) is increasing when $u_m \geq \mu_m$, while the second term is decreasing when $u_{m^*} \leq \mu_{m^*}$. For this reason, we must have $u_m = u_{m^*}$ at optimality. It is, therefore, sufficient to minimize

$$I(u) = \frac{1}{2}\left(p_m\frac{(u-\mu_m)^2}{\sigma_m^2} + p_{m^*}\frac{(u-\mu_{m^*})^2}{\sigma_{m^*}^2}\right).$$

After a bit of algebra, we arrive at the large deviations law

$$\lim_{N\to\infty} \frac{1}{N} \log P\left(\theta_m^N \geq \theta_{m^*}^N\right) = -\frac{(\mu_m - \mu_{m^*})^2}{2\left(\frac{\sigma_m^2}{p_m} + \frac{\sigma_{m^*}^2}{p_{m^*}}\right)}, \tag{3.16}$$

which again involves an expression very similar to the two-sample test statistic from (3.1).

One can analogously derive rate exponents for non-normal distributions. Glynn and Juneja (2004) provide the derivation for Bernoulli distributions, while Gao and Gao (2016) consider exponential distributions. Chi-square distributions, which arise when we wish to identify the largest variance rather than the largest population mean, are handled in Hunter and McClosky (2016). Shin et al. (2016) considered the problem of finding largest quantiles. Very recently, Zhou and Ryzhov (2022) derived a large deviations law for the ordinary least squares estimator under normally distributed residual noise; we will return to this setting in Sect. 3.5.1. These references are left to the interested reader, and our discussion will now turn to how large deviations laws may be used to optimize allocations.

### 3.3.4 Optimal Allocations

As discussed previously, results of the form (3.8) imply that

$$\lim_{N\to\infty} \frac{1}{N} \log P(E) = -\min_m \Gamma_m(p),$$

where we have made the dependence of $\Gamma_m$ on $p$ explicit. The error probability vanishes faster when the rate exponent increases. Consequently, the best possible convergence rate is achieved by solving the optimization problem

$$\max_p \min_m \Gamma_m(p)$$

subject to the linear constraints $\sum_{m=1}^M p_m = 1$ and $p \geq 0$ on $p \in \mathbb{R}^M$. Since there are finitely many alternatives, one can use a standard technique to linearize the objective and obtain the problem

$$\begin{aligned} \max_{p,z} \quad & z \\ \text{s.t.} \quad & z \leq \Gamma_m(p), \qquad m \neq m^*, \\ & \textstyle\sum_{m=1}^M p_m = 1, \\ & p_m \geq 0, \qquad m = 1, \ldots, M. \end{aligned} \tag{3.17}$$

It can be shown that the rate exponent $\Gamma_m$ is a concave function of $p$ (for general sampling distributions), so (3.17) is a concave optimization problem. Therefore, the optimal $p$ is unique and obeys the first-order optimality conditions

$$\sum_{m \neq m^*} \frac{\partial \Gamma_m(p)/\partial p_{m^*}}{\partial \Gamma_m(p)/\partial p_m} = 1, \tag{3.18}$$

$$\Gamma_m(p) = \Gamma_{m'}(p), \qquad m, m' \neq m^*. \tag{3.19}$$

Equations (3.19) follow intuitively from the max-min objective of (3.17). Increasing $p_m$ will improve $\Gamma_m$ since the pairwise comparison between $m$ and $m^*$ becomes more accurate, but simultaneously $\Gamma_{m'}$ will become worse for other $m' \neq m^*$ since fewer samples remain for those comparisons. Thus, every pairwise comparison should have the same rate exponent at optimality. Equation (3.18) determines how large this exponent can be, as a result of the normalization constraint on $p$. We call (3.18)–(3.19) the "total" and "individual" balance conditions, respectively.

Let us see once more how the general forms of these conditions simplify in the case where $F_m$ is $\mathcal{N}\left(\mu_m, \sigma_m^2\right)$. Then, (3.18)–(3.19) become

$$\frac{p_{m^*}^2}{\sigma_{m^*}^2} = \sum_{m \neq m^*} \frac{p_m^2}{\sigma_m^2}, \tag{3.20}$$

$$\frac{(\mu_m - \mu_{m^*})^2}{\frac{\sigma_m^2}{p_m} + \frac{\sigma_{m^*}^2}{p_{m^*}}} = \frac{(\mu_{m'} - \mu_{m^*})^2}{\frac{\sigma_{m'}^2}{p_{m'}} + \frac{\sigma_{m^*}^2}{p_{m^*}}}, \qquad m, m' \neq m^*. \tag{3.21}$$

Note that, in the special case where $M = 2$, the optimality conditions reduce to (3.4), which we have already seen. Furthermore, in the case where $p_{m^*} \gg p_m$ for all $m \neq m^*$, (3.21) reduces to

$$\frac{p_m}{p_{m'}} = \frac{\sigma_m^2 (\mu_{m'} - \mu_{m^*})^2}{\sigma_{m'}^2 (\mu_m - \mu_{m^*})^2}. \tag{3.22}$$

Equation (3.22) is widely known in the simulation literature as the OCBA (Optimal Computing Budget Allocation) ratio, first derived by Chen et al. (2000) using an approximation of the error probability. An entire literature on OCBA is available; the monograph by Chen and Lee (2010) offers a summary.

None of these conditions, however, gives us a budget allocation that we can implement directly. In order to solve (3.20)–(3.21), we must know the true values $\mu_m$ (also the variances $\sigma_m^2$), but these are precisely the quantities that we are trying to learn. The standard workaround adopted in Chen and Lee (2010) and related papers is described in Fig. 3.1. For simplicity, suppose that the variances $\sigma_m^2$ are known and we only need to estimate the means. Essentially, this procedure numerically solves (3.20)–(3.21) using plug-in estimators of the unknown parameters and uses

Step 0: Collect $n_0$ samples of each alternative $m = 1, ..., M$. Set $n = n_0 M$ and compute sample averages $\theta_m^n$. Fix some positive integer $\Delta$.
Step 1: Replace $\mu_m$ in (3.20)-(3.21) by $\theta_m^n$ and let $p_m^n$ be the solution (computed using some numerical root-finding method).
Step 2: Collect $p_m^n \cdot \Delta$ samples of each alternative $m$.
Step 3: Increment $n$ by $\Delta$ and update sample averages, then return to step 1.

**Fig. 3.1** Description of sequential OCBA algorithm for normal sampling distributions

the resulting approximate proportions to allocate a portion of the budget consisting of $\Delta$ samples. This process is repeated; as $\theta_m^n \to \mu_m$, the approximate proportions also converge to the true optimal allocation.

The algorithm in Fig. 3.1 is cumbersome, as it requires us to solve a sequence of difficult root-finding problems. However, it illustrates an important concept: the theoretical characterization of the optimal solution of (3.17) is used to guide a sequential algorithm that learns this solution over time. We will see later on that the optimal allocation also underlies some algorithms that had not been designed with it in mind.

It is worth noting that, conceptually, the approach presented here is not greatly different from classical design of experiments as described in Sect. 3.2. Problem (3.17) is quite similar to (3.7), except that it uses a different optimality criterion. But, as we have seen, the literature on design of experiments also considers a wide variety of criteria. The main distinction here is that, through large deviations theory, we have obtained a criterion that explicitly depends on the value of an alternative (that is, $\Gamma_m$ depends on $\mu_m$ as well as $p_m$), a notion that is generally absent from classical design of experiments. The optimal allocation can be expressed as the solution to a *static* optimization problem, much as experimental designs are obtained from a single mathematical program. However, since we do not actually know the values of the alternatives, the optimal design begins to play a different role: instead of giving us a directly implementable course of action, it provides a goal to work toward as we gradually learn the values.

Large deviations theory can be used to characterize optimal budget allocations for problems that go beyond the R&S framework. Pasupathy et al. (2014) were an important advance in this direction, together with other related papers by Hunter and Pasupathy (2013), Zhang et al. (2016), and Applegate et al. (2020). Such problems continue to be an active area of research.

## 3.4 Sequential Algorithms

This section will discuss R&S algorithms that, on the surface, do not seem to be connected to the large deviations analysis described in Sect. 3.3. Nonetheless, such connections exist and will become apparent as the discussion progresses. Two

popular methodologies will be discussed: Sect. 3.4.1 covers value of information methods, while Sect. 3.4.2 focuses on Thompson sampling. Finally, Sect. 3.4.3 will discuss rate-balancing procedures that are more explicitly inspired by optimal designs, and Sect. 3.4.4 will discuss important nuances of how these procedures perform relative to the optimal designs themselves.

### 3.4.1 Value of Information Methods

The value of information methodology dates back to at least Gupta and Miescke (1996) and Jones et al. (1998) and is one of the most enduring and popular algorithmic concepts for R&S. This approach uses a Bayesian statistical model, in which the unknown values are modeled as random variables. Focusing on normal distributions for simplicity, let us suppose that $\mu_m \sim \mathcal{N}\left(\theta_m^0, \left(\sigma_m^0\right)^2\right)$, with $\mu_m$ independent of $\mu_{m'}$ for any $m \neq m'$. Given a sequence $\{m^n\}_{n=0}^{\infty}$ of alternatives, we observe $\left\{W_{m^n}^{n+1}\right\}_{n=0}^{\infty}$, with each $W_m^n = \mu_m + \varepsilon_m^n$ and $\varepsilon_m^n \sim \mathcal{N}\left(0, \sigma_m^2\right)$ being an independent noise term. Again, to keep the presentation simple we assume that $\sigma_m^2$ is known. Information is collected sequentially: thus, every $m^n$ may depend on the information set $\mathcal{F}^n = \left\{m^1, W_{m^1}^1, \ldots, m^{n-1}, W_{m^{n-1}}^n\right\}$. Our state of knowledge about $\mu_m$ is represented by the posterior mean and variance of this quantity given $\mathcal{F}^n$.

Under the non-informative prior $\theta_m^0 \equiv 0, \sigma_m^0 \equiv \infty$, this statistical model becomes almost identical to the one we used earlier. Let $N_m^n = \sum_{n'=0}^{n-1} 1_{\left\{m^{n'}=m\right\}}$ be the number of samples of alternative $m$ collected up to time $n$. Given $\mathcal{F}^n$, the conditional distribution of $\mu_m$ is normal with parameters

$$\theta_m^n = \frac{1}{N_m^n} \sum_{n'=0}^{n-1} W_{m^{n'}}^{n'+1} 1_{\left\{m^{n'}=m\right\}},$$

$$\sigma_m^n = \frac{\sigma_m^2}{N_m^n},$$

which are identical to the usual frequentist sample mean and its variance. In other words, the true values are estimated in exactly the same way by the Bayesian model as by the earlier frequentist one. The difference is in how the Bayesian model makes predictions: given $\mathcal{F}^n$, the posterior distribution of $\mu_m$ assigns a precise numerical quantity to the likelihood with which $\mu_m$ takes on any value.

This probabilistic prediction can be used to design sampling criteria. Perhaps the best-known of these is the expected improvement criterion of Jones et al. (1998). Letting $m^{*,n} = \arg\max_m \theta_m^n$ be the index of the alternative believed to be the best at time $n$, we compute

$$\nu_m^n = \mathbb{E}\left(\max\left\{\mu_m - \theta_{m^{*,n}}^n, 0\right\} \mid \mathcal{F}^n,\ m^n = m\right), \tag{3.23}$$

which measures the amount by which $\mu_m$ is expected (based on the most recent information) to exceed the current estimate of the highest value. The larger this quantity, the more likely it is that alternative $m$ is better than we think. We then allocate the next sample to $m^n = \arg\max_m \nu_m^n$, observe $W_{m^n}^{n+1}$, update our posterior parameters, and repeat the process.

Equation (3.23) is attractive as a sampling criterion because it can be computed in closed form, with

$$\nu_m^n = \sigma_m^n f\left(-\frac{\left|\theta_m^n - \theta_{m^{*,n}}^n\right|}{\sigma_m^n}\right), \tag{3.24}$$

and $f(z) = z\Phi(z) + \phi(z)$, with $\phi$, $\Phi$ being the standard normal density and CDF. The allocation decision at time $n$ can thus be computed very efficiently, but unlike the optimal allocations studied in Sect. 3.3, it is completely myopic, using only a rough forecast of $\mu_m$ based only on the information available at that moment. At first glance, it is difficult to see how it might be related to the optimal allocation or to the analysis we have previously developed.

Nonetheless, there is such a connection. Ryzhov (2016) showed that, for $m, m' \neq m^*$, expected improvement leads to

$$\frac{N_m^n}{N_{m'}^n} \to \frac{\sigma_m^2 (\mu_{m'} - \mu_{m^*})^2}{\sigma_{m'}^2 (\mu_m - \mu_{m^*})^2},$$

which is exactly identical to (3.22). Although the expected improvement criterion was not developed with experimental design in mind, it nonetheless provably converges to the same allocation as the OCBA approach discussed in Sect. 3.3.4. The reason for this is because the expected improvement quantity (3.23) is reduced to zero with enough samples, i.e., $\nu_m^n \to 0$ as $n \to \infty$. Since we always allocate the next measurement to the alternative with the largest expected improvement, this has the effect of forcing $\nu_m^n$ to decline to zero at the same rate across all $m$. But the declining behavior of $\nu_m^n$ is determined by the tails of the function $f$ in (3.24). In order for $\nu_m^n$ to converge at the same rate, the arguments of $f$ in (3.24) have to be approximately equal, meaning that

$$\frac{\left|\theta_m^n - \theta_{m^*}^n\right|}{\sigma_m^n} \approx \frac{\left|\theta_{m'}^n - \theta_{m^*}^n\right|}{\sigma_{m'}^n}$$

for large values of $n$ (when $m^{*,n} = m^*$). But since $\sigma_m^n = \frac{\sigma_m}{\sqrt{N_m^n}}$, this leads to the same result as in (3.22). Thus, it appears that the myopic structure of expected improvement is really another way of achieving the same goal as sequential methods that are based on optimal designs. In fact, if we can modify (3.24) so that the tails of

$f$ vanish at the same rates as the error probabilities in Sect. 3.3, the above arguments suggest that we may be able to recover the *optimal* allocation.

Recent work has shown that this is indeed the case. Salemi et al. (2014) proposed a variant of expected improvement in which (3.23) is replaced by

$$\begin{aligned}\bar{\nu}_m^n &= \mathbb{E}\left(\max\left\{\mu_m - \mu_{m^{*,n}}, 0\right\} \mid \mathcal{F}^n,\ m^n = m\right), \\ &= \sqrt{\left(\sigma_m^n\right)^2 + \left(\sigma_{m^{*,n}}^n\right)^2} f\left(-\frac{\left|\theta_m^n - \theta_{m^{*,n}}^n\right|}{\sqrt{\left(\sigma_m^n\right)^2 + \left(\sigma_{m^{*,n}}^n\right)^2}}\right). \end{aligned} \tag{3.25}$$

This version of the sampling criterion includes uncertainty in the values of both $m$ and $m^{*,n}$. From (3.25), it is clear that the argument of $f$ now behaves like the rate exponent $\Gamma_m$ that we derived for normal distributions in (3.16). Chen and Ryzhov (2019b) then integrated this criterion into a simple algorithm, described in Fig. 3.2, which is guaranteed to converge to the solution of (3.20)–(3.21) as $n \to \infty$.

Unlike the algorithm in Fig. 3.1, this procedure is trivial to implement. It does not require us to run any nonlinear optimization (or root-finding) method, and has no tunable parameters. The notion of an optimal allocation now becomes more powerful—although we cannot implement the solution to (3.17) directly, we can efficiently learn it over time. Furthermore, value of information methods are known to yield superlative practical performance even for small learning budgets, as has been repeatedly observed by Branke et al. (2007), Chick et al. (2010), Han et al. (2016), and others. Thus, a static allocation derived through experimental design provides useful guidance for a sequential method that also performs well for small sampling budgets.

### 3.4.2 Thompson Sampling

The idea behind Thompson sampling dates back to Thompson (1933), but this method has enjoyed a recent surge in popularity due to the seminal paper of Russo

Step 0: Let $n = 0$, initialize $\theta^0 \equiv 0$, $\sigma^0 \equiv \infty$, $N^0 \equiv 0$.

Step 1: Check whether

$$\left(\frac{N_{m^{*,n}}^n}{\sigma_{m^{*,n}}}\right)^2 < \sum_{m \neq m^{*,n}} \left(\frac{N_m^n}{\sigma_m}\right)^2. \tag{3.26}$$

If (3.26) holds, assign $m^n = m^{*,n}$. Otherwise, assign $m^n = \arg\max_{m \neq m^{*,n}} \bar{\nu}_m^n$ using the formula in (3.25).

Step 2: Observe $W_{m^n}^{n+1}$, update $\theta^n$, $\sigma^n$, $N^n$. Increment $n$ by 1 and return to Step 1.

**Fig. 3.2** Modified expected improvement algorithm of Chen and Ryzhov (2019b)

and Van Roy (2014). Like value of information, Thompson sampling is based on Bayesian statistics, so we can carry over the setting of Sect. 3.4.1 unchanged.

Rather than taking expectations, as in (3.23), we adopt a randomized approach. Given $\mathcal{F}^n$, let $\hat{\mu}_m^n \sim \mathcal{N}\left(\theta_m^n, \left(\sigma_m^n\right)^2\right)$ be a sample from the current posterior distribution of $\mu_m$. The next allocation decision is then made using

$$m^n = \arg\max_m \hat{\mu}_m^n.$$

We deliberately introduce a certain amount of noise into our decision; however, all else being equal, alternatives with larger $\theta_m^n$ and/or larger $\sigma_m^n$ will be more likely to be sampled. Value of information methods have much the same regularity, since (3.23) and similar criteria also favor alternatives with better estimated values (because they appear to be good) or higher uncertainty (because they are more likely to be better than we think). Thompson sampling has the advantage of being very user-friendly, because it is often much easier to sample from a posterior distribution than it is to take expectations over it.

As $n$ increases, the posterior distribution of alternative $m$ will concentrate around $\mu_m$. Supposing that $\mu_m > \mu_{m'}$, the event that $\hat{\mu}_m^n \leq \hat{\mu}_{m'}^n$ again occurs on the tail of the joint distribution of $\left(\hat{\mu}_m^n, \hat{\mu}_{m'}^n\right)$. This again suggests a connection with the previous large deviations-theoretic analysis. Indeed, Russo (2020) showed that a modified Thompson sampling procedure ("top-two Thompson sampling") also provably converges to the optimal solution of (3.17). This algorithm is given in Fig. 3.3.

Like the modified expected improvement algorithm in Fig. 3.2, top-two Thompson sampling introduces special logic, not present in the original Thompson sampling procedure, to decide whether to sample $m^{*,n}$. In Fig. 3.3, this is done by simply flipping a biased coin with some fixed probability $\rho$. If we decide not to sample $m^{*,n}$, we can then use Thompson sampling (or value of information, in Fig. 3.2) to choose among the suboptimal alternatives. The main difference between the algorithms is that, in Fig. 3.2, the decision to sample $m^{*,n}$ was automated using condition (3.26), whereas in top-two Thompson sampling it is necessary to prespecify $\rho$. The algorithm will then converge to the solution of (3.20)–(3.21) if this parameter is chosen correctly, but it requires tuning.

---

Step 0: Let $n = 0$, initialize $\theta^0 \equiv 0$, $\sigma^0 \equiv \infty$, $N^0 \equiv 0$.
Step 1: With probability $\rho$, let $m^n = m^{*,n}$. With probability $1 - \rho$, do the following:

Step 1a: Generate $\hat{\mu}_m \sim \mathcal{N}\left(\theta_m^n, \left(\sigma_m^n\right)^2\right)$ for all $m$. Let $\hat{m} = \arg\max_m \hat{\mu}_m$.
Step 1b: If $\hat{m} \neq m^{*,n}$, let $m^n = \hat{m}$ and continue. Otherwise, return to Step 1a.

Step 2: Observe $W_{m^n}^{n+1}$, update $\theta^n$, $\sigma^n$, $N^n$. Increment $n$ by 1 and return to Step 1.

---

**Fig. 3.3** Top-two Thompson sampling algorithm of Russo (2020)

Neither expected improvement nor Thompson sampling is able to learn the optimal allocation in its original, unmodified version. Both criteria are effective in choosing between suboptimal alternatives, but in both cases, additional logic is needed to decide between $m^{*,n}$ and some $m \neq m^{*,n}$. This reflects the fact that, in the original optimal design problem, a separate total balance condition (3.18) has to be satisfied in addition to the individual balance conditions (3.19). Essentially, the original versions of both sequential algorithms are able to satisfy (3.19), but modifications are needed in order to handle (3.18).

### 3.4.3 *Rate-Balancing Methods*

Having now seen two completely different methodologies that both arrive at the same destination (despite starting from very different origins), we might ask if any of these criteria—value of information, Thompson sampling, or something else—is really necessary. If we are to end up at the optimal design, we can reach it more easily by reverse-engineering (3.18)–(3.19) directly.

For normal sampling distributions, the way to do this is already suggested by the structure of Fig. 3.2. Indeed, Shin et al. (2018) proposed precisely such an algorithm. At time $n$, one first checks (3.26), exactly as in Fig. 3.2. If this inequality holds, we assign $m^n = m^{*,n}$ as before. If the inequality does not hold, we assign

$$m^n = \arg\min_m \frac{\left(\theta_m^n - \theta_{m^{*,n}}^n\right)^2}{\frac{\sigma_m^2}{N_m^n} + \frac{\sigma_{m^{*,n}}^2}{N_{m^{*,n}}^n}}. \tag{3.27}$$

The function $f$ in (3.23) and (3.25) is monotonic, so there is no real difference between maximizing $f$ and minimizing its argument—which, again, is none other than the two-sample test statistic for comparing the values of $m$ and $m^{*,n}$. As $n$ increases, the value of this statistic also tends to increase, so by choosing $m$ for which this statistic has the smallest value, we can ensure that all of the statistics increase at the same rate, thus satisfying (3.21) asymptotically. Condition (3.26) is needed to handle (3.20).

Chen and Ryzhov (2019a) explained how this concept could be used to solve (3.18)–(3.19), the general form of the optimality conditions. We now make the dependence of $\Gamma_m$ on the population means explicit: let $\Gamma_m(p\,;\,\theta)$ be the value of the $m$th rate exponent under allocation $p$ and with $\theta$ standing in for the true values. Then, $\Gamma_m(p\,;\,\mu)$ is the true rate exponent, and $\Gamma_m(p\,;\,\theta^n)$ uses plug-in estimates of the population means. As shown in Fig. 3.4, we first use condition (3.28), analogous to (3.26) in the normal case, to determine whether to sample $m^{*,n}$. If we do not do so, we then use (3.29), by analogy with (3.27), as the criterion for selecting a suboptimal alternative. Chen and Ryzhov (2022) prove that this algorithm always learns the solution to (3.18)–(3.19), without tuning.

Step 0: Let $n = 0$, initialize $\theta^0 \equiv 0$, $N^0 \equiv 0$, set arbitrary $p_m^0 > 0$ with $\sum_m p_m^0 = 1$.
Step 1: Check whether

$$\sum_{m \neq m^*} \frac{\partial \Gamma_m \left(p^n ; \theta^n\right) / \partial p_{m^{*,n}}}{\partial \Gamma_m \left(p^n ; \theta^n\right) / \partial p_m} > 1. \tag{3.28}$$

If (3.28) holds, assign $m^n = m^{*,n}$. Otherwise, assign

$$m^n = \arg \min_{m \neq m^{*,n}} \Gamma_m \left(p^n ; \theta^n\right). \tag{3.29}$$

Step 2: Observe $W_{m^n}^{n+1}$, update $\theta^n$, $N^n$, compute $p^n$ from $N^n$. Increment $n$ by 1 and return to Step 1.

**Fig. 3.4** Balancing Optimal Large Deviations (BOLD) algorithm of Chen and Ryzhov (2019a)

One limitation of this approach (which, however, it shares with the vast majority of existing algorithms for this problem) is that it requires us to know the distributional family of the samples. This is necessary in order to be able to evaluate $\Gamma_m$ and its partial derivatives. If the distributional family is unknown, the problem becomes far more difficult. Gao et al. (2017) sketches out a similar algorithm, based on an estimator of the moment-generating function described in Glynn and Juneja (2004), that potentially could be applied in a setting where no information about the distributional family is available, but this paper does not give a convergence proof. Conversely, Russo (2020) handles general distributional families, but this comes at the cost of having to tune a parameter. Regardless of the theoretical issues, however, it is not clear that any algorithm would be practical in a setting where one must store and update empirical estimators of the sampling distribution (or its moment-generating functions). Even an algorithm like Thompson sampling, which is among the easiest to run, would require a complicated Markov chain Monte Carlo model to store the posterior distribution. Most R&S algorithms that are used in practice, such as OCBA (Lin et al., 2013), simply assume normal distributions.

In any case, the preceding discussion shows that we are now completely free of any need to solve sequences of convex programs or root-finding problems. We can first use the philosophy of design of experiments to derive an optimal allocation, and then construct a sequential procedure along the lines of Fig. 3.4 to adaptively learn it over time. Sections 3.4.1 and 3.4.2 show that, essentially, sequential methods are just trying to learn this allocation in different ways, which lends support to the idea of cutting out the middleman and simply learning it directly.

### *3.4.4 Discussion*

The theoretical framework in Sect. 3.3.2 hinges on the assumption that the allocation $p$ is static (pre-specified). Only then is the log of the moment-generating function of

$\left(\theta_m^N, \theta_{m^*}^N\right)$ separable in $m$ and $m^*$, leading to the exponential convergence rate $e^{-\Gamma \cdot N}$ for the error probability $P(E)$. One naturally wonders if this result is preserved under a sequential algorithm that only learns the optimal $p$ asymptotically. Under such an algorithm, the sample means are no longer independent because the decision to sample $m$ at time $n$ is based on all of the information in $\mathcal{F}^n$, so such a rate cannot be straightforwardly obtained from the arguments we have presented.

In fact, it appears that exponential convergence is lost when we pass from a static to a dynamic allocation. Wu and Zhou (2018) show examples where an sequential OCBA-like allocation leads to polynomial, rather than exponential, convergence. Nonetheless, the optimal static allocation still plays an important role in characterizing the performance of a dynamic procedure. Qin et al. (2017) and Russo (2020) find that, while convergence to the optimal $p$ is not *sufficient* for optimal performance in a sequential setting, it is *necessary*. Thus, whatever limitations the design of experiments approach to R&S may have, the optimal static allocation obtained through this approach continues to underlie virtually all of the state-of-the-art algorithmic work on this problem.

## 3.5 Recent Advances

In this section, we discuss two examples of very recent research where the concepts we presented earlier are used outside of R&S. Section 3.5.1 describes a new large deviations analysis of linear regression models, thus returning to the classical setting of design of experiments. Section 3.5.2 describes a recently proposed idea for budget allocation in approximate dynamic programming.

### *3.5.1 A New Optimal Design for Linear Regression*

Let us return to the setting of Sect. 3.2, but now, let us interpret the expected response $\beta^\top x$ as the "value" of the feature vector $x$. We suppose that higher values are better, introducing a notion of priority into our optimal design. We are no longer interested in reducing the variance of every prediction uniformly—we only care about accurately identifying the "best" feature vector out of some set of interest. We will assume that the residual noise is i.i.d. $\mathcal{N}\left(0, \sigma^2\right)$, the most classical OLS setting.

The following discussion is a summary of the work by Chen and Ryzhov (2022). We make the crucial assumption that $\boldsymbol{A}^N \to \boldsymbol{A}$, where $\boldsymbol{A}^N$ is as in (3.6), and $\boldsymbol{A}$ is a symmetric positive definite matrix. This condition is sufficient for consistency of the OLS estimator $\theta^N$ (Lai & Wei, 1982), and can be viewed as a "law of large numbers" for the sequence $\{x^n\}$ of data vectors. In the language of Sect. 3.3, this condition is analogous to requiring $p_m > 0$ for all $m$. We treat $\{x^n\}$ as a deterministic sequence, similarly to how Sect. 3.3 views $p$ as a fixed, deterministic vector. One

can equivalently view it as a sequence of random vectors sampled independently from a distribution with $\mathbb{E}\left(xx^\top\right) = \boldsymbol{A}$. All of the following results will also hold under this interpretation, as long as this sampling distribution is independent of the observations $\{y^n\}$.

With these assumptions, Zhou and Ryzhov (2021) derive the rate function of $\theta^N$ as

$$I(u) = \frac{1}{2\sigma^2}(u-\beta)^\top \boldsymbol{A}(u-\beta).$$

Recall that the asymptotic behavior of a probability $P\left(\theta^N \in \mathcal{E}\right)$ can be characterized by minimizing $I(u)$ over $u \in \mathcal{E}$. In this setting, the error event is constructed as follows. Let $x^* \in \mathbb{R}^d$ be a fixed vector representing some sort of "reference," relative to which other choices of $x$ are evaluated. It may be that $x^*$ is the solution to some optimization problem, but we will not model any such problem explicitly. We are only concerned with correctly identifying $x^*$ relative to other $x$ for which $\beta^\top(x^* - x) > 0$, that is, these $x$ have lower values than $x^*$.

Letting $v = x^* - x$, we can define $\mathcal{E}_v = \left\{u : u^\top v \leq 0\right\}$ to be the set of all possible values of $\theta^N$ that lead us to falsely identify $x$ as being higher-valued than $x^*$. For any such $v$, we proceed along the lines of Sect. 3.3.2 and obtain the large deviations law

$$\lim_{N\to\infty} \frac{1}{N} \log P\left(\theta^N \in \mathcal{E}_v\right) = -\frac{1}{2\sigma^2} \frac{\left(\beta^\top v\right)^2}{v^\top \boldsymbol{A}^{-1} v},$$

which is readily seen to be a generalization of (3.16). Note that the rate exponent is invariant with respect to the magnitude of $v$, so we can restrict ourselves to the unit sphere $\|v\| = 1$ without loss of generality. By analogy with Sect. 3.3.4, one can define an optimal design as the solution to the problem

$$\max_{\boldsymbol{A}:\mathrm{tr}(\boldsymbol{A})=1} \min_{v\in V_\delta} \frac{\left(\beta^\top v\right)^2}{v^\top \boldsymbol{A}^{-1} v}, \tag{3.30}$$

where the design itself is completely characterized by the limiting matrix $\boldsymbol{A}$, and $V_\delta = \left\{v : \beta^\top v \geq \delta,\ \|v\| = 1\right\}$. The goal is to maximize the smallest rate exponent among all possible $v$. In a continuous design space, however, it is possible to find $x$ whose value is arbitrarily close to $x^*$, so we introduce a threshold $\delta$ to make (3.30) well-defined. The constraint on the trace of $\boldsymbol{A}$ is likewise imposed to normalize the problem, so that we are not able to make $\boldsymbol{A}$ arbitrarily large.

Since $\boldsymbol{A}$ is positive definite, we can write

$$\boldsymbol{A} = \sum_{j=1}^{d} p_j \zeta_j \zeta_j^\top,$$

where $(\zeta_1, \ldots, \zeta_d)$ comprise an orthonormal basis for $\mathbb{R}^d$. Zhou and Ryzhov (2021) characterizes the optimal $\boldsymbol{A}$ as follows. First, we set $\zeta_1 = \beta$, which determines the remaining basis vectors up to multiplication by $\pm 1$. We then let

$$p_1 = \frac{\sqrt{(d-1)\,\Delta}}{(d-1) + \sqrt{(d-1)\,\Delta}}, \tag{3.31}$$

$$p_j = \frac{1}{(d-1) + \sqrt{(d-1)\,\Delta}}, \qquad j = 2, \ldots, d, \tag{3.32}$$

where $\Delta = \frac{\delta^2}{1-\delta^2}$. Thus, if we view $\boldsymbol{A}$ as an expected value, the optimal design can be viewed as assigning a proportion $p_j$ of the learning budget to the basis vector $\zeta_j$.

It is interesting to compare this design with the classical D-optimal method. When $\text{tr}\,(A) = 1$, a D-optimal $\boldsymbol{A}$ matrix can easily be obtained by sampling the data from a uniform distribution on the unit sphere. The large deviations-based design, however, is *almost* uniform, but we sample less often along the direction $\beta$. This offers a clean illustration of how the introduction of the notion of the value of $x$ changes the priority with which we wish to learn about different $x$ vectors. It is also worth noting that, in (3.30), the denominator of the rate exponent is connected to the variance of prediction, and is essentially the G-optimal design criterion (a connection also made by Fiez et al., 2019). However, the vector $\beta$, which determines the value of $x$, is present in the numerator, which again shows how standard optimal design concepts are modified when the notion of value is introduced.

Another interesting and unusual insight is that neither the convergence rate of the error probability nor the optimal design (3.31)–(3.32) depends on $x^*$ in any way. The convergence rate is only affected by the gap $x^* - x$, not by $x^*$ itself. In a manner of speaking, our design provides the same amount of information about any $(x, x^*)$ pair with the same $x^* - x$ value. This is quite different from R&S, where we had a separate optimality condition (3.18) governing the proportion of the learning budget to assign to alternative $m^*$. In order to satisfy this condition, we also had to identify $m^*$. The sequential methods discussed in Sect. 3.4 all replace $m^*$ by $m^{*,n}$, the alternative believed to be the best at time $n$. This introduces additional error into finite-time performance, as our approximate solution of (3.18)–(3.19) will be completely wrong if $m^{*,n} \neq m^*$.

In linear regression, however, this issue never arises because we do not even need to know what $x^*$ is. We do not need to assign any part of the learning budget to sampling $x^*$ directly. Instead of allocating the budget to different $x$ values, we instead divide it between the basis vectors $(\beta, \zeta_2, \ldots, \zeta_d)$. We handle the continuous design space by refocusing the problem around the finite set of basis vectors. In the optimal design, the only unknown quantity is $\beta$ itself—the probabilities $p_j$ are given in closed form, and the other basis vectors are easily found from $\beta$. Thus, the optimal design becomes exceptionally easy to implement sequentially, using $\theta^N$ in place of $\beta$, completing an orthonormal basis from $\theta^N$, and randomly choosing a basis vector according to (3.31)–(3.32).

### 3.5.2 Optimal Budget Allocation in Approximate Dynamic Programming

A very interesting perspective on OCBA was recently put forth by Zhu et al. (2019) in the context of approximate dynamic programming. We give a brief description of this research, which connects optimal learning, optimal design, and reinforcement learning in novel ways.

Consider a classical Markov decision process model (Puterman, 2014) with finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, and transition probabilities $P\left(s' \mid s, a\right)$ for $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. Let $C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be a reward function. The goal in dynamic programming is to solve

$$\sup_{\pi \in \Pi} \sum_{n=0}^{\infty} \gamma^n C\left(s^n, \pi\left(s^n\right)\right),$$

where $s^0$ and $0 < \gamma < 1$ are given, $\Pi$ is the space of functions $\pi$ which map $s \in \mathcal{S}$ to $\pi\left(s\right) \in \mathcal{A}$, and $s^{n+1} = s$ with probability $P\left(s \mid s^n, \pi\left(s^n\right)\right)$ for all $s \in \mathcal{S}$ and all $n > 0$. It is well-known that the optimal policy $\pi^*$ satisfies

$$\pi^*\left(s\right) = \arg\max_{a \in \mathcal{A}} C\left(s, a\right) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) V\left(s'\right),$$

where $V$ is the unique solution to Bellman's equation:

$$V\left(s\right) = \max_{a \in \mathcal{A}} C\left(s, a\right) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) V\left(s'\right). \tag{3.33}$$

In virtually any practical application, the sum over transition probabilities in (3.33) is intractable. Often, the transition probabilities themselves are unknown, though it is possible to sample from the transition distribution in a black-box fashion. In such situations, $V$ may be learned asymptotically using the Q-learning algorithm (Tsitsiklis, 1994). We first define

$$\begin{aligned} Q\left(s, a\right) &= C\left(s, a\right) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) V\left(s'\right) \\ &= C\left(s, a\right) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \max_{a' \in \mathcal{A}} Q\left(s', a'\right). \end{aligned} \tag{3.34}$$

Suppose that we are given some approximation $\bar{Q}$ of $Q$, as well as some fixed $(s, a)$. We do not know the transition probabilities, but can simulate a new state $\hat{s}'$ according to the distribution $P\left(\hat{s}' = s'\right) = P\left(s' \mid s, a\right)$ for all $s' \in \mathcal{S}$. Then, the quantity

Step 0: Let $n = 0$, initialize $\bar{Q}^0$, choose a suitable stepsize sequence $\{\alpha_n\}$.
Step 1: Generate $(s^n, a^n)$ from some suitable distribution.
Step 2: Generate $\hat{s}$ according to the distribution $P(\hat{s} = s) = P(s \mid s^n, a^n)$.
Step 3: Compute

$$\hat{q}^n = C(s^n, a^n) + \gamma \max_{a \in \mathcal{A}} \bar{Q}^n(\hat{s}, a).$$

Step 4: Update

$$\bar{Q}^{n+1}(s^n, a^n) = (1 - \alpha_n) Q^n(s^n, a^n) + \alpha_n \hat{q}^n.$$

Step 5: Increment $n$ by 1 and return to Step 1.

**Fig. 3.5** Basic Q-learning algorithm

$$\hat{q} = C(s, a) + \gamma \max_{a' \in \mathcal{A}} \bar{Q}(\hat{s}', a') \tag{3.35}$$

can be viewed as an *approximate* observation of $Q(s, a)$. If we can collect many such observations and average them, this can be viewed as a Monte Carlo estimator of the expected value over the transition distribution in (3.34). However, the estimator is biased because, in (3.34), this expectation is of $Q(s', a')$, and $Q$ is precisely what we wish to find. We use $\bar{Q}$ as a stand-in for $Q$, for lack of anything better. Through the max operator in (3.35), however, we believe that $\hat{q}$ will provide us with useful information about $Q(s, a)$ that can be averaged in with our old estimate $\bar{Q}(s, a)$.

Figure 3.5 formally states this algorithm. In every iteration, Steps 2–4 describe the process we have just discussed. Given $(s^n, a^n)$, we simulate a "downstream" state $\hat{s}$, compute the approximate (biased!) observation $\hat{q}^n$, and average it in with an existing approximation $\bar{Q}^n(s^n, a^n)$ using the stepsize $\alpha_n$, which is chosen to satisfy the usual conditions imposed on stochastic approximation methods (Kushner & Yin, 2003). The profound insight of Tsitsiklis (1994) is that the bias in the observations is attenuated over time, leading to $\bar{Q}^n \to Q$.

It is important to note that, once $\hat{q}^n$ has been computed, the simulated state $\hat{s}$ has served its purpose. In particular, the next state $s^{n+1}$ that we visit need not be the state $\hat{s}$ generated in the previous iteration. We can discard $\hat{s}$ and generate $(s^{n+1}, a^{n+1})$ from some completely unrelated distribution; the results of Tsitsiklis (1994) hold as long as every state-action pair $(s, a)$ is visited infinitely often. Potentially, we can view the state-action pairs as "alternatives" in an R&S-like problem, which leads to the question of allocating a learning budget (i.e., deciding how often we visit $(s, a)$ over time).

Zhu et al. (2019) approach this problem by first deriving a central limit theorem on the approximation $\bar{Q}^n$. It is shown that $\sqrt{n}(\bar{Q}^n - Q) \overset{d}{\Rightarrow} \mathcal{N}(0, \Sigma)$, where the covariance matrix $\Sigma$ depends in a complicated way on the proportions of iterations in which we observe certain states, actions, or state-action pairs. By extending the analysis from Sects. 3.3.3–3.3.4, one can choose these proportions in a way that minimizes the probability that $\bar{Q}^n(s, a) \geq \bar{Q}^n(s, \pi^*(s))$ for $a \neq$

$\pi^*(s)$. In other words, the error event here is the event that we falsely identify action $a$ as being better than the optimal action. Although, as we have discussed, the central limit theorem may not give an accurate picture of the asymptotic convergence rates of tail probabilities, the asymptotic normality established in Zhu et al. (2019) offers a way to make the problem tractable; as we have mentioned, most practical implementations of OCBA or other R&S policies tend to assume normal distributions for ease of computation. In this way, the concepts of optimal design are finding new applications far beyond the domain that originally motivated their development.

## 3.6 Conclusion

We hope that we have successfully made the case that optimal learning and optimal design are much more closely connected than might seem at first glance. Even though optimal learning is "sequential" while optimal design is "static," in fact a certain kind of optimal design can be viewed as the end goal for virtually any rigorous sequential method. The principles of design of experiments can also serve as a starting point for a learning problem—if one can characterize the optimal design, one can then construct a sequential method to learn it.

Section 3.5 has only given a very brief glimpse of the various research directions for this area. In Sect. 3.5.1, we saw how ideas from ranking and selection (specifically, the notion of value used to compare alternatives, and the ensuing definition of error events) can be brought back into the most classical setting of the design of experiments literature. There are many more opportunities along these lines, a major one being the development of optimal designs for continuous black-box optimization using Gaussian process regression. Some ideas from design of experiments, such as D-optimal designs, have been extended to Gaussian process models (Harari & Steinberg, 2014), but the focus of all this work has remained on uncertainty reduction or geometric space-filling. In many applications of Gaussian process regression, however (such as hyperparameter tuning), the goal is not merely to accurately interpolate the data but also to identify an optimal solution. The notion of an "optimal design" in such a context has yet to be characterized.

An important direction for further work is to remove the need for distributional assumptions, which we had imposed in Sect. 3.4.3. Essentially, one would have to estimate and simultaneously balance the large deviations rate functions based purely on samples. At the heart of this problem is the challenge of efficiently estimating moment-generating functions, which is currently unaddressed. Of course, parametric methods would likely continue to be widely used, due to their reduced computational cost and ease of implementation.

## References

Ahipaşaoğlu, S. D. (2015). A first-order algorithm for the A-optimal experimental design problem: A mathematical programming approach. *Statistics and Computing, 25*(6), 1113–1127.

Applegate, E. A., Feldman, G., Hunter, S. R., & Pasupathy, R. (2020). Multi-objective ranking and selection: Optimal sampling laws and tractable approximations via SCORE. *Journal of Simulation, 14*(1), 21–40.

Branke, J., Chick, S. E., & Schmidt, C. (2007). Selecting a selection procedure. *Management Science, 53*(12), 1916–1932.

Chau, M., Fu, M. C., Qu, H., & Ryzhov, I. O. (2014). Simulation optimization: a tutorial overview and recent developments in gradient-based methods. In A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, & J. A. Miller (Eds.), *Proceedings of the 2014 Winter Simulation Conference* (pp. 21–35).

Chen, C. H., Chick, S. E., Lee, L. H., & Pujowidianto, N. A. (2015). Ranking and selection: Efficient simulation budget allocation. In M. C. Fu (Ed.), *Handbook of simulation optimization* (pp. 45–80). Springer.

Chen, C. H., & Lee, L. H. (2010). *Stochastic simulation optimization: An optimal computing budget allocation*. World Scientific.

Chen, C. H., Lin, J., Yücesan, E., & Chick, S. E. (2000). Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems, 10*(3), 251–270.

Chen, Y., & Ryzhov, I. O. (2019a). Balancing optimal large deviations in ranking and selection. In N. Mustafee, K. H. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, & Y. J. Son (Eds.), *Proceedings of the 2019 Winter Simulation Conference* (pp. 3368–3379).

Chen, Y., & Ryzhov, I. O. (2019b). Complete expected improvement converges to an optimal budget allocation. *Advances in Applied Probability, 51*(1), 209–235.

Chen, Y. & Ryzhov, I. O. (2022). *Balancing optimal large deviations in sequential selection*. Management Science (to appear).

Chick, S. E., Branke, J., & Schmidt, C. (2010). Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing, 22*(1), 71–80.

Dannenberg, O., Dette, H., & Munk, A. (1994). An extension of Welch's approximate t-solution to comparative bioequivalence trials. *Biometrika, 81*(1), 91–101.

Dembo, A., & Zeitouni, O. (2009). *Large Deviations Techniques and Applications* (2nd ed.). Springer.

Dette, H. (1997). Designing experiments with respect to 'standardized' optimality criteria. *Journal of the Royal Statistical Society, B59*(1), 97–110.

Eitrich, T., & Lang, B. (2006). Efficient optimization of support vector machine learning parameters for unbalanced datasets. *Journal of Computational and Applied Mathematics, 196*(2), 425–436.

Fiez, T., Jain, L., Jamieson, K. G., & Ratliff, L. (2019). Sequential experimental design for transductive linear bandits. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32, pp. 10667–10677).

Gao, F., & Gao, S. (2016). Optimal computing budget allocation with exponential underlying distribution. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, & S. E. Chick (Eds.), *Proceedings of the 2016 Winter Simulation Conference* (pp. 682–689).

Gao, S., Chen, W., & Shi, L. (2017). A new budget allocation framework for the expected opportunity cost. *Operations Research, 65*(3), 787–803.

Garivier, A., & Kaufmann, E. (2016). Optimal best arm identification with fixed confidence. In V. Feldman, A. Rakhlin, & O. Shamir (Eds.), *Proceedings of the 29th Annual Conference on Learning Theory* (pp. 998–1027).

Glynn, P. W., & Juneja, S. (2004). A large deviations perspective on ordinal optimization. In R. Ingalls, M. D. Rossetti, J. S. Smith, & B. A. Peters (Eds.), *Proceedings of the 2004 Winter Simulation Conference* (pp. 577–585).

Goos, P., Jones, B., & Syafitri, U. (2016). I-optimal design of mixture experiments. *Journal of the American Statistical Association, 111*(514), 899–911.

Gupta, S. S., & Miescke, K. J. (1996). Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference, 54*(2), 229–244.

Han, B., Ryzhov, I. O., & Defourny, B. (2016). Optimal learning in linear regression with combinatorial feature selection. *INFORMS Journal on Computing, 28*(4), 721–735.

Harari, O., & Steinberg, D. M. (2014). Optimal designs for Gaussian process models via spectral decomposition. *Journal of Statistical Planning and Inference, 154*, 87–101.

Hong, L. J., & Nelson, B. L. (2009). A brief introduction to optimization via simulation. In M. Rosetti, R. Hill, B. Johansson, A. Dunkin, & R. Ingalls (Eds.), *Proceedings of the 2009 Winter Simulation Conference* (pp. 75–85).

Hunter, S. R., & McClosky, B. (2016). Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. *IIE Transactions, 48*(6), 565–578.

Hunter, S. R., & Pasupathy, R. (2013). Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing, 25*(3), 527–542.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference, 26*(2), 131–148.

Jones, B., Allen-Moyer, K., & Goos, P. (2020). A-optimal versus D-optimal design of screening experiments. *Journal of Quality Technology, 53*(4), 369–382.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization, 13*(4), 455–492.

Kiefer, J. (1971). The role of symmetry and approximation in exact design optimality. In S. S. Gupta, & J. Yackel (Eds.), *Statistical decision theory and related topics* (pp. 109–118).

Kushner, H., & Yin, G. (2003). *Stochastic approximation and recursive algorithms and applications* (2nd ed.). Springer Science and Business Media.

Lai, T. L., & Wei, C. Z. (1982). Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics, 10*(1), 154–166.

Lin, J. T., Chen, C. M., Chiu, C. C., & Fang, H. Y. (2013). Simulation optimization with PSO and OCBA for semiconductor back-end assembly. *Journal of Industrial and Production Engineering, 30*(7), 452–460.

Lu, H., Freund, R. M., & Nesterov, Y. (2018). Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization, 28*(1), 333–354.

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference, 43*(3), 381–402.

Nasrollahzadeh, A. A., & Khademi, A. (2020). Optimal stopping of adaptive dose-finding trials. *Service Science, 12*(2–3), 80–99.

O'Brien, T. E., & Funk, G. M. (2003). A gentle introduction to optimal design for regression models. *The American Statistician 57*(4), 265–267.

Pasupathy, R., Hunter, S. R., Pujowidianto, N. A., Lee, L. H., & Chen, C. H. (2014). Stochastically constrained ranking and selection via SCORE. *ACM Transactions on Modeling and Computer Simulation, 25*(1), 1:1–1:26.

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley.

Qin, C., Klabjan, D., & Russo, D. (2017). Improving the expected improvement algorithm. In: I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates.

Qu, H., Ryzhov, I. O., Fu, M. C., & Ding, Z. (2015). Sequential selection with unknown correlation structures. *Operations Research, 63*(4), 931–948.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Rodriguez, M., Jones, B., Borror, C. M., & Montgomery, D. C. (2010). Generating and assessing exact G-optimal designs. *Journal of Quality Technology, 42*(1), 3–20.

Russo, D. (2020). Simple Bayesian algorithms for best-arm identification. *Operations Research, 68*(6), 1625–1647.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243.

Ryzhov, I. O. (2016). On the convergence rates of expected improvement methods. *Operations Research, 64*(6), 1515–1528.

Sagnol, G., & Harman, R. (2015). Computing exact D-optimal designs by mixed integer second-order cone programming. *The Annals of Statistics, 43*(5), 2198–2224.

Salemi, P., Nelson, B. L., & Staum, J. (2014). Discrete optimization via simulation using Gaussian Markov random fields. In A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, & J. A. Miller (Eds.), *Proceedings of the 2014 Winter Simulation Conference* (pp. 3809–3820).

Shin, D., Broadie, M., & Zeevi, A. (2016). Tractable sampling strategies for quantile-based ordinal optimization. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, & S. E. Chick (Eds.), *Proceedings of the 2016 Winter Simulation Conference* (pp. 847–858).

Shin, D., Broadie, M., & Zeevi, A. (2018). Tractable sampling strategies for ordinal optimization. *Operations Research, 66*(6), 1693–1712.

Soare, M., Lazaric, A., & Munos, R. (2014). Best-arm identification in linear bandits. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27, pp. 828–836). Curran Associates.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3–4), 285–294.

Tsitsiklis, J. N. (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning, 16*(3), 185–202.

Wu, D., & Zhou, E. (2018). Analyzing and provably improving fixed budget ranking and selection algorithms. *Preprint arXiv:181112183*.

Zhang, S., Lee, L. H., Chew, E. P., Xu, J., & Chen, C. H. (2016). A simulation budget allocation procedure for enhancing the efficiency of optimal subset selection. *IEEE Transactions on Automatic Control, 61*(1), 62–75.

Zhou, J., & Ryzhov, I. O. (2021). *A new rate-optimal design for linear regression*. Technical Report, University of Maryland.

Zhou, J. & Ryzhov, I. O. (2022). *A new rate-optimal sampling allocation for linear belief models*. Operations Research (to appear).

Zhu, Y., Dong, J., & Lam, H. (2019). Efficient inference and exploration for reinforcement learning. *Preprint arXiv:191005471*.

# Part II
# Price Optimization

# Chapter 4
# Dynamic Pricing with Demand Learning: Emerging Topics and State of the Art


**Arnoud V. den Boer and Nuri Bora Keskin**

## 4.1 Introduction

Trying to determine the "right" or "optimal" price for a product is an activity that probably exists since the time that mankind started to engage in trading commodities. More recently—that is, only a few centuries ago—people started to document in some detail how product quantity correlated with its market price; a well-known example is the so-called King-Davenant law (Davenant, 1699) that documents how deficiencies in the yearly supply of corn affected its price relative to the common rate. Clearly, accurate information about the relation between demand, supply, and selling price are essential when one aims to *optimize* decisions such as price or production quantity. For centuries, obtaining high-quality data to underpin these decisions has in many contexts been a challenging and highly non-trivial task, with the consequence that "learning" optimal selling prices from data was simply not possible.

In contrast, taking a big leap to the 21st century, abundant availability of information is perhaps one of the key characteristics of our time. Using cookies and other techniques, sellers continuously collect detailed data on the browsing and shopping behavior of their customers; not only on an aggregate but even on an individual level, enabling firms to personalize product recommendations and discounts. Furthermore, the fact that decisions such as selling prices are not static

A. V. den Boer
Korteweg-de Vries Institute for Mathematics and Amsterdam Business School, University of Amsterdam, Amsterdam, GE, Netherlands
e-mail: boer@uva.nl

N. B. Keskin (✉)
Duke University, Fuqua School of Business, Durham, NC, USA
e-mail: bora.keskin@duke.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_4

decisions but can often be changed any moment, without any additional costs, enables sellers to dynamically *learn* how the demand for their products is affected by selling prices and respond to this stream of statistical information by continuously and dynamically updating their prices.

Given the complexity of the task of taking optimal, data-driven pricing decisions, it may come as no surprise that algorithms play a key role in this process. A substantial stream of academic literature has emerged in recent years that designs and analyzes such algorithms that learn optimal selling prices from accumulating sales data. The goal of this chapter is to summarize a number of recent results in this area. In addition, since learning-and-earning is far from a finished research topic, we point to a number of emerging topics in the pricing-and-learning literature that may become important research directions in the upcoming years.

## 4.2 Model

We consider a seller who is selling a single type of product and who needs to determine the selling price of the product during subsequent discrete periods. That is, at the beginning of each period $t \in \mathbb{N}$, the seller determines a selling price $p_t \in [p_{\min}, p_{\max}]$, where $p_{\min}$ and $p_{\max}$ are given lower and upper bounds, respectively, on the possible selling prices, and $0 \leq p_{\min} < p_{\max}$. After setting the selling price, the seller observes the demand $D_t$ that is realized during that period, collects revenue $p_t D_t$, and moves on to the next period. The demand is of the form

$$D_t = d(p_t) + \epsilon_t \text{ for all } t \in \mathbb{N},$$

where

$$d : [p_{\min}, p_{\max}] \to [0, \infty)$$

is a continuous and nonincreasing mapping called the *demand function*, and $\{\epsilon_t : t \in \mathbb{N}\}$ is comprised of light-tailed random variables that are independent of $p_{t+1}, p_{t+2}, \ldots$ for all $t \in \mathbb{N}$ and satisfy $\mathbb{E}[\epsilon_t \mid p_1, \ldots, p_t] = 0$ a.s. and $\sup_{t\in\mathbb{N}} \mathbb{E}[\epsilon_t^2 \mid p_1, \ldots, p_t] \leq \sigma^2$ a.s. for some $\sigma > 0$. Thus, $d(p)$ is the expected demand given selling price $p$, and $\epsilon_t$ is a random demand shock in period $t$. Observe that the demand noise terms $\{\epsilon_t : t \in \mathbb{N}\}$ are not necessarily identically distributed. This allows their distribution to depend on the charged price; an example is where each $D_t$ is Bernoulli distributed with mean $d(p_t)$.

The seller is interested in determining a price that maximizes the expected revenue function

$$r(p) := pd(p) \text{ for } p \in [p_{\min}, p_{\max}].$$

For ease of exposition we assume that the marginal costs of the product are equal to zero, so that revenue is equal to profit. Non-zero marginal costs $c$ can be incorporated in the model by maximizing $(p - c)d(p)$ instead of $pd(p)$. We also assume that all demand can be satisfied by the seller, so that stock-outs do not occur. Models with inventory restrictions are considered in the next chapter of this book.

Crucially, the demand function $d(\cdot)$ is assumed to be *unknown* to the seller, so that the revenue function $r(\cdot)$ cannot be maximized directly. Instead, the seller faces the task of *learning* an optimal price from accumulating sales data in an efficient way. To this end, the seller needs to determine for each possible data set or *history* $(p_1, D_1, \ldots, p_{t-1}, D_{t-1})$ of previously used prices and corresponding demand observations, which price will be charged in the next period $t$, for all $t \in \mathbb{N}$. Prices may be random, and thus have a distribution: we denote by $\pi(\cdot \mid h)$ the probability distribution of $p_t$ conditional on $(p_1, D_1, \ldots, p_{t-1}, D_{t-1}) = h$, for all histories $h$ in the set of possible histories

$$H := \bigcup_{t \in \mathbb{N}} \Big\{ [p_{\min}, p_{\max}] \times \mathcal{D} \Big\}^{t-1},$$

where $\mathcal{D} \subset \mathbb{R}$ denotes the set of values that demand can attain (for example, if $D_t$ is Bernoulli distributed for all $t \in \mathbb{N}$, then $\mathcal{D} = \{0, 1\}$). The pricing decisions of the seller are fully specified by the collection $\{\pi(\cdot \mid h) : h \in H)\}$. This collection is called an *admissible policy*, and we denote by $\Pi$ the space of all admissible policies. Note that the empty set $\emptyset$ is an element of $H$, corresponding to the distribution $\pi(\cdot \mid \emptyset)$ of the first price $p_1$ (which is determined when no sales data are available).

It is worth mentioning that policies are usually not specified in a completely formal way. For example, if a policy stipulates "$p_1 = p_{\min}$," then this should be interpreted as $\pi(\cdot \mid \emptyset)$ being a degenerate distribution that puts all probability mass on $p_{\min}$. Furthermore, it is possible that each $\pi(\cdot \mid h)$ is a degenerate distribution that puts all probability mass on a single price. In this case $\pi$ is called a *deterministic* or *non-random* policy: each history then uniquely determines the next price that will be charged, and the policy can be construed as a mapping $\pi : H \to [p_{\min}, p_{\max}]$ from histories to prices. Policies that are not deterministic are called random or randomized policies. To emphasize that the distribution of the price and demand vector $(p_t, D_t : t \in \mathbb{N})$ depends both on the policy $\pi$ and the demand function $d(\cdot)$, we denote the probability measure governing this distribution by $\mathbb{P}_d^\pi\{\cdot\}$ and the associated expectation operator by $\mathbb{E}_d^\pi\{\cdot\}$.

Because the seller is interested in maximizing the expected revenue function $r(\cdot)$, we measure the quality of a policy by the expected revenue loss caused by charging sub-optimal prices. Formally, the regret of an admissible policy $\pi$ after $T$ periods is defined as

$$R_d^\pi(T) := T \max_{p \in [p_{\min}, p_{\max}]} \{r(p)\} - \mathbb{E}_d^\pi \left[ \sum_{t=1}^{T} p_t D_t \right] \quad \text{for } T \in \mathbb{N}.$$

Regret is always non-negative, and the better the policy, the lower the regret.

Ideally, the seller would be able to determine a policy $\pi \in \Pi$ that minimizes its regret $R_d^\pi(T)$ for a given $T \in \mathbb{N}$ and for a large class of demand functions $d(\cdot)$. However, this turns out to be an intractable problem: except for very simple cases (e.g., with $T = 1$), it is generally not possible to compute an optimal policy. For this reason, much research focuses on determining a policy $\pi$ for which the *growth rate* of the regret, as a function of $T$, is as small as possible. Such policies are then called *asymptotically optimal*. Determining asymptotically optimal policies involves two tasks: proving a *lower bound* on the regret of any admissible policy, and constructing a particular admissible policy and proving an *upper bound* on its regret that matches the growth rate of the lower bound in $T$.

*Remark 4.1* The model described above adopts a frequentist approach to uncertainty: it is assumed that there is a single, fixed, non-random demand function $d(\cdot)$ that (partly) determines how expected demand depends on price. Alternatively one could adopt a Bayesian approach and assume that $d(\cdot)$ itself is randomly selected from a set of possible demand functions. A typical approach is then to start with a prior distribution $F_0$ on $d(\cdot)$, update this distribution using Bayes' rule to compute the posterior distribution $F_t$ of $d(\cdot)$ after having observed the data $p_1, D_1, \ldots, p_t, D_t$, and determine $p_{t+1}$ based on this posterior distribution. The quality of a policy in the Bayesian framework is usually measured by the Bayesian regret $\mathbb{E}_{d\sim F_0}[R_d^\pi(T)]$: the expectation of regret with respect to the prior distribution of $d(\cdot)$. This chapter focuses predominantly on frequentist approaches—however, in Sect. 4.3.3, we point to a number of important Bayesian contributions.

*Remark 4.2* It is worth emphasizing that the model described above differs in a number of ways from traditional multi-armed bandit settings. First, the set of feasible actions is not finite but an interval, $[p_{\min}, p_{\max}]$. Second, the expected reward function has a particular structure of the form $r(p) = pd(p)$, where $d(\cdot)$ is a continuous and nonincreasing function, and often more assumptions added to ensure that $r(\cdot)$ has a unique maximizer. Third, observations are made on the demand instead of the revenue. In the single-product setting above, this may seem like a minor difference, but in multi-product extensions, observing the demand for each product separately instead of only observing the aggregate revenue makes a substantial difference.

## 4.3 Asymptotically Optimal Pricing Policies

In this section, we discuss different approaches to the pricing-and-learning problem described in the previous section. Parametric approaches are described in Sect. 4.3.1, nonparametric approaches in Sect. 4.3.2, and references for important extensions and generalizations are given in Sect. 4.3.3.

### 4.3.1 Parametric Approaches

#### 4.3.1.1 Model and Estimation

Several papers study the learning-and-earning problem under the additional assumption that the demand function is linear. This means that

$$d(p) = \theta_1 + \theta_2 p \quad \text{for } p \in [p_{\min}, p_{\max}],$$

where $\theta_1$ and $\theta_2$ are *unknown* parameters. We write $\theta = (\theta_1, \theta_2)$ and assume that $\theta$ lies in a compact set $\Theta := [\theta_{1,\min}, \theta_{1,\max}] \times [\theta_{2,\min}, \theta_{2,\max}]$ for some known parameter bounds $\theta_{1,\min}$, $\theta_{1,\max}$, $\theta_{2,\min}$, $\theta_{2,\max}$ satisfying $0 < \theta_{1,\min} < \theta_{1,\max}$ and $\theta_{2,\min} < \theta_{2,\max} < 0$. In addition, we occasionally write

$$r(p, \vartheta) := p(\vartheta_1 + \vartheta_2 p) \quad \text{for } p \in [p_{\min}, p_{\max}] \text{ and } \vartheta = (\vartheta_1, \vartheta_2) \in \Theta,$$

to emphasize the dependence of the revenue function on both the price and the parameters. If needed, we could assume that $\theta_{1,\min} + \theta_{2,\min} p_{\max} \geq 0$ to ensure that the expected demand is always non-negative, but from a mathematical perspective this assumption is not always necessary. For ease of exposition, we do assume, however, that $\vartheta_1/(-2\vartheta_2) \in (p_{\min}, p_{\max})$ for $(\vartheta_1, \vartheta_2) \in \Theta$. This assumption ensures that the price $\psi(\vartheta)$ that maximizes $r(p, \vartheta)$ with respect to $p$, given by

$$\psi(\vartheta) := \frac{\vartheta_1}{-2\vartheta_2} \quad \text{for } \vartheta \in \Theta,$$

lies in the interior of the feasible price range. Because the unknown demand function is completely characterized by the parameter vector $\theta$, we use in this subsection $\mathbb{P}_\theta^\pi\{\cdot\}$ and $\mathbb{E}_\theta^\pi\{\cdot\}$ instead of $\mathbb{P}_d^\pi\{\cdot\}$ and $\mathbb{E}_d^\pi\{\cdot\}$, respectively.

The unknown parameters of the linear demand model can conveniently be estimated using ordinary least squares (OLS). The unconstrained OLS estimator $\hat{\vartheta}(t)$ of $\theta$, based on data from the first $t$ periods, is given by

$$\hat{\vartheta}(t) := \operatorname*{arg\,min}_{\vartheta \in \mathbb{R}^2} \left\{ \sum_{s=1}^{t} (D_s - \vartheta_1 - \vartheta_2 p_s)^2 \right\} \quad \text{for all } t \in \mathbb{N} \tag{4.1}$$

and is well-defined if not all prices $p_1, \ldots, p_t$ are the same. Because there are no guarantees that all components of $\hat{\vartheta}(t)$ have the correct sign, and because the true parameter vector $\theta$ lies in $\Theta$, we project the unconstrained OLS estimator to $\Theta$:

$$\hat{\theta}(t) := \begin{pmatrix} \mathcal{P}_{[\theta_{1,\min}, \theta_{1,\max}]} \hat{\vartheta}_1(t) \\ \mathcal{P}_{[\theta_{2,\min}, \theta_{2,\max}]} \hat{\vartheta}_2(t) \end{pmatrix} \quad \text{for all } t \in \mathbb{N}, \tag{4.2}$$

where $\mathcal{P}_{[l,u]}(x) := \min\{u, \max\{l, x\}\}$ for all $l, u, x \in \mathbb{R}$ with $l \leq u$.

#### 4.3.1.2 Certainty-Equivalence Pricing and Incomplete Learning

Perhaps the most intuitive pricing policy would be to simply always set the price that is optimal with respect to the available parameter estimates. More formally, choose $p_1, p_2 \in [p_{\min}, p_{\max}]$ with $p_1 \neq p_2$ to ensure that the OLS estimator is defined, and for all $t \geq 3$, set

$$p_t = \psi(\hat{\theta}(t-1)).$$

Thus, this policy simply uses the estimated optimal decision $\psi(\hat{\theta}(t-1))$ in all periods except the first two that are meant for initializing the OLS estimator. Observe that our assumptions on $\Theta$ ensure that $p_t \in [p_{\min}, p_{\max}]$ for all $t$.

The principle of always choosing an action that maximizes the estimated objective function (except in a few initial periods) can be viewed as a *myopic* or *greedy* policy, also known as *passive learning* or *certainty-equivalence control* in general. This principle is very simple, and in some settings, its performance is excellent (see Broder & Rusmevichientong, 2012, section 4; den Boer & Zwart, 2015, section 3; Keskin & Zeevi, 2018, section 4.2.4; Keskin & Birge, 2019, section 5). However, in many settings such as the dynamic pricing-and-learning problem described above, this approach unfortunately performs very poorly (see, e.g., Lai & Robbins, 1982, section 2; Harrison et al. 2012, section 4; den Boer & Zwart, 2014, section 3.1; Keskin & Zeevi, 2014, section 3, den Boer & Keskin, 2022, section 4.4). Building on the analysis of Lai and Robbins (1982), den Boer and Zwart (2014) show that prices generated by the certainty-equivalence policy may converge to a price different from the optimal price, leading to linearly growing regret.

**Proposition 4.1 (den Boer and Zwart (2014, proposition 1))** *Under the certainty-equivalence pricing policy, $p_t$ converges with positive probability to a price different from the optimal price $\psi(\theta)$ as $t \to \infty$.*

In Fig. 4.1, we show sample paths of prices from several simulations of the certainty-equivalence pricing policy. The figure illustrates that the prices do not converge to the optimal price $\psi(\theta)$ as $t \to \infty$, and that the limit price is in fact random. Figure 4.2 shows the limit values of the OLS estimates in these simulations, represented by small circles. The limit values all lie on the curve defined by

$$\theta_1 + \theta_2 \psi(\vartheta) = \vartheta_1 + \vartheta_2 \psi(\vartheta). \tag{4.3}$$

For parameter estimates $\vartheta \in \Theta$ that satisfy (4.3), the *true expected demand* under the estimated optimal price $\psi(\vartheta)$ is equal to the *estimated expected demand* under this price. In other words, the observed expected demand when using price $\psi(\vartheta)$ seems to "confirm" the correctness of the estimates $\vartheta$, even though $\vartheta$ might be different from the true parameter $\theta$.

The phenomenon that parameter estimates in a dynamic decision problem with parameter uncertainty converge with positive probability to an incorrect value is

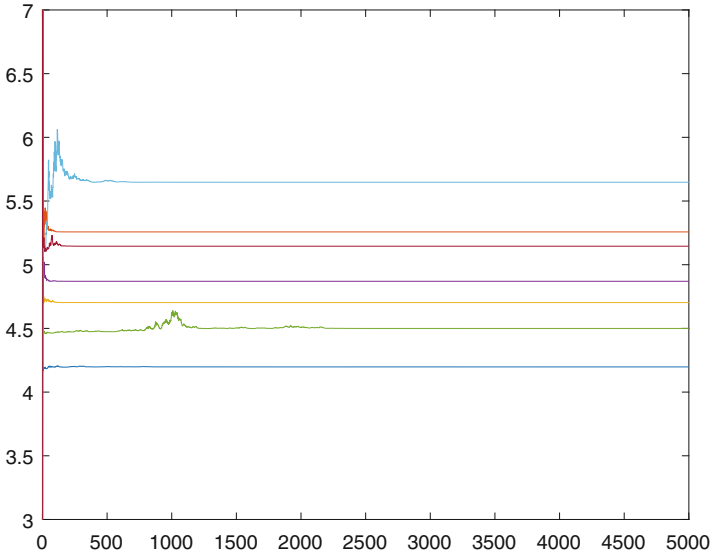


**Fig. 4.1** Sample paths of prices from different simulations of the certainty-equivalence pricing policy, with $\theta_1 = 10$, $\theta_2 = -1$, $p_{\min} = 0$, $p_{\max} = 25$, $p_1 = 3$, $p_2 = 7$, $\theta_{1,\min} = 1$, $\theta_{1,\max} = 20$, $\theta_{2,\min} = -2$, $\theta_{2,\max} = -0.5$, and $\epsilon_t \sim N(0, \sigma^2)$ with $\sigma = 0.5$. Observe that $p_{\min} < \psi(\vartheta) \leq \theta_{1,\max}/(-2\theta_{2,\max}) < p_{\max}$ for all $\vartheta \in \Theta$. The optimal price in this instance is $\psi(10, -1) = 5$

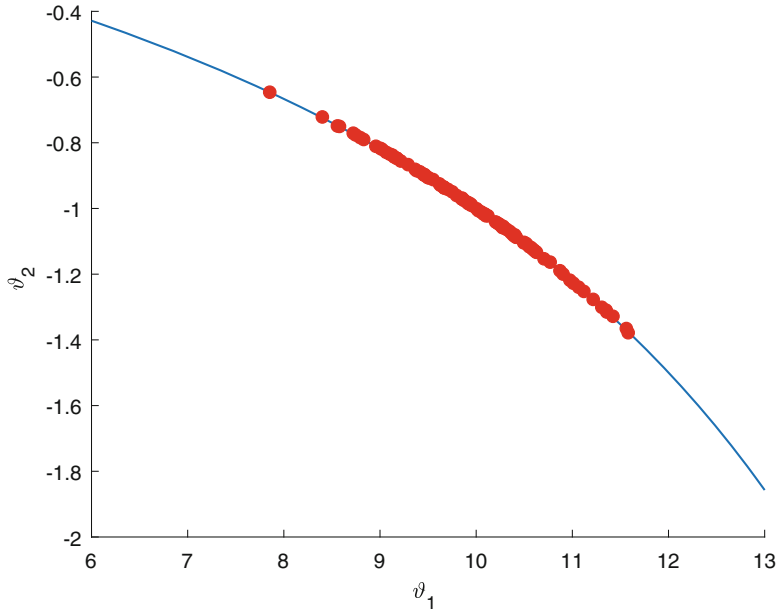


**Fig. 4.2** Each red dot represents the limit value of $\hat{\theta}(t)$ for a simulation of the certainty-equivalence pricing policy, with same parameters as in Fig. 4.1. The solid line represents solutions to (4.3)

called *incomplete learning*, and certainty-equivalence pricing is an example where this phenomenon occurs (see Keskin & Zeevi, 2018).

#### 4.3.1.3 Asymptotically Optimal Policies

An intuitive explanation for why, under the certainty-equivalence policy, the parameter estimates $\hat{\theta}(t)$ do not converge to the true parameter $\theta$ (and the prices $\psi(\hat{\theta}(t))$ do not converge to the true optimal price $\psi(\theta)$) is that certainty-equivalence pricing does not induce sufficient price dispersion: in a sense, the prices converge "too fast."

Dispersion in the covariates is crucial for the consistency of OLS. This is easily seen in a simple, one-dimensional model: let $y_t = \gamma x_t + \epsilon_t$ for all $t \in \mathbb{N}$, where $\{\epsilon_t : t \in \mathbb{N}\}$ are i.i.d. standard Gaussian random variables, and $\{x_t : t \in \mathbb{N}\}$ are non-random real numbers not all equal to zero. The OLS estimate $\hat{\gamma}(t)$ of $\gamma$ based on the data $\{(x_s, y_s) : 1 \leq s \leq t\}$ is equal to $\hat{\gamma}(t) = (\sum_{s \leq t} x_s y_s)/(\sum_{s \leq t} x_s^2)$, which is normally distributed with mean $\gamma$ and variance $1/(\sum_{s \leq t} x_s^2)$. Thus, $\hat{\gamma}(t)$ converges in probability to $\gamma$ as $t \to \infty$ if and only if $\sum_{s=1}^{\infty} x_s^2 = \infty$. For example, if $x_t = 1/t^2$ for all $t \in \mathbb{N}$, then the covariates converge "too quickly" to zero, so that $\sum_{s=1}^{\infty} x_s^2 < \infty$, and $\hat{\gamma}(t)$ does not converge in probability to $\gamma$ as $t \to \infty$.

In our pricing problem, the (unconstrained) OLS estimates $\hat{\vartheta}(t)$ are given by

$$\hat{\vartheta}(t) = \theta + \mathcal{J}_t^{-1} \begin{pmatrix} \sum_{s=1}^{t} \epsilon_s \\ \sum_{s=1}^{t} p_s \epsilon_s \end{pmatrix} \quad \text{for all } t \geq 3,$$

where

$$\mathcal{J}_t := \begin{pmatrix} t & \sum_{s=1}^{t} p_s \\ \sum_{s=1}^{t} p_s & \sum_{s=1}^{t} p_s^2 \end{pmatrix}.$$

The variation of prices is measured by the smallest eigenvalue $\lambda_{\min}(\mathcal{J}_t)$ of this matrix. For practitioners, this quantity might be difficult to interpret. It is related, however, to a quantity that is more intuitive to interpret, namely the variance of $p_1, \ldots, p_t$, as follows:

$$\frac{t\mathrm{Var}(p_1, \ldots, p_t)}{(1 + p_{\max}^2)} \leq \frac{\det(\mathcal{J}_t)/t}{\mathrm{tr}(\mathcal{J}_t)/t} \leq \frac{\det(\mathcal{J}_t)}{\lambda_{\max}(\mathcal{J}_t)} = \lambda_{\min}(\mathcal{J}_t),$$

and

$$\lambda_{\min}(\mathcal{J}_t) = \frac{\det(\mathcal{J}_t)}{\lambda_{\max}(\mathcal{J}_t)} \leq \frac{t^2\mathrm{Var}(p_1, \ldots, p_t)}{\mathrm{tr}(\mathcal{J}_t)/2} \leq \frac{t\mathrm{Var}(p_1, \ldots, p_t)}{(1 + p_{\min}^2)/2},$$

so that $t\mathrm{Var}(p_1, \ldots, p_t) \asymp \lambda_{\min}(\mathcal{J}_t)$. The following result shows that a guaranteed lower bound on $\lambda_{\min}(\mathcal{J}_t)$, or equivalently, on $t\mathrm{Var}(p_1, \ldots, p_t)$, implies a high-probability bound on the OLS estimation error:

**Proposition 4.2 (Keskin and Zeevi (2014, lemma 3))** *There exist positive constants $\rho$ and $k$ such that, under any pricing policy $\pi$,*

$$\mathbb{P}_\theta^\pi \left( ||\vartheta(t) - \theta|| > \delta \text{ and } \lambda_{\min}(\mathcal{J}_t) \geq m \right) \leq kt \exp(-\rho \min(\delta, \delta^2)m),$$

*for all $\delta, m > 0$ and all $t \geq 2$.*

Proposition 4.2 implies that a pricing policy that induces a sufficient amount of price dispersion throughout all periods (i.e., ensuring that $\lambda_{\min}(\mathcal{J}_t)$ or $t\mathrm{Var}(p_1, \ldots, p_t)$ grows sufficiently fast) will generate consistent estimates $\hat{\theta}(t)$ of $\theta$. If, in addition, the policy ensures that the charged prices $p_t$ are also "sufficiently close" to the estimated optimal price $\psi(\hat{\theta}(t-1))$ for all $t$, then prices $p_t$ will converge in probability to the optimal price $\psi(\theta)$. The following result makes this condition more precise and also establishes an upper bound on the resulting regret of the policy:

**Theorem 4.1 (Keskin and Zeevi (2014, theorem 2))** *Let $\kappa_0$, $\kappa_1$ be positive constants, and let $\pi$ be a pricing policy that satisfies*

*(i)* $\lambda_{\min}(\mathcal{J}_t) \geq \kappa_0 \sqrt{t}$,
*(ii)* $\sum_{s=t_0}^{t} \left( \psi(\hat{\theta}(s)) - p_{s+1} \right)^2 \leq \kappa_1 \sqrt{t}$,

*almost surely for some $t_0 \in \mathbb{N}$ and all $t \geq t_0$. Then, there exists a constant $C > 0$ such that $R_\theta^\pi(T) \leq C\sqrt{T} \log T$ for all $T \geq 3$.*

The regret growth rate $R_\theta^\pi(T)$ of $\sqrt{T} \log T$ can hardly be improved. It can be shown by application of the van Trees inequality (Gill & Levit, 1995), a multivariate and Bayesian generalization of the Cramér–Rao lower bound, that $\sqrt{T}$ is the best possible growth rate of regret:

**Theorem 4.2 (Keskin and Zeevi (2014, theorem 1))** *There is a $c > 0$ such that, for all policies $\pi$ and all $T \geq 3$,*

$$\sup_{\theta \in \Theta} R_\theta^\pi(T) \geq c\sqrt{T}.$$

An alternative proof of this result, based on inequalities in hypothesis testing, can be found in Broder and Rusmevichientong (2012, theorem 3.1).

Thus, any policy that satisfies the conditions in Theorem 4.1 is asymptotically optimal in the sense that the growth rate of the regret is optimal, up to logarithmic factors.

We now give three concrete examples of pricing policies that satisfy the criteria for asymptotic optimality. The first example is "controlled variance pricing" (den

Boer & Zwart, 2014), which is also called "constrained iterated least squares" (Keskin & Zeevi, 2014).

Controlled variance pricing

Let $c_1 > 0$ and $c_2 \in (0, (p_{\max} - p_{\min})/2)$.
Let $p_1, p_2 \in [p_{\min}, p_{\max}]$ with $p_1 \neq p_2$.

For all $t \geq 3$:

- Write $\psi_t := \psi(\hat{\theta}(t-1))$.
- If $\text{Var}(p_1, \ldots, p_{t-1}, \psi_t) \geq c_1 t^{-1/2}$ then choose $p_t = \psi_t$;
- Otherwise, choose $p_t = \psi_t \pm c_2 t^{-1/4}$ such that $p_t \in [p_{\min}, p_{\max}]$.

The key idea of this policy is to charge the estimated optimal price $\psi_t$ at each period $t \geq t_0$, except when this price induces insufficient price dispersion to ensure that $\text{Var}(p_1, \ldots, p_t)$ is at least $c_1 t^{-1/2}$; in this case, a small perturbation of $c_2 t^{-1/4}$ is added to or subtracted from $\psi_t$.

Our second example of an asymptotically optimal pricing policy is "MLE-cycle" pricing (Broder & Rusmevichientong, 2012) and is also known under the name "ILS with deterministic testing" (Keskin & Zeevi, 2014). The policy description uses the notation $\hat{\theta}(\mathcal{T})$, for $\mathcal{T} \subset \mathbb{N}$, which denotes the projected OLS estimate based only on the data from the periods in $\mathcal{T}$. Thus, with this notation, $\hat{\theta}(t)$ is shorthand for $\hat{\theta}(\{1, \ldots, t\})$.

MLE-cycle

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be disjoint subsets of $\mathbb{N}$ such that $1 \in \mathcal{T}_1$, $2 \in \mathcal{T}_2$, and

$$\inf_{t \geq 3, i \in \{1,2\}} \left\{ t^{-1/2} |\mathcal{T}_i \cap \{1, \ldots, t\}| \right\} > 0.$$

For all $t \in \mathbb{N}$:

- If $t \in \mathcal{T}_1$, choose $p_t = p_1$;
- If $t \in \mathcal{T}_2$, choose $p_t = p_2$;
- If $t \notin \mathcal{T}_1 \cup \mathcal{T}_2$, choose $p_t = \psi(\hat{\theta}(\{s \in \mathcal{T}_1 \cup \mathcal{T}_2 : s \leq t-1\}))$.

This policy ensures sufficient price dispersion by devoting predetermined portions of the time horizon to price experiments: for all $t \in \mathcal{T}_1$, the price is set to a fixed price $p_1$, and for all $t \in \mathcal{T}_2$, the price is set to a fixed price $p_2 \neq p_1$. In all other periods, the estimated optimal price $\psi(\hat{\theta}(t-1))$ is charged. The sets $\mathcal{T}_1$ and $\mathcal{T}_2$ are

chosen such that the number of exploration periods in periods 1 through $t$ is at least a positive constant times $\sqrt{t}$. This can, for example, be achieved by choosing, as in Broder and Rusmevichientong (2012),

$$\mathcal{T}_1 = \left\{1 + 2c + \frac{1}{2}c(c+1)c_1 : c \in \mathbb{N} \cup \{0\}\right\}, \tag{4.4}$$

$$\mathcal{T}_2 = \left\{2 + 2c + \frac{1}{2}c(c+1)c_1 : c \in \mathbb{N} \cup \{0\}\right\}, \tag{4.5}$$

for some $c_1 \in \mathbb{N}$. The estimate of $\theta$ at each $t \notin \mathcal{T}_1 \cup \mathcal{T}_2$ is based only on the data from the exploration periods $\{s \in \mathcal{T}_1 \cup \mathcal{T}_2 : s \leq t-1\}$. This simplifies the mathematical analysis of the estimator. It is also possible to estimate $\theta$ based on *all* available data, by replacing $\psi(\hat{\theta}(\{s \in \mathcal{T}_1 \cup \mathcal{T}_2 : s \leq t-1\}))$ in the policy description with $p_t = \psi(\hat{\theta}(t-1))$. This modification of MLE-cycle is called "MLE-cycle-s" in Broder and Rusmevichientong (2012). Intuitively one might expect that including more data can only improve the quality of estimators, but this is not true in general (den Boer, 2013).

Our third example of an asymptotically optimal pricing policy is the "semi-myopic pricing scheme" introduced in Besbes and Zeevi (2015). We here call it the "geometric-cycle" policy, since the policy keeps the prices fixed during periods of geometrically increasing duration.

Geometric-cycle

Let $c_1 > 1$, $c_2 \in (0, p_{\max} - p_{\min})/2)$, and $\hat{\theta}^{(0)} \in \Theta$.
For all $c \in \mathbb{N}$, let $n_c := \lceil c_1^c \rceil$, $N_c := \sum_{k=1}^{c-1} 2n_k$, and $\delta_c := c_2 n_c^{-1/4}$. Let $N_0 := 0$

For all $t \in \mathbb{N}$:

$$p_t = \max\{p_{\min}, \psi_{c-1} - \delta_c\} \text{ for all } t = N_{c-1}+1, \ldots, N_{c-1}+n_c,$$
$$p_t = \min\{p_{\max}, \psi_{c-1} + \delta_c\} \text{ for all } t = N_{c-1}+n_c+1, \ldots, N_c,$$

where

$$\psi_{c-1} := \psi(\hat{\theta}(\{N_{c-2}+1, \ldots, N_{c-1}\})) \text{ for } c \geq 2,$$

and $\psi_0 := \psi(\hat{\theta}^{(0)})$.

The geometric-cycle policy divides the time horizon into consecutive cycles indexed by $c \in \mathbb{N}$. The selling price during the first half of the cycle is fixed at the estimated optimal price minus a small perturbation, and to the estimated

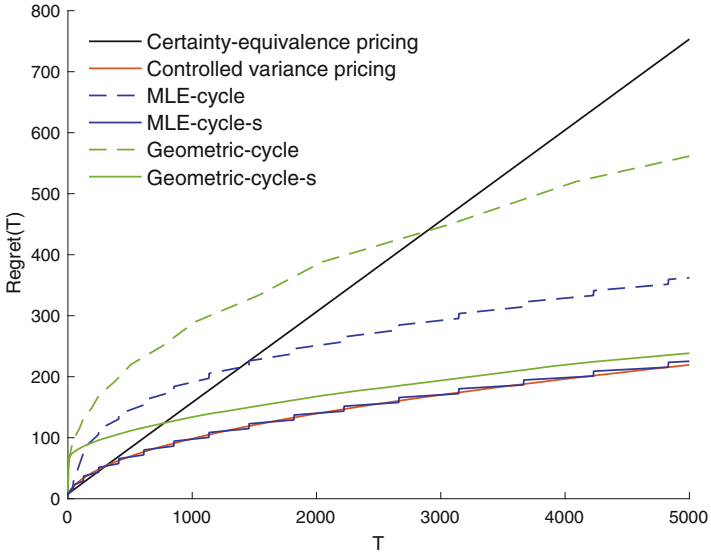


**Fig. 4.3** Regret as function of $T$, for six different pricing policies: certainty-equivalence pricing, controlled variance pricing, MLE-cycle, MLE-cycle-s, geometric-cycle, and geometric-cycle-s, with same parameters as in Fig. 4.1. The pricing policies are implemented with hyper-parameters $p_1 = 3$ and $p_2 = 7$ for certainty-equivalence pricing; $p_1 = 3$, $p_2 = 7$, $c_1 = 2$, and $c_2 = 4$ for controlled variance pricing; $p_1 = 3$, $p_2 = 7$, and $\mathcal{T}_1, \mathcal{T}_2$ as in (4.5) with $c_1 = 40$ for MLE-cycle and MLE-cycle-s; $c_1 = 1$, $c_2 = 1$, and $\hat{\theta}(0)$ drawn uniformly at random from $\Theta$ for geometric-cycle; and $c_1 = 1$, $c_2 = 0.5$, and $\hat{\theta}(0)$ determined similarly for geometric-cycle-s

optimal price plus a small perturbation in the second half of the cycle. Exploration and exploitation are balanced by optimally tuning the length of the cycles, $n_c$, and the magnitude of perturbation from the estimated optimal price, $\delta_c$. The estimated optimal price $\psi_{c-1}$ is based only on the data from cycle $c-1$; this simplifies analysis of the OLS estimator and can be beneficial in scenarios where the sales data is believed to misspecified or contaminated. It is possible to include all available data in the estimator, by replacing $\psi_{c-1}$ in the policy description with $\psi_{c-1} = \psi(\hat{\theta}(N_{c-1}))$.

Figure 4.3 shows the regret as function of $T$ for the six policies described above, namely certainty-equivalence pricing, controlled variance pricing, MLE-cycle, MLE-cycle-s, geometric-cycle, and a variant of geometric-cycle that uses all available data to compute the OLS estimates—we call this policy "geometric-cycle-s." The figure illustrates that the regret of certainty-equivalence pricing grows linearly in $T$, while the regret of the other policies grows akin to $\sqrt{T}$. The regret of MLE-cycle and geometric-cycle policies are higher than that of their counterparts that use all available data to compute the OLS estimates. The performance of the policies might be further improved by fine-tuning the hyper-parameters of the policies.

#### 4.3.1.4 Extensions to Generalized Linear Models

All results in the preceding section can be extended to the case where the demand follows a *generalized linear model* (GLM) instead of a linear one. This means that the expected demand is a general function of $\theta_1 + \theta_2 p$, and the variance of demand is a function of its mean:

$$\mathbb{E}[D_t \mid p_t = p] = h(\theta_1 + \theta_2 p),$$
$$\mathrm{Var}(D_t \mid p_t = p) = v(h(\theta_1 + \theta_2 p)),$$

for all $t \in \mathbb{N}$ and $p \in [p_{\min}, p_{\max}]$, and for some sufficiently smooth known functions $h$ and $v$. The inverse of the function $h(\cdot)$ is usually called the link function of the GLM. Examples that are captured by this model are (i) Bernoulli distributed demand with logit demand function: $h(x) = 1/(1 + \exp(-x))$, $v(x) = x(1 - x)$, (ii) Poisson distributed demand with exponential demand function: $h(x) = \exp(x)$, $v(x) = x$, and (iii) normally distributed demand with linear demand function: $h(x) = x$, $v(x) = \sigma^2$ for some $\sigma > 0$. The unknown parameter vector of the model, $(\theta_1, \theta_2)$, can be estimated by maximum likelihood estimation, and concentration inequalities such as Proposition 4.2 remains valid; the same holds for the upper bound on regret in Theorem 4.1. The proof of this bound uses the fact that, for linear demand functions, the instantaneous revenue loss caused by using an estimated optimal price instead of the true optimal price is quadratic in the estimation error:

$$\psi(\theta) d(\psi(\theta)) - \psi(\hat{\theta}) d(\psi(\hat{\theta})) = O(||\hat{\theta} - \theta||^2). \tag{4.6}$$

This property, and, therefore, the bound in Theorem 4.1, remains valid if the linear demand model is replaced by a generalized linear model.

#### 4.3.1.5 Extensions to Multiple Products

The results in Sect. 4.3.1.3 can be extended to the settings where the seller has multiple products for sale. Let $i \in \{1, \ldots, n\}$ be an index to denote $n$ different products, and write $[n] := \{1, \ldots, n\}$, where $n \in \mathbb{N}$. The set of feasible prices is of the form $\prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}]$ such that for all $i \in [n]$, $p_{i,\min}$ and $p_{i,\max}$ are given lower and upper bounds, respectively, which satisfy $0 \leq p_{i,\min} < p_{i,\max}$. We here discuss the case of linear demand functions, but extensions can be made to generalized linear models (see den Boer, 2014) or multinomial logit models, by replacing the OLS estimator with the maximum likelihood estimator. In each period $t$, the vector of demands for products 1 through $n$, $\mathbf{D}(t) = (D_1(t), \ldots, D_n(t))^\top$, is given by

$$\mathbf{D}(t) = \mathbf{d}(\mathbf{p}(t)) + \boldsymbol{\epsilon}(t) \text{ for all } t \in \mathbb{N},$$

with

$$\mathbf{d}(\mathbf{p}) := \mathbf{a} + \mathbf{B}\mathbf{p} \text{ for } \mathbf{p} \in \prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}],$$

where $\mathbf{a}$ is an unknown vector with strictly positive components, $\mathbf{B} = (b_{ij})$ is an unknown $n \times n$ matrix with strictly negative diagonal elements, and $|b_{ii}| > \sum_{j \neq i} |b_{ij}|$ for all $i \in [n]$, $\mathbf{p}(t) = (p_1(t), \ldots, p_n(t))$ is the vector of prices for products $1, \ldots, n$ in period $t$, and $\boldsymbol{\epsilon}(t)$ is a vector of light-tailed random disturbance terms that satisfy $\mathbb{E}[\boldsymbol{\epsilon}(t) \mid \mathbf{p}(1), \ldots, \mathbf{p}(t)] = 0$ a.s. and $\sup_{t \in \mathbb{N}} \mathbb{E}[||\boldsymbol{\epsilon}(t)||^2 \mid \mathbf{p}(1), \ldots, \mathbf{p}(t)] \leq \sigma^2$ a.s. for some $\sigma > 0$. The expected revenue function, which is given by

$$r(\mathbf{p}) := \mathbf{p}^\top \mathbf{d}(\mathbf{p}) \text{ for } \mathbf{p} \in \prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}],$$

is then strictly concave, with unique maximizer $(\mathbf{B} + \mathbf{B}^\top)^{-1}\mathbf{a}$. Extending the single-product definitions, an admissible policy in the multiple product setting, denoted as $\pi(\cdot \mid h)$, is a collection of probability distributions on the feasible price set $\prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}]$ for each history $h = (\mathbf{p}(1), \mathbf{D}(1), \ldots, \mathbf{p}(t-1), \mathbf{D}(t-1))$.

The regret $R_{\mathbf{d}}^{\pi}(T)$ of a policy $\pi$ after $T$ periods, with unknown demand function $\mathbf{d}$, is defined as

$$R_{\mathbf{d}}^{\pi}(T) = T \max_{\mathbf{p} \in \prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}]} \{r(\mathbf{p})\} - \mathbb{E}_{\mathbf{d}}^{\pi} \left[ \sum_{t=1}^{T} \mathbf{p}(t)^\top \mathbf{D}(t) \right].$$

If we write $x_{\mathbf{p}(t)} := [1 \ \mathbf{p}(t)^\top]^\top$ for all $t \in \mathbb{N}$ and $\boldsymbol{\theta}_i := [a_i \ b_{i,1} \ \ldots \ b_{i,n}]^\top$ for all $i \in [n]$, then for all $i \in [n]$, the OLS estimator of $\boldsymbol{\theta}_i$ based on the transaction data collected in the first $t$ periods is equal to

$$\hat{\boldsymbol{\vartheta}}_i(t) := \arg\min_{\vartheta \in \mathbb{R}^{n+1}} \left\{ \sum_{s=1}^{t} \left( D_i(s) - \vartheta^\top x_{\mathbf{p}(s)} \right)^2 \right\},$$

which is well-defined if the matrix

$$\mathcal{J}(t) := \sum_{s=1}^{t} x_{\mathbf{p}(s)} x_{\mathbf{p}(s)}^\top$$

is invertible. Let $\boldsymbol{\theta}$ be the matrix whose $i$-th row is equal to $\boldsymbol{\theta}_i^\top$ for all $i \in [n]$, and similarly let $\hat{\boldsymbol{\vartheta}}(t)$ denote the matrix whose $i$-th row is equal to $\hat{\boldsymbol{\vartheta}}_i(t)^\top$ for all $i \in [n]$. Let $\Theta$ be a compact set of feasible parameter values containing $\boldsymbol{\theta}$ such that for all $\tilde{\boldsymbol{\theta}} = [\tilde{\mathbf{a}} \ \tilde{\mathbf{B}}] \in \Theta$, the corresponding optimal price vector,

$$\psi(\tilde{\boldsymbol{\theta}}) := (\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)^{-1}\tilde{\mathbf{a}},$$

is well-defined and lies in the feasible price set $\prod_{i=1}^{n}[p_{i,\min}, p_{i,\max}]$. Finally, let $\hat{\boldsymbol{\theta}}(t)$ denote the entrywise projection of the matrix $\hat{\boldsymbol{\vartheta}}(t)$ onto $\Theta$ for all $t \in \mathbb{N}$. For all $\tilde{\boldsymbol{\theta}} \in \Theta$, it holds similarly to (4.6) that instantaneous revenue losses are quadratic in the estimation error. In addition, the squared estimation error $||\hat{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}||^2$ is again closely related to the smallest eigenvalue of $\mathcal{J}(t)$ (den Boer, 2014, proposition 4), similar to the single-product case. As a result of these observations, it can be shown that analogous to Theorem 4.1, if a pricing policy is such that there are $\kappa_0 > 0$, $\kappa_1 > 0$, and $t_0 \in \mathbb{N}$ satisfying

$$\lambda_{\min}(\mathcal{J}(t)) \geq \kappa_0\sqrt{t},$$

and

$$\sum_{s=t_0}^{t} ||\psi(\hat{\boldsymbol{\theta}}(t)) - \mathbf{p}(s+1)||^2 \leq \kappa_1\sqrt{t},$$

almost surely for all $t \geq t_0$, then the regret of the policy is $O(\sqrt{T}\log T)$ (see Keskin and Zeevi (2014, theorem 6), for a proof within a class of "orthogonal" policies).

An example of a policy that satisfies these requirements is the "adaptive pricing policy" (den Boer, 2014), also known as "multivariate CILS" (Keskin & Zeevi, 2014), which extends controlled variance pricing to multiple dimensions:

Multivariate CILS for $n$ products

Let $p_1, \ldots, p_{n+1} \in \prod_{i=1}^{n}[p_{i,\min}, p_{i,\max}]$ such that $\mathcal{J}(n+1)$ is invertible.
Let $c_1 \in \left(0, (n+1)^{-1/2}\lambda_{\min}(\mathcal{J}(n+1))\right)$.
For all $t \geq n+2$:

- Write $\psi_t := \psi(\hat{\boldsymbol{\theta}}(t-1))$ and

$$\mathcal{S}_t := \left\{\mathbf{p} \in \prod_{i=1}^{n}[p_{i,\min}, p_{i,\max}] : \lambda_{\min}\left(\mathcal{J}(t-1) + \mathbf{p}\mathbf{p}^\top\right) \geq c_1 t^{1/2}\right\}.$$

- If $\mathcal{S}_t \neq \emptyset$, then choose

$$\mathbf{p}(t) \in \arg\min\{||\psi_t - \mathbf{p}||^2 : \mathbf{p} \in \mathcal{S}_t\}. \tag{4.7}$$

- Otherwise, choose $\mathbf{p}(t) = \mathbf{p}(t-1)$.

The description above is a simplification from the original policy given in den Boer (2014). There, instead of ensuring a lower bound on the smallest eigenvalue of $\mathcal{J}(t)$, a lower bound on the inverse of the trace of the inverse of $\mathcal{J}(t)$ is ensured; this is based on the fact that $\text{tr}(A^{-1})^{-1} \in [\lambda_{\min}(A), m\lambda_{\min}(A)]$ for a symmetric positive definite $d \times d$ matrix $A$. An advantage of this alternative characterization is computational tractability: while it may not be obvious how to solve $\mathbf{p}(t)$ in (4.7), the condition based on the trace of the inverse of $\mathcal{J}(t)$ implies that the price can be determined by solving a quadratic optimization problem with a single, non-convex quadratic constraint. This type of problems can be solved efficiently (Boyd & Vandenberghe, 2004, appendix B).

It is also possible to extend the ILS/MLE-cycle policy to multiple products. Instead of two different test prices, we now use $n + 1$ test price vectors $\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(n+1)}$ that are linearly independent.

Multivariate ILS with deterministic testing for $n$ products

Let $\mathcal{T}_1, \ldots, \mathcal{T}_{n+1}$ be disjoint subsets of $\mathbb{N}$ such that $i \in \mathcal{T}_i$ for all $i \in [n+1]$, and

$$\inf_{t \geq n+2, i \in \{1,\ldots,n+1\}} \left\{ t^{-1/2} |\mathcal{T}_i \cap \{1, \ldots, t\}| \right\} > 0.$$

Let $\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(n+1)} \in \prod_{i=1}^{n} [p_{i,\min}, p_{i,\max}]$ such that

$$\lambda_{\min} \left( \sum_{s=1}^{n+1} x_{\mathbf{p}(s)} x_{\mathbf{p}(s)}^{\top} \right) > 0.$$

For all $t \in \mathbb{N}$:

- If $t \in \mathcal{T}_i$, choose $\mathbf{p}(t) = \mathbf{p}^{(i)}$, for all $i \in [n]$;
- If $t \notin \bigcup_{i \in [n]} \mathcal{T}_i$, choose $\mathbf{p}(t) = \psi(\hat{\boldsymbol{\theta}}(t-1))$.

The geometric-cycle policy from Sect. 4.3.1.3 can also be extended to multiple products. The perturbations $\pm\delta_c$ around the estimated optimal price $\psi_{c-1}$ should then be extended to all dimensions, for example, by using each of the prices $\psi_{c-1} \pm \delta_c \mathbf{e}_1, \ldots, \psi_{c-1} \pm \delta_c \mathbf{e}_n$ in total $n_c$ times during cycle $c$, where $\mathbf{e}_1, \ldots, \mathbf{e}_n$ are the unit vectors in $\mathbb{R}^n$. Yang et al. (2020) study this type of policy in a nonparametric setting with competition.

### 4.3.2 Nonparametric Approaches

Instead of assuming that the expected demand function has a particular parametric form (e.g., a linear or generalized linear model, with finitely many unknown parameters), it is also possible to take a nonparametric approach and only assume certain smoothness and regularity conditions on the demand and revenue function.

An important class of policies designed to maximize a function (in this case, the revenue function) is known as *gradient ascent* policies. These policies are based on the intuitive idea of "hill-climbing": taking successive steps in the direction where the objective function is increasing, i.e., in the direction of the gradient. In price optimization problems, this gradient is not directly observable from sales data and, therefore, has to be estimated. The literature on the analysis of these types of algorithms, starting with the seminal work by Kiefer and Wolfowitz (1952), is very rich. The following policy description is based on Hong et al. (2020), who present and analyze the Kiefer–Wolfowitz recursion in a pricing context. Let $\mathbf{e}_j$ denote the $j$-th unit vector in $\mathbb{R}^n$, and let $\mathcal{P}_A(x)$ denote the projection of $x$ on $A$. Consider the same setting and notation as in Sect. 4.3.1.5, but without assuming that the demand function $\mathbf{d}$ is linear. For $t \in \mathbb{N}$, let $r_t := \mathbf{D}(t)^\top \mathbf{p}(t)$ denote the revenue earned in period $t$.

Kiefer–Wolfowitz policy for $n$ products

Let $\{a_k\}_{k\in\mathbb{N}}$ and $\{c_k\}_{k\in\mathbb{N}}$ be sequences of positive numbers.
Let $\widehat{\mathbf{p}}(0) \in \prod_{i=1}^n [p_{i,\min}, p_{i,\max}]$.

For all $k \in \mathbb{N} \cup \{0\}$, let

$$\mathbf{p}(t) = \widehat{\mathbf{p}}(k) \qquad \text{if } t = (n+1)k + 1,$$
$$\mathbf{p}(t) = \widehat{\mathbf{p}}(k) + c_k \mathbf{e}_j \qquad \text{if } t = (n+1)k + 1 + j \text{ for some } j \in \{1, \ldots, n\}.$$

If $t$ is a multiple of $(n+1)$, then compute

$$\hat{\mathbf{G}}(k) = \frac{1}{c_k} \left( r_{(n+1)k+2} - r_{(n+1)k+1}, \ldots, r_{(n+1)k+1+n} - r_{(n+1)k+1} \right)$$

and choose

$$\widehat{\mathbf{p}}(k+1) = \mathcal{P}_{\prod_{i=1}^n [p_{i,\min}, p_{i,\max}]} \left\{ \widehat{\mathbf{p}}(k) - a_k \hat{\mathbf{G}}(k) \right\}.$$

If the sequences $\{a_k\}_{k\in\mathbb{N}}$ and $\{c_k\}_{k\in\mathbb{N}}$ are carefully tuned, then the price $\widehat{\mathbf{p}}(k)$ converges in mean squared error to the optimal price, and the regret is $O(\sqrt{T})$:

**Theorem 4.3 (Hong et al. 2020, theorem 3)** *Let $\pi$ correspond to the Kiefer–Wolfowitz policy with step sizes $\{a_k\}_{k\in\mathbb{N}}$ and $\{c_k\}_{k\in\mathbb{N}}$. Suppose that*

*(i)* $\mathbf{d}(\mathbf{p})$ *is twice continuously differentiable with*

$$\max_{\mathbf{p}'\in\prod_{i=1}^{n}[p_{i,\min},p_{i,\max}]} \mathbb{E}[||\mathbf{D}(t)||^2 \mid \mathbf{p}(t)=\mathbf{p}'] < \infty;$$

*(ii) there is a $B_1 > 0$ such that, for all $\mathbf{p}, \mathbf{p}' \in \prod_{i=1}^{n}[p_{i,\min}, p_{i,\max}]$,*

$$r(\mathbf{p}') \le r(\mathbf{p}) + \nabla r(\mathbf{p})^\top(\mathbf{p}'-\mathbf{p}) - \frac{1}{2}B_1||\mathbf{p}'-\mathbf{p}||^2;$$

*(iii)* $\mathbf{r}$ *has unique maximizer $\mathbf{p}^* \in \prod_{i=1}^{n}(p_{i,\min}, p_{i,\max})$;*
*(iv) $a_k = \alpha k^{-1}$ and $c_k = \gamma k^{-1/4}$ for all $k \in \mathbb{N}$ and some $\alpha, \gamma > 0$ satisfying $(4B_1)^{-1} < \alpha < (2B_1)^{-1}$.*

*Then, there are positive constants $\kappa_0, \kappa_1$ such that*

$$\mathbb{E}[||\widehat{\mathbf{p}}(k) - \mathbf{p}^*||^2] \le \kappa_0 k^{-1/2},$$

*for all $k \in \mathbb{N}$, and*

$$R_{\mathbf{d}}^{\pi}(T) \le \kappa_1 T^{1/2},$$

*for all $T \in \mathbb{N}$.*

Alternative conditions on the objective function that ensure similar rates of convergence can be found in Broadie et al. (2011).

The continuous-armed bandit literature proposes several alternative policies to learn the maximum of an objective function, which can also be applied to maximize the expected revenue or profit as function of price: see, e.g., Kleinberg and Leighton (2003), Auer et al. (2007), Cope (2009), Combes and Proutiere (2014), Trovò et al. (2018), Misra et al. (2019). The algorithms proposed in these papers usually make regularity assumptions on the unknown demand function that implies existence of a unique optimal price vector. An exception is Wang et al. (2021), who analyze the case where the revenue function may have multiple local maxima.

### *4.3.3 Extensions and Generalizations*

The literature on dynamic pricing with incomplete information is vast and growing; countless variants and generalizations to the "base" problem described above have been studied. For example, Cheung et al. (2017) determine asymptotically optimal policies in case the seller is allowed to make only a bounded number of price

changes; Birge et al. (2019) consider markdown pricing strategies for forward-looking customers; den Boer and Keskin (2020); Cesa-Bianchi et al. (2019) allow for discontinuities in the demand function; Nyarko (1991), Besbes and Zeevi (2015), Nambiar et al. (2019) study the effect of various forms of model misspecification. We refer to the survey papers by Chen and Chen (2014) and den Boer (2015) for more references to important pricing problems. Our discussion focuses on a frequentist approach; important contributions from a Bayesian viewpoint include Rothschild (1974), McLennan (1984), Cope (2007), Harrison et al. (2012), Kao et al. (2020). Robust optimization approaches to demand uncertainty are also considered; see Bergemann and Schlag (2011). Other important variants of the problem are pricing with finite inventory, contextual information, or changing environments; these three topics are discussed in detail in other chapters of this book.

## 4.4 Emerging Topics and Generalizations

### *4.4.1 Product Differentiation*

Joint management of pricing and product differentiation offers ample opportunities for companies to tailor their products and services to their customers' needs and thereby increase their profits. Inspired by this, an emerging area of research is concerned with the generalization of dynamic pricing-and-learning problems to allow for product differentiation. Studies in this area focus on either horizontal or vertical differentiation strategies.

Ulu et al. (2012) and den Boer et al. (2020) consider the problem of optimally pricing and positioning *horizontally* differentiated products, based on a locational choice model (Hotelling, 1929; Lancaster, 1966, 1975). In such choice models, customers and products are represented by a point on the unit interval, and a customer's utility of purchasing a product depends on its price and its distance to the customer. Ulu et al. (2012) assume discrete support and construct a Bayesian dynamic program to determine the optimal price and locations; den Boer et al. (2020) adopt a frequentist approach, allow customers and products to be located on the whole continuum, and design asymptotically optimal decision policies.

Keskin and Birge (2019) and Keskin and Li (2020) study the optimal pricing of *vertically* differentiated product offerings in the presence of model uncertainty and learning. Motivated by insurance, consumer lending, and telecommunications applications in practice, Keskin and Birge (2019) consider a dynamic learning problem where a firm faces uncertainty about the cost of service quality and analyze how this cost uncertainty influences the firm's vertically differentiated menu of products. Keskin and Li (2020) study the dynamic pricing of vertically differentiated products in a Markov-modulated demand environment.

### 4.4.2 Online Marketplaces

Another emerging area of research considers the design and management of online marketplaces, which enable a multitude of sellers and buyers to conduct business with each other (see, e.g., Cachon et al., 2017; Bai et al., 2018; Taylor, 2018; Gurvich et al., 2019; Bernstein et al., 2021; Huang et al., 2020). Within this research area, the optimal design of pricing-and-learning strategies for marketplaces has recently attracted attention. For example, Birge et al. (2021), Birge et al. (2021) investigate how strategic interactions between marketplace participants influence a market maker's pricing-and-learning strategy.

Two possible directions for future research involve expanding this literature to consider investment strategies (see, e.g., Johari et al., 2010) and matching decisions (see, e.g., Özkan & Ward, 2020). Other directions worth investigating in the context of marketplaces include big data applications (Ban & Keskin, 2021; Keskin et al. 2020), inventory constraints (den Boer et al. 2018; Keskin et al. 2022; Avramidis & den Boer, 2021), and seller collusion (Meylahn & den Boer, 2022).

### 4.4.3 Continuous-Time Approximations

An interesting direction to expand the theory of dynamic pricing with demand learning is the analysis of Brownian models. Using stochastic control theory (see, e.g., Harrison & Sunar, 2015; Sunar et al. 2019, 2021, it is possible to characterize optimal pricing-and-learning policies in certain continuous-time approximations (Keller & Rady, 1999; Keskin 2014). One way to expand the above literature is to study the Brownian counterparts of the aforementioned problems and investigate whether continuous-time approximation offers new insights for policy design.

## References

Auer, P., Ortner, R., & Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In N. Bshouty & C Gentile (Eds.) *Learning Theory. COLT 2007. Lecture Notes in Computer Science* (Vol. 4539, pp. 454–468). Berlin, Heidelberg: Springer.

Avramidis, A. N., & den Boer, A. V. (2021). Dynamic pricing with finite price sets: A non-parametric approach. *Mathematical Methods of Operations Research, 94*(1), 1–34.

Bai, J., So, K. C., Tang, C. S., Chen, X., & Wang, H. (2018). Coordinating supply and demand on on-demand service platform with impatient customers. *Manufacturing and Service Operations Management, 21*(3), 556–570.

Ban, G. Y., & Keskin, N. B. (2021). Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. *Management Science, 67*(9), 5549–5568.

Bergemann, D., & Schlag, K. (2011). Robust monopoly pricing. *Journal of Economic Theory, 146*(6), 2527–2543.

Bernstein, F., DeCroix, G. A., & Keskin, N. B. (2021). Competition between two-sided platforms under demand and supply congestion effects. *Manufacturing & Service Operations Management, 23*(5), 1043–1061.

Besbes, O., & Zeevi, A. (2015). On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science, 61*(4), 723–739.

Birge, J. R., Chen, H., & Keskin, N. B. (2019). Markdown policies for demand learning with forward-looking customers. https://ssrn.com/abstract=3299819

Birge, J. R., Feng, Y., Keskin, N. B., & Schultz, A. (2021). Dynamic learning and market making in spread betting markets with informed bettors. *Operations Research, 69*(6), 1746–1766.

Birge, J. R., Chen, H., Keskin, N. B., & Ward, A. (2021). To interfere or not to interfere: Information revelation and price-setting incentives in a multiagent learning environment. https://ssrn.com/abstract=3864227

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge: Cambridge University Press.

Broadie, M., Cicek, D., & Zeevi, A. (2011). General bounds and finite-time improvement for the Kiefer-Wolfowitz stochastic approximation algorithm. *Operations Research, 59*(5), 1211–1224.

Broder, J., & Rusmevichientong, P. (2012). Dynamic pricing under a general parametric choice model. *Operations Research, 60*(4), 965–980.

Cachon, G. P., Daniels, K. M., & Lobel, R. (2017). The role of surge pricing on a service platform with self-scheduling capacity. *Manufacturing & Service Operations Management, 19*(3), 368–384.

Cesa-Bianchi, N., Cesari, T., & Perchet, V. (2019). Dynamic pricing with finitely many unknown valuations. In A. Garivier & S. Kale (Eds.) *Algorithmic Learning Theory, ALT 2019, Proceedings of Machine Learning Research, PMLR* (Vol. 98, pp. 247–273)

Chen, M., & Chen, Z. L. (2014). Recent developments in dynamic pricing research: Multiple products, competition, and limited demand information. *Production and Operations Management, 24*(5), 704–731.

Cheung, W. C., Simchi-Levi, D., & Wang, H. (2017). Dynamic pricing and demand learning with limited price experimentation. *Operations Research, 65*(6), 1722–1731.

Combes, R., & Proutiere, A. (2014). Unimodal bandits: Regret lower bounds and optimal algorithms. In E. P. Xing & T. Jebara (Eds.) *Proceedings of the 31st International Conference on International Conference on Machine Learning, PMLR* (Vol. 32, pp. 521–529).

Cope, E. (2007). Bayesian strategies for dynamic pricing in e-commerce. *Naval Research Logistics, 54*(3), 265–281.

Cope, E. (2009). Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control, 54*(6), 1243–1253.

Davenant, C. (1699). *An essay upon the probable methods of making a people gainers in the balance of trade*. London: James Knapton.

den Boer, A. V. (2013). Does adding data always improve linear regression estimates? *Statistics & Probability Letters, 83*(3), 829–835.

den Boer, A. V. (2014). Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of Operations Research, 39*(3), 863–888.

den Boer, A. V. (2015). Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science, 20*(1), 1–18.

den Boer, A. V., & Keskin, N. B. (2020). Discontinuous demand functions: Estimation and pricing. *Management Science, 66*(10), 4516–4534.

den Boer, A. V., & Keskin, N. B. (2022). Dynamic pricing with demand learning and reference effects. *Management Science* (in press).

den Boer, A. V., & Zwart, B. (2014). Simultaneously learning and optimizing using controlled variance pricing. *Management Science, 60*(3), 770–783.

den Boer, A. V., & Zwart, B. (2015). Dynamic pricing and learning with finite inventories. *Operations Research, 63*(4), 965–978.

den Boer, A., Perry, O., & Zwart, B. (2018). Dynamic pricing policies for an inventory model with random windows of opportunities. *Naval Research Logistics (NRL), 65*(8), 660–675.

den Boer, A. V., Chen, B., & Wang, Y. (2020). Pricing and positioning of horizontally differentiated products with incomplete demand information. https://ssrn.com/abstract=3682921

Gill, R. D., & Levit, B. Y. (1995). Applications of the van Trees inequality: A Bayesian Cramér-Rao bound. *Bernoulli, 1*(1/2), 59.

Gurvich, I., Lariviere, M., & Moreno, A. (2019). Operations in the on-demand economy: Staffing services with self-scheduling capacity. In M. Hu (Ed.) *Sharing Economy. Springer Series in Supply Chain Management* (Vol. 6, pp. 249–278). Cham: Springer.

Harrison, J. M., & Sunar, N. (2015). Investment timing with incomplete information and multiple means of learning. *Operations Research, 63*(2), 442–457.

Harrison, J. M., Keskin, N. B., & Zeevi, A. (2012). Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science, 58*(3), 570–586.

Hong, L. J., Li, C., & Luo, J. (2020). Finite-time regret analysis of Kiefer-Wolfowitz stochastic approximation algorithm and nonparametric multi-product dynamic pricing with unknown demand. *Naval Research Logistics, 67*(5), 368–379.

Hotelling, H. (1929). Stability in competition. *The Economic Journal, 39*(153), 41.

Huang, H., Sunar, N., & Swaminathan, J. M. (2020). Do noisy customer reviews discourage platform sellers? Empirical analysis of an online solar marketplace. https://ssrn.com/abstract=3645605

Johari, R., Weintraub, G. Y., & Van Roy, B. (2010). Investment and market structure in industries with congestion. *Operations Research, 58*(5), 1303–1317.

Kao, Y. M., Keskin, N. B., & Shang, K. (2020). Bayesian dynamic pricing and subscription period selection with unknown customer utility. https://ssrn.com/abstract=3722376

Keller, G., & Rady, S. (1999). Optimal experimentation in a changing environment. *The Review of Economic Studies, 66*(3), 475–507.

Keskin, N. B. (2014). Optimal dynamic pricing with demand model uncertainty: A squared-coefficient-of-variation rule for learning and earning. https://ssrn.com/abstract=2487364

Keskin, N. B., & Birge, J. R. (2019). Dynamic selling mechanisms for product differentiation and learning. *Operations Research, 67*(4), 1069–1089.

Keskin, N. B., & Li, M. (2020). Selling quality-differentiated products in a Markovian market with unknown transition probabilities. https://ssrn.com/abstract=3526568

Keskin, N. B., & Zeevi, A. (2014). Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research, 62*(5), 1142–1167.

Keskin, N. B., & Zeevi, A. (2018). On incomplete learning and certainty-equivalence control. *Operations Research, 66*(4), 1136–1167.

Keskin, N. B., Li, Y., & Sunar, N. (2020). Data-driven clustering and feature-based retail electricity pricing with smart meters. https://ssrn.com/abstract=3686518

Keskin, N. B., Li, Y., & Song, J. S. J. (2022). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science, 68*(3), 1938–1958.

Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics, 23*(3), 462–466.

Kleinberg, R., & Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03* (pp. 594–605). Washington, DC: IEEE Computer Society.

Lai, T., & Robbins, H. (1982). Iterated least squares in multiperiod control. *Advances in Applied Mathematics, 3*(1), 50–73.

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of Political Economy, 74*(2), 132–157.

Lancaster, K. J. (1975). Socially optimal product differentiation. *American Economic Review, 65*(4), 567–585.

McLennan, A. (1984). Price dispersion and incomplete learning in the long run. *Journal of Economic Dynamics and Control, 7*(3), 331–347.

Meylahn, J., & den Boer, A. (2022). Learning to collude in a pricing duopoly. *Manufacturing & Service Operations Management* (in press).

Misra, K., Schwartz, E. M., & Abernethy, J. (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science, 38*(2), 226–252.

Nambiar, M., Simchi-Levi, D., & Wang, H. (2019). Dynamic learning and pricing with model misspecification. *Management Science, 65*(11), 4980–5000.

Nyarko, Y. (1991). Learning in mis-specified models and the possibility of cycles. *Journal of Economic Theory, 55*(2), 416–427.

Özkan, E., & Ward, A. R. (2020). Dynamic matching for real-time ride sharing. *Stochastic Systems, 10*(1), 29–70.

Rothschild, M. (1974). A two-armed bandit theory of market pricing. *Journal of Economic Theory, 9*(2), 185–202.

Sunar, N., Birge, J. R., & Vitavasiri, S. (2019). Optimal dynamic product development and launch for a network of customers. *Operations Research, 67*(3), 770–790.

Sunar, N., Yu, S., & Kulkarni, V. G. (2021). Competitive investment with Bayesian learning: Choice of business size and timing. *Operations Research, 69*(5), 1430–1449.

Taylor, T. (2018). On-demand service platforms. *Manufacturing and Service Operations Management, 20*(4), 704–720.

Trovò, F., Paladino, S., Restelli, M., & Gatti, N. (2018). Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning, 98*, 196–235.

Ulu, C., Honhon, D., & Alptekinoğlu, A. (2012). Learning consumer tastes through dynamic assortments. *Operations Research, 60*(4), 833–849.

Wang, Y., Chen, B., & Simchi-Levi, D. (2021). Multimodal dynamic pricing. *Management Science, 67*(10), 6136–6152.

Yang, Y., Lee, Y. C., & Chen, P. A. (2020). Competitive demand learning: A data-driven pricing algorithm. https://arxiv.org/abs/2008.05195

# Chapter 5
# Learning and Pricing with Inventory Constraints


**Qi (George) Chen, He Wang, and Zizhuo Wang**

## 5.1 Introduction

In this chapter, we consider learning and pricing problems with inventory constraints: given an initial inventory of one or multiple products and finite selling season, a seller must choose prices dynamically to maximize revenue over the course of the season. Inventory constraints are prevalent in many business settings. For most goods and services, there is limited inventory due to supply constraint, sellers' budget constraint, or limited storage space. Therefore, one must consider the impact of inventory constraints when learning demand functions and setting prices.

Dynamic pricing with inventory constraints has been extensively studied in the revenue management literature, often under the additional assumption that the demand function (i.e., the relationship between demand and price) is known to the seller prior to the selling season. However, when the demand function is unknown, the seller faces a trade-off commonly referred to as the *exploration–exploitation trade-off*. Toward the beginning of the selling season, the seller may offer different prices to try to learn and estimate the demand rate at each price ("exploration" objective). Over time, the seller can use these demand rate estimates to set prices that

Q. (George) Chen
London Business School, London, UK
e-mail: gchen@london.edu

H. Wang (✉)
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA
e-mail: he.wang@isye.gatech.edu

Z. Wang
School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
e-mail: wangzizhuo@cuhk.edu.cn



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_5

maximize revenue throughout the remainder of the selling season ("exploitation" objective). With limited inventory, pursuing the exploration objective comes at the cost of not only lowering revenue but also diminishing valuable inventory. Simply put, if inventory is depleted while exploring different prices, there is no inventory left to exploit the knowledge gained.

In this chapter, we will study how one should design learning algorithm in the presence of inventory constraints. Specifically, we will study how one can overcome the additional challenges brought forth by the limited inventory and still design efficient algorithms for learning demand functions with regret guarantees. In what follows, we will first discuss the simplest case in this setting in Sect. 5.2, i.e., the learning and pricing problem of a single product with an inventory constraint. Then, in Sect. 5.3, we discuss the problem of learning and pricing with multiple products under inventory constraints. Finally, in Sect. 5.4, we consider a Bayesian learning setting with inventory constraints. In each of the sections, we describe the model and the challenges and then present the algorithms and analysis for respective problems. In Sect. 5.5, we present concluding remarks and some further readings for this chapter.

## 5.2 Single Product Case

In this section, we consider the problem of a monopolist selling a single product in a finite selling season $T$. We assume that the seller has a fixed inventory $x$ at the beginning and no replenishment can be made during the selling season. During the selling season, customers arrive according to a Poisson process with an instantaneous demand rate $\lambda_t$ at time $t$.[1] In our model, we assume that $\lambda_t$ is solely dependent on the price the seller offers at time $t$. That is, we can write $\lambda_t = \lambda(p(t))$, where $p(t)$ is the price offered at time $t$. The sales will be terminated at time $T$, and there is no salvage value for the remaining items.

In our model, we assume that the set of feasible prices is an interval $[\underline{p}, \overline{p}]$ with an additional cut-off price $p_\infty$ such that $\lambda(p_\infty) = 0$. The demand rate function $\lambda(p)$ is assumed to be strictly decreasing in $p$ and has an inverse function $p = \gamma(\lambda)$. We define a revenue rate function $r(\lambda) = \lambda\gamma(\lambda)$, which captures the expected revenue when the price is chosen such that the demand is $\lambda$. We further assume $r(\lambda)$ is concave in $\lambda$. These assumptions on demand functions are quite standard and are called the *regular* assumptions in the revenue management literature (Gallego and van Ryzin, 1994).

In addition to the above, we make the following assumptions on the demand rate function $\lambda(p)$ and the revenue rate function $r(\lambda)$:

***Assumption 1*** For some positive constants $M$, $K$, $m_L$, and $m_U$,

[1] Our analysis and result also work if we discretize the time horizon and assume at each time period $t$, there is a probability $\lambda_t$ such that a customer arrives.

1. Boundedness: $|\lambda(p)| \leq M$ for all $p \in [\underline{p}, \overline{p}]$.
2. Lipschitz continuity: $\lambda(p)$ and $r(\lambda(p))$ are Lipschitz continuous with respect to $p$ with factor $K$. Also, the inverse demand function $p = \gamma(\lambda)$ is Lipschitz continuous in $\lambda$ with factor $K$.
3. Strict concavity and differentiability: $r''(\lambda)$ exists and $-m_L \leq r''(\lambda) \leq -m_U < 0$ for all $\lambda$ in the range of $\lambda(p)$ for $p \in [\underline{p}, \overline{p}]$.

In the following, we use $\Gamma = \Gamma(M, K, m_L, m_U)$ to denote the set of demand functions satisfying the above assumptions with the corresponding coefficients. In our model, the seller does not know the true demand function $\lambda$. The only knowledge the seller has is that the demand function belongs to $\Gamma$. Note that $\Gamma$ does not need to have any parametric representation. We note that Assumption 1 is quite mild, and it is satisfied for many commonly used demand function classes including linear, exponential, and logit demand functions.

To evaluate the performance of any pricing algorithm, we adopt the minimax regret objective. We call a pricing policy $\pi = (p(t) : 0 \leq t \leq T)$ admissible if (1) it is a non-anticipating price process that is defined on $[\underline{p}, \overline{p}] \cup \{p_\infty\}$ and (2) it satisfies the inventory constraint, that is, $\int_0^T dN^\pi(s) \leq x$, with probability 1, where $N^\pi(t) = N\left(\int_0^t \lambda(p(s))ds\right)$ denotes the cumulative demand up to time $t$ using policy $\pi$.

We denote the set of admissible pricing policies by $\mathcal{P}$. We define the expected revenue generated by a policy $\pi$ by

$$J^\pi(x, T; \lambda) = E\left[\int_0^T p(s)dN^\pi(s)\right]. \tag{5.1}$$

Given a demand rate function $\lambda$, there exists an optimal admissible policy $\pi^*$ that maximizes (5.1). In our model, since we do not know $\lambda$ in advance, we seek $\pi \in \mathcal{P}$ that performs as close to $\pi^*$ as possible.

However, even if the demand function $\lambda$ is known, computing the expected value of the optimal policy is computationally prohibitive. It involves solving a continuous-time dynamic program exactly. Fortunately, as shown in Gallego and van Ryzin (1994), there exists an upper bound for the expected value of any policy based on the following deterministic optimization problem:

$$\begin{aligned} J^D(x, T; \lambda) = \sup & \int_0^T r(\lambda(p(s)))ds \\ \text{s.t.} & \int_0^T \lambda(p(s))ds \leq x \\ & p(s) \in [\underline{p}, \overline{p}] \cup \{p_\infty\}, \quad \forall s \in [0, T]. \end{aligned} \tag{5.2}$$

Gallego and van Ryzin (1994) showed that $J^D(x, T; \lambda)$ provides an upper bound on the expected revenue generated by any admissible pricing policy $\pi$, that is,

$J^{\pi}(x, T; \lambda) \leq J^{D}(x, T; \lambda)$, for all $\lambda \in \Gamma$ and $\pi \in \mathcal{P}$. With this upper bound, we define the regret $R^{\pi}(x, T; \lambda)$ for any given demand function $\lambda \in \Gamma$ and policy $\pi \in \mathcal{P}$ by

$$R^{\pi}(x, T; \lambda) = 1 - \frac{J^{\pi}(x, T; \lambda)}{J^{D}(x, T; \lambda)}. \tag{5.3}$$

Clearly, $R^{\pi}(x, T; \lambda)$ is always greater than 0. Furthermore, the smaller $R^{\pi}(x, T; \lambda)$ is, the closer the performance of $\pi$ is to that of the optimal policy. However, since the decision-maker does not know the true demand function, it is attractive to have a pricing policy $\pi$ that achieves small regrets across all possible underlying demand functions $\lambda \in \Gamma$. To capture this, we consider the worst-case regret. Specifically, the decision-maker chooses a pricing policy $\pi$, and the nature picks the worst possible demand function for that policy and our goal is to minimize the worst-case regret:

$$\inf_{\pi \in \mathcal{P}} \sup_{\lambda \in \Gamma} R^{\pi}(x, T; \lambda). \tag{5.4}$$

Unfortunately, it is hard to evaluate (5.4) for any finite size problem. In order to obtain theoretical guarantee of proposed policies, we adopt a widely used asymptotic performance analysis. Particularly, we consider a regime in which both the size of the initial inventory and the demand rate grow proportionally. Specifically, in a problem with size $n$, the initial inventory and the demand function are given by

$$x_n = nx \text{ and } \lambda_n(\cdot) = n\lambda(\cdot).$$

Define $J_n^{D}(x, T; \lambda) = J^{D}(nx, T, n\lambda) = nJ^{D}(x, T, \lambda)$ to be the optimal value to the deterministic problem with size $n$ and $J_n^{\pi}(x, T; \lambda) = J^{\pi}(nx, T, n\lambda)$ to be the expected value of a pricing policy $\pi$ when it is applied to a problem with size $n$. The regret for the size-$n$ problem $R_n^{\pi}(x, T; \lambda)$ is therefore

$$R_n^{\pi}(x, T; \lambda) = 1 - \frac{J_n^{\pi}(x, T; \lambda)}{J_n^{D}(x, T; \lambda)},$$

and our objective is to study the asymptotic behavior of $R_n^{\pi}(x, T; \lambda)$ as $n$ grows large and design an algorithm with small asymptotic regret.

### 5.2.1 Dynamic Pricing Algorithm

In this section, we introduce a dynamic pricing algorithm, which achieves near-optimal asymptotic regret for the aforementioned problem. To start with, we first

consider the full-information deterministic problem (5.2). As shown in Besbes and Zeevi (2009), the optimal solution to (5.2) is given by

$$p(t) = p^D = \max\{p^u, p^c\} \tag{5.5}$$

where

$$p^u = \arg \max_{p\in[\underline{p},\overline{p}]} \{r(\lambda(p))\}, \quad p^c = \arg \min_{p\in[\underline{p},\overline{p}]} \left|\lambda(p) - \frac{x}{T}\right|. \tag{5.6}$$

The following important lemma is proved in Gallego and van Ryzin (1994).

**Lemma 1** *Let $p^D$ be the optimal deterministic price when the underlying demand function is $\lambda$. Let $\pi^D$ be the pricing policy that uses the deterministic optimal price $p^D$ throughout the selling season until there is no inventory left. Then, $R_n^{\pi^D}(x, T, \lambda) = O(n^{-1/2})$.*

Lemma 1 states that if one knows $p^D$ in advance, then simply applying this price throughout the entire time horizon can achieve asymptotically optimal performance. Therefore, the idea of our algorithm is to find an estimate of $p^D$ that is close enough to the true one efficiently, using empirical observations on hand. In particular, under Assumption 1, we know that if $p^D = p^u > p^c$, then

$$\left|r(p) - r(p^D)\right| \le \frac{1}{2} m_L (p - p^D)^2 \tag{5.7}$$

for $p$ close to $p^D$, while if $p^D = p^c \ge p^u$, then

$$\left|r(p) - r(p^D)\right| \le K \left|p - p^D\right| \tag{5.8}$$

for $p$ close to $p^D$. In the following discussion, without loss of generality, we assume $p^D \in (\underline{p}, \overline{p})$. Note that this can always be achieved by choosing a large interval of $[\underline{p}, \overline{p}]$.

We now state the main result in this section. We use the notation $f(n) = O^*(g(n))$ to denote there is a constant $C$ and $k$ such that $f(n) \le C \cdot g(n) \cdot \log^k n$.

**Theorem 1** *Let Assumption 1 hold for $\Gamma = \Gamma(M, K, m_L, m_U)$. Then, there exists an admissible policy $\pi$ generated by Algorithm 1, such that for all $n \ge 1$,*

$$\sup_{\lambda\in\Gamma} R_n^{\pi}(x, T; \lambda) = O^*\left(n^{-1/2}\right).$$

A corollary of Theorem 1 follows from the relationship between the nonparametric model and the parametric one:

**Corollary 1** *Assume that $\Gamma$ is a parameterized demand function family satisfying Assumption 1. Then, there exists an admissible policy $\pi$ generated by Algorithm 1, such that for all $n \ge 1$,*

$$\sup_{\lambda \in \Gamma} R_n^{\pi}(x, T; \lambda) = O^*\left(n^{-1/2}\right).$$

Now, we explain the meaning of Theorem 1 and Corollary 1. First, as we will show a matching lower bound in Theorem 2, the result in Theorem 1 is the best asymptotic regret that one can achieve in this setting. Another consequence of our result is that it shows that there is no performance gap between parametric and nonparametric settings in the asymptotic sense, implying that the value of knowing the parametric form of the demand function is marginal in this problem when the best algorithm is adopted. In this sense, our algorithm could save firms' efforts in searching for the right parametric form of the demand functions.

Now, we describe the dynamic pricing algorithm. As mentioned earlier, we aim to learn $p^D$ through price experimentations. Specifically, the algorithm will be able to distinguish whether $p^u$ or $p^c$ is optimal. Meanwhile, it keeps a shrinking interval containing the optimal price with high probability until a certain accuracy is achieved.

Now, we explain the ideas behind the Algorithm 1. In the algorithm, we divide the selling season into a carefully selected set of time periods. In each time period, we test a set of prices within a certain price interval. Based on the empirical observations, we shrink the price interval to a smaller subinterval that still contains the optimal price with high probability and enter the next time period with a smaller price range. We repeat the shrinking procedure until the price interval is small enough so that the desired accuracy is achieved.

Recall that the optimal deterministic price $p^D$ is equal to the maximum of $p^u$ and $p^c$, where $p^u$ and $p^c$ are solved from (5.6). As shown in (5.7) and (5.8), the local behavior of the revenue rate function is quite different around $p^u$ and $p^c$: the former one resembles a quadratic function, while the latter one resembles a linear function (this is an important feature due to the inventory constraint). This difference requires us to use different shrinking strategies for the cases when $p^u > p^c$ and $p^c > p^u$. This is why we have two learning steps (Steps 2 and 3) in our algorithm. Specifically, in Step 2, the algorithm works by shrinking the price interval until either a transition condition (5.10) is triggered or the learning phase is terminated. We show that when the transition condition (5.10) is triggered, with high probability, the optimal solution to the deterministic problem is $p^c$. Otherwise, if we terminate learning before the condition is triggered, we know that $p^u$ is either the optimal solution to the deterministic problem or it is close enough so that using $p^u$ can yield a near-optimal revenue. When (5.10) is triggered, we switch to Step 3, in which we use a new set of shrinking and price testing parameters. Note that in Step 3, we start from the initial price interval rather than the current interval obtained. This is not necessary but solely for the ease of analysis. Both Step 2 and Step 3 (if it is invoked) must terminate in a finite number of iterations.

In the end of the algorithm, a fixed price is used for the remaining selling season (Step 4) until the inventory runs out. In fact, instead of applying a fixed price in

**Algorithm 1** Dynamic pricing algorithm (DPA)

**Step 1. Initialization:**

(a) Consider a sequence of $\tau_i^u, \kappa_i^u$, $i = 1, 2, \ldots, N^u$ and $\tau_i^c, \kappa_i^c$, $i = 1, 2, \ldots, N^c$ ($\tau$ and $\kappa$ represent the length of each learning period and the number of different prices to be tested in each learning period, respectively. Their values along with the values of $N^u$ and $N^c$ are defined in (5.22)–(5.27), (5.17) and (5.21)). Define $\underline{p}_1^u = \underline{p}_1^c = \underline{p}$ and $\overline{p}_1^u = \overline{p}_1^c = \overline{p}$. Define $t_i^u = \sum_{j=1}^{i} \tau_j^u$, for $i = 0$ to $N^u$ and $t_i^c = \sum_{j=1}^{i} \tau_j^c$, for $i = 0$ to $N^c$;

**Step 2. Learn $p^u$ or determine $p^c > p^u$:**
For $i = 1$ to $N^u$ do

(a) Divide $[\underline{p}_i^u, \overline{p}_i^u]$ into $\kappa_i^u$ equally spaced intervals and let $\{p_{i,j}^u, j = 1, 2, \ldots, \kappa_i^u\}$ be the left endpoints of these intervals;
(b) Divide the time interval $[t_{i-1}^u, t_i^u]$ into $\kappa_i^u$ equal parts and define

$$\Delta_i^u = \frac{\tau_i^u}{\kappa_i^u}, \qquad t_{i,j}^u = t_{i-1}^u + j\Delta_i^u, \qquad j = 0, 1, \ldots, \kappa_i^u;$$

(c) For $j$ from 1 to $\kappa_i^u$, apply $p_{i,j}^u$ from time $t_{i,j-1}^u$ to $t_{i,j}^u$. If inventory runs out, then apply $p_\infty$ until time $T$ and STOP;
(d) Compute

$$\hat{d}(p_{i,j}^u) = \frac{\text{total demand over } [t_{i,j-1}^u, t_{i,j}^u)}{\Delta_i^u}, \quad j = 1, \ldots, \kappa_i^u;$$

(e) Compute

$$\hat{p}_i^u = \arg \max_{1 \le j \le \kappa_i^u} \{p_{i,j}^u \hat{d}(p_{i,j}^u)\} \quad \text{and} \quad \hat{p}_i^c = \arg \min_{1 \le j \le \kappa_i^u} \left| \hat{d}(p_{i,j}^u) - x/T \right|; \tag{5.9}$$

(f) If

$$\hat{p}_i^c > \hat{p}_i^u + 2\sqrt{\log n} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \tag{5.10}$$

then break from Step 2, enter Step 3 and set $i_0 = i$;
Otherwise, set $\hat{p}_i = \max\{\hat{p}_i^c, \hat{p}_i^u\}$. Define

$$\underline{p}_{i+1}^u = \hat{p}_i - \frac{\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \text{ and } \overline{p}_{i+1}^u = \hat{p}_i + \frac{2\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}. \tag{5.11}$$

And define the price range for the next iteration

$$I_{i+1}^u = [\underline{p}_{i+1}^u, \overline{p}_{i+1}^u].$$

Here we truncate the interval if it does not lie inside the feasible set $[\underline{p}, \overline{p}]$;
(g) If $i = N^u$, then enter Step 4(a);

(continued)

**Algorithm 1** (continued)

**Step 3. Learn $p^c$ when $p^c > p^u$:**
For $i = 1$ to $N^c$ do

(a) Divide $[\underline{p}_i^c, \overline{p}_i^c]$ into $\kappa_i^c$ equally spaced intervals and let $\{p_{i,j}^c, j = 1, 2, \ldots, \kappa_i^c\}$ be the left endpoints of these intervals;
(b) Define

$$\Delta_i^c = \frac{\tau_i^c}{\kappa_i^c}, \qquad t_{i,j}^c = t_{i-1}^c + j\Delta_i^c + t_{i_0}^u, \qquad j = 0, 1, \ldots, \kappa_i^c;$$

(c) For $j$ from 1 to $\kappa_i^c$, apply $p_{i,j}^c$ from time $t_{i,j-1}^c$ to $t_{i,j}^c$. If inventory runs out, then apply $p_\infty$ until time $T$ and STOP;
(d) Compute

$$\hat{d}(p_{i,j}^c) = \frac{\text{total demand over } [t_{i,j-1}^c, t_{i,j}^c)}{\Delta_i^c}, \quad j = 1, \ldots, \kappa_i^c;$$

(e) Compute

$$\hat{q}_i = \arg \min_{1 \le j \le \kappa_i^c} \left| \hat{d}(p_{i,j}^c) - x/T \right|. \tag{5.12}$$

Define

$$\underline{p}_{i+1}^c = \hat{q}_i - \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \text{ and } \overline{p}_{i+1}^c = \hat{q}_i + \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}. \tag{5.13}$$

And define the price range for the next iteration

$$I_{i+1}^c = [\underline{p}_{i+1}^c, \overline{p}_{i+1}^c].$$

Here, we truncate the interval if it does not lie inside the feasible set of $[\underline{p}, \overline{p}]$;
(f) If $i = N^c$, then enter Step 4(b);

**Step 4. Apply the learned price:**

(a) Define $\tilde{p} = \hat{p}_{N^u} + 2\sqrt{\log n} \cdot \frac{\overline{p}_{N^u}^u - \underline{p}_{N^u}^u}{\kappa_{N^u}^u}$. Use $\tilde{p}$ for the rest of the selling season until the inventory runs out;
(b) Define $\tilde{q} = \hat{q}_{N^c}$. Use $\tilde{q}$ for the rest of the selling season until the inventory runs out.

Step 4, one may continue learning using our shrinking strategy. However, it will not further improve the asymptotic performance of our algorithm.

In the following, we define $\tau_i^u$, $\kappa_i^u$, $N^u$, $\tau_i^c$, $\kappa_i^c$ and $N^c$. Without loss of generality, we assume $T = 1$ and $\overline{p} - \underline{p} = 1$ in the following discussion. We first provide a set of relations we want $(\tau_i^u, \kappa_i^u)$ and $(\tau_i^c, \kappa_i^c)$ to satisfy. Then, we explain the meaning of each relations and derive a set of parameters that satisfy these relations. We use the notation $f \sim g$ to mean that $f$ and $g$ are of the same order in $n$.

The relations that we want $(\tau_i^u, \kappa_i^u)_{i=1}^{N^u}$ to satisfy are

$$\left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}\right)^2 \sim \sqrt{\frac{\kappa_i^u}{n\tau_i^u}}, \quad \forall i = 2, \ldots, N^u, \tag{5.14}$$

$$\overline{p}_{i+1}^u - \underline{p}_{i+1}^u \sim \log n \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}, \quad \forall i = 1, \ldots, N^u - 1, \tag{5.15}$$

$$\tau_{i+1}^u \cdot \left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}\right)^2 \cdot \sqrt{\log n} \sim \tau_1^u, \quad \forall i = 1, \ldots, N^u - 1. \tag{5.16}$$

Also, we define

$$N^u = \min_l \left\{ l : \left(\frac{\overline{p}_l^u - \underline{p}_l^u}{\kappa_l^u}\right)^2 \sqrt{\log n} < \tau_1^u \right\}. \tag{5.17}$$

Next, we state the set of relations we want $(\tau_i^c, \kappa_i^c)_{i=1}^{N^c}$ to satisfy

$$\frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \sim \sqrt{\frac{\kappa_i^c}{n\tau_i^c}}, \quad \forall i = 2, \ldots, N^c, \tag{5.18}$$

$$\overline{p}_{i+1}^c - \underline{p}_{i+1}^c \sim \log n \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}, \quad \forall i = 1, \ldots, N^c - 1, \tag{5.19}$$

$$\tau_{i+1}^c \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \cdot \sqrt{\log n} \sim \tau_1^c, \quad \forall i = 1, \ldots, N^c - 1. \tag{5.20}$$

Also, we define

$$N^c = \min_l \left\{ l : \frac{\overline{p}_l^c - \underline{p}_l^c}{\kappa_l^c} \sqrt{\log n} < \tau_1^c \right\}. \tag{5.21}$$

To understand the above relations, it is useful to examine the source of revenue losses in our algorithm. First, there is an *exploration loss* in each period—the prices tested are not optimal, resulting in suboptimal revenue rate or suboptimal inventory consumption rate. The magnitude of such losses in each period is roughly the deviation of the revenue rate (or the inventory consumption rate) multiplied by the time length of the period. Second, there is a *deterministic loss* due to the limited learning capacity—we only test a grid of prices in each period and may never use the exact optimal price. Third, since the demand follows a stochastic process, the observed demand rate may deviate from the true underlying demand rate, resulting in a *stochastic loss*. Note that these three losses also exist in the learning algorithm

proposed in Besbes and Zeevi (2009). However, in dynamic learning, the loss in one period does not simply appear once, it may have impact on all the future periods. The design of our algorithm tries to balance these losses in each step to achieve the maximum efficiency of learning. With these in mind, we explain the meaning of each equation above in the following:

- The first relation (5.14) ((5.18), respectively) balances the deterministic loss induced by only considering the grid points (the grid granularity is $\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}$ ($\frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}$, resp.)) and the stochastic loss induced in the learning period which will be shown to be $\sqrt{\frac{\kappa_i^u}{n\tau_i^u}}$ ($\sqrt{\frac{\kappa_i^c}{n\tau_i^c}}$, respectively). Due to the relation in (5.7) and (5.8), the loss is quadratic in the price granularity in Step 2 and linear in Step 3.
- The second relation (5.15) ((5.19), respectively) makes sure that with high probability, the price intervals $I_i^u$ ($I_i^c$, respectively) contain the optimal price $p^D$. This is necessary, since otherwise a constant loss will be incurred in all periods afterward.
- The third relation (5.16) ((5.20), respectively) bounds the exploration loss for each learning period. This is done by considering the multiplication of the revenue rate deviation (also demand rate deviation) and the length of the learning period, which in our case can be upper bounded by $\tau_{i+1}^u \sqrt{\log n} \cdot \left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}\right)^2$ ($\tau_{i+1}^c \sqrt{\log n} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}$, respectively). We want this loss to be of the same order for each learning period (and all equal to the loss in the first learning period, which is $\tau_1$) to achieve the maximum efficiency of learning.
- Formula (5.17) ((5.21), respectively) determines when the price we obtain is close enough to optimal such that we can apply this price in the remaining selling season. We show that $\sqrt{\log n} \cdot \left(\frac{\overline{p}_l^u - \underline{p}_l^u}{\kappa_l^u}\right)^2$ ($\sqrt{\log n} \cdot \frac{\overline{p}_l^c - \underline{p}_l^c}{\kappa_l^c}$, respectively) is an upper bound of the revenue rate and demand rate deviations of price $\hat{p}_l$. When this is less than $\tau_1$, we can simply apply $\hat{p}_l$ and the loss will not exceed the loss of the first learning period.

Now, we solve the relations (5.14)–(5.16) and obtain a set of parameters that satisfy them:

$$\tau_1^u = n^{-\frac{1}{2}} \cdot (\log n)^{3.5} \text{ and } \tau_i^u = n^{-\frac{1}{2}\cdot(\frac{3}{5})^{i-1}} \cdot (\log n)^5, \ \forall i = 2, \ldots, N^u, \quad (5.22)$$

$$\kappa_i^u = n^{\frac{1}{10}\cdot(\frac{3}{5})^{i-1}} \cdot \log n, \quad \forall i = 1, 2, \ldots, N^u. \quad (5.23)$$

As a by-product, we have

$$\overline{p}_i^u - \underline{p}_i^u = n^{-\frac{1}{4}(1-(\frac{3}{5})^{i-1})}, \quad \forall i = 1, 2, \ldots, N^u. \quad (5.24)$$

Similarly, we solve the relations (5.18)–(5.20) and obtain a set of parameters that satisfy them:

$$\tau_1^c = n^{-\frac{1}{2}} \cdot (\log n)^{2.5} \text{ and } \tau_i^c = n^{-\frac{1}{2}\cdot(\frac{2}{3})^{i-1}} \cdot (\log n)^3, \ \forall i = 2, \ldots, N^c, \quad (5.25)$$

$$\kappa_i^c = n^{\frac{1}{6}\cdot(\frac{2}{3})^{i-1}} \cdot \log n, \quad \forall i = 1, 2, \ldots, N^c \quad (5.26)$$

and

$$\overline{p}_i^c - \underline{p}_i^c = n^{-\frac{1}{2}(1-(\frac{2}{3})^{i-1})}, \quad \forall i = 1, \ldots, N^c. \quad (5.27)$$

Note that by (5.24) and (5.27), the price intervals defined in our algorithm indeed shrink in each iteration.

### 5.2.2 *Lower Bound Example*

In the last section, we proposed a dynamic pricing algorithm and proved an upper bound of $O^*(n^{-1/2})$ on its regret in Theorem 1. In this section, we show that there exists a class of demand functions satisfying our assumptions such that no pricing policy can achieve an asymptotic regret less than $O^*(n^{-1/2})$. This lower bound example provides a clear evidence that the upper bound is tight. Therefore, our algorithm achieves nearly the best performance among all possible algorithms and closes the performance gap for this problem. Because our algorithm and the lower bound example apply for both parametric and nonparametric settings, it also closes the gap for the problem with a known parametric demand function.

**Theorem 2 (Lower Bound Example)** *Let $\lambda(p; z) = 1/2 + z - zp$, where $z$ is a parameter taking values in $Z = [1/3, 2/3]$ (we denote this demand function set by $\Lambda$). Let $\underline{p} = 1/2$, $\overline{p} = 3/2$, $x = 2$, and $T = 1$. Then, we have the following:*

- *This class of demand function satisfies Assumption 1. Furthermore, for any $z \in [1/3, 2/3]$, the optimal price $p^D$ always equals $p^u$ and $p^D \in [7/8, 5/4]$.*
- *For any admissible pricing policy $\pi$ and all $n \geq 1$,*

$$\sup_{z\in Z} R_n^\pi(x, T; z) \geq \frac{1}{12(48)^2\sqrt{n}}.$$

We first explain some intuitions behind this example. Note that all the demand functions in $\Lambda$ cross at one common point, that is, when $p = 1$, $\lambda(p; z) = 1/2$. Such a price is called an *uninformative* price in Broder and Rusmevichientong (2012). When there exists an uninformative price, experimenting at that price will not gain

information about the demand function. Therefore, in order to learn the demand function (i.e., the parameter $z$) and determine the optimal price, one must at least perform some price experiments at prices away from the uninformative price; on the other hand, when the optimal price is indeed the uninformative price, doing price experiments away from the optimal price will incur some revenue losses. This tension is the key reason for such a lower bound for the loss, and mathematically it is reflected in statistical bounds on hypothesis testing. For the proof of Theorem 2, we refer the readers to Wang et al. (2014).

## 5.3 Multiproduct Setting

In this section, we consider a multiple product and multiple resource generalization of the problem introduced in the previous section. This more general problem, also known as the price-based Network Revenue Management (NRM) problem with learning, considers a setting in which a seller sells to incoming customers $n$ types of products, each of which is made up from a subset of $m$ types of resources, during a finite selling season which consists of $T$ *decision* periods. Denote by $A = [A_{ij}] \in \mathbb{R}_+^{m\times n}$ the *resource consumption matrix*, which indicates that a single unit of product $j$ requires $A_{ij}$ units of resource $i$. Denote by $C \in \mathbb{R}_+^m$ the vector of initial capacity levels of all resources at the beginning of the selling season which cannot be replenished and have zero salvage value at the end of the selling season. At the beginning of period $t \in [T]$, the seller first decides the price $p_t = (p_{t,1}; \ldots; p_{t,n})$ for his products, where $p_t$ is chosen from a convex and compact set $\mathcal{P} = \otimes_{l=1}^n [\underline{p}_l, \bar{p}_l] \subseteq \mathbb{R}^n$ of feasible price vectors. Let $D_t(p_t) = (D_{t,1}(p_t); \ldots; D_{t,n}(p_t)) \in \mathcal{D} := \{(d_1; \ldots; d_n) \in \{0,1\}^n : \sum_{i=1}^n d_i \leq 1\}$ denote the vector of realized demand in period $t$ under price $p_t$. For simplicity, we assume that at most one sale for one of the products occurs in each period. We assume that the purchase probability vector for all products under any price $p_t$, i.e., $\lambda^*(p_t) := \mathbf{E}[D_t(p_t)]$ is unknown to the seller, and this relationship $\lambda^*(.)$, also known as the demand function, needs to be estimated from the data the seller observes during the finite selling season. Define the revenue function $r^*(p) := p \cdot \lambda^*(p)$ to be the one-period expected revenue that the seller can earn under price $p$. It is typically assumed in the literature that $\lambda^*(.)$ is invertible (see the regularity assumptions below). By abuse of notation, we can then write $r^*(p) = p \cdot \lambda^*(p) = \lambda \cdot p^*(\lambda) = r^*(\lambda)$ to emphasize the dependency of revenue on demand rate instead of on price. We make the following regularity assumptions about $\lambda^*(.)$ and $r^*(.)$ which can be viewed as multidimensional counterparts of Assumption 1.

**Regularity Assumptions**

R1. $\lambda^*(.)$ is twice continuously differentiable and it has an inverse function $p^*(.)$ which is also twice continuously differentiable.

R2. There exists a set of turnoff prices $p_j^\infty \in \mathbb{R}_+ \cap \{\infty\}$ for $j = 1, \ldots, n$ such that for any $p = (p_1; \ldots; p_n)$, $p_j = p_j^\infty$ implies that $\lambda_j^*(p) = 0$.
R3. $r^*(.)$ is bounded and strongly concave in $\lambda$.

Compared to the single product setting, the NRM setting imposes two challenges: first, the nice solution structure for single product setting breaks down in the presence of multiple types of products and resources, and second, the approach of estimating the deterministic optimal prices and then applying this learned price may not be sufficient to get tight regret bound since ensuring the same estimation error of the deterministic optimal prices in multidimensional setting requires significantly more observations which in turn affects the best achievable regret bound of this approach. The goal of this section is twofold. First, we introduce two settings of NRM where the demand function possesses some additional structural properties, i.e., the parametric setting where demand function comes from a family of functions parameterized by a *finite* number of parameters and the nonparametric setting where demand function is sufficiently smooth. Second, we introduce an adaptive exploitation pricing scheme which help achieve tight regret bound for the two settings. In the remainder of this section, after introducing some additional preliminary results in Sect. 5.3.1, we will first investigate parametric setting in Sect. 5.3.2 and then investigate the nonparametric setting in Sect. 5.3.3.

### 5.3.1 Preliminaries

Let $D_{1:t} := (D_1, D_2, \ldots, D_t)$ denote the history of the demand realized up to and including period $t$. Let $\mathcal{H}_t$ denote the $\sigma$-field generated by $D_{1:t}$. We define a *control* $\pi$ as a sequence of functions $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$, where $\pi_t$ is a $\mathcal{H}_{t-1}$-measurable real function that maps the history $D_{1:t-1}$ to $\otimes_{j=1}^n [\underline{p}_j, \bar{p}_j] \cup \{p_j^\infty\}$. This class of controls is often referred to as *non-anticipating controls* because the decision in each period depends only on the accumulated observations up to the beginning of the period. Under policy $\pi$, the seller sets the price in period $t$ equal to $p_t^\pi = \pi_t(D_{1:t-1})$ almost surely (a.s.). Let $\Pi$ denote the set of all *admissible controls*:

$$\Pi := \left\{ \pi : \sum_{t=1}^{T} A D_t(p_t^\pi) \preceq C \ \text{ and } \ p_t^\pi = \pi_t(\mathcal{H}_{t-1}) \ \ a.s. \right\}.$$

Note that even though the seller does not know the underlying demand function, the existence of the turnoff prices $p_1^\infty, \ldots, p_n^\infty$ guarantees that this constraint can be satisfied if the seller applies $p_j^\infty$ for product $j$ as soon as the remaining capacity at hand is not sufficient to produce one more unit of product $j$. Let $\mathbf{P}_t^\pi$ denote the induced probability measure of $D_{1:t} = d_{1:t}$ under an admissible control $\pi \in \Pi$, i.e.,

$$\mathbf{P}_t^{\pi}(d_{1:t}) = \prod_{s=1}^{t} \left[ \left( 1 - \sum_{j=1}^{n} \lambda_j^*(p_s^{\pi}) \right)^{\left(1-\sum_{j=1}^{n} d_{s,j}\right)} \prod_{j=1}^{n} \lambda_j^*(p_s^{\pi})^{d_{s,j}} \right],$$

where $p_s^{\pi} = \pi_s(d_{1:s-1})$ and $d_s = [d_{s,j}] \in \mathcal{D}$ for all $s = 1, \ldots, t$. (By definition of $\lambda^*(.)$, the term $1 - \sum_{j=1}^{n} \lambda_j^*(p_s^{\pi})$ can be interpreted as the probability of no-purchase in period $s$ under price $p_s^{\pi}$.) Denote by $\mathbf{E}^{\pi}$ the expectation with respect to the probability measure $\mathbf{P}_t^{\pi}$. The total expected revenue under $\pi \in \Pi$ is then given by

$$R^{\pi} = \mathbf{E}^{\pi} \left[ \sum_{t=1}^{T} p_t^{\pi} \cdot D_t(p_t^{\pi}) \right].$$

The multidimensional version of the deterministic problem in the previous section can be formulated as follows:

$$\text{(P)} \qquad J^D := \max_{p_t, t \in [T]} \left\{ \sum_{t=1}^{T} r^*(p_t) : \sum_{t=1}^{T} A\lambda^*(p_t) \preceq C \right\},$$

$$\text{or equivalently,} \quad (\mathrm{P}_\lambda) \qquad J^D := \max_{\lambda_t, t \in [T]} \left\{ \sum_{t=1}^{T} r^*(\lambda_t) : \sum_{t=1}^{T} A\lambda_t \preceq C \right\}.$$

By assumption R3, $\mathrm{P}_\lambda$ is a convex program and it can be shown that $J^D$ is an upper bound for the total expected revenue under any admissible control, i.e., $R^{\pi} \leq J^D$ for all $\pi \in \Pi$. This allows us to define the regret of an admissible control $\pi \in \Pi$ as $\rho^{\pi} := J^D - R^{\pi}$. Let $\lambda^D$ denote the optimal solution of $\mathrm{P}_\lambda$, and let $p^D = p^*(\lambda^D)$ denote the corresponding optimal deterministic price. (Since $r^*(\lambda)$ is strongly concave with respect to $\lambda$, by Jensen's inequality, the optimal solution is static, i.e., $\lambda_t = \lambda^D$ for all $t$.) Let $\mathsf{Ball}(x, r)$ be a closed Euclidean ball centered at $x$ with radius $r$. We state our fourth regularity assumption below which essentially states that the static price should neither be too low that it attracts too much demand nor too high that it induces no demand:

R4. There exists $\phi > 0$ such that $\mathsf{Ball}(p^D, \phi) \subseteq \mathcal{P}$.

Finally, we will consider a sequence of problems where the length of the selling season and the initial capacity levels are scaled proportionally by a factor $k > 0$. One can interpret $k$ as the *size* of the problem. One can show that the optimal deterministic solution in the scaled problems remains $\lambda^D$. Let $\rho^{\pi}(k)$ denote the regret under an admissible control $\pi \in \Pi$ for the problem with scaling factor $k$. We use the asymptotic order of $\rho^{\pi}(k)$ as the metric for heuristic performance.

### 5.3.2 Parametric Case

In the parametric setting, the functional form of the demand is known, but the finite parameters which pin down the function are unknown. Mathematically, let $\Theta$ be a compact subset of $\mathbb{R}^q$, where $q \in \mathbb{Z}_{++}$ is the number of unknown parameters. Under the parametric demand case, the seller knows that the underlying demand function $\lambda^*(.)$ equals $\lambda(.;\theta)$ for some $\theta \in \Theta$. Although the function $\lambda(.;\theta)$ is known, the true parameter vector $\theta^*$ is unknown and needs to be estimated from the data. The one-period expected revenue function is given by $r(p;\theta) := p \cdot \lambda(p;\theta)$. To leverage the parametric structure of the unknown function, we will focus primarily on *Maximum Likelihood* (ML) estimation which not only has certain desirable theoretical properties but is also widely used in practice. As shown in the statistics literature, to guarantee the regular behavior of ML estimator, certain statistical conditions need to be satisfied. To formalize these conditions in our context, it is convenient to first consider the distribution of a sequence of demand realizations when a sequence of $\tilde{q} \in \mathbb{Z}_{++}$ *fixed* price vectors $\tilde{p} = (\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(\tilde{q})}) \in \mathcal{P}^{\tilde{q}}$ have been applied. For all $d_{1:\tilde{q}} \in \mathcal{D}^{\tilde{q}}$, we define

$$\mathbf{P}^{\tilde{p},\theta}(d_{1:\tilde{q}}) := \prod_{s=1}^{\tilde{q}} \left[ \left( 1 - \sum_{j=1}^{n} \lambda_j(\tilde{p}^{(s)};\theta) \right)^{\left(1-\sum_{j=1}^{n} d_{s,j}\right)} \prod_{j=1}^{n} \lambda_j(\tilde{p}^{(s)};\theta)^{d_{s,j}} \right]$$

and denote by $\mathbf{E}_{\theta}^{\tilde{p}}$ the expectation with respect to $\mathbf{P}^{\tilde{p},\theta}$. In addition to the regularity assumptions R1–R4, we impose additional properties to ensure that the function class $\{\lambda(.;\theta)\}_{\theta\in\Theta}$ is well-behaved.

**Parametric Family Assumptions**

A1 $\lambda(p;\theta)$ and $\frac{\partial \lambda_j}{\partial p_i}(p;\theta)$ for all $i, j \in [n]$ and $i \neq j$ are continuously differentiable in $\theta$.

A2 R1 and R3 hold for all $\theta \in \Theta$.

A3 There exists $\tilde{p} = (\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(\tilde{q})}) \in \mathcal{P}^{\tilde{q}}$ such as for all $\theta \in \Theta$,

i. $\mathbf{P}^{\tilde{p},\theta}(.) \neq \mathbf{P}^{\tilde{p},\theta'}(.)$ for all $\theta' \in \Theta$ and $\theta' \neq \theta$.

ii. For all $k \in [\tilde{q}]$ and $j \in [n]$, $\lambda_j(\tilde{p}^{(k)};\theta) > 0$ and $\sum_{j=1}^{n} \lambda_j(\tilde{p}^{(k)};\theta) < 1$.

iii. The minimum eigenvalue of the matrix $\mathcal{I}(\tilde{p},\theta) := [\mathcal{I}_{i,j}(\tilde{p},\theta)] \in \mathbb{R}^{q\times q}$ where

$$\mathcal{I}_{i,j}(\tilde{p},\theta) = \mathbf{E}_{\theta}^{\tilde{p}} \left[ -\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log \mathbf{P}^{\tilde{p},\theta}(D_{1:\tilde{q}}) \right]$$

is bounded from below by a positive number.

Note that A1 and A2 are quite natural assumptions satisfied by many demand functions such as linear, multinomial logit, and exponential demand. We call $\tilde{p}$ in A3 *exploration prices*. A3 ensures that there exists a set of price vectors (e.g., $\tilde{p}$), which, when used repeatedly, would allow the seller to use ML estimator to statistically identify the true demand parameter. Note that the symmetric matrix $\mathcal{I}(\tilde{p}, \theta)$ defined in A3-iii is known as the *Fisher information* matrix in the literature, and it captures the amount of information that the seller obtains about the true parameter vector using the exploration prices $\tilde{p}$. A3-iii requires the Fisher matrix to be strongly positive definite; this is needed to guarantee that the seller's information about the underlying parameter vector strictly increases as he observes more demand realizations under $\tilde{p}$. We want to point out that it is easy to find exploration prices for the commonly used demand function families. For example, for linear and exponential demand function families, any $\tilde{q} = n+1$ price vectors $\tilde{p}^{(1)}, \ldots, \tilde{p}^{(n+1)}$ constitute a set of exploration prices if (a) they are all in the interior of $\mathcal{P}$ and (b) the vectors $(1; \tilde{p}^{(1)}), \ldots, (1; \tilde{p}^{(n+1)}) \in \mathbb{R}^{n+1}$ are linearly independent. For the multinomial logit demand function family, any $\tilde{q} = 2$ price vectors $\tilde{p}^{(1)}, \tilde{p}^{(2)}$ constitute a set of exploration prices if (a) they are both in the interior of $\mathcal{P}$ and (b) $\tilde{p}_i^{(1)} \neq \tilde{p}_i^{(2)}$ for all $i = 1, \ldots, n$.

Next, we develop a heuristic called Parametric Self-adjusting Control (PSC). In PSC, the selling season is divided into an *exploration* stage followed by an *exploitation* stage. The exploration stage lasts for $L$ periods ($L$ is a tuning parameter to be selected by the seller) where the seller alternates among exploration prices to learn the demand function. At the end of the exploration stage, the seller computes his ML estimate of $\theta^*$, denoted by $\hat{\theta}_L$ (in case the maximum of the likelihood function is not unique, take any maximum as the ML estimate), based on all his observations so far, and solves $\mathrm{P}_\lambda(\hat{\theta}_L)$ for its solution $\lambda^D(\hat{\theta}_L)$ as an estimate of the deterministically optimal demand rate $\lambda^D(\theta^*)$. Then, for the remaining $(T-L)$-period exploitation stage, the seller uses price vectors according to a simple adaptive rule which we explain in more detail below. Define $\hat{\Delta}_t(p_t; \hat{\theta}_L) := D_t - \lambda(p_t; \hat{\theta}_L)$, and let $C_t$ denote the remaining capacity at the *end* of period $t$. The complete PSC procedure is given in Algorithm 2.

In contrast to many proposed heuristics that use the learned deterministic optimal price for exploitation, PSC uses the adaptive price adjustment rule in (5.28) for exploitation. To see the idea behind this design, suppose the estimate of the parameter vector is accurate (Jasin, 2014). In that setting, $\hat{\Delta}_t$ equals the stochastic variability in demand arrivals $\Delta_t := D_t - \lambda(p_t; \theta^*)$, and the pricing rule in (5.28) reduces to adjusting the prices in each period $t$ to achieve a *target demand rate*, i.e., $\lambda^D(\theta^*) - \sum_{s=L+1}^{t-1} \frac{\Delta_s}{T-s}$. The first part of this expression, $\lambda^D(\theta^*)$, is the optimal demand rate if there were no stochastic variability, and we use it as a *base rate*; the second part of the expression, on the other hand, works as a fine adjustment to the base rate in order to mitigate the observed stochastic variability. To see how such adjustment works, consider the case with a single product: if there is more demand than what the seller expects in period $s$, i.e., $\Delta_s > 0$, then the pricing rule automatically accounts for it by reducing the target demand rate for all remaining

**Algorithm 2** Parametric self-adjusting control (PSC)

Tuning Parameter: $L$

1. *Stage 1 (Exploration)*
   a. Determine the exploration prices $\{\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(\tilde{q})}\}$.
   b. For $t = 1$ to $L$, do:
      - If $C_{t-1} \succeq A_j$ for all $j$, apply price $p_t = \tilde{p}^{(\lfloor (t-1)\tilde{q}/L \rfloor + 1)}$ in period $t$.
      - Otherwise, apply price $p_{t',j} = p_j^{\infty}$ for all $j$ and $t' \geq t$; then terminate PSC.
   c. At the end of period $L$:
      - Compute the ML estimate $\hat{\theta}_L$ based on $p_{1:L}$ and $D_{1:L}$
      - Solve $\mathrm{P}_\lambda(\hat{\theta}_L)$ for $\lambda^D(\hat{\theta}_L)$.

2. *Stage 2 (Exploitation)*
   For $t = L + 1$ to $T$, compute:

$$\hat{p}_t = p\left(\lambda^D(\hat{\theta}_L) - \sum_{s=L+1}^{t-1} \frac{\hat{\Delta}_s(p_s; \hat{\theta}_L)}{T - s}; \hat{\theta}_L\right). \tag{5.28}$$

   - If $C_{t-1} \succeq A_j$, and $\hat{p}_t \in \mathcal{P}$, apply price $p_t = \hat{p}_t$ in period $t$
   - Otherwise, for product $j = 1$ to $n$, do:
     – If $C_{t-1} \prec A_j$, apply price $p_{t,j} = p_j^{\infty}$.
     – Otherwise, apply price $p_{t,j} = p_{t-1,j}$

$(T - s)$-period; moreover, the target demand rate adjustment is made *uniformly* across all $(T-s)$-period so as to minimize unnecessary price variations. Jasin (2014) has shown that the ability to *accurately* mitigate the stochastic variability allows this self-adjusting pricing rule be effective *when the parameter vector is known*. However, as one can imagine, such *precise* adjustment is not possible when the parameter vector is subject to estimation error. Indeed, when $\hat{\theta}_L \neq \theta^*$, the seller can only adjust target demand rate based on an estimate of $\Delta_s$, i.e., $\hat{\Delta}_s$; moreover, the seller can no longer correctly find out the price vector that accurately induces (on average) the target demand rate since the inverse demand function is also subject to estimation error. Can this pricing rule work well when the underlying demand parameter is subject to estimation error? The answer is yes, and the key observation is that these two sources of systematic biases push the price decisions on opposing directions and their impact is thus reduced. To see that, consider a single product case where the seller overestimates demand for all prices, i.e., $\lambda(p; \hat{\theta}_L) > \lambda(p; \theta^*)$ for all $p$: on the one hand, since the seller would underestimate the stochastic variation that he needs to adjust (i.e., $\hat{\Delta}_s = D_s - \lambda(p_s; \hat{\theta}_L) < D_s - \lambda(p_s; \theta^*) = \Delta_s$), this would push up the target demand rate (which would push down the price) than if there were no estimation error; on the other hand, since $p(\lambda; \hat{\theta}_L) > p(\lambda; \theta^*)$, for

a given target demand rate, the presence of estimation error would push the price up. Quite interestingly, these opposing mechanisms are sufficient for PSC to achieve the optimal rate of regret.

**Theorem 3** *Suppose that R1–R4 and A1–A3 hold. Set* $L = \lceil \sqrt{kT} \rceil$. *Then, there exists a constant* $M_1 > 0$ *independent of* $k \geq 1$ *such that* $\rho^{PSC}(k) \leq M_1\sqrt{k}$ *for all* $k \geq 1$.

Note that in light of the lower bound example in the previous section, PSC achieves the best achievable regret. The reason PSC achieves this tight bound can be briefly explained as follows. First, it leverages the fact that the demand model is fully determined by a *finite-dimensional* vector $\theta^*$, which can be efficiently estimated by ML estimation. Under ML, roughly speaking, to obtain an estimation error in the order of $\epsilon$, the seller needs to spend roughly $\Theta(\epsilon^{-2})$ periods exploring the demand curve with exploration prices which are not necessarily optimal. Second, the self-adjusting pricing rule in (5.28) helps reduce the impact of estimation error on revenue obtained during exploitation compared to using the learned deterministic price directly. To see that, suppose that the true parameter vector is misestimated by a *small* error $\epsilon$, then one can show that $\lambda^D(\hat{\theta}_L)$ is roughly $\epsilon$ away from $\lambda^D(\theta^*)$. If the seller simply uses the learned deterministic optimal price $p^D(\hat{\theta}_L)$ throughout the exploitation stage, then the one-period regret is roughly $r(\lambda^D(\theta^*); \theta^*) - r(\lambda^D(\hat{\theta}_L); \theta^*) \approx \nabla_\lambda r(\lambda^D(\theta^*); \theta^*) \cdot (\lambda^D(\theta^*) - \lambda^D(\hat{\theta}_L)) \approx \Theta(\epsilon)$ (note that a tighter bound cannot be obtained since the gradient at the constrained optimal solution is not necessarily zero). In PSC, as mentioned above, the pricing rule (5.28) introduces opposing mechanisms to mitigate the impact of systematic error $\epsilon$ on regret which results in a one-period regret of $\Theta(\epsilon^2)$. Thus, the total regret in both exploration and exploitation is bounded by $\Theta(L) + \Theta(\epsilon^2(kT - L)) = O(\epsilon^{-2} + \epsilon^2 kT)$, which is bounded by $O(\sqrt{kT})$ after optimally tuning $\epsilon$ (or equivalently, $L$).

### 5.3.3 Nonparametric Case

The setting in Sect. 5.3.2 assumes that the seller has a good prior knowledge of the functional form of the demand function which may not be appropriate in cases such as new product launch where no historically relevant data is available. Blindly assuming a parametric demand model may be inappropriate and could potentially result in significant revenue loss if the parametric form is misspecified, e.g., a seller who uses linear model to fit the data generated by a logit model. An alternative setting, also known as the *nonparametric approach*, is one where the seller has no prior knowledge of the functional form but tries to estimate the demand directly. The challenge of this approach is that, instead of estimating a finite number of parameters, the seller now needs to directly estimate the demand function value at different price vectors to get an idea of the shape of the demand curve; thus, the number of point estimates needed to ensure low estimation error increases exponentially as the number of products increases. To keep the estimation

problem tractable, a common assumption made in the statistics literature for nonparametric approaches is to impose smoothness conditions of the underlying demand functions (Gyorfi et al., 2002). To that end, let $\bar{s}$ denote the largest integer such that $|\partial^{a_1,\ldots,a_n}\lambda_j^*(p)/\partial p_1^{a_1}\ldots\partial p_n^{a_n}|$ is uniformly bounded for all $j \in [n]$ and $0 \leq a_1,\ldots,a_n \leq \bar{s}$. We call $\bar{s}$ the *smoothness index*. We make the following smoothness assumptions:

**Nonparametric Function Smoothness Assumptions**

N1. $\bar{s} \geq 2$.

N2. $\left|\frac{\partial^{a_1,\ldots,a_n}\lambda_j^*(p)}{\partial p_1^{a_1}\ldots\partial p_n^{a_n}}\right|$ is uniformly bounded for all $j \in [n]$, $p \in \mathcal{P}, 0 \leq a_1,\ldots,a_n \leq \bar{s}$.

The above assumptions are fairly mild and are satisfied by most commonly used demand functions, including linear, polynomial with higher degree, logit, and exponential with a bounded domain of feasible prices. The smoothness index $\bar{s}$ indicates the level of difficulty in estimating the corresponding demand function: the larger the value of $\bar{s}$, the smoother the demand function, and the easier it is to estimate its shape because its value cannot have a drastic local change.

The idea of the nonparametric approach to be introduced later in this section is to replace the ML estimator in PSC by a nonparametric estimation procedure. One such approach is to use a linear combination of spline functions to approximate the underlying demand function which we introduce below. Spline functions have been widely used in engineering to approximate complicated functions, and their popularity is primarily due to their flexibility in effectively approximating complex curve shapes (Schumaker, 2007). This flexibility lies in the piecewise nature of spline functions—a spline function is constructed by attaching piecewise polynomial functions with a certain degree, and the coefficients of these polynomials are computed in such a way that a sufficiently high degree of smoothness is ensured in the places where the polynomials are connected. More formally, for all $l \in [n]$, let $\underline{p}_l = x_{l,0} < x_{l,1} \cdots < x_{l,d} < x_{l,d+1} = \bar{p}_l$ be a partition that divides $[\underline{p}_l, \bar{p}_l]$ into $d+1$ subintervals of equal length where $d \in \mathbb{Z}_{++}$. Let $\mathcal{G} := \otimes_{l=1}^n \mathcal{G}_l$ denote a set of grid points, where $\mathcal{G}_l = \{x_{l,i}\}_{i=0}^{d+1}$. We define the function space of *tensor-product polynomial splines of order* $(s;\ldots;s) \in \mathbb{R}^n$ with a set of grid points $\mathcal{G}$ as $\mathsf{S}(\mathcal{G},s) := \otimes_{l=1}^n \mathsf{S}_l(\mathcal{G}_l,s)$, where $\mathsf{S}_l(\mathcal{G}_l,s) := \{f \in \mathsf{C}^{s-2}([\underline{p}_l,\bar{p}_l]) : f$ is a single-variate polynomial of degree $s-1$ on each subinterval $[x_{l,i-1},x_{l,i})$, for all $i \in [d]$ and $[x_{l,d},x_{l,d+1}]\}$. One of the key questions that spline approximation theory addresses is the following: given an arbitrary function $\lambda$ that satisfies N1-N2, find a spline function $g^* \in \mathsf{S}(\mathcal{G},s)$ that approximates $\lambda$ well. Among the various approaches, one of the most popular approximations is using the so-called *tensor-product B-Spline basis functions* (Schumaker, 2007). This approach is based on using the linear combinations of a collection of $(s+d)^n$ tensor-product B-Spline basis functions, denoted by $\{N_{i_1,\ldots,i_n}(x_1,\ldots,x_n)\}_{i_1=1,\ldots,i_n=1}^{s+d,\ldots,s+d}$, which span the functional space $\mathsf{S}(\mathcal{G},s)$, to approximate the target function $\lambda$. Therefore, the problem of finding $g^*$ is reduced to the problem of computing the coefficients for

representing $g^*$. Schumaker (2007) proposed an explicit formula for computing these coefficients when the value of $\lambda$ is perfectly observable, and the coefficients depend on $\lambda(.)$ only via its function value evaluated on a finite number of price vectors in $\mathcal{P}$ (i.e., the $(s+d)^n s^n$ price vectors in $\tilde{\mathcal{G}}$ defined in Algorithm 3); the details for the formula are bit technical, but we provide these in Algorithm 3 for completeness. In our problem setting, finding an approximation for $\lambda_j^*(.)$ for all $j \in [n]$ is more challenging since we observe noisy observations of the function value, so we use empirical mean of demand realizations as a surrogate for $\lambda_j^*(p)$ and propose the following *Spline Estimation* algorithm in Algorithm 4 to estimate the demand, which involves observing $\tilde{L}_0 := L_0(s+d)^n s^n$ samples.

Let $\tilde{\lambda}(.)$ denote the spline function computed via Algorithm 4. It can be shown that with high probability, the approximation error of $\tilde{\lambda}(.)$ converges to zero at a slightly slower rate than the ML estimator in the parametric case. While one may be tempted to directly apply the exploitation method in PSC, i.e., the pricing rule in (5.28), the analysis of such approach is quite difficult since, given the nature of B-spline functions and the estimation procedure, $\tilde{\lambda}(.)$ may lose some of the regularity properties that $\lambda^*(.)$ possesses. Thus, we introduce two more functional approximations on $\tilde{\lambda}(.)$ before applying the self-adjusting pricing procedure for exploitation. To that end, we introduce a quadratic program approximation of P

**Algorithm 3** Spline approximation

Input function: $\lambda \in C^0(\mathcal{P})$ and $\lambda$ satisfies N1 and N2
Output function: $g^* \in \mathsf{S}(\mathcal{G}, s)$

1. For $l \in [n]$, $i \in [s+d]$, define $\{y_{l,i}\}_{i=1}^{2s+d}$ as follows

$$y_{l,1} = \cdots = y_{l,s} = x_{l,0},$$
$$y_{l,s+1} = x_{l,1},\ y_{l,s+2} = x_{l,2},\ \ldots,\ y_{l,s+d} = x_{l,d},$$
$$y_{l,s+d+1} = \cdots = y_{l,2s+d} = x_{l,d+1};$$

moreover, compute the following:

$$\tau_{l,i,j} = y_{l,i} + (y_{l,i+s} - y_{l,i})\tfrac{j-1}{s-1} \quad \text{and} \quad \beta_{l,i,j} = \textstyle\sum_{v=1}^{j} \frac{(-1)^{v-1}}{(s-1)!}\phi_{l,i,s}^{(s-v)}(0)\psi_{l,i,j}^{(v-1)}(0), \quad \text{for } j \in [s],$$

where $\phi_{l,i,s}(t) = \prod_{r=1}^{s-1}(t - y_{l,i+r})$, $\psi_{l,i,j}(t) = \prod_{r=1}^{j-1}(t - \tau_{l,i,r})$, $\psi_{l,i,1}(t) \equiv 1$. Let $\tilde{\mathcal{G}} := \{(\tau_{1,i_1,j_1}; \ldots; \tau_{n,i_n,j_n}) : i_l \in [s+d],\ j_l \in [s] \text{ for all } l \in [n]\}$.
2. Define $g^*$ as follows:

$$g^*(x_1, \ldots, x_n) = \sum_{i_1=1}^{s+d} \cdots \sum_{i_n=1}^{s+d} \gamma_{i_1,\ldots,i_n} N_{i_1,\ldots,i_n}(x_1, \ldots, x_n),$$

$$\text{where} \quad \gamma_{i_1,\ldots,i_n} = \sum_{j_1=1}^{s}\sum_{r_1=1}^{j_1} \cdots \sum_{j_n=1}^{s}\sum_{r_n=1}^{j_n} \frac{\lambda(\tau_{1,i_1,r_1}, \ldots, \tau_{n,i_n,r_n}) \prod_{l=1}^{n} \beta_{l,i_l,j_l}}{\prod_{l=1}^{n}\prod_{s_l=1, s_l \neq r_l}^{j_l}(\tau_{l,i_l,r_l} - \tau_{l,i_l,s_l})}$$

**Algorithm 4** Spline estimation

Input Parameter: $L_0, n, s$ Tuning Parameter: $d$

1. Estimate $\lambda^*(p)$ at points $p \in \tilde{\mathcal{G}}$. For each $p \in \tilde{\mathcal{G}}$
   a. Apply price $p$ $L_0$ times
   b. Let $\tilde{\lambda}(p)$ be the sample mean of the $L_0$ observations
2. Construct spline approximation
   a. For all $j \in \overline{[1,n]}$ and $i_l \in \overline{[1,s+d]}$, $l \in \overline{[1,n]}$, calculate coefficients $c^j_{i_1,\ldots,i_n}$ as:

$$c^j_{i_1,\ldots,i_n} = \sum_{j_1=1}^{s} \sum_{r_1=1}^{j_1} \cdots \sum_{j_n=1}^{s} \sum_{r_n=1}^{j_n} \frac{\tilde{\lambda}_j(\tau_{1,i_1,r_1}, \ldots, \tau_{n,i_n,r_n}) \prod_{l=1}^{n} \beta_{l,i_l,j_l}}{\prod_{l=1}^{n} \prod_{s_l=1, s_l \neq r_l}^{j_l} (\tau_{l,i_l,r_l} - \tau_{l,i_l,s_l})}.$$

   b. Construct a tensor-product spline function $\tilde{\lambda}(p) = (\tilde{\lambda}_1(p); \ldots; \tilde{\lambda}_n(p))$, where

$$\tilde{\lambda}_j(p) = \sum_{i_1=1}^{s+d} \cdots \sum_{i_n=1}^{s+d} c^j_{i_1,\ldots,i_n} N_{i_1,\ldots,i_n}(p).$$

in which we approximate the constraints of P with linear functions and its objective with a quadratic function. First, to linearize the constraints of P, since the capacity constraints form an affine transformation of the demand function, we will simply linearize the demand function. For any $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times n}$, let $B_1, \ldots, B_n$ be the columns of $B$ and define $\theta_\iota = (a; B_1; \ldots; B_n) \in \mathbb{R}^{n^2+n}$, where the subscript $\iota$ stands for *linear demand*. We denote a linear demand function by $\lambda(p; \theta_\iota) = a + B'p$. Next, we explain how we use a quadratic function to approximate the objective of P. For any $E \in \mathbb{R}$, $F \in \mathbb{R}^n$, $G \in \mathbb{R}^{n \times n}$, let $G_1, \ldots, G_n$ denote the columns of $G$ and define $\theta_o = (E; F; G_1; \ldots; G_n) \in \mathbb{R}^{n^2+n+1}$, where the subscript $o$ stands for *objective*. We denote the resulting quadratic function by $q(p; \theta_o) = E + F'p + \frac{1}{2}p'Gp$. Finally, let $\theta = (\theta_o; \theta_\iota) \in \mathbb{R}^{2n^2+2n+1}$. For any $\theta \in \mathbb{R}^{2n^2+2n+1}$, $\delta \in \mathbb{R}^m$, we can define a quadratic program $\mathbf{QP}(\theta; \delta)$ as follows:

$$(\mathbf{QP}(\theta; \delta)) \qquad \max_{p \in \mathcal{P}} \left\{ q(p; \theta_o) : \ A\lambda(p; \theta_\iota) \preceq \frac{C}{T} - \delta \right\}.$$

It can be shown that quadratic program will have the same optimal solution as P and will possess some very useful stability properties if the parameters of the quadratic and linear functions are chosen as follows: for linear demand function, let $\theta^*_\iota = (a^*; B^*_1; \ldots; B^*_n)$, where $B^* := \nabla\lambda^*(p^D)$ and $a^* := \lambda^D - (B^*)'p^D$; for the quadratic objective function, let $\theta^*_o = (E^*; F^*; G^*_1; \ldots; G^*_n)$ where

$$E^* := \frac{1}{2}(p^D)'H^*p^D, \quad F^* := a^* - H^*p^D, \quad G^* := B^* + (B^*)' + H^*,$$

where $H^*$ is an $n$ by $n$ symmetric matrix defined as $H^* := B^* \nabla^2 r^*_\lambda(\lambda^D)(B^*)' - B^* - (B^*)'$. Finally, let $\theta^* := (\theta^*_o; \theta^*_t)$. Note that $\mathbf{QP}(\theta^*; \mathbf{0})$ is a very intuitive approximation of P since the function $\lambda(p; \theta^*_t) = a^* + (B^*)'p = \lambda^D + (B^*)'(p - p^D)$ can be viewed as a linearization of $\lambda^*(.)$ at $p^D$. Note also that the gradients of the objective function and the constraints in $\mathbf{QP}(\theta^*; \mathbf{0})$ at $p^D$ coincide with those in P. By Karush–Kuhn–Tucker (KKT) optimality conditions, it can be shown that the optimal solution of $\mathbf{QP}(\theta^*; \mathbf{0})$ is the same as the optimal solution of P.

We are now ready to describe *Nonparametric Self-adjusting Control* (NSC) and discuss its asymptotic performance. NSC consists of an exploration procedure and an exploitation procedure. The exploration procedure uses the Spline Estimation algorithm in Algorithm 4 to construct a spline approximation $\tilde{\lambda}(.)$ of the underlying demand function $\lambda^*(.)$. This function $\tilde{\lambda}(.)$ is then used to construct a linear function $\lambda(.; \hat{\theta}_l)$ that closely approximates $\lambda(.; \theta^*_l)$ in the neighborhood of $p^D$ and a quadratic program that closely approximates P. During the exploitation phase, we use the optimal solution of the approximate quadratic program as baseline control and automatically adjust the price according to a version of (5.28). Further details will be provided below. Recall that $\tilde{L}_0$ is the duration of the Spline Estimation algorithm. Let $C_t$ denote the remaining capacity at the *end* of period $t$. Let $\hat{\theta} := (\hat{\theta}_o; \hat{\theta}_l)$, where $\hat{\theta}_l := (\hat{a}; \hat{B}_1; \ldots; \hat{B}_n)$, $\hat{\theta}_o := (\hat{E}; \hat{F}; \hat{G}_1; \ldots; \hat{G}_n)$ and

$$\hat{B} := \nabla\tilde{\lambda}(\tilde{p}^D),\ \hat{a} := \tilde{\lambda} - \hat{B}'\tilde{p}^D,\ \hat{E} := \tfrac{1}{2}(\tilde{p}^D)'\hat{H}\tilde{p}^D,\ \hat{F} := \hat{a} - \hat{H}\tilde{p}^D,$$

$$\hat{G} := \hat{B} + \hat{B}' + \hat{H}, \quad \text{and}$$

$$\hat{H} = [\hat{H}_{ij}] \text{ where } \hat{H}_{ij} := -\hat{u}'_{ij}\hat{B}^{-1}\tilde{\lambda}^D \text{ and } \hat{u}_{ij} := \left[\frac{\partial^2\tilde{\lambda}_1(\tilde{p}^D)}{\partial p_i \partial p_j}; \ldots; \frac{\partial^2\tilde{\lambda}_n(\tilde{p}^D)}{\partial p_i \partial p_j}\right].$$

(Note that $\tilde{p}^D$ is the deterministic optimal solution of a version of P, where $\lambda^*$ is replaced by $\tilde{\lambda}$.) The details of NSC is given in Algorithm 5.

The following result states that the performance of NSC is close to the best achievable (asymptotic) performance bound.

**Theorem 4** *Suppose that* $s \geq 4$, $L_0 = \lceil (kT)^{(s+n/2)/(2s+n-2)} (\log(kT))^{(2s+n-4)/(2s+n-2)} \rceil$ *and* $d = \lceil (L_0^{1/2}/\log(kT))^{1/(s+n/2)} \rceil$. *There exists a constant* $M_1 > 0$ *independent of* $k > 3$ *such that for all* $s \geq 4$, *we have*

$$\rho^{NSC}(k) \leq M_1 k^{\frac{1}{2}+\epsilon(n,s,\bar{s})} \log k, \quad \textit{where } \epsilon(n, s, \bar{s}) = \frac{1}{2}\left(\frac{2s - 2(s \wedge \bar{s}) + n + 2}{2s + n - 2}\right).$$

Note that since most commonly used demand functions such as polynomial with arbitrary degree, logit, and exponential are infinitely differentiable (i.e., $\bar{s}$ can be arbitrarily large), for any fixed $\epsilon > 0$, we can select integers $s \geq (n+2)/(4\epsilon) - (n-2)/2$ such that the performance under NSC is $O(k^{1/2+\epsilon} \log k)$. Theoretically, this means that the asymptotic performance of NSC is very close to the best achievable performance lower bound of $\Omega(\sqrt{k})$. By comparing the algorithm and the analysis

**Algorithm 5** Nonparametric self-adjusting control (NSC)

Input Parameters: $n$, $s$ Tuning Parameter: $d$, $L_0$

1. *Stage 1 (Exploration Phase 1 - Spline Estimation)*

   a. For $t = 1$ to $\tilde{L}_0 \wedge T$

   - If $C_{t-1} \prec A_j$ for some $j = 1, \ldots, n$, set $p_{t,j} = p_j^\infty$ for all $j = 1, \ldots, n$.
   - Otherwise, follow Step 1 in *Spline Estimation* algorithm.

   b. At the end of period $\tilde{L}_0 \wedge T$, do:

   - If $\tilde{L}_0 \geq T$, terminate NSC.
   - If $\tilde{L}_0 < T$ and $C_{\tilde{L}_0} \prec A_j$ for some $j = 1, \ldots, n$:
     - For all $t > \tilde{L}_0$, set $p_{t,j} = p_j^\infty$ for all $j = 1, \ldots, n$.
     - Terminate NSC.
   - If $\tilde{L}_0 < T$ and $C_{\tilde{L}_0} \succeq A_j$ for all $j = 1, \ldots, n$:
     - Follow Step 2 in *Spline Estimation* algorithm to get $\tilde{\lambda}(.)$.
     - Go to Stage 2 below.

2. *Stage 2 (Exploration Phase 2 - Function Approximation)*

   a. Solve $\tilde{\mathbf{P}}$ and obtain the optimizer $\tilde{p}^D$.
   b. Let $\delta := C/T - C_{\tilde{L}_0}/(T - \tilde{L}_0)$.
   c. Compute $\hat{a}$, $\hat{B}$, $\hat{E}$, $\hat{F}$, $\hat{G}$, $\hat{H}$ and $\hat{\theta} = (\hat{\theta}_o; \hat{\theta}_t)$.

   - If $\hat{B}$ is invertible, go to Stage 2(d) below.
   - Otherwise, for $t = \tilde{L}_0 + 1$ to $T$:
     - If $C_{t-1} \succeq A_j$ for $j = 1, \ldots, n$, apply $p_t = \tilde{p}^D$.
     - Otherwise, for product $j = 1$ to $n$, do:
       - If $C_{t-1} \prec A_j$, set $p_{t,j} = p_j^\infty$.
       - Otherwise, set $p_{t,j} = p_{t-1,j}$.

   d. Solve $\mathbf{QP}(\hat{\theta}; \delta)$ for its static price $p_\delta^D(\hat{\theta})$.

3. *Stage 3 (Exploitation)*
   For $t = \tilde{L}_0 + 1$ to $T$:

   - Compute: $\hat{p}_t = p_\delta^D(\hat{\theta}) - \nabla_\lambda p(\lambda_\delta^D(\hat{\theta}); \hat{\theta}_t) \cdot \sum_{s=\tilde{L}_0+1}^{t-1} \frac{\tilde{\Delta}_s}{T-s}$, where $\tilde{\Delta}_t := D_t - \lambda(p_t; \hat{\theta}_t)$.
   - If $\hat{p}_t \in \mathcal{P}$ and $C_{t-1} \succeq A_j$ for $j = 1, \ldots, n$, apply $p_t = \hat{p}_t$.
   - Otherwise, for product $j = 1$ to $n$, do:
     - If $C_{t-1} \prec A_j$, set $p_{t,j} = p_j^\infty$.
     - Otherwise, set $p_{t,j} = p_{t-1,j}$.

of PSC and NSC, the extra $\epsilon$ in the exponent of the regret bound of NSC is driven by the slightly slower rate of convergence of the nonparametric approach for estimating

demand function. It remains an open question whether there exists a nonparametric approach for the NRM setting with a continuum of feasible price vectors which attains a regret bound of $O(\sqrt{k})$.

## 5.4 Bayesian Learning Setting

The multi-armed bandit (MAB) problem is often used to model the exploration–exploitation trade-off in the dynamic learning and pricing model *without* inventory constraints (see Chap. 1 for an overview of the MAB problem). In one of the earliest papers on the multi-armed bandit problem, Thompson (1933) proposed a novel randomized Bayesian algorithm, which has since been referred to as the *Thompson sampling* algorithm. The basic idea of Thompson sampling is that at each time period, random numbers are sampled according to the posterior distributions of the reward for each action, and then the action with the highest sampled reward is chosen. In a revenue management setting, each "action" or "arm" is a price, and "reward" refers to the revenue earned by offering that price. Thus, in the original Thompson sampling algorithm—in the absence of inventory constraints—random numbers are sampled according to the posterior distributions of the mean demand rates for each price, and the price with the highest sampled revenue (i.e., price times sampled demand) is offered.

In this section, we develop a class of Bayesian learning algorithms for the multiproduct pricing problem with inventory constraints. This class of algorithms extends the powerful machine learning technique known as Thompson sampling to address the challenge of balancing the exploration–exploitation trade-off under the presence of inventory constraints. We focus on a model with discrete price sets and present two algorithms (the algorithm can also be used for continuous price sets, see Ferreira et al. (2018)). The first algorithm adapts Thompson sampling by adding a linear programming (LP) subroutine to incorporate inventory constraints. The second algorithm builds upon our first; specifically, in each period, we modify the LP subroutine to further account for the purchases made to date. Both of the algorithms contain two simple steps in each iteration: sampling from a posterior distribution and solving a linear program. As a result, the algorithms are easy to implement in practice.

### *5.4.1 Model Setting*

We consider a retailer who sells $N$ products, indexed by $i \in [N]$, over a finite selling season. (Below, we denote by $[x]$ the set $\{1, 2, \ldots, x\}$.) These products consume $M$ resources, indexed by $j \in [M]$. Specifically, we assume that one unit of product $i$ consumes $a_{ij}$ units of resource $j$, where $a_{ij}$ is a fixed constant. The selling season is divided into $T$ periods. There are $I_j$ units of initial inventory for each resource

$j \in [M]$, and there is no replenishment during the selling season. We define $I_j(t)$ as the inventory at the end of period $t$, and we denote $I_j(0) = I_j$. In each period $t \in [T]$, the following sequence of events occurs:

1. The retailer offers a price for each product from a finite set of admissible price vectors. We denote this set by $\{p_1, p_2, \ldots, p_K\}$, where $p_k$ ($\forall k \in [K]$) is a vector of length $N$ specifying the price of each product. More specifically, we have $p_k = (p_{1k}, \ldots, p_{Nk})$, where $p_{ik}$ is the price of product $i$, for all $i \in [N]$. Following the tradition in dynamic pricing literature, we also assume that there is a "shut-off" price $p_\infty$ such that the demand for any product under this price is zero with probability one. We denote by $P(t) = (P_1(t), \ldots, P_N(t))$ the prices chosen by the retailer in this period, and require that $P(t) \in \{p_1, p_2, \ldots, p_K, p_\infty\}$.
2. Customers then observe the prices chosen by the retailer and make purchase decisions. We denote by $D(t) = (D_1(t), \ldots, D_N(t))$ the demand of each product at period $t$. We assume that given $P(t) = p_k$, the demand $D(t)$ is sampled from a probability distribution on $\mathbb{R}_+^N$ with joint cumulative distribution function (CDF) $F(x_1, \ldots, x_N; p_k, \theta)$, indexed by a parameter (or a vector of parameters) $\theta$ that takes values in the parameter space $\Theta \subset \mathbb{R}^l$. The distribution is assumed to be subexponential; note that many commonly used demand distributions such as normal, Poisson, exponential and all bounded distributions belong to the family of subexponential distributions. We also assume that $D(t)$ is independent of the history $\mathcal{H}_{t-1} = (P(1), D(1), \ldots, P(t-1), D(t-1))$ given $P(t)$.

   Depending on whether there is sufficient inventory, one of the following events happens:

   (a) If there is enough inventory to satisfy all demand, the retailer receives an amount of revenue equal to $\sum_{i=1}^N D_i(t) P_i(t)$, and the inventory level of each resource $j \in [M]$ diminishes by the amount of each resource used such that $I_j(t) = I_j(t-1) - \sum_{i=1}^N D_i(t) a_{ij}$.
   (b) If there is not enough inventory to satisfy all demand, the demand is partially satisfied and the rest of demand is lost. Let $\tilde{D}_i(t)$ be the demand satisfied for product $i$. We require $\tilde{D}_i(t)$ to satisfy three conditions: $0 \le \tilde{D}_i(t) \le D_i(t), \forall i \in [N]$; the inventory level for each resource at the end of this period is nonnegative: $I_j(t) = I_j(t-1) - \sum_{i=1}^N \tilde{D}_i(t) a_{ij} \ge 0, \forall j \in [M]$; there exists at least one resource $j' \in [M]$ whose inventory level is zero at the end of this period, i.e. $I_{j'}(t) = 0$. Besides these natural conditions, we do not require any additional assumption on how demand is specifically fulfilled. The retailer then receives an amount of revenue equal to $\sum_{i=1}^N \tilde{D}_i(t) P_i(t)$ in this period.

We assume that the demand parameter $\theta$ is fixed but *unknown* to the retailer at the beginning of the season, and the retailer must learn the true value of $\theta$ from demand data. That is, in each period $t \in [T]$, the price vector $P(t)$ can only be chosen based on the observed history $\mathcal{H}_{t-1}$, but cannot depend on the unknown value $\theta$ or any event in the future. The retailer's objective is to maximize expected revenue over the course of the selling season given the prior distribution on $\theta$.

We use a parametric Bayesian approach in our model, where the retailer has a *known* prior distribution of $\theta \in \Theta$ at the beginning of the selling season. However, our model allows the retailer to choose an arbitrary prior. In particular, the retailer can assume an arbitrary parametric form of the demand CDF, given by $F(x_1, \ldots, x_N; p_k, \theta)$. This joint CDF parametrized by $\theta$ can parsimoniously model the correlation of demand among products. For example, the retailer may specify products' joint demand distribution based on some discrete choice model, where $\theta$ is the unknown parameter in the multinomial logit function. Another benefit of the Bayesian approach is that the retailer may choose a prior distribution over $\theta$ such that demand is correlated for different prices, enabling the retailer to learn demand for all prices, not just the offered price. e selling season as inventory is depleted; this latter idea is incorporated into the second algorithm that we will present later.

### 5.4.2 Thompson Sampling with Fixed Inventory Constraints

We now present the first version of the Thompson sampling-based pricing algorithm. For each resource $j \in [M]$, we define a fixed constant $c_j := I_j/T$. Given any demand parameter $\rho \in \Theta$, we define the mean demand under $\rho$ as the expectation associated with CDF $F(x_1, \ldots, x_N; p_k, \rho)$ for each product $i \in [N]$ and price vector $k \in [K]$. We denote by $d = \{d_{ik}\}_{i\in[N],k\in[K]}$ the mean demand under the *true* model parameter $\theta$.

The Thompson sampling with Fixed Inventory Constraints (TS-fixed) algorithm is shown in Algorithm 6. Here, "TS" stands for Thompson sampling, while "fixed" refers to the fact that we use fixed constants $c_j$ for all time periods as opposed to updating $c_j$ over the selling season as inventory is depleted; this latter idea is incorporated into the second algorithm that we will present later.

Steps 1 and 4 are based on the Thompson sampling algorithm for the classical multi-armed bandit setting, whereas Steps 2 and 3 are added to incorporate inventory constraints. In Step 1 of the algorithm, we randomly sample parameter $\theta(t)$ according to the posterior distribution of unknown demand parameter $\theta$. This step is motivated by the original Thompson sampling algorithm for the classical multi-armed bandit problem. The key idea of the Thompson sampling algorithm is to use random sampling from the posterior distribution to balance the exploration–exploitation trade-off. The algorithm differs from the ordinary Thompson sampling in Steps 2 and 3. In Step 2, the retailer solves a linear program, LP($d(t)$), which identifies the optimal mixed price strategy that maximizes expected revenue given the sampled parameters. The first constraint specifies that the average resource consumption in this time period cannot exceed $c_j$, the average inventory available per period. The second constraint specifies that the sum of probabilities of choosing a price vector cannot exceed one. In Step 3, the retailer randomly offers one of the $K$ price vectors (or $p_\infty$) according to probabilities specified by the optimal solution of LP($d(t)$). Finally, in Step 4, the algorithm updates the posterior distribution of $\theta$ given $\mathcal{H}_t$. Such Bayesian updating is a simple and powerful tool to update belief

**Algorithm 6** Thompson sampling with fixed inventory constraints (TS-fixed)

Repeat the following steps for all periods $t = 1, \ldots, T$:

1. *Sample Demand*: Sample a random parameter $\theta(t) \in \Theta$ according to the posterior distribution of $\theta$ given history $\mathcal{H}_{t-1}$. Let the mean demand under $\theta(t)$ be $d(t) = \{d_{ik}(t)\}_{i\in[N],k\in[K]}$.
2. *Optimize Prices given Sampled Demand*: Solve the following linear program, denoted by $\mathsf{LP}(d(t))$:

$$\begin{aligned} \mathsf{LP}(d(t)): \quad \max_{x} \ & \sum_{k=1}^{K}\Big(\sum_{i=1}^{N} p_{ik} d_{ik}(t)\Big)x_k \\ \text{subject to } & \sum_{k=1}^{K}\Big(\sum_{i=1}^{N} a_{ij} d_{ik}(t)\Big)x_k \le c_j, \ \forall j \in [M] \\ & \sum_{k=1}^{K} x_k \le 1 \\ & x_k \ge 0, \ k \in [K]. \end{aligned}$$

   Let $x(t) = (x_1(t), \ldots, x_K(t))$ be the optimal solution to $\mathsf{LP}(d(t))$.
3. *Offer Price*: Offer price vector $P(t) = p_k$ with probability $x_k(t)$, and choose $P(t) = p_\infty$ with probability $1 - \sum_{k=1}^{K} x_k(t)$.
4. *Update Estimate of Parameter*: Observe demand $D(t)$. Update the history $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{P(t), D(t)\}$ and the posterior distribution of $\theta$ given $\mathcal{H}_t$.

probabilities as more information—customer purchase decisions in our case—becomes available. By employing Bayesian updating in Step 4, we are ensured that as any price vector $p_k$ is offered more and more times, the sampled mean demand associated with $p_k$ for each product $i$ becomes more and more centered around the true mean demand, $d_{ik}$.

We note that the LP defined in Step 2 is closely related to the LP used by Gallego and Van Ryzin (1997), where they consider a network revenue management problem in the case of known demand. Essentially, their pricing algorithm is a special case of Algorithm 6 where they solve $\mathsf{LP}(d)$, i.e., $\mathsf{LP}(d(t))$ with $d(t) = d$, in every time period.

Next, we illustrate the application of our TS-fixed algorithm by providing one concrete example. For simplicity, in this example, we assume that the prior distribution of demand for different prices is independent; however, the definition of TS-fixed is quite general and allows the prior distribution to be *arbitrarily correlated* for different prices. As mentioned earlier, this enables the retailer to learn the mean demand not only for the offered price but also for prices that are not offered.

*Example (Bernoulli Demand with Independent Uniform Prior)* We assume that for all prices, the demand for each product is Bernoulli distributed. In this case, the unknown parameter $\theta$ is just the mean demand of each product. We use a beta

posterior distribution for each $\theta$ because it is conjugate to the Bernoulli distribution. We assume that the prior distribution of mean demand $d_{ik}$ is uniform in [0, 1] (which is equivalent to a Beta(1, 1) distribution) and is independent for all $i \in [N]$ and $k \in [K]$. In this example, the posterior distribution is very simple to calculate. Let $N_k(t-1)$ be the number of time periods that the retailer has offered price $p_k$ in the first $t-1$ periods, and let $W_{ik}(t-1)$ be the number of periods that product $i$ is purchased under price $p_k$ during these periods. In Step 1 of TS-fixed, the posterior distribution of $d_{ik}$ is Beta$(W_{ik}(t-1)+1, N_k(t-1)-W_{ik}(t-1)+1)$, so we sample $d_{ik}(t)$ independently from a Beta$(W_{ik}(t-1)+1, N_k(t-1)-W_{ik}(t-1)+1)$ distribution for each price $k$ and each product $i$. In Steps 2 and 3, LP$(d(t))$ is solved and a price vector $p_{k'}$ is chosen; then, the customer demand $D_i(t)$ is revealed to the retailer. In Step 4, we then update $N_{k'}(t) \leftarrow N_{k'}(t-1)+1$, $W_{ik'}(t) \leftarrow W_{ik'}(t-1)+D_i(t)$ for all $i \in [N]$. The posterior distributions associated with the $K-1$ unchosen price vectors ($k \neq k'$) are not changed.

### 5.4.3 Thompson Sampling with Inventory Constraint Updating

Now, we propose the second Thompson sampling-based algorithm. Recall that in TS-fixed, we use fixed inventory constants $c_j$ in every period. Alternatively, we can update $c_j$ over the selling season as inventory is depleted, thereby incorporating real-time inventory information into the algorithm.

In particular, we recall that $I_j(t)$ is the inventory level of resource $j$ at the end of period $t$. Define $c_j(t) = I_j(t-1)/(T-t+1)$ as the average inventory for resource $j$ available from period $t$ to period $T$. We then replace constants $c_j$ with $c_j(t)$ in LP$(d(t))$ in step 2 of TS-fixed, which gives us the Thompson sampling with Inventory Constraint Updating algorithm (TS-update for short) shown in Algorithm 7. The term "update" refers to the fact that in every iteration, the algorithm updates inventory constants $c_j(t)$ in LP$(d(t))$ to incorporate real-time inventory information.

In the revenue management literature, the idea of using updated inventory rates like $c_j(t)$ has been previously studied in various settings (Jasin and Kumar, 2012; Jasin, 2014). TS-update is an algorithm that incorporates real-time inventory updating when the retailer faces an exploration–exploitation trade-off with its pricing decisions. Although intuitively incorporating updated inventory information into the pricing algorithm should improve the performance of the algorithm, Cooper (2002) provides a counterexample where the expected revenue is reduced after the updated inventory information is included. Therefore, it is not immediately clear if TS-update would achieve higher revenue than TS-fixed. We will rigorously analyze the performance of both TS-fixed and TS-update in the next section; our numerical simulation shows that in fact there are situations where TS-update outperforms TS-fixed and vice versa.

**Algorithm 7** Thompson sampling with inventory constraint updating (TS-update)

Repeat the following steps for all periods $t = 1, \ldots, T$:

1. *Sample Demand*: Sample a random parameter $\theta(t) \in \Theta$ according to the posterior distribution of $\theta$ given history $\mathcal{H}_{t-1}$. Let the mean demand under $\theta(t)$ be $d(t) = \{d_{ik}(t)\}_{i\in[N],k\in[K]}$.
2. *Optimize Prices given Sampled Demand*: Solve the following linear program, denoted by $\mathsf{LP}(d(t), c(t))$:

$$\begin{aligned} \mathsf{LP}(d(t), c(t)): \quad \max_{x} \ & \sum_{k=1}^{K}\Big(\sum_{i=1}^{N} p_{ik} d_{ik}(t)\Big) x_k \\ \text{subject to } & \sum_{k=1}^{K}\Big(\sum_{i=1}^{N} a_{ij} d_{ik}(t)\Big) x_k \le c_j(t), \ \forall j \in [M] \\ & \sum_{k=1}^{K} x_k \le 1 \\ & x_k \ge 0, \ k \in [K]. \end{aligned}$$

   Let $x(t) = (x_1(t), \ldots, x_K(t))$ be the optimal solution to $\mathsf{LP}(d(t), c(t))$.
3. *Offer Price*: Offer price vector $P(t) = p_k$ with probability $x_k(t)$, and choose $P(t) = p_\infty$ with probability $1 - \sum_{k=1}^{K} x_k(t)$.
4. *Update Estimate of Parameter*: Observe demand $D(t)$. Update the history $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{P(t), D(t)\}$ and the posterior distribution of $\theta$ given $\mathcal{H}_t$.

### 5.4.4 Performance Analysis

To evaluate the proposed Bayesian learning algorithms, we compare the retailer's revenue with a benchmark where the true demand distribution is known a priori. We define the retailer's *regret* over the selling horizon as

$$\text{Regret}(T, \theta) = \text{E}[\text{Rev}^*(T) \mid \theta] - \text{E}[\text{Rev}(T) \mid \theta],$$

where $\text{Rev}^*(T)$ is the revenue achieved by the optimal policy if the demand parameter $\theta$ is known a priori, and $\text{Rev}(T)$ is the revenue achieved by an algorithm that may not know $\theta$. The conditional expectation is taken on random demand realizations given $\theta$ and possibly on some external randomization used by the algorithm (e.g., random samples in Thompson sampling). In words, the regret is a nonnegative quantity measuring the retailer's revenue loss due to not knowing the latent demand parameter.

We also define the *Bayesian regret* (also known as *Bayes risk*) by

$$\text{BayesRegret}(T) = E[\text{Regret}(T, \theta)],$$

where the expectation is taken over the prior distribution of $\theta$.

We now prove regret bounds for TS-fixed and TS-update under the realistic assumption of bounded demand. Specifically, in the following analysis, we further assume that for each product $i \in [N]$, the demand $D_i(t)$ is bounded by $D_i(t) \in [0, \bar{d}_i]$ under any price vector $p_k, \forall k \in [K]$. However, the result can be generalized when the demand is unbounded and follows a sub-Gaussian distribution. We also define the constants

$$p_{\max} := \max_{k\in[K]} \sum_{i=1}^{N} p_{ik}\bar{d}_i, \quad p^j_{\max} := \max_{i\in[N]:a_{ij}\neq 0, k\in[K]} \frac{p_{ik}}{a_{ij}}, \ \forall j \in [M],$$

where $p_{\max}$ is the maximum revenue that can possibly be achieved in one period, and $p^j_{\max}$ is the maximum revenue that can possibly be achieved by adding one unit of resource $j, \forall j \in [M]$.

**Theorem 5** *The Bayesian regret of* TS-fixed *is bounded by*

$$\text{BayesRegret}(T) \leq \left(18 p_{\max} + 37 \sum_{i=1}^{N}\sum_{j=1}^{M} p^j_{\max} a_{ij} \bar{d}_i\right) \sqrt{TK \log K}.$$

**Theorem 6** *The Bayesian regret of* TS-update *is bounded by*

$$\text{BayesRegret}(T) \leq \left(18 p_{\max} + 40 \sum_{i=1}^{N}\sum_{j=1}^{M} p^j_{\max} a_{ij} \bar{d}_i\right) \sqrt{TK \log K} + p_{\max} M.$$

The results above state that the Bayesian regrets of both TS-fixed and TS-update are bounded by $O(\sqrt{TK \log K})$, where $K$ is the number of price vectors that the retailer is allowed to use and $T$ is the number of time periods. Moreover, the regret bounds are *prior-free* as they do not depend on the prior distribution of parameter $\theta$; the constants in the bounds can be computed explicitly without knowing the demand distribution.

It has been shown that for a multi-armed bandit problem with reward in [0, 1]—a special case of our model with no inventory constraints—no algorithm can achieve a prior-free Bayesian regret smaller than $\Omega(\sqrt{KT})$ (see Theorem 3.5, Bubeck and Cesa-Bianchi 2012). In that sense, the above regret bounds are optimal with respect to $T$ and cannot be improved by any other algorithm by more than $\sqrt{\log K}$.

Note that the regret bound of TS-update is slightly worse than the regret bound of TS-fixed. Although intuition would suggest that updating inventory information in TS-update will lead to better performance than TS-fixed, this intuition is somewhat surprisingly not always true—we can find counterexamples where updating inventory information actually deteriorates the performance for any given horizon length $T$.

The detailed proofs of Theorems 5 and 6 are omitted. We briefly summarize the intuition behind the proofs. For both Theorems 5 and 6, we first assume an "ideal" scenario where the retailer is able to collect revenue even after inventory runs out. We show that if prices are given according to the solutions of TS-fixed or TS-update, the expected revenue achieved by the retailer is within $O(\sqrt{T})$ compared to the optimal revenue $Rev^*(T)$. However, this argument overestimates the expected revenue. In order to compute the actual revenue given constrained inventory, we should account for the amount of revenue that is associated with lost sales. For Theorem 5 (TS-fixed), we prove that the amount associated with lost sales is no more than $O(\sqrt{T})$. For Theorem 6 (TS-update), we show that the amount associated with lost sales is no more than $O(1)$.

## 5.5 Remarks and Further Reading

The content of Sect. 5.2 is based on Wang (2012) and Wang et al. (2014). For the proofs of the main results, the readers are referred to Wang et al. (2014). In Wang et al. (2014), there are also implementation suggestions for the proposed algorithms. Note that in practical implementation, the algorithm can be made more efficient by relaxing some requirements stated in the Algorithm 1. Extensive numerical experiments and comparison with other algorithms can be found in Wang (2012) and Wang et al. (2014). Later, Lei et al. (2014) improve the result of Theorem 1 to remove the logarithmic factor in the worst-case regret using a bisection type of method. For details of the algorithm and the analysis, we refer the readers to Lei et al. (2014).

Section 5.3 is adapted from Chen et al. (2019) and Chen et al. (2021), which contain full proofs of the theorems presented and additional numerical studies. Chen et al. (2021) further considers a well-separated condition of demand functions and derive a much sharper $O(\log^2 k)$ regret than the $O(\sqrt{k})$ regret in the general demand case.

Section 5.4 is primarily based on Ferreira et al. (2018). The definition of Bayesian regret used in this section is a standard metric for the performance of online Bayesian algorithms, see Russo and Van Roy (2014). Ferreira et al. (2018) also developed the Thompson sampling algorithms for the linear demand case and the *bandits with knapsack* problem, see Badanidiyuru et al. (2013).

Other methods have been proposed in the literature to address learning and pricing problems in the constrained inventory setting. One approach is to separate the selling season ($T$ periods) into a disjoint exploration phase (say, from period 1 to $\tau$) and exploitation phase (from period $\tau + 1$ to $T$) (Besbes and Zeevi, 2009, 2012). One drawback of this strategy is that it does not use purchasing data after period $\tau$ to continuously refine demand estimates. Furthermore, when there is very limited inventory, this approach is susceptible to running out of inventory during the exploration phase before any demand learning can be exploited. Another approach is to use multi-armed bandit methods such as the upper confidence bound (UCB)

algorithm (Auer et al., 2002) to make pricing decisions in each period. The UCB algorithm creates a confidence interval for unknown demand using purchase data and then selects a price that maximizes revenue among all parameter values in the confidence set. We refer the readers to Badanidiyuru et al. (2013) and Agrawal and Devanur (2014) for UCB algorithms with constrained inventory.

**Acknowledgments** This chapter is partially based on material copyrighted by INFORMS and is republished with permission.

## References

Agrawal, S., & Devanur, N. R. (2014). Bandits with concave rewards and convex knapsacks. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation* (pp. 989–1006)

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*(2–3), 235–256.

Badanidiyuru, A., Kleinberg, R., & Slivkins, A. (2013). Bandits with knapsacks. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 207–216).

Besbes, O., & Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, *57*(6), 1407–1420.

Besbes, O., & Zeevi, A. (2012). Blind network revenue management. *Operations Research*, *60*(6), 1537–1550.

Broder, J., & Rusmevichientong, P. (2012). Dynamic pricing under a general parametric choice model. *Operations Research*, *60*(4), 965–980.

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, *5*(1), 1–122.

Chen, Q., Jasin, S., & Duenyas, I. (2019). Nonparametric self-adjusting control for joint learning and optimization of multiproduct pricing with finite resource capacity. *Mathematics of Operations Research*, *44*(2), 601–631.

Chen, Q., Jasin, S., & Duenyas, I. (2021). Joint learning and optimization of multi-product pricing with finite resource capacity and unknown demand parameters. *Operations Research*, *69*(2), 560–573.

Cooper, W. L. (2002). Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, *50*(4), 720–727.

Ferreira, K. J., Simchi-Levi, D., & Wang, H. (2018). Online network revenue management using Thompson sampling. *Operations Research*, *66*(6), 1586–1602.

Gallego, G., & van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, *40*(8), 999–1029.

Gallego, G., & Van Ryzin, G. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, *45*(1), 24–41.

Gyorfi, L., Kohler, M., Krzyzak, A., & Walk, H. (2002). *A distribution-free theory of nonparametric regression*. Springer.

Jasin, S. (2014). Reoptimization and self-adjusting price control for network revenue management. *Operations Research*, *62*(5), 1168–1178.

Jasin, S., & Kumar, S. (2012). A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, *37*(2), 313–345.

Lei, Y. M., Jasin, S., & Sinha, A. (2014). Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint, in *Working Paper*.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, *39*(4), 1221–1243.

Schumaker, L. (2007). *Spline functions: Basic theory* (3rd ed.). Cambridge University Press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3/4), 285–294.

Wang, Z. (2012). *Dynamic learning mechanism in revenue management problems*. PhD thesis, Stanford University, Palo Alto.

Wang, Z., Deng, S., & Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, *62*(2), 318–331.

# Chapter 6
# Dynamic Pricing and Demand Learning in Nonstationary Environments


**Arnoud V. den Boer and Nuri Bora Keskin**

## 6.1 Introduction

The demand for a seller's product can be nonstationary in many business settings. This could be due to exogenous factors such as macroeconomic issues and fashion trends. It could also be based on an endogenous mechanism that depends on pricing decisions—examples of this include reference-price effects and price competition.

As noted in preceding chapters, learning the relationship between price and demand while simultaneously trying to earn revenues is a key challenge, even in stationary demand environments. In nonstationary demand environments, this challenge would also entail judiciously filtering obsolete historical information. There are several ways to accomplish this task, depending on the nature of changes. For exogenous changes, statistical filtering methods such as change-point detection and smoothing can be useful. For endogenous changes, the seller would need to take additional care in controlling the price process. The goal of this chapter is to provide an overview of the state-of-the-art models and methods for dynamic pricing and demand learning in different kinds of changing demand environments, as well as to provide some research directions for future work.

A. V. den Boer
Korteweg-de Vries Institute for Mathematics and Amsterdam Business School, University of Amsterdam, Amsterdam, Netherlands
e-mail: boer@uva.nl

N. B. Keskin (✉)
Fuqua School of Business, Duke University, Durham, NC, USA
e-mail: bora.keskin@duke.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_6

## 6.2 Problem Formulation

The distinguishing feature of the problem formulations we consider in this chapter is the *non-stationarity* of the demand environment. We consider a seller offering a product for sale over a discrete time horizon. In each period $t \in \mathbb{N}$, the seller first chooses a selling price $p_t \in [p_{\min}, p_{\max}]$ for the product, where $0 \leq p_{\min} < p_{\max}$. After that, the seller observes the demand $D_t$ realized in response to price $p_t$ and collects revenue $p_t D_t$. The demand realizations are given by

$$D_t = d_t(p_t) + \epsilon_t \text{ for all } t \in \mathbb{N},$$

where

$$\boldsymbol{d} := \{d_t(\cdot) : t \in \mathbb{N}\}$$

is a sequence of demand functions such that for all $t \in \mathbb{N}$, $d_t : [p_{\min}, p_{\max}] \to [0, \infty)$ is a continuous and nonincreasing mapping, and $\{\epsilon_t : t \in \mathbb{N}\}$ is a sequence of unobservable demand shocks. Suppose that $\{\epsilon_t : t \in \mathbb{N}\}$ consists of independent and identically distributed random variables with zero mean and variance equal to $\sigma^2$ for some $\sigma > 0$, and that there exists $x_0 > 0$ satisfying $\mathbb{E}[\exp(x\epsilon_t)] < \infty$ for all $|x| \leq x_0$ and all $t$. In this construction, $d_t(p)$ represents the expected demand in period $t$ as a function of price $p$. Accordingly, the seller's expected revenue in period $t$, as a function of price $p$, is

$$r_t(p) := p d_t(p) \text{ for } p \in [p_{\min}, p_{\max}].$$

The sequence of demand functions, $\boldsymbol{d} = \{d_t(\cdot) : t \in \mathbb{N}\}$, is unknown to the seller. Therefore, to *earn* higher revenues, the seller needs to *learn* the demand function sequence. As discussed below, different studies consider different nonstationary families of demand function sequences, resulting in several distinct ways to balance learning and earning. To choose the selling price $p_t$ in period $t$, the seller uses the *history* $(p_1, D_1, \ldots, p_{t-1}, D_{t-1})$ of past prices and demand realizations. To be precise, we let $\pi(\cdot \mid h_{t-1})$ be the probability distribution of $p_t$ conditional on the history $h_{t-1} = (p_1, D_1, \ldots, p_{t-1}, D_{t-1}) \in H = \bigcup_{t \in \mathbb{N}} \{[p_{\min}, p_{\max}] \times \mathcal{D}\}^{t-1}$, where $\mathcal{D} \subset \mathbb{R}$ is the set of all possible demand realizations. The seller's price decisions over the time horizon are characterized by the collection $\{\pi(\cdot \mid h) : h \in H\}$. We refer to this collection as an *admissible policy*, and let $\Pi$ denote the space of all admissible policies. The vector of prices and demand realizations, $(p_t, D_t : t \in \mathbb{N})$, has a distribution that depends on both the policy $\pi$ and the demand function sequence $\boldsymbol{d}$. We write $\mathbb{P}^{\pi}_{\boldsymbol{d}}\{\cdot\}$ to denote the probability measure governing this distribution and $\mathbb{E}^{\pi}_{\boldsymbol{d}}[\cdot]$ to denote the associated expectation operator.

The seller aims to minimize the expected revenue loss due to not knowing the underlying demand function sequence $\boldsymbol{d}$. In accordance with this, we measure the performance of a policy by its *regret* after $T$ periods, which is defined as

$$R_{\boldsymbol{d}}^{\pi}(T) := \sum_{t=1}^{T} \max_{p \in [p_{\min}, p_{\max}]} \{r_t(p)\} - \mathbb{E}_{\boldsymbol{d}}^{\pi}\left[\sum_{t=1}^{T} p_t D_t\right] \quad \text{for } T \in \mathbb{N}.$$

Note that lower values of regret are more desirable to the seller. Due to the high-dimensionality of the dynamic pricing problem in changing environments, it is prohibitively difficult to obtain an exactly optimal policy except in a few special cases. Consequently, we are primarily interested in finding *asymptotically optimal* policies that minimize the growth rate of regret in $T$ under different assumptions on the sequence of demand functions.

In what follows, we examine various approaches used for modeling nonstationary demand environments. At a high level, the non-stationarity of a demand environment can be of two forms: the environment can change either *exogenously* or *endogenously*, depending on whether the changes are independent of the seller's pricing decisions or not. Exogenously changing demand environments include change-point detection models, finite-state-space Markov chains, and autoregressive models. Recent studies also consider more general exogenously changing environments that encapsulate the aforementioned settings. On the other hand, endogenously changing demand environments are concerned with dynamic pricing in the presence of reference effects, competition, multi-agent learning, and forward-looking customers. We discuss all of these cases in the following two sections.

## 6.3 Exogenously Changing Demand Environments

### *6.3.1 Change-Point Detection Models*

Change-point detection research focuses on identifying changes in a time series. The early statistics literature on this subject is primarily motivated by military and quality control applications; see the surveys by Lai (1995) and Shiryaev (2010). In the context of dynamic pricing, the change-point detection framework can be generalized to identify temporal shifts in a demand function. Suppose that the demand function sequence $\boldsymbol{d} = \{d_t(\cdot) : t \in \mathbb{N}\}$ introduced in the preceding section is a constant sequence except at one period. That is, there exists a period $\tau_0 \in \mathbb{N}$ and two distinct demand functions $f_0(\cdot)$ and $f_1(\cdot)$ such that for all $t \in \mathbb{N}$,

$$d_t(\cdot) \equiv \begin{cases} f_0(\cdot) & \text{if } t < \tau_0, \\ f_1(\cdot) & \text{if } t \geq \tau_0. \end{cases}$$

Besbes and Zeevi (2011) consider a version of this problem where $f_0(\cdot)$ and $f_1(\cdot)$ are known to the seller but $\tau_0$ is unknown. They show that it is possible to achieve a $T$-period regret in the order of $\log T$, which is the smallest possible growth rate of regret in their setting. To be precise, Besbes and Zeevi (2011) propose a passive

detection policy $\pi$ that repeatedly checks whether there is a statistically significant shift in the expected demand under a fixed price. They prove that there is a finite and positive constant $C$ such that $R_{\boldsymbol{d}}^{\pi}(T) \leq C \log T$ for all $T = 2, 3, \ldots$ (Besbes & Zeevi, 2011, section 4.2). They also provide a lower bound on regret of matching order, indicating that passive detection is asymptotically optimal in this setting (Besbes & Zeevi, 2011, section 4.3).

Keskin and Zeevi (2017) study a generalized version of the above problem where multiple change points are allowed and the seller knows neither the possible demand functions, nor when the changes happen, nor the number of changes. As a result, the seller needs to simultaneously *learn* the demand functions and *detect* potential changes. Keskin and Zeevi (2017) design a joint learning-and-detection policy and show that this policy achieves a $T$-period regret in the order of $\sqrt{T}$, up to logarithmic terms (Keskin & Zeevi, 2017, section 4.2). Based on earlier lower bounds on regret (e.g., Keskin & Zeevi, 2014, section 3.1), this establishes that the joint learning-and-detection policy of Keskin and Zeevi (2017) is asymptotically optimal.

In a recent study, den Boer and Keskin (2020) generalize this research stream to analyze *discontinuous* demand functions, which arise in network pricing problems as well as online marketplaces featuring price-based rankings (den Boer & Keskin, 2020, section 1.2). They consider demand functions with multiple discontinuities whose locations and magnitudes are unknown and may change over time. den Boer and Keskin (2020) develop a policy that efficiently estimates potential discontinuities in the demand function, while jointly learning the demand function and detecting potential changes. They prove that the $T$-period regret of this policy is of order $\sqrt{T}$, up to logarithmic terms (den Boer & Keskin, 2020, section 4). Thus, the generalized discontinuity-estimation policy of den Boer and Keskin (2020) achieves exhibits asymptotically optimal regret performance, in light of the aforementioned lower bound of Keskin and Zeevi (2014, section 3.1).

Keskin et al. (2022) further extend this literature to the case of joint pricing and inventory decisions. The introduction of inventory management to this problem formulation makes the seller's regret more sensitive to the assumptions on temporal demand shocks, $\{\epsilon_t : t \in \mathbb{N}\}$. Keskin et al. (2022) consider both nonparametric and parametric demand shock distributions and develop a distinct regret bound for each case (Keskin et al., 2022, section 4.1).

### 6.3.2 Finite-State-Space Markov Chains

A common way to model a nonstationary environment is to use a Markov chain. Consider two distinct demand functions $f_0(\cdot)$ and $f_1(\cdot)$, and a demand function sequence $\boldsymbol{d} = \{d_t(\cdot) : t \in \mathbb{N}\}$ that evolves as a discrete-time Markov chain on the state space $\{f_0(\cdot), f_1(\cdot)\}$. To be more precise, let

$$M_t := \mathbb{I}\{d_t(\cdot) \equiv f_1(\cdot)\} \text{ for all } t \in \mathbb{N},$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function (i.e., given condition $A$, $\mathbb{I}\{A\} = 1$ if $A$ holds and 0 otherwise). Suppose that under any admissible policy $\pi$,

$$\mathbb{P}_{\boldsymbol{d}}^{\pi}\{M_{t+1} = j \mid M_t = i\} = \rho_{i,j} \quad \text{for all } t \in \mathbb{N},$$

where $\rho_{i,j} \in (0, 1)$ for $i, j \in \{0, 1\}$ such that $\sum_{j\in\{0,1\}} \rho_{i,j} = 1$ for $i \in \{0, 1\}$.

Rustichini and Wolinsky (1995) consider a seller who receives demand in the form of a step function; i.e., for all $t \in \mathbb{N}$ and $p \in [p_{\min}, p_{\max}]$, $d_t(p) = \mathbb{I}\{w_t \geq p\}$ for some $w_t > 0$. They assume that the sequence $\{w_t : t \in \mathbb{N}\}$ follows a two-state discrete-time Markov chain and examine the structural properties of the seller's optimal Bayesian policy. Keller and Rady (1999) consider linear demand functions in a continuous-time version of this Markovian setting; i.e., for all $t \in \mathbb{N}$ and $p \in [p_{\min}, p_{\max}]$, $d_t(p) = \alpha_t - \beta_t p$ for some $\alpha_t, \beta_t > 0$, and $\{(\alpha_t, \beta_t) : t \in \mathbb{N}\}$ evolves as a two-state continuous-time Markov chain. Using stochastic control theory, Keller and Rady (1999) characterize the optimal Bayesian policy and study how this policy uses experimentation in different scenarios.

As the state space of the underlying Markov chain grows, it becomes prohibitively difficult to compute an optimal policy. Because of this, Aviv and Pazgal (2005) focus on developing approximately optimal pricing policies when the Markovian demand environment has a larger state space. In particular, they derive an upper bound on the seller's optimal cumulative revenue, and use this upper bound to construct an approximately optimal policy based on dynamic programming. Chen and Farias (2013) consider a continuous-time formulation in which the market size evolves as a Gaussian process while the price sensitivity of demand does not change over time. In this setting, they show that a policy that frequently reoptimizes prices based on most recent information can perform well. In a recent study, Keskin and Li (2020) analyze dynamic pricing in a Markovian demand environment with unknown transition probabilities. They prove that *bounding* the seller's belief process can yield asymptotically optimal regret performance. Specifically, Keskin and Li (2020) establish that the $T$-period regret of their bounded learning policy is of order $\sqrt{nT}$, where $n$ is an upper bound the expected number of changes. They also show that the $T$-period regret of any admissible policy must be at least in the order of $\sqrt{nT}$, which indicates that the bounded learning policy is asymptotically optimal (Keskin & Li, 2020, section 4).

One possible way to extend the above literature is to investigate the impact of Markov-modulated unit costs on the regret results for dynamic pricing and inventory management with demand learning (e.g., as in den Boer et al., 2018).

### 6.3.3 *Autoregressive Models*

Non-stationarity can also be modeled via a parametric demand function whose parameters follow an autoregressive process. Suppose that for all $t \in \mathbb{N}$, the demand function in period $t$ satisfies

$$d_t(p) = f(p, \theta_t) \text{ for } p \in [p_{\min}, p_{\max}],$$

where $f(\cdot)$ is a known parametric function, and

$$\boldsymbol{\theta} := \{\theta_t : t \in \mathbb{N}\}$$

is a sequence of unknown demand parameter vectors in $\mathbb{R}^d$ for some $d \in \mathbb{N}$.

An important special case of this setting is the linear demand model with time-varying parameters: for all $t \in \mathbb{N}$ and $p \in [p_{\min}, p_{\max}]$, $d_t(p) = f(p, \theta_t) = \alpha_t - \beta_t p$ for some $\theta_t = (\alpha_t, \beta_t)$ with $\alpha_t, \beta_t > 0$. Balvers and Cosimano (1990) consider a variant of this case in which the intercept sequence $\{\alpha_t : t \in \mathbb{N}\}$ is a first-order autoregressive process (i.e., $\alpha_t = \rho\alpha_{t-1} + \xi_t$ for $t = 2, 3, \ldots$, where $\rho \in (0, 1]$ and $\{\xi_t : t \in \mathbb{N}\}$ is a sequence of independent noise terms) and the slope sequence $\{\beta_t : t \in \mathbb{N}\}$ is a Gaussian random walk. In this setting, Balvers and Cosimano (1990) derive an implicit expression for optimal pricing decisions and use this expression to generate insights on the optimal policy.

Beck and Wieland (2002) consider another variant in which demand realizations follow a first-order autoregressive process; i.e., $d_t = \alpha_t - \beta_t p_t + \rho d_{t-1} + \epsilon_t$ for $t = 2, 3, \ldots$, where the intercept sequence $\{\alpha_t : t \in \mathbb{N}\}$ is constant over time, the slope sequence $\{\beta_t : t \in \mathbb{N}\}$ is a Gaussian random walk, and $\rho \in (0, 1]$. Beck and Wieland (2002) characterize the optimal policy in their setting and compare it with different heuristic policies.

An interesting direction for future research is developing asymptotically optimal policies for dynamic pricing and demand learning when the unknown demand parameters evolve according to a general autoregressive process.

### 6.3.4 General Changing Environments

Recent studies on dynamic pricing consider more general frameworks for nonstationary demand environments. For example, den Boer (2015b) analyzes a demand environment in which the market size is unknown and nonstationary whereas the price sensitivity of demand is known. He develops policies that hedge against potential demand changes, deriving upper bounds on the long-run average regret of these policies.

Keskin and Zeevi (2017) study a general changing environment where both the market size and the price sensitivity are unknown and nonstationary. In the setting of Keskin and Zeevi (2017), the underlying changes are allowed to have any pattern that satisfies a cumulative variation budget. Without knowing the variation budget, the seller needs to learn the demand function while filtering obsolete information. Keskin and Zeevi (2017) show that in this environment, the $T$-period regret of the seller is at least in the order of $T^{2/3}$ (Keskin & Zeevi, 2017, section 3.1). They also design policies that use moving window and decaying weights to discount

older information to achieve a $T$-period regret of order $T^{2/3}$, which corresponds to asymptotically optimal regret performance (Keskin & Zeevi, 2017, section 3.3).

Another issue investigated by Keskin and Zeevi (2017) is how regret depends on whether the underlying changes are gradual or abrupt. It turns out that this distinction can significantly affect asymptotically optimal performance (Keskin & Zeevi, 2017, section 4). Chen et al. (2019) further investigate this issue by developing a unified approach that combines change-point detection with upper confidence bound (UCB) policies seen in the bandit literature (Chen et al., 2019, section 4). They also prove that their approach exhibits asymptotically optimal regret performance (Chen et al., 2019, section 5).

The aforementioned developments in dynamic pricing in general nonstationary environments also influence the recent work on other operations problems. Chen (2021) applies this approach to dynamic inventory control, deriving asymptotically optimal regret bounds. Keskin and Li (2020) formulate and study a nonstationary newsvendor problem, extending the earlier work on data-driven learning in stationary newsvendor problems (see, e.g., Besbes & Muharremoglu, 2013; Levi et al., 2015).

While usually viewed as a challenge, nonstationary environments occasionally improve a policy's performance. For instance, due to a lack of forced exploration, passive learning policies typically suffer from incomplete learning in stationary environments (Lai & Robbins, 1982; Harrison et al., 2012; den Boer & Zwart, 2014; Keskin & Zeevi, 2018). However, Keskin and Zeevi (2018) show that passive learning policies do not suffer from this issue in certain nonstationary environments that evolve in an *unbounded* manner (Keskin & Zeevi, 2018, sections 4.2.3 and 4.2.4). They also show that incomplete learning persists in *boundedly* changing environments (Keskin & Zeevi, 2018, section 4.2.1).

### *6.3.5 Contextual Pricing*

Another cause of exogenous changes in demand environments is contextual information that varies over time. Examples include detailed information on the customers and products of an online retailer. Such contextual information typically leads to high-dimensional pricing problems based on stochastic features (see, e.g., Nambiar et al., 2019; Ban & Keskin, 2021; Miao et al., 2022; Keskin et al., 2020). We refer readers to the next chapter of this book for a discussion on high-dimensional pricing problems.

## 6.4 Endogenously Changing Demand Environments

### *6.4.1 Reference-Price Effects*

The demand for a product can sometimes be subject to customers' behavioral biases; e.g., the customers may form a price expectation in the form of a *reference price*. In this case, the customers view price increases/decreases relative to the reference price as losses/gains, which subsequently influences demand. Since reference-price formation depends on past prices, dynamic pricing with reference effects leads to endogenous changes in a demand environment. Optimal control of these endogenous changes is extensively studied in the dynamic pricing literature (see, e.g., Fibich et al., 2003; Popescu & Wu, 2007; Chen et al., 2017, and the references therein). In a recent study, den Boer and Keskin (2022) extend this literature to the case of demand learning. They show that if the customers are loss-averse, then a slow-moving pricing policy is asymptotically optimal (den Boer & Keskin, 2022, section 3). On the other hand, if the customers are gain-seeking, then a cyclical pricing policy is asymptotically optimal, and the best achievable performance and the optimal cycle length are parameter-dependent (den Boer & Keskin, 2022, section 5). An interesting extension to this work would be the analysis of more general reference-price formation processes that capture different degrees of customer memory.

### *6.4.2 Competition and Collusion*

Competitors changing their prices cause non-stationarity in a demand environment. There is a vast literature on pricing with incomplete information in a competitive market, which roughly can be classified into theoretical studies that analyze the convergence behavior of pricing algorithms, and simulation studies that assess the numerical performance of policies. An in-depth discussion of this literature is beyond the scope of this chapter; for a review, we refer readers to den Boer (2015a, section 6.2).

A key part in analyzing the performance of a dynamic pricing-and-learning policy in the presence of competition is the assumptions on competitors' actions. It is often assumed that all players in a market use the same policy (see, e.g., Yang et al., 2020). A drawback of this assumption that it does not address the case where competitors might use different pricing policies. One can also take an adversarial approach, e.g., by discretizing prices and using an adversarial-bandit algorithm as in Auer et al. (2002). A potential drawback of this approach is that this may generate prices that are too conservative because, in practice, sellers usually maximize their own profits rather than trying to minimize competitors' profits. The third approach is to simply ignore the presence of competitors in a market, acting as a "monopolist" who is oblivious to competition. Cooper et al. (2015) show that

if two sellers in a duopoly use an iterated least squares policy that neglects the presence of a competitor, then the limit prices are random and potentially higher or lower than the Nash-equilibrium prices. On the other hand, Meylahn and den Boer (2022) show that the prices generated by a Kiefer-Wolfowitz recursion that does not take competition into account still converge to a Nash equilibrium if used by both players. They also prove that if the competitor's actions are determined by a reaction function, the aforementioned Kiefer-Wolfowitz price process converges to a best response to the competitor's price. These results indicate that under certain conditions, ignoring the presence of competition is not necessarily harmful. Apart from the above, the fourth approach is to use one of the methods designed for exogenously changing environments discussed in the preceding section.

A question of recent attention is whether self-learning algorithms are capable of learning to collude instead of compete with each other. Legal scholars are worried that algorithmic pricing could result in supra-competitive prices that are harmful for consumer welfare and that existing competition law is ill-suited to deal with algorithmic collusion (Ezrachi & Stucke, 2016, 2020; Gal, 2018, 2019; Harrington, 2018; Mehra, 2016; Smejkal, 2017), although some economists are skeptical about the need to change the law (Kühn & Tadelis, 2017; Schrepel, 2017; Schwalbe, 2018). Simulations by Cooper et al. (2015), building on the work by Kirman (1975), indicate that a greedy iterated least squares policy used by both players in a duopoly generates limit prices and profits that, with positive probability, are component-wise larger than competitive Nash-equilibrium prices and profits. Similar observations are made on simulation studies of Q-learning (Calvano et al., 2020; Klein, 2018). Figures 6 and 7 of Cooper et al. (2015) show that the converse can also happen: limit prices and profits that, with positive probability, are component-wise smaller than Nash-equilibrium prices and profits. Instead of "accidentally" arising supra-competitive limit prices, Meylahn and den Boer (2022) show that algorithms may also be explicitly designed to learn to collude: they construct such an algorithm for a duopoly and prove convergence results that guarantee supra-competitive prices and profits when the algorithm is used by both players in a duopoly and the cartel price limit is mutually beneficial.

One way to expand this literature is to examine joint pricing and capacity expansion in the presence of competition and demand learning. In a recent study, Sunar et al. (2021b) study competitive capacity expansion with dynamic learning, generalizing the earlier work by Harrison and Sunar (2015) and Qi et al. (2017). Analyzing the extension to capture pricing decisions and developing asymptotically optimal policies in this setting is a possible direction for future research.

### *6.4.3 Platforms and Multi-Agent Learning*

The increasing prevalence of online marketplace platforms in practice makes them a focus of attention in the pricing literature (see, e.g., Weyl, 2010; Banerjee et al., 2015; Bai et al., 2018; Taylor, 2018; Bimpikis et al., 2019; Bernstein et al., 2021;

Huang et al., 2020). Data-driven learning in such platforms creates an endogenously changing demand environment. The reason is that many online marketplaces have a large number of participants, and simultaneous decision making of these participants leads to a nonstationary market environment where past decisions of participants can influence future payoffs. This type of intertemporal dependencies are usually studied in the literature on multi-agent learning (see, e.g., Zhou et al., 2018; Mertikopoulos & Zhou, 2019).

In a recent study on learning in platforms, Feng et al. (2020) consider a two-sided mobile-promotion platform where online advertisers and publishers participate. The platform dynamically receives online ad campaigns from the advertisers and procures impressions from the publishers to fulfill campaigns through real-time bidding. The probability of winning an impression as a function of bid price is unknown to the platform and must be learned from data. In this setting, Feng et al. (2020) design a cyclical policy that dynamically allocates bids while learning the win probabilities, and prove that this policy is asymptotically optimal.

Birge et al. (2021) analyze an online marketplace setting where a platform and its sellers have limited information on how demand depends on the sellers' prices. They show that sharing no information with the sellers does not necessarily result in poor revenue performance for the platform. Birge et al. (2021) also prove that the platform can avoid large losses by sharing all of its demand information with the sellers. Based on these results, they design a policy that strategically reveals the platform's demand information to the sellers to achieve asymptotically optimal performance in general.

A possible direction for future research is expanding this literature to consider provider and customer networks in online marketplaces. With regard to recent related work, see, e.g., Sunar et al. (2019) for optimal product development and launch for a customer network, and Kao et al. (2020) for optimal design and pricing of subscription services for a finite population of customers.

### *6.4.4 Forward-Looking and Patient Customers*

Customer patience is another source of endogenous changes in a demand environment. A customer's willingness to wait for multiple sales opportunities from a seller results in an intertemporal dependency between price and demand. When customers look forward and evaluate future sales opportunities, the seller's price affects the demand function for subsequent sales, leading to a nonstationary demand environment.

There is a rich literature on pricing with patient customers (see, e.g., Besbes & Lobel, 2015; Liu & Cooper, 2015; Lobel, 2020) with a recent stream of research extending this literature to dynamic learning (see, e.g., Zhang & Jasin, 2022; Birge et al., 2019). Zhang and Jasin (2022) analyze cyclical pricing-and-learning policies in the presence of patient customers. They show that a cyclical price skimming policy can exhibit asymptotically optimal regret performance. Birge et al. (2019)

study the design of markdown pricing policies for patient customers in the presence of limited demand information. They show that customer memory plays a significant role in determining the best achievable revenue performance in this context.

Recent work also studies the case of patient customers who act rationally (see Birge et al., 2021; Golrezaei et al., 2019, 2021). Birge et al. (2021) study the dynamic pricing problem of a market maker facing an informed and strategic market participant. They design an inertial policy that uses small price increments over time, proving that this policy can help the market maker guard against potential manipulations of the strategic market participant. Golrezaei et al. (2019, 2021) analyze dynamic learning in repeated contextual second-price auctions. They construct learning policies that are robust to strategic bidding behavior, and show that their policies exhibit near-optimal revenue performance. An interesting future direction for this area is the analysis of how forward-looking customers might impose externalities on each other in dynamic pricing-and-learning settings, and especially, how this interaction affects social welfare (e.g., as in Sunar et al., 2021a).

## References

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing, 32*(1), 48–77.

Aviv, Y., & Pazgal, A. (2005) A partially observed Markov decision process for dynamic pricing. *Management Science, 51*(9), 1400–1416.

Bai, J., So, K. C., Tang, C. S., Chen, X., & Wang, H. (2018). Coordinating supply and demand on on-demand service platform with impatient customers. *Manufacturing and Service Operations Management, 21*(3), 556–570.

Balvers, R. J., & Cosimano, T. F. (1990). Actively learning about demand and the dynamics of price adjustment. *The Economic Journal, 100*(402), 882–898.

Ban, G. Y., & Keskin, N. B. (2021). Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. *Management Science, 67*(9), 5549–5568.

Banerjee, S., Johari, R., & Riquelme, C. (2015). Pricing in ride-sharing platforms: A queueing-theoretic approach. In M. Feldman, T. Roughgarden, & M. Schwarz (Eds.), *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (p. 639). ACM.

Beck, G. W., & Wieland, V. (2002). Learning and control in a changing economic environment. *Journal of Economic Dynamics and Control, 26*(9–10), 1359–1377.

Bernstein, F., DeCroix, G. A., & Keskin, N. B. (2021). Competition between two-sided platforms under demand and supply congestion effects. *Manufacturing & Service Operations Management, 23*(5), 1043–1061.

Besbes, O., & Lobel, I. (2015). Intertemporal price discrimination: Structure and computation of optimal policies. *Management Science, 61*(1), 92–110.

Besbes, O., & Muharremoglu, A. (2013). On implications of demand censoring in the newsvendor problem. *Management Science, 59*(6), 1407–1424.

Besbes, O., & Zeevi, A. (2011). On the minimax complexity of pricing in a changing environment. *Operations Research, 59*(1), 66–79.

Bimpikis, K., Candogan, O., & Saban, D. (2019). Spatial pricing in ride-sharing networks. *Operations Research, 67*(3), 744–769.

Birge, J. R., Chen, H., & Keskin, N. B. (2019). Markdown policies for demand learning with forward-looking customers. https://ssrn.com/abstract=3299819

Birge, J. R., Chen, H., Keskin, N. B., & Ward, A. (2021). To interfere or not to interfere: Information revelation and price-setting incentives in a multiagent learning environment. https://ssrn.com/abstract=3864227

Birge, J. R., Feng, Y., Keskin, N. B., & Schultz, A. (2021). Dynamic learning and market making in spread betting markets with informed bettors. *Operations Research, 69*(6), 1746–1766.

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review, 110*(10), 3267–3297.

Chen, B. (2021). Data-driven inventory control with shifting demand. *Production and Operations Management, 30*(5), 1365–1385.

Chen, X., Hu, P., &Hu, Z. (2017). Efficient algorithms for the dynamic pricing problem with reference price effect. *Management Science, 63*(12), 4389–4408.

Chen, Y., & Farias, V. F. (2013). Simple policies for dynamic pricing with imperfect forecasts. *Operations Research, 61*(3), 612–624.

Chen, Y., Wen, Z., & Xie, Y. (2019). Dynamic pricing in an evolving and unknown marketplace. https://ssrn.com/abstract=3382957

Cooper, W. L., Homem-de Mello, T., & Kleywegt, A. J. (2015). Learning and pricing with models that do not explicitly incorporate competition. *Operations Research, 63*(1), 86–103.

den Boer, A., Perry, O., & Zwart, B. (2018). Dynamic pricing policies for an inventory model with random windows of opportunities. *Naval Research Logistics (NRL), 65*(8), 660–675.

den Boer, A. V. (2015a). Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science, 20*(1), 1–18.

den Boer, A. V. (2015b). Tracking the market: Dynamic pricing and learning in a changing environment. *European Journal of Operational Research, 247*(3), 914–927.

den Boer, A. V., & Keskin, N. B. (2020). Discontinuous demand functions: Estimation and pricing. *Management Science, 66*(10), 4516–4534.

den Boer, A. V., & Keskin, N. B. (2022). Dynamic pricing with demand learning and reference effects. *Management Science*, (in press).

den Boer, A. V., & Zwart, B. (2014). Simultaneously learning and optimizing using controlled variance pricing. *Management Science, 60*(3), 770–783.

Ezrachi, A., & Stucke, M. (2016). Virtual competition: The promise and perils of the algorithm-driven economy. Cambridge, Massachusetts: Harvard University Press.

Ezrachi, A., & Stucke, M. E. (2020). Sustainable and unchallenged algorithmic tacit collusion. *Northwestern Journal of Technology and Intellectual Property, 17*(2), 217–260.

Feng, Z., Dawande, M., Janakiraman, G., & Qi, A. (2020). An asymptotically tight learning algorithm for mobile-promotion platforms. https://ssrn.com/abstract=3523491

Fibich, G., Gavious, A., & Lowengart, O. (2003). Explicit solutions of optimization models and differential games with nonsmooth (asymmetric) reference-price effects. *Operations Research, 51*(5), 721–734.

Gal, M. S. (2018). Illegal pricing algorithms. *Communications of the ACM, 62*(1), 18–20.

Gal, M. S. (2019). Algorithms as illegal agreements. *Berkeley Technology Law Journal, 34*(1), 67.

Golrezaei, N., Jaillet, P., & Liang, J. C. N. (2019). Incentive-aware contextual pricing with non-parametric market noise. https://arxiv.org/abs/1911.03508

Golrezaei, N., Javanmard, A., & Mirrokni, V. (2021). Dynamic incentive-aware learning: Robust pricing in contextual auctions. *Operations Research, 69*(1), 297–314.

Harrington Jr, J. (2018). Developing competition law for collusion by autonomous price-setting agents. *Journal of Competition Law and Economics, 14*(3), 331–363.

Harrison, J. M., Keskin, N. B., & Zeevi, A. (2012). Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science, 58*(3), 570–586.

Harrison, J. M., & Sunar, N. (2015). Investment timing with incomplete information and multiple means of learning. *Operations Research, 63*(2), 442–457.

Huang, H., Sunar, N., & Swaminathan, J. M. (2020). Do noisy customer reviews discourage platform sellers? Empirical analysis of an online solar marketplace. https://ssrn.com/abstract=3645605

Kao, Y. M., Keskin, N. B., & Shang, K. (2020). Bayesian dynamic pricing and subscription period selection with unknown customer utility. https://ssrn.com/abstract=3722376

Keller, G., & Rady, S. (1999). Optimal experimentation in a changing environment. *The Review of Economic Studies, 66*(3), 475–507.

Keskin, N. B., & Li, M. (2020). Selling quality-differentiated products in a Markovian market with unknown transition probabilities. https://ssrn.com/abstract=3526568

Keskin, N. B., Li, Y., & Song, J. S. J. (2022). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science, 68*(3), 1938–1958.

Keskin, N. B., Li, Y., & Sunar, N. (2020). Data-driven clustering and feature-based retail electricity pricing with smart meters. https://ssrn.com/abstract=3686518

Keskin, N. B., Min, X., & Song, J. S. J. (2021). The nonstationary newsvendor: Data-driven nonparametric learning. https://ssrn.com/abstract=3866171

Keskin, N. B., & Zeevi, A. (2014). Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research, 62*(5), 1142–1167.

Keskin, N. B., & Zeevi, A. (2017). Chasing demand: Learning and earning in a changing environment. *Mathematics of Operations Research, 42*(2), 277–307.

Keskin, N. B., & Zeevi, A. (2018). On incomplete learning and certainty-equivalence control. *Operations Research, 66*(4), 1136–1167.

Kirman, A. P. (1975). Learning by firms about demand conditions. In R. H. Day, & T. Groves (Eds.), *Adaptive Economic Models* (pp. 137–156). Elsevier.

Klein, T. (2018). Assessing autonomous algorithmic collusion: Q-learning under short-run price commitments. Amsterdam Law School Research Paper No. 2018-15, Amsterdam Center for Law & Economics Working Paper No. 2018-05.

Kühn, K. U., & Tadelis, S. (2017). *Algorithmic Collusion*. https://www.ebos.com.cy/cresse2013/uploadfiles/2017_sps5_pr2.pdf

Lai, T., & Robbins, H. (1982). Iterated least squares in multiperiod control. *Advances in Applied Mathematics, 3*(1), 50–73.

Lai, T. L. (1995). Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society: Series B (Methodological), 57*(4), 613–644.

Levi, R., Perakis, G., & Uichanco, J. (2015). The data-driven newsvendor problem: New bounds and insights. *Operations Research, 63*(6), 1294–1306.

Liu, Y., & Cooper, W. L. (2015). Optimal dynamic pricing with patient customers. *Operations Research, 63*(6), 1307–1319.

Lobel, I. (2020). Dynamic pricing with heterogeneous patience levels. *Operations Research, 68*(4), 1038–1046.

Mehra, S. (2016). Antitrust and the Robo-Seller: Competition in the time of algorithms. *Minnesota Law Review, 100*, 1323–1375.

Mertikopoulos, P., & Zhou, Z. (2019). Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming, 173*(1), 465–507.

Meylahn, J., & den Boer, A. (2022). Learning to collude in a pricing duopoly. *Manufacturing & Service Operations Management* (in press).

Miao, S., Chen, X., Chao, X., Liu, J., & Zhang, Y. (2022). Context-based dynamic pricing with online clustering. *Production and Operations Management* (in press).

Nambiar, M., Simchi-Levi, D., & Wang, H. (2019). Dynamic learning and pricing with model misspecification. *Management Science, 65*(11), 4980–5000.

Popescu, I., & Wu, Y. (2007). Dynamic pricing strategies with reference effects. *Operations Research, 55*(3), 413–429.

Qi, A., Ahn, HS., & Sinha, A. (2017). Capacity investment with demand learning. *Operations Research, 65*(1), 145–164.

Rustichini, A., & Wolinsky, A. (1995). Learning about variable demand in the long run. *Journal of Economic Dynamics and Control, 19*(5–7), 1283–1292.

Schrepel, T. (2017). Here's why algorithms are NOT (really) a thing. Concurrentialiste. https://leconcurrentialiste.com/algorithms-based-practices-antitrust

Schwalbe, U. (2018). Algorithms, machine learning and collusion. *Journal of Competition Law & Economics, 14*(4), 568–607.

Shiryaev, A. N. (2010) Quickest detection problems: Fifty years later. *Sequential Analysis, 29*(4), 345–385.

Smejkal, V. (2017). Cartels by robots – Current antitrust law in search of an answer. InterEULawEast. *Journal for the International and European Law, Economics and Market Integrations, 4*(2), 1–18.

Sunar, N., Birge, J. R., & Vitavasiri, S. (2019). Optimal dynamic product development and launch for a network of customers. *Operations Research, 67*(3), 770–790.

Sunar, N., Tu, Y., & Ziya, S. (2021a). Pooled vs. dedicated queues when customers are delay-sensitive. *Management Science, 67*(6), 3785–3802.

Sunar, N., Yu, S., & Kulkarni, V. G. (2021b). Competitive investment with Bayesian learning: Choice of business size and timing. *Operations Research, 69*(5), 1430–1449.

Taylor, T. (2018). On-demand service platforms. *Manufacturing and Service Operations Management, 20*(4), 704–720.

Weyl, E. G. (2010). A price theory of multi-sided platforms. *American Economic Review, 100*(4), 1642–1672.

Yang, Y., Lee, Y. C., & Chen, P. A. (2020). Competitive demand learning: A data-driven pricing algorithm. https://arxiv.org/abs/2008.05195

Zhang, H., & Jasin, S. (2022). Online learning and optimization of (some) cyclic pricing policies in the presence of patient customers. *Manufacturing & Service Operations Management, 24*(2), 1165–1182.

Zhou, Z., Mertikopoulos, P., Bambos, N., Glynn, P., & Tomlin, C. (2018). Multi-agent online learning with imperfect information. Working paper, Stanford University.

# Chapter 7
# Pricing with High-Dimensional Data


**Gah-Yi Ban**

## 7.1 Introduction

From the mid-1990s, companies that acquire and interact with customers primarily online were born. Many of these companies are now household names, such as Amazon (est. 1994), Netflix (est. 1997), Google (est. 1998), Facebook (est. 2004), and Airbnb (est. 2008), having disrupted many traditional industries including retail, advertising, and entertainment.

The resulting decades of e-commerce has led to an explosion of business-generated data, which in turn have been used to further enhance and grow the business. A celebrated example of using such data is for personalization of recommendations—be it for products, advertisements, or consumable media. More recently, the Operations Research/Management Science community has been exploring the use of potentially large amounts of data beyond recommendation systems, e.g., for inventory and supply chain decisions (Ban and Rudin, 2019; Ban et al., 2019; Mandl and Minner, 2020), medical decision-making (Bastani and Bayati, 2020), and pricing and revenue optimization (Ferreira et al., 2016; Qing and Bayati, 2016; Javanmard and Nazerzadeh, 2019; Qu et al., 2020; Cohen et al., 2020; Chen et al., 2022; Ban and Keskin, 2021; Chen et al., 2020).

In this chapter, we review recent theoretical developments in using high-dimensional data (usually, information pertaining to customers and/or the product) in pricing. The chapter is structured as follows. In Sect. 7.2, we provide a brief background on high-dimensional statistics. In Sects. 7.3 and 7.4, respectively, we review a static and a dynamic pricing model that incorporate high-dimensional data. In Sect. 7.5, we discuss future directions for research in this sphere.

G.-Y. Ban (✉)
Robert H. Smith School of Business, University of Maryland, College Park, MD, USA
e-mail: gban@umd.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_7

## 7.2 Background: High-Dimensional Statistics

High-dimensional statistics is the study and analysis of data where the number of dimensions observed (typically denoted by $d$ or $p$, here we use $d$) is comparable to, or exceeds, the number of observations (typically denoted by $n$). This contrasts with classical statistics, where $d$ is assumed fixed and small compared to $n$.

Even before the availability of large datasets in many domains, interest in high-dimensional statistics grew from the 1950s when researchers such as Rao, Wigner, Kolmogorov, and Huber (to name a few) recognized that standard statistical methods and theory may fail in a high-dimensional regime. To illustrate, consider the classical ordinary least-squares regression problem, with where $n$ response observations $y_1, \ldots, y_n$ are regressed on $d$-dimensional regressors (also known as explanatory or independent variables, covariates, or features) $\mathbf{x}_1, \ldots, \mathbf{x}_n$:

$$\min_{\alpha \in \mathbb{R}, \beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \alpha - \boldsymbol{\beta}^\top \mathbf{x}_i)^2. \tag{7.1}$$

Equation (7.1) is a convex optimization problem, so the optimal coefficient estimates $(\hat{\alpha}, \hat{\boldsymbol{\beta}}) \in \mathbb{R} \times \mathbb{R}^d$ can be found by solving the first-order optimality equations

$$\begin{aligned} \textstyle\sum_{i=1}^n 2(y_i - \alpha - \boldsymbol{\beta}^\top \mathbf{x}_i) &= 0 \\ \textstyle\sum_{i=1}^n 2(y_i - \alpha - \boldsymbol{\beta}^\top \mathbf{x}_i) x_{i1} &= 0 \\ \vdots \qquad\qquad\qquad &\quad \vdots \\ \textstyle\sum_{i=1}^n 2(y_i - \alpha - \boldsymbol{\beta}^\top \mathbf{x}_i) x_{id} &= 0, \end{aligned}$$

which in matrix form is

$$\boldsymbol{X}^\top \boldsymbol{X} \begin{pmatrix} \alpha \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix} = \boldsymbol{X}^\top \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \tag{7.2}$$

where $\boldsymbol{X}$ is the $n \times (d+1)$ matrix given by

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \ldots & x_{nd} \end{pmatrix}.$$

Thus, the existence of a unique solution $(\hat{\alpha}, \hat{\boldsymbol{\beta}}) \in \mathbb{R} \times \mathbb{R}^d$ depends on the rank of $\boldsymbol{X}$. If $d + 1 > n$, then the rank of $\boldsymbol{X}$ is at most $n$, so $\boldsymbol{X}^\top \boldsymbol{X}$ is not invertible and a unique solution to (7.2) cannot be found. In the current age of "big data," it is fairly common to have a large $d$ as data collection and recording has become cheap and

easy. Furthermore, if interaction and nonlinear effects of a starting set of features are considered, then the number of regressors can quickly grow, even if the starting set may not be too large.

A popular remedy that has emerged for doing regression with high-dimensional data is by perturbing the objective in (7.1) by a *regularization* function:

$$\min_{\alpha\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^d} \sum_{i=1}^{n}(y_i - \alpha - \boldsymbol{\beta}^\top \mathbf{x}_i)^2 + \lambda R(\alpha, \boldsymbol{\beta}), \tag{7.3}$$

where $\lambda \geq 0$ is a constant that controls the degree of regularization, and $R(\cdot) : \mathbb{R}^{d+1} \to \mathbb{R}_+$ is a positive penalty function. The most popular choice for the regularization function is the $L_1$ norm penalty ($R(\mathbf{z}) = \|\mathbf{z}\|_1$), referred to as "lasso regression." More broadly, when lasso regularization is used in general estimation problems (not just least-squares regression), it leads to sparse coefficient estimates because lasso is a convex relaxation of the $L_0$ norm (which counts the number of nonzero elements of a vector). This in turn leads to more interpretable models; as such, lasso regularization has been the focus of intense research in recent decades (Hastie et al., 2009; Bühlmann and Van De Geer, 2011; Wainwright, 2019). In pricing, Qu et al. (2020), Chen et al. (2022), and Ban and Keskin (2021) exemplify using lasso regularization in real-data case studies; Qu et al. (2020) for estimating the demand for multiple, heterogeneous products in business-to-business pricing, Chen et al. (2022) for customized pricing for airline priority seating, and Ban and Keskin (2021) for estimating the demand for car loans at the individual customer level for an auto loan company.

Alternatively, non-parametric "machine learning" methods of discovering the relationship between a response variable and high-dimensional explanatory variables have also emerged in recent decades. In particular, tree-based methods such as *random forest* (Breiman, 2001) are proving to be very effective "off-the-shelf" methods for prediction accuracy. In pricing, Ferreira et al. (2016) found regression trees with bagging (which is similar to random forests but more interpretable) to be superior to a number of other methods (least squares, principal components, partial least squares, multiplicative, and semi-logarithmic regression) for predicting the demand for new products in a real-data case study for a fashion retailer.

This is just one illustration of how high-dimensional data revolutionized a classical statistical method (least-squares regression), leading to new methods, empirical insights, and theory. We refer to Hastie et al. (2009), Bühlmann and Van De Geer (2011), and Wainwright (2019) for the readers interested in learning further about high-dimensional statistics. In the rest of this chapter, we review the theory of pricing with high-dimensional data.

## 7.3 Static Pricing with High-Dimensional Data

In this section, we review the theoretical results in Chen et al. (2022), which analyzes a static pricing problem with high-dimensional data using statistical learning theory.

### *7.3.1 Feature-Dependent Choice Model*

Consider a firm selling a set $\mathcal{J}$ of $J$ products. Prior to pricing the items, the firm observes a $d$-dimensional vector, $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^d$ of customer features (characteristics), which are assumed to be normalized so that $\|\mathbf{z}\|_\infty \leq \infty$. Once $\mathbf{z}$ is observed, the firm chooses a price $p_j \in [p_{min}, p_{max}] =: \mathcal{P}$ for each product $j \in \mathcal{J}$. Let $\mathbf{p} := [p_1, \ldots, p_J]$ denote the vector of prices for all $J$ products.

For each product $j \in \mathcal{J}$, Chen et al. (2022) proposes the following personalized utility model for a customer with feature vector $\mathbf{z}$:

$$U_j(\mathbf{z}, p_j, \theta_j) = V_j(\mathbf{z}, p_j, \theta_j) + \varepsilon_j, \tag{7.4}$$

where $V_j(p_j, \mathbf{z}, \theta_j)$, the deterministic part of the utility model, is specified to be linear in the customer features and the price:

$$V_j(\mathbf{z}, p_j, \theta_j) = \gamma_j^\top \mathbf{z} + \beta_j^\top p_j, \tag{7.5}$$

for some constants $\gamma_j \in \mathbb{R}^d$ and $\beta_j \in \mathbb{R}$, captured together by $\theta_j := [\gamma_j, \beta_j] \in \mathbb{R}^{d+1}$, and $\varepsilon_j$, independent across the $j$'s, are specified to follow the Gumbel distribution.

By discrete-choice theory (Train, 2009), this leads to a personalized logit function for the probability of the customer purchasing product $j \in \mathcal{J}$:

$$\mathbb{P}(j; \mathbf{z}, \mathbf{p}, \boldsymbol{\theta}) = \frac{e^{V_j(\mathbf{z}, p_j, \theta_j)}}{1 + \sum_{k=1}^{J} e^{V_k(\mathbf{z}, p_k, \theta_k)}}, \tag{7.6}$$

where $\boldsymbol{\theta} := \{\theta_1, \ldots, \theta_J\}$.

Denote the customer's decision with $y$, where $y \in \{0, 1, \ldots, J\}$ such that $y = j$ corresponds to the purchase of product $j$, and $y = 0$ denotes no purchase.

The firm's decision objective is to find the personalized price that maximizes the expected revenue, $r(\mathbf{p}, \mathbf{z}, \boldsymbol{\theta})$:

$$\mathbf{p}^*(\mathbf{z}) = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmax}}\, r(\mathbf{z}, \mathbf{p}, \boldsymbol{\theta}) := \sum_{j \in \mathcal{J}} p_j \mathbb{P}(j; \mathbf{z}, \mathbf{p}, \boldsymbol{\theta}). \tag{7.7}$$

It is straightforward to show that (7.7) is a convex optimization problem.

### 7.3.2 Estimation Method

In practice, a firm would not know the true parameters $\boldsymbol{\theta}$. Suppose the decision-maker has access to $n$ past customer features, prices, and purchase decisions, $\mathcal{D} = \{(\mathbf{z}_1, \mathbf{p}_1, y_1), \ldots, (\mathbf{z}_n, \mathbf{p}_n, y_n)\}$. Chen et al. (2022) considers a static learning setting, meaning all $n$ periods of past data are available at once, and the prices were not personalized in the past—i.e., the price sequence $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ is independent of the customer feature sequence $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. This would reflect a scenario where a firm did not personalize prices in the past and is considering doing so now. Alternatively, this also reflects a scenario where a firm already does some personalized pricing, but not for all, and so $\mathcal{D}$ would capture a subset of historical data for which prices were blind to the customer features.

In such a setting, Chen et al. (2022) proposes estimating $\boldsymbol{\theta}$ by maximum likelihood estimation with $L_1$ regularization:

$$\begin{aligned} \hat{\boldsymbol{\theta}}(R) = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\ \ell_n(\mathcal{D}, \boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{P}(y_i; \mathbf{p}_i, \mathbf{z}_i, \boldsymbol{\theta}) \\ s.t. \\ \|\boldsymbol{\theta}\|_1 \leq R, \end{aligned} \tag{7.8}$$

where $R > 0$ is a tuning parameter that controls the model complexity. Equation (7.8) is a convex optimization problem, so can be solved efficiently using widely available solvers.

The firm can thus estimate the optimal personalized price $\hat{\mathbf{p}}(\mathbf{z})$ by plugging-in $\hat{\boldsymbol{\theta}}(R)$ to the pricing problem (7.7):

$$\hat{\mathbf{p}}(\mathbf{z}) = \underset{\mathbf{p}\in\mathcal{P}}{\operatorname{argmax}} \sum_{j\in\mathcal{J}} p_j \mathbb{P}(j; \mathbf{p}, \mathbf{z}, \hat{\boldsymbol{\theta}}(R)). \tag{7.9}$$

In Chen et al. (2022), the estimation method above is numerically evaluated on simulated data and real data from a European airline carrier. In the rest of this section, however, we focus on the theoretical performance analysis.

### 7.3.3 Performance Guarantees

A central question for a firm considering implementing the estimation method from Sect. 7.3.2 would be how close the revenue of the estimated price $\hat{\mathbf{p}}(\mathbf{z})$ would be to the optimal revenue from $\mathbf{p}^*(\mathbf{z})$. Chen et al. (2022) provides high-probability guarantees for the difference using the theory of M-estimation from classical statistics. First, the following assumptions are needed.

***Assumption 1 (Conditional Independence of Purchase Decisions)*** The purchase decisions, $y_i, i = 1, \ldots, n$, are independent of each other given each $\mathbf{z}_i$ and $\mathbf{p}_i$. □

Now, let $\mathbf{x}_{ij} := (\mathbf{z}_i, p_{ij})^\top$ be a $(d+1)$-dimensional composite vector for $j \in \mathcal{J}$ and $i \in \mathbb{Z}_+$. Also, let $\Sigma := n^{-1} \sum_{i=1}^{n} \mathbb{E}[x_{ij} x_{ij}^\top]$, and let $\lambda_{\min}(\cdot)$ denote the minimum eigenvalue function.

***Assumption 2***

(a) For each $j \in \mathcal{J}$, the vectors $\{\mathbf{x}_{ij}\}_{i=1}^{n}$ are independent across $i$.
(b) For each $j \in \mathcal{J}$, the vectors $\{\mathbf{x}_{ij}\}_{i=1}^{n}$ are sub-Gaussian with the uniform sub-Gaussian norm $\psi$ given by

$$\psi(X) := \inf \left\{ t \geq 0 : \mathbb{E}\left[ \exp\left( X^2 / t^2 \right) \leq 2 \right] \right\}.$$

(c) There exists a constant $\rho > 0$ such that $\lambda_{\min}(\Sigma) \geq \rho$ for all $j \in \mathcal{J}$. Furthermore,

$$\max_{i,j} \lambda_{\min}\left( \mathbb{E}[x_{ij} x_{ij}^\top] \right) > 0.$$

**Remarks on Assumptions 1 and 2** Assumption 1 is standard in the revenue management literature. Assumption 2 (a) means customers arrive independently of each other and is a standard assumption in statistical learning. Assumption 2 (b) is common in regression analysis because it captures a wide range of multivariate distributions. Finally, Assumption 2 (c) stipulates that the feature vectors and the pricing decisions are not collinear.

The following performance guarantee can be shown for $\hat{\mathbf{p}}(\mathbf{z})$:

**Theorem 1 (Theorem 2, Chen et al., 2022 )** *Under Assumptions 1 and 2, for* $n \geq \frac{4C(\psi,R)\log(n)}{\min(\rho,1)^2}$ *and any feature vector* $\mathbf{z}$*, the expected revenue gap between the optimal personalized price* $\mathbf{p}^*(\mathbf{z})$ *and its estimate* $\hat{\mathbf{p}}(\mathbf{z})$ *can be bounded with high probability as follows:*

$$r(\mathbf{z}, \mathbf{p}^*, \boldsymbol{\theta}) - r(\mathbf{z}, \hat{\mathbf{p}}, \boldsymbol{\theta}) \leq \frac{C(\psi, R)}{\rho} J^4 (d+1) \sqrt{\frac{\log(2nJ(d+1))}{n}}, \tag{7.10}$$

where $C(\psi, R)$ is a constant depending only on $\psi$ and $R$.

Theorem 1 shows that the estimation approach described in Sect. 7.3.2 is well-justified, because the expected revenue of the estimated personalized price $\hat{\mathbf{p}}$ converges to that of the optimal price $\mathbf{p}^*$ at $\sqrt{n}$ rate, up to logarithmic factors. Theorem 1 also makes explicit the effect of key parameters on the revenue bound, which can guide practice. For example, a firm with a given number of relevant data $n$ may wish to try personalized pricing first on a smaller number of products first, as the effect of $J$ on the revenue bound is so large. Theorem 1 also makes the

effect of the dimensionality explicit; note that the bound in (7.10) can be very loose if $(d+1)/n$ is large. In such a high-dimensional setting, the bound (7.10) could be tightened by making a sparsity assumption such as $\|\boldsymbol{\theta}\|_1 \leq R$ (which implies $\|\boldsymbol{\theta}\|_0 \leq R$), with $R < (d+1)$, in which case $(d+1)$ could be replaced by $R$ with a slight change to the preamble to the theorem.

The proof of Theorem 1 relies on the following two results.

**Lemma 1 (Proposition 1, Chen et al., 2022 )** *For $n \geq \frac{4C(\psi,R)\log(n)}{\min(\rho,1)^2}$ and for any price vector $\mathbf{p} \in \mathcal{P}$, the error in the revenue forecast as a fraction of the maximal price can be bounded with high probability as follows:*

$$|r(\mathbf{z},\mathbf{p},\boldsymbol{\theta}) - r(\mathbf{z},\mathbf{p},\hat{\boldsymbol{\theta}}(R))| \leq \frac{C(\psi,R)}{2\rho} J^4(d+1)\sqrt{\frac{\log(2nJ(d+1))}{n}}.$$

***Proof (of Lemma 1)*** We have

$$\begin{aligned}
|r(\mathbf{z},\mathbf{p},\boldsymbol{\theta}) - r(\mathbf{z},\mathbf{p},\hat{\boldsymbol{\theta}}(R))| &\overset{(i)}{\leq} \sum_{j\in\mathcal{J}} p_j |\mathbb{P}(j;\mathbf{z},\mathbf{p},\boldsymbol{\theta}) - \mathbb{P}(j;\mathbf{z},\mathbf{p},\hat{\boldsymbol{\theta}}(R))| \\
&\overset{(ii)}{\leq} \sum_{j\in\mathcal{J}} \frac{1}{4} p_j \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_1 \\
&\overset{(iii)}{\leq} \frac{J p_{max}}{4} \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_1 \\
&\overset{(iv)}{\leq} \frac{J p_{max}}{4} \sqrt{J(d+1)} \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2,
\end{aligned}$$

where (i) is due to the triangle inequality; (ii) results from the fact that

$$\|\nabla \mathbb{P}(j;\mathbf{z},\mathbf{p},\boldsymbol{\theta})\|_\infty \leq 1/4$$

since we can show, for each $j \in \mathcal{J}$,

$$\frac{\delta}{\delta\theta_{jk}} \mathbb{P}(j;\mathbf{z},\mathbf{p},\theta) \leq \frac{1}{4}\|z\|_\infty \leq \frac{1}{4} \; \forall\, k \in \mathcal{J}, z \in \mathcal{Z}, \boldsymbol{\theta} \in \Theta;$$

(iii) is simply upper-bounding $p_j$'s by $p_{max}$ and adding up the sum; and (iv) is due to the Cauchy–Schwarz inequality. □

**Theorem 2 (Theorem 1, Chen et al., 2022 )** *Under Assumptions 1 and 2 and for $n \geq \frac{4C_\psi \log(n)}{\min(\rho,1)^2}$ for some constant $C_\psi$, the following holds with probability at least $[1-(1+2J)/n)]$:*

$$\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2 \leq 2\frac{[1 + \exp(-R) + (J-1)\exp(R)]^2}{\exp(-R)\rho}\sqrt{\frac{2J(d+1)\log(2J(d+1))}{n}}.$$

The proof of Theorem 2 is rather long, so we defer the reader to Chen et al. (2022) for the details.

***Proof (of Theorem 1)*** By adding and subtracting the same expression $r(\mathbf{z}, \hat{\mathbf{p}}, \hat{\boldsymbol{\theta}}(R))$, we can rewrite the revenue difference into two separate differences, which can be bounded as follows:

$$\begin{aligned} r(\mathbf{z}, \mathbf{p}^*, \boldsymbol{\theta}) - r(\mathbf{z}, \hat{\mathbf{p}}, \boldsymbol{\theta}) &= [r(\mathbf{z}, \mathbf{p}^*, \boldsymbol{\theta}) - r(\mathbf{z}, \hat{\mathbf{p}}, \hat{\boldsymbol{\theta}}(R))] + [r(\mathbf{z}, \hat{\mathbf{p}}, \hat{\boldsymbol{\theta}}(R)) - r(\mathbf{z}, \hat{\mathbf{p}}, \boldsymbol{\theta})] \\ &\leq [r(\mathbf{z}, \mathbf{p}^*, \boldsymbol{\theta}) - r(\mathbf{z}, \mathbf{p}^*, \hat{\boldsymbol{\theta}}(R))] + [r(\mathbf{z}, \hat{\mathbf{p}}, \hat{\boldsymbol{\theta}}(R)) - r(\mathbf{z}, \hat{\mathbf{p}}, \boldsymbol{\theta})] \\ &\leq \frac{C(\psi, R)}{\rho} J^4(d+1)\sqrt{\frac{\log(2nJ(d+1))}{n}}, \end{aligned}$$

where the first inequality results from the fact that $\hat{\mathbf{p}}$ is the maximal price for $r(\mathbf{z}, \cdot, \hat{\boldsymbol{\theta}}(R))$, and the final inequality results from applying Lemma 1 and Theorem 2 twice. □

## 7.4 Dynamic Pricing with High-Dimensional Data

In many situations, a company that wishes to use high-dimensional data (whether it be customer characteristics or product features) for pricing decisions may not have collected such data in the past. In this situation, the static setting of Sect. 7.3 does not apply, and so the company would need to balance data collection and learning about the best price to charge (exploration) with charging the best current estimate of the optimal price (exploitation).

Several recent works (Qing and Bayati, 2016; Javanmard and Nazerzadeh, 2019; Cohen et al., 2020; Ban and Keskin, 2021) investigate the problem of dynamic pricing with high-dimensional data. In this section, we first review the theoretical results in Ban and Keskin (2021), then discuss how this work contrasts with the others. Note any notations introduced in this section are independent of notations used in previous sections.

### *7.4.1 Feature-Dependent Demand Model*

Ban and Keskin (2021) consider a firm that offers a product for sale to $T$ customers who arrive sequentially. Each sales opportunity is considered to be a separate time

period and the firm has a discrete time horizon of $T$ periods and can dynamically adjust the product's price over the time horizon.

At the beginning of period $t = 1, 2, \ldots, T$, the firm observes a $d$-dimensional vector of features pertaining to the customer arriving in period $t$. Ban and Keskin (2021) denote this random vector by $Z_t = (Z_{t1}, Z_{t2}, \ldots, Z_{td})$ and assumes that $\{Z_t,\ t = 1, 2, \ldots, T\}$ are independent and identically distributed with a compact support $\mathcal{Z} \subseteq B_0(z_{\max}) \subset \mathbb{R}^d$, where $B_0(z_{\max})$ is the $d$-dimensional ball of radius $z_{\max} > 0$, and $\mathbb{E}[Z_t]$ is, without loss of generality, assumed to be normalized to 0. Denote by $\Sigma_Z = \mathbb{E}[Z_t Z_t^\top]$ the covariance matrix of $\{Z_t\}$ and assume that $\Sigma_Z$ is a symmetric and positive definite matrix. Note that the firm need not know $\Sigma_Z$.

Ban and Keskin (2021) allow $Z_t$ to include individual customer characteristics, features about the product, and macroeconomic factors that may be both categorical (e.g., postal code and income bracket) and continuous (e.g., credit score). This means that some components of $Z_t$ are continuous random variables and others discrete random variables. For the features modeled as continuous random variables, the only assumption Ban and Keskin (2021) make is that they have positive measure in the interior of their domains and zero on the boundary.

For convenience, let $X_t := \left[\begin{smallmatrix} 1 \\ Z_t \end{smallmatrix}\right] \in \mathbb{R}^{d+1}$. Accordingly, denote the support of $X_t$ by $\mathcal{X} = \{1\} \times \mathcal{Z}$ and the expectation over the product measure on $X_1 \times \cdots \times X_T$ by $\mathbb{E}_X\{\cdot\}$.

Upon observing $X_t = x_t$, the firm chooses a price $p_t \in [\ell, u]$ to be offered to the customer arriving in period $t$, where $0 < \ell < u < \infty$. Then, the firm observes this customer's demand in response to $p_t$, which is given by

$$D_t = g\big(\alpha \cdot x_t + (\beta \cdot x_t)\, p_t\big) + \varepsilon_t \quad \text{for } t = 1, 2, \ldots, T, \tag{7.11}$$

where $\alpha, \beta \in \mathbb{R}^{d+1}$ are demand parameter vectors unknown to the firm, $g(\cdot)$ is a known function, $\varepsilon_t$ is the unobservable and idiosyncratic demand shock of the customer arriving in period $t$, and $u \cdot v = \sum_{i=1}^{d+1} u_i v_i$ denotes the inner product of vectors $u$ and $v$. Note that the demand model (7.11) captures feature-dependent customer taste and potential market size (through $\alpha \cdot x_t$) as well as feature-dependent price sensitivity (through $\beta \cdot x_t$).

Let $\theta := (\alpha, \beta)$ be the vector of all unknown demand parameters and $\Theta$ be a compact rectangle in $\mathbb{R}^{2(d+1)}$ from which the value of $\theta$ is chosen. The dimension $d$ is allowed to be large, possibly larger than the selling horizon $T$, but it is also assumed that a smaller subset of the $d$ features have a sizable effect in the demand model. Ban and Keskin (2021) denote this sparsity structure as follows: $\mathcal{S}_\alpha := \{i = 1, \ldots, d+1 : \alpha_i \neq 0\}$, $\mathcal{S}_\beta := \{i = 1, \ldots, d+1 : \beta_i \neq 0\}$, and $\mathcal{S} := \mathcal{S}_\alpha \cup \mathcal{S}_\beta$. Note that $\mathcal{S}$ contains the indices of all nonzero components of $\alpha$ and $\beta$. For notational convenience, use the set $\mathcal{S}$ to express the sparsity structure in the unknown parameter vector $\theta = (\alpha, \beta)$. (If the nonzero components of $\alpha$ and $\beta$ are distinct, one could use $\mathcal{S}_\alpha$ and $\mathcal{S}_\beta$ to express the sparsity structures in $\alpha$ and $\beta$ separately; the analysis is valid for that case because $\mathcal{S}$ already includes all components that influence demand.) Define $\alpha_\mathcal{S} = (\alpha_i)_{i \in \mathcal{S}}$ and $\beta_\mathcal{S} = (\beta_i)_{i \in \mathcal{S}}$ as the vectors consisting of the components of $\alpha$ and $\beta$, respectively, whose indices

are in $\mathcal{S}$, and $\theta_{\mathcal{S}} = (\alpha_{\mathcal{S}}, \beta_{\mathcal{S}})$. Note that $\theta_{\mathcal{S}}$ is a compressed vector that contains all nonzero components of $\theta$; hence, refer to $\theta_{\mathcal{S}}$ as the *compressed version* of $\theta$. Let $s \in \{1, \ldots, d+1\}$ be the cardinality of $\mathcal{S}$, and denote the compressed versions of the key quantities defined earlier with a subscript $\mathcal{S}$. Thus, the compressed version of $\Theta$ is $\Theta_{\mathcal{S}} = \{\theta_{\mathcal{S}} : \theta \in \Theta\} \subset \mathbb{R}^{2s}$. For $t = 1, \ldots, T$, the compressed versions of $Z_t$ and $X_t$ are $Z_{\mathcal{S},t} \in \mathcal{Z}_{\mathcal{S}} \subset \mathbb{R}^s$ and $X_{\mathcal{S},t} = \left[\begin{smallmatrix} 1 \\ Z_{\mathcal{S},t} \end{smallmatrix}\right] \in \mathcal{X}_{\mathcal{S}} \subset \mathbb{R}^{s+1}$, respectively, where $\mathcal{Z}_{\mathcal{S}} = \{(z_i)_{i \in \mathcal{S}} : z \in \mathcal{Z}\}$ and $\mathcal{X}_{\mathcal{S}} = \{1\} \times \mathcal{Z}_{\mathcal{S}}$. The firm is not assumed to know the sparsity structure a priori.

The demand function in (7.11) is known as a generalized linear model (GLM) because, given $x \in \mathcal{X}$, the function that maps price $p$ to expected demand is the composition of the function $g : \mathbb{R} \to \mathbb{R}$ and the linear function $p \mapsto \alpha \cdot x + (\beta \cdot x)\, p$. In this relationship, the function $g(\cdot)$ is referred to as the "link" function that captures potential nonlinearities in the demand–price relationship. Ban and Keskin (2021) assume that $g(\cdot)$ is differentiable and increasing; this is satisfied for a broad family of functions including linear, logit, probit, and exponential demand functions. It also implies that the link function has bounded derivatives over its compact domain.[1]

Ban and Keskin (2021) assume that $\{\varepsilon_t,\ t = 1, 2, \ldots\}$ is a sub-Gaussian martingale difference sequence; that is, $\mathbb{E}[\varepsilon_t | \mathcal{F}_{t-1}] = 0$, and there exist positive constants $\sigma_0$ and $\eta_0$ such that $\mathbb{E}[\varepsilon_t^2 | \mathcal{F}_{t-1}] \le \sigma_0^2$ and $\mathbb{E}[e^{\eta \varepsilon_t} | \mathcal{F}_{t-1}] < \infty$ for all $\eta$ satisfying $|\eta| < \eta_0$, where $\mathcal{F}_t = \sigma(p_1, \ldots, p_t, \varepsilon_1, \ldots, \varepsilon_t, X_1, \ldots, X_{t+1})$ and the construction of admissible price sequences $\{p_t,\ t = 1, 2, \ldots\}$ is specified below. (A simple example of this setting is where $\{\varepsilon_t\}$ are bounded and have zero mean.) We note that the distribution of $\varepsilon_t$ can depend on price and feature observations. This implies that the idiosyncratic demand shocks of customers are allowed to be dependent on prices and customer features in this formulation, which contrasts with the static pricing model of Sect. 7.3. Also note that the generality of the above demand-shock distribution allows for continuous as well as discrete demand distributions. A noteworthy example within discrete demand distributions is the binary customer response model, where $\{\varepsilon_t\}$ are such that $D_t \in \{0, 1\}$ for all $t$. In this case, the event $\{D_t = 1\}$ corresponds to a sale at the offered price $p_t$, whereas $\{D_t = 0\}$ corresponds to no sale.

Given $\theta = (\alpha, \beta) \in \Theta$ and $x = \left[\begin{smallmatrix} 1 \\ z \end{smallmatrix}\right] \in \mathcal{X}$, the firm's expected single-period revenue is

$$r(p, \theta, x) = p\big[g\big(\alpha \cdot x + (\beta \cdot x)p\big)\big] \text{ for } p \in [\ell, u]. \tag{7.12}$$

Let $\varphi(\theta, x) = \operatorname{argmax}_p\{r(p, \theta, x)\}$ denote the unconstrained revenue-maximizing price in terms of $\theta \in \Theta$ and $x \in \mathcal{X}$. Ban and Keskin (2021) assume that $\varphi(\theta, x)$ is in the interior of the feasible set $[\ell, u]$ for all $\theta \in \Theta$ and $x \in \mathcal{X}$.

[1] That is, there exist $\tilde{\ell}, \tilde{u} \in \mathbb{R}$ satisfying $0 < \tilde{\ell} \le |g'(\xi)| \le \tilde{u} < \infty$ for all $\xi = \alpha \cdot x + (\beta \cdot x)\, p$ such that $(\alpha, \beta) \in \Theta$, $x \in \mathcal{X}$, and $p \in [\ell, u]$ (here and later, a prime denotes a derivative).

### 7.4.2 Learning-and-Earning Algorithm

Let $H_0 = X_1$, and for $t = 1, 2, \ldots, T$, let $H_t$ be a vector consisting of the observations until just after the beginning of period $t + 1$, when the feature vector for period $t + 1$ has been revealed but before the pricing decision; i.e., $H_t = (p_1, \ldots, p_t, D_1, \ldots, D_t, X_1, \ldots, X_{t+1})$. Define an *admissible policy* as a sequence of functions $\pi = (\pi_1, \pi_2, \ldots)$, where $\pi_t : \mathbb{R}^{(d+3)t-2} \to [\ell, u]$ is a measurable function that maps $H_{t-1}$ to the price to be offered in period $t$. Thus, $p_t = \pi_t(H_{t-1})$ for all $t = 1, 2, \ldots, T$, under policy $\pi$. Denote by $\Pi$ the set of all admissible pricing policies. Given $\pi \in \Pi$ and $\theta = (\alpha, \beta) \in \Theta$, define a probability measure $\mathbb{P}_\theta^\pi\{\cdot\}$ on the sample space of demand sequences $D = (D_1, D_2, \ldots)$ such that

$$\mathbb{P}_\theta^\pi\{D_1 \in d\xi_1, \ldots, D_T \in d\xi_T\} = \prod_{t=1}^{T} \mathbb{P}_\varepsilon\left\{g\left(\alpha \cdot X_t + (\beta \cdot X_t)p_t\right) + \varepsilon_t \in d\xi_t \,\middle|\, H_{t-1}\right\}$$
$$\text{for } \xi_1, \xi_2, \ldots, \xi_T \in \mathbb{R},$$

where $\mathbb{P}_\varepsilon\{\cdot\}$ is the probability measure governing $\{\varepsilon_t, t = 1, 2, \ldots\}$. The firm's conditional expected revenue loss in $T$ periods relative to a clairvoyant who knows the underlying demand parameter vector $\theta$ is defined as

$$\Delta_\theta^\pi(T; \boldsymbol{X}_T) = \mathbb{E}_\theta^\pi\left\{\sum_{t=1}^{T}\left[r^*(\theta, X_t) - r(p_t^\pi, \theta, X_t)\right] \,\middle|\, \boldsymbol{X}_T\right\} \tag{7.13}$$

for $\theta \in \Theta$, $\pi \in \Pi$, and $\boldsymbol{X}_T = (X_1, \ldots, X_T) \in \mathcal{X}^T$, where $\mathbb{E}_\theta^\pi\{\cdot\}$ is the expectation operator associated with $\mathbb{P}_\theta^\pi\{\cdot\}$, $r^*(\theta, x) = r(\varphi(\theta, x), \theta, x)$ is the maximum single-period revenue function, and $p_t^\pi$ is the price charged in period $t$ under policy $\pi$. This performance metric is the firm's $T$-period *conditional regret*, which is a random variable that depends on the realization of $\boldsymbol{X}_T = (X_1, \ldots, X_T)$. The firm's objective is to minimize its $T$-period *expected regret*, given by

$$\Delta_\theta^\pi(T) = \mathbb{E}_X\left\{\Delta_\theta^\pi(T; \boldsymbol{X}_T)\right\} \tag{7.14}$$

for $\theta \in \Theta$ and $\pi \in \Pi$, where $\mathbb{E}_X\{\cdot\}$ is the expectation operator associated with the probability measure governing $\{X_t, t = 1, 2, \ldots\}$. Throughout the sequel, we use the expectation notation $\mathbb{E}_{X,\theta}^\pi\{\cdot\} := \mathbb{E}_X\{\mathbb{E}_\theta^\pi\{\cdot\}\}$, and let $\mathbb{P}_{X,\theta}^\pi\{\cdot\}$ be the probability measure associated with $\mathbb{E}_{X,\theta}^\pi\{\cdot\}$. Finally, Ban and Keskin (2021) focus on the firm's *worst-case expected regret*, defined as $\Delta^\pi(T) = \sup\{\Delta_\theta^\pi(T) : \theta \in \Theta\}$ to analyze the complexity of the learning problem.

Given a history of feature vectors $(X_1, \ldots, X_t) = (x_1, \ldots, x_t)$, let

$$Q_t(\tilde{\theta}, \tilde{\lambda}) := \sum_{k=1}^{t} \chi_k \int_{D_k}^{g(\tilde{\theta} \cdot u_k)} \frac{D_k - y}{\nu(y)}\, dy - \tilde{\lambda}\|\tilde{\theta}\|_1 \quad \text{for } \tilde{\theta} \in \mathbb{R}^{2(d+1)} \text{ and } \tilde{\lambda} \geq 0, \tag{7.15}$$

where $\chi_k = \mathbb{I}\{k \in M\}$ and $u_k = \begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes x_k$ for $k \in \{1, 2, \ldots\}$, $\nu(y) = g'\big(g^{-1}(y)\big)$ for $y \in \mathbb{R}$, and $\|\tilde{\theta}\|_1 = \sum_{i=1}^{2(d+1)} |\tilde{\theta}_i|$ denotes the $\ell_1$-norm of $\tilde{\theta}$. The function $Q_t(\cdot, \cdot)$ in (7.15) is a lasso-regularized quasi-likelihood function for the firm's observations in the first $t$ periods (for a reference on maximum quasi-likelihood estimation, see Nelder and Wedderburn (1972)). This function subsumes the lasso regression estimation objective and standard maximum likelihood estimation with lasso regularization. It is also worth noting that, given $\tilde{\lambda} \geq 0$, the mapping $\tilde{\theta} \mapsto Q_t(\tilde{\theta}, \tilde{\lambda})$ is strictly concave and has a unique maximizer.

Ban and Keskin (2021) propose the following learning-and-earning algorithm, called *iterated lasso-regularized quasi-likelihood regression with price experimentation* (abbreviated ILQX). Upon observing the feature vector $X_t = x_t$ in period $t$, the ILQX policy with nonnegative parameters $m_1$ and $m_2$, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots)$, denoted by ILQX$(m_1, m_2, \boldsymbol{\lambda})$, charges the price

$$p_t = \begin{cases} m_1 & \text{if } t \in M_1, \\ m_2 & \text{if } t \in M_2, \\ \varphi\big(\hat{\vartheta}_t^{(\text{lasso})}(\lambda_t), x_t\big) & \text{otherwise,} \end{cases} \tag{7.16}$$

where $\hat{\vartheta}_t^{(\text{lasso})}(\lambda_t)$ is given by the following maximum quasi-likelihood estimation:

$$\hat{\theta}_{t+1}^{(\text{lasso})}(\tilde{\lambda}) = \text{argmax}_{\tilde{\theta} \in \mathbb{R}^{2(d+1)}} \big\{ Q_t(\tilde{\theta}, \tilde{\lambda}) \big\}, \tag{7.17}$$

with $\hat{\vartheta}_{t+1}^{(\text{lasso})}(\tilde{\lambda})$ being the truncated estimate satisfying $\hat{\vartheta}_{t+1}^{(\text{lasso})}(\tilde{\lambda}) = \mathcal{P}_\Theta\{\hat{\theta}_{t+1}^{(\text{lasso})}(\tilde{\lambda})\}$ for $\tilde{\lambda} \geq 0$, where $\mathcal{P}_\Theta : \mathbb{R}^{2(d+1)} \to \Theta$ denotes the projection mapping from $\mathbb{R}^{2(d+1)}$ onto $\Theta$.

The prices $m_1$ and $m_2$ are two distinct *experimental prices* in $[\ell, u]$, such that the number of price experiments conducted over periods $\{1, 2, \ldots, t\}$ is at least in the order of $\sqrt{t}$, the reason for which will be made clear in Sect. 7.4.3. For instance, the following scheme would work: for $i \in \{1, 2\}$, the set of periods in which the experimental price $m_i$ is charged could be

$$M_i = \{t = L^2 + i - 1 : L = 1, 2, \ldots\}. \tag{7.18}$$

Denote by $M = M_1 \cup M_2$ the set of all experimentation periods. This price experimentation scheme ensures that, for all $t \geq 5$, each experimental price is charged at least $\frac{1}{4}\sqrt{t}$ times. This scheme uses two prices for experimentation—one needs at least two distinct experimental prices to ensure that regression estimates are well defined in all periods. Ban and Keskin (2021) use two experimental prices throughout the paper; however, all the results remain valid if more than two experimental prices are used.

In Ban and Keskin (2021), the demand model (7.11) and the learning-and-earning algorithm ILQX are numerically evaluated on simulated data and real data from an online car loan company. In what follows, however, we focus on the theoretical performance analysis. Specifically, we present the bounds on the expected regret derived in Ban and Keskin (2021). In contrast to Sect. 7.3, here we are able to present both a lower bound and an upper bound on the revenue gap. The lower

bound is universal in that it specifies a limit to how fast *any* admissible pricing policy could learn the optimal personalized price, for *any* differentiable and increasing link function $g(\cdot)$ in the demand model (7.11). The upper bound is specific to the ILQX algorithm described in Sect. 7.4.2, but it matches the universal lower bound to logarithmic terms, so by deduction, the ILQX algorithm must be rate-optimal with respect to measuring the performance by expected regret.

### *7.4.3 A Universal Lower Bound on the Regret*

To characterize the complexity of the problem in terms of the best achievable regret performance, Ban and Keskin (2021) focus on a special case of the general demand model (7.11) by letting the expected demand be a linear function of the price; i.e., $g(\xi) = \xi$ for all $\xi \in \mathbb{R}$. In this case, the demand in period $t$ is given by

$$D_t = \alpha \cdot x_t + (\beta \cdot x_t)\, p_t + \varepsilon_t \quad \text{for } t = 1, 2, \ldots, T. \tag{7.19}$$

Note that (7.19) is a high-dimensional personalized version of the well-known linear demand model. Also, note that the unconstrained revenue-maximizing price is $\varphi(\theta, x) = -(\alpha \cdot x)/(2\beta \cdot x)$ for $\theta = (\alpha, \beta) \in \Theta$ and $x \in \mathcal{X}$.

Ban and Keskin (2021) derive the following lower bound on the firm's expected regret under any admissible policy.

**Theorem 3 (Theorem 1, Ban and Keskin, 2021)** *Let $\{D_t\}$ be given by the linear demand model (7.19), and $\varepsilon_t$, $t \in \mathbb{Z}_+$, are independent and identically distributed from an exponential family of distributions. Then, there exists a finite positive constant $c$ such that*

$$\Delta^{\pi}(T) \geq cs\sqrt{T} \quad \textit{for all } \pi \in \Pi \textit{ and } T \geq 2.$$

We note that the restriction to the linear demand model in the statement of Theorem 3 is not prohibitive because the result implies

$$\sup_{g \in G} \inf_{\pi \in \Pi} \{\Delta^{g,\pi}(T)\} \geq cs\sqrt{T}, \ \forall T \geq 2,$$

where $G$ denotes the set of all differentiable and increasing functions, and $\Delta^{g,\pi}(T)$ is the $T$-period expected regret of policy $\pi$, with its dependence on the link function $g(\cdot)$ expressed explicitly. Thus, the lower bound in Theorem 3 is a worst-case lower bound on the minimum regret for a broad class of link functions.

Theorem 3 characterizes the complexity of the personalized dynamic pricing problem of Ban and Keskin (2021). It states that the expected regret of any admissible policy must grow at least in the order of $s\sqrt{T}$. It is worth mentioning here that, due to the sparsity assumption, this limit to the rate of learning does not depend on $d$.

The proof of Theorem 3 requires the following lemma.

**Lemma 2 (Lemma EC.1, Ban and Keskin, 2021)** *There exist finite positive constants $c_0$ and $c_1$ such that*

$$\sup_{\theta\in\Theta}\left\{\sum_{t=2}^{T}\mathbb{E}_X\mathbb{E}_\theta^\pi[(p_t-\varphi(\theta,X_t))^2]\right\} \geq \sum_{t=2}^{T}\frac{c_0}{c_1+\sup_{\theta\in\Theta}\left\{\mathbb{E}_X\left[C_t(\theta,\boldsymbol{X}_t)\,\mathbb{E}_\theta^\pi[\mathcal{J}_{t-1}(\boldsymbol{X}_{t-1})]\,C_t(\theta,\boldsymbol{X}_t)^\intercal\right]\right\}},$$

*where $C_t(\cdot\,,\cdot)$ is a $2(d+1)$-dimensional function on $\Theta\times\mathcal{X}^t$ such that*

$$C_t(\theta,\boldsymbol{x}_t)=\left[-\textstyle\sum_{k=1}^t\varphi(\theta,x_k)\,v_k^\intercal \quad \sum_{k=1}^t v_k^\intercal\right],$$

*and $v_k\in\mathbb{R}^{d+1}$ are (column) vectors constructed as follows: for $k=1,\ldots,t$, let $v_k:=\sum_{\ell=1}^t\gamma_{k\ell}\,x_\ell$, where $\{\gamma_{k\ell},\ \ell=1,\ldots,t\}$ solve the following $t$ equations:*

$$\sum_{\ell=1}^{t}\gamma_{k\ell}\,x_\ell^\intercal\,x_{\ell'}=\begin{cases}1 & \text{if } \ell'=k,\\ 0 & \text{otherwise.}\end{cases}$$

***Proof (of Lemma 2)*** Let $\mu$ be an absolutely continuous density on $\Theta$, taking positive values on the interior of $\Theta$ and zero on its boundary, and let $\mathbb{E}_\mu\{\cdot\}$ be the expectation operator associated with the density $\mu$. We consider estimating the vector $[\varphi(\theta,X_1),\ldots,\varphi(\theta,X_T)]$.

Given that the components of $X_t$ are continuous random variables, with positive measure in the interior of $\mathcal{X}$ and zero on the boundary, the multivariate van Trees inequality (Gill and Levit, 1995) implies that

$$\mathbb{E}_{\mu,X}\left\{\mathbb{E}_\theta^\pi\left[(p_t-\varphi(\theta,X_t))^2\right]\right\}\geq\frac{\left(\mathbb{E}_{\mu,X}\{\mathsf{tr}[C_t(\theta,\boldsymbol{X}_t)(\partial\varphi(\theta,X_t)/\partial\theta)^\intercal]\}\right)^2}{\mathbb{E}_{\mu,X}\{\mathsf{tr}[C_t(\theta,\boldsymbol{X}_t)\mathcal{I}_{t-1}^\pi(\boldsymbol{X}_t)C_t(\theta,\boldsymbol{X}_t)^\intercal]\}+\tilde{\mathcal{I}}(\mu)},\tag{7.20}$$

where $\tilde{\mathcal{I}}(\mu)$ is a constant that depends on $\mu$, and $\mathbb{E}_{\mu,X}=\mathbb{E}_\mu\{\mathbb{E}_X(\cdot)\}$. Since

$$\frac{\partial\varphi(\theta,x_t)}{\partial\theta}=\left[-\frac{x_t}{2\beta\cdot x_t}\quad -\frac{\varphi(\theta,x_t)x_t}{\beta\cdot x_t}\right],$$

we have

$$\mathsf{tr}\left[C_t(\theta,\boldsymbol{x}_t)\frac{\partial\varphi(\theta,x_t)^\intercal}{\partial\theta}\right]=-\frac{\varphi(\theta,x_t)}{2\beta\cdot x_t}.$$

By (7.22), we have $\mathcal{I}^{\pi}_{t-1}(\boldsymbol{X}_t) = \zeta(\phi)\,\mathbb{E}^{\pi}_{\theta}[\mathcal{J}_{t-1}(\boldsymbol{X}_t)] = \zeta(\phi)\,\mathbb{E}^{\pi}_{\theta}[\mathcal{J}_{t-1}(\boldsymbol{X}_{t-1})]$. Using these facts and summing up over $t = 2, \ldots, T$,

$$\sum_{t=2}^{T} \mathbb{E}_{\mu,X}\left\{\mathbb{E}^{\pi}_{\theta}\left[(p_t - \varphi(\theta, X_t))^2\right]\right\} \geq \sum_{t=2}^{T} \frac{\left(\mathbb{E}_{\mu,X}\left[\dfrac{\varphi(\theta, X_t)}{2\beta \cdot X_t}\right]\right)^2}{\zeta(\phi)\,\mathbb{E}_{\mu,X}\{\mathsf{tr}[C_t(\theta, \boldsymbol{X}_t)\mathbb{E}^{\pi}_{\theta}[\mathcal{J}_{t-1}(\boldsymbol{X}_{t-1})]C_t(\theta, \boldsymbol{X}_t)^{\intercal}]\} + \tilde{\mathcal{I}}(\mu)},$$

and since $\mathbb{E}_{\mu}\{\cdot\}$ is a monotone operator,

$$\sup_{\theta\in\Theta} \sum_{t=2}^{T} \mathbb{E}_X\{\mathbb{E}^{\pi}_{\theta}[(p_t - \varphi(\theta, X_t))^2]\} \geq \sum_{t=2}^{T} \frac{\inf\limits_{\theta\in\Theta} \mathbb{E}_X\left(\dfrac{\varphi(\theta, X_t)}{2\beta \cdot X_t}\right)^2}{\zeta(\phi) \sup\limits_{\theta\in\Theta} \mathbb{E}_X\{\mathsf{tr}\left[C_t(\theta, \boldsymbol{X}_t)\mathbb{E}^{\pi}_{\theta}[\mathcal{J}_{t-1}(\boldsymbol{X}_{t-1})]C_t(\theta, \boldsymbol{X}_t)^{\intercal}\right]\} + \tilde{\mathcal{I}}(\mu)}.$$

Because $0 < \ell \leq \varphi(\theta, x)$ for all $\theta$ and $x$, the numerator of the right-hand side of the preceding inequality is greater than or equal to $\ell^2/[4\beta^2_{\max}(\max\{1, z_{\max}\})^2]$, where $\beta_{\max} = \max_{(\alpha,\beta)\in\Theta}\{\|\beta\|\}$. Thus, letting $c_0 = \ell^2/[4\zeta(\phi)\beta^2_{\max}(\max\{1, z_{\max}\})^2]$ and $c_1 = \tilde{\mathcal{I}}(\mu)/\zeta(\phi)$, we arrive at the desired result. □

***Proof (of Theorem 3)*** First, assume that all components of $X_t$ are continuous random variables, with positive measure in the interior of $\mathcal{X}$ and zero on the boundary, and show at the end that this can be generalized to $X_t$ with discrete components.

Ban and Keskin (2021) derive the lower bound for the more general case where the distribution of $\{\varepsilon_t\}$ is from the exponential family of distributions; that is, $\{\varepsilon_t\}$ are independent and identically distributed random variables whose density has the following parametric form: $f_{\varepsilon}(\xi \mid \phi) = e^{\phi\cdot\boldsymbol{T}(\xi) - A(\phi) + B(\xi)}$, where $\phi \in \mathbb{R}^n$ is the vector of distribution parameters, $\boldsymbol{T} : \mathbb{R} \to \mathbb{R}^n$, $A : \mathbb{R}^n \to \mathbb{R}$, and $B : \mathbb{R} \to \mathbb{R}$ are differentiable functions, and $n$ is a natural number that represents the number of distribution parameters. Note that the case where $\varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_0^2)$ is a special case of the above setting, obtained by letting $n = 1$, $\phi = -\frac{1}{2\sigma_0^2}$, $\boldsymbol{T}(\xi) = \xi^2$ and $B(\xi) = -\frac{1}{2}\log(2\pi)$ for all $\xi$, and $A(\phi) = \frac{1}{2}\log\left(-\frac{1}{2\phi}\right)$ for all $\phi$.

Given $\theta = (\alpha, \beta) \in \Theta$ and conditional on $\boldsymbol{X}_T = \boldsymbol{x}_T$, the density of the history vector $H_t = (p_1, \ldots, p_t, D_1, \ldots, D_t, X_1, \ldots, X_{t+1})$ is given by

$$\ell_t(H_t, \theta, \boldsymbol{x}_T) = \prod_{k=1}^{t} f_\varepsilon\big(D_k - \alpha \cdot x_k - (\beta \cdot x_k) p_k \,\big|\, \phi\big) \text{ for } t = 1, 2, \ldots, T. \tag{7.21}$$

By elementary analysis, (7.21) implies that $H_t$ has the following Fisher information matrix under any given admissible policy $\pi \in \Pi$:

$$\begin{aligned} \mathcal{I}_t^\pi(\boldsymbol{x}_T) :&= \mathbb{E}_\theta^\pi \left\{ \left[ \frac{\partial \log \ell_t(H_t, \theta, \boldsymbol{x}_T)}{\partial \theta} \right]^\mathsf{T} \left[ \frac{\partial \log \ell_t(H_t, \theta, \boldsymbol{x}_T)}{\partial \theta} \right] \right\} \\ &= \zeta(\phi)\, \mathbb{E}_\theta^\pi[\mathcal{J}_t(\boldsymbol{x}_T)], \end{aligned} \tag{7.22}$$

where $\zeta(\phi) = \mathbb{E}_\theta^\pi[\phi \cdot \nabla \boldsymbol{T}(\varepsilon_1) + B'(\varepsilon_1)]$, $\nabla \boldsymbol{T}(\xi) = \big(\frac{\partial}{\partial \xi}\boldsymbol{T}_1(\xi), \frac{\partial}{\partial \xi}\boldsymbol{T}_2(\xi), \ldots, \frac{\partial}{\partial \xi}\boldsymbol{T}_n(\xi)\big)$ and $B'(\xi) = \frac{\partial}{\partial \xi} B(\xi)$ for all $\xi$, $\mathcal{J}_t(\boldsymbol{x}_T)$ is the empirical Fisher information matrix given by

$$\mathcal{J}_t(\boldsymbol{x}_T) = \begin{bmatrix} \sum_{k=1}^t x_k x_k^\mathsf{T} & \sum_{k=1}^t p_k x_k x_k^\mathsf{T} \\ \sum_{k=1}^t p_k x_k x_k^\mathsf{T} & \sum_{k=1}^t p_k^2 x_k x_k^\mathsf{T} \end{bmatrix} = \sum_{k=1}^{t} \left( \begin{bmatrix} 1 \\ p_k \end{bmatrix} \cdot \begin{bmatrix} 1 \\ p_k \end{bmatrix}^\mathsf{T} \right) \otimes x_k x_k^\mathsf{T},$$

and $\otimes$ denotes the Kronecker product of matrices. In the remainder of the proof, we consider two cases:

*Case 1:* $d + 1 \geq T$. In this case, we use the following lemma.

For each $k = 1, \ldots, t$, the constants $\{\gamma_{k\ell},\ \ell = 1, \ldots, t\}$ in Lemma 2 are found by solving the following system of linear equations:

$$\underline{\mathbf{X}}_t^\mathsf{T} \underline{\mathbf{X}}_t \gamma_k = \mathbf{e}_k, \tag{7.23}$$

where $\underline{\mathbf{X}}_t = [x_1, \ldots, x_t]$ is the $(d+1) \times t$ matrix of feature vectors up to time $t$, $\gamma_k = (\gamma_{k1}, \ldots, \gamma_{kt}) \in \mathbb{R}^t$, and $\mathbf{e}_k \in \mathbb{R}^t$ is the $k$-th basis vector in $\mathbb{R}^t$. Because $d + 1 \geq T$, the matrix $\underline{\mathbf{X}}_t^\mathsf{T} \underline{\mathbf{X}}_t$ is full-rank; hence, there exists a unique solution for $\gamma_k$. Note that

$$\begin{aligned} &C_t(\theta, \boldsymbol{x}_t)\, \mathcal{J}_{t-1}(\boldsymbol{x}_t)\, C_t(\theta, \boldsymbol{x}_t)^\mathsf{T} \\ &= \begin{bmatrix} -\sum_{k=1}^t \varphi(\theta, x_k)\, v_k^\mathsf{T} & \sum_{k=1}^t v_k^\mathsf{T} \end{bmatrix} \begin{bmatrix} \sum_{k=1}^{t-1} x_k x_k^\mathsf{T} & \sum_{k=1}^{t-1} p_k x_k x_k^\mathsf{T} \\ \sum_{k=1}^{t-1} p_k x_k x_k^\mathsf{T} & \sum_{k=1}^{t-1} p_k^2 x_k x_k^\mathsf{T} \end{bmatrix} \\ &\quad \times \begin{bmatrix} -\sum_{k=1}^t \varphi(\theta, x_k) v_k \\ \sum_{k=1}^t v_k \end{bmatrix} \\ &= \begin{bmatrix} -\sum_{k=1}^t \varphi(\theta, x_k)\, v_k^\mathsf{T} & \sum_{k=1}^t v_k^\mathsf{T} \end{bmatrix} \begin{bmatrix} -\sum_{k=1}^{t-1} \sum_{k'=1}^t \{\varphi(\theta, x_{k'}) x_k x_k^\mathsf{T} v_{k'} + p_k x_k x_k^\mathsf{T} v_{k'}\} \\ -\sum_{k=1}^{t-1} \sum_{k'=1}^t \{p_k \varphi(\theta, x_{k'}) x_k x_k^\mathsf{T} v_{k'} + p_k^2 x_k x_k^\mathsf{T} v_{k'}\} \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
&\overset{(a)}{=} \left[ -\sum_{k=1}^{t} \varphi(\theta, x_k)\, v_k^{\mathsf{T}} \;\; \sum_{k=1}^{t} v_k^{\mathsf{T}} \right] \begin{bmatrix} -\sum_{k=1}^{t-1} \{\varphi(\theta, x_k) x_k + p_k x_k\} \\ -\sum_{k=1}^{t-1} \{p_k \varphi(\theta, x_k) x_k + p_k^2 x_k\} \end{bmatrix} \\
&= \sum_{k'=1}^{t} \sum_{k=1}^{t-1} \{\varphi(\theta, x_{k'})\varphi(\theta, x_k) v_{k'}^{\mathsf{T}} x_k - 2 p_k \varphi(\theta, x_{k'}) v_{k'}^{\mathsf{T}} x_k + p_k^2 v_{k'}^{\mathsf{T}} x_k\} \\
&\overset{(b)}{=} \sum_{k=1}^{t-1} \{p_k - \varphi(\theta, x_k)\}^2,
\end{aligned}$$

where (a) and (b) follow because, by construction, $v_{k'}^{\mathsf{T}} x_k = 0$ unless $k = k'$, in which case $v_k^{\mathsf{T}} x_k = 1$. Thus,

$$\begin{aligned}
C_t(\theta, \boldsymbol{x}_t)\, \mathbb{E}_\theta^\pi [\mathcal{J}_{t-1}(\boldsymbol{x}_t)]\, C_t(\theta, \boldsymbol{x}_t)^{\mathsf{T}} &= \mathbb{E}_\theta^\pi [C_t(\theta, \boldsymbol{x}_t) \mathcal{J}_{t-1}(\boldsymbol{x}_t) C_t(\theta, \boldsymbol{x}_t)^{\mathsf{T}}] \\
&= \sum_{k=1}^{t-1} \mathbb{E}_\theta^\pi \left[ \{p_k - \varphi(\theta, x_k)\}^2 \right].
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
\Delta^\pi(T) &= \sup_{\theta \in \Theta} \left\{ \sum_{t=1}^{T} \mathbb{E}_X \mathbb{E}_\theta^\pi \left[ -(\beta^{\mathsf{T}} X_t)(p_t - \varphi(\theta, X_t))^2 \right] \right\} \\
&\overset{(c)}{\geq} |\beta_{\min}| \sup_{\theta \in \Theta} \left\{ \sum_{t=1}^{T} \mathbb{E}_X \left\{ ||X_{S,t}||_1 \mathbb{E}_\theta^\pi \left[ (p_t - \varphi(\theta, X_t))^2 \right] \right\} \right\} \\
&\geq |\beta_{\min}| \sup_{\theta \in \Theta} \left\{ \sum_{t=1}^{T} \mathbb{E}_X \left\{ X_{\min} \mathbb{E}_\theta^\pi \left[ (p_t - \varphi(\theta, X_t))^2 \right] \right\} \right\},
\end{aligned}$$

where $\beta_{\min} = \min_{(\alpha,\beta) \in \Theta} \{\|\beta\|\}$, $||X_{S,t}||_1 = \sum_{i=1}^{s} |X_{S,t}^i|$ is the $\ell_1$-norm of the compressed feature vector $X_{S,t}$; $X_{S,t}^i$ is the $i$-th component of $X_{S,t}$ for $i = 1, \ldots, s$, $X_{\min} := \min\{||X_{S,1}||_1, \ldots, ||X_{S,T}||_1\}$, and (c) follows because

$$\tilde{\beta} = |\beta_{\min}| [\operatorname{sgn} X_{S,t}^1, \ldots, \operatorname{sgn} X_{S,t}^s]$$

is a feasible solution to the supremum problem in the first line. Now, since no component of $X_{S,t}$ is almost surely zero, there is a positive constant

$$c_{\min} = \min_{i \in \{1, \ldots, s\}} \{\mathbb{E}|X_{S,t}^i|\}.$$

Then, $X_{\min} \geq c_{\min} s$, and we get

$$\Delta^{\pi}(T) \geq |\beta_{\min}|\, c_{\min}\, s \sup_{\theta \in \Theta} \left\{ \sum_{t=1}^{T} \mathbb{E}_X \mathbb{E}_{\theta}^{\pi} \left[ (p_t - \varphi(\theta, X_t))^2 \right] \right\}.$$

Combining the above with Lemma 2, we can lower bound the worst-case regret by

$$\Delta^{\pi}(T) \geq \beta_{\min}^2 c_{\min}^2 s^2 \sum_{t=2}^{T} \frac{c_0}{c_1 |\beta_{\min}| c_{\min} s + \Delta^{\pi}(t-1)}.$$

Letting $K_1 = c_0 \beta_{\min}^2$ and $K_2 = c_1 |\beta_{\min}|$, we further obtain the following:

$$\Delta^{\pi}(T) \overset{\text{(d)}}{\geq} \frac{K_1 c_{\min}^2 s^2 (T-1)}{K_2 c_{\min} s + \Delta^{\pi}(T)} \overset{\text{(e)}}{\geq} \frac{s K_1 c_{\min}^2 s^2 T}{2\Delta^{\pi}(T)(K_2 c_{\min} s / \Delta^{\pi}(T) + 1)},$$

where (d) follows because $\Delta^{\pi}(T) \geq \Delta^{\pi}(t-1)$ for $t \in \{1, \ldots, T\}$, and (e) follows because $T \geq 2$. Now,

$$\Delta^{\pi}(T) \geq \Delta^{\pi}(1) \geq |\beta_{\min}| c_{\min} s (u - \ell)^2 / 4.$$

Thus, letting $K_3 = \frac{K_2}{|\beta_{\min}|(u-\ell)^2/4} + 1$, we get

$$\Delta^{\pi}(T) \geq \left(\frac{K_1}{2K_3}\right)^{1/2} c_{\min} s \sqrt{T}.$$

*Case 2:* $d + 1 < T$. In this case, the $t$ systems of linear equations (7.23) may become inconsistent by the Rouché–Capelli theorem, because the right-hand side of (7.23) spans the entire $\mathbb{R}^t$ space, but the rank of $\underline{\mathbf{X}}_t^{\mathsf{T}} \underline{\mathbf{X}}_t$ may be less than $t$. To avoid such inconsistencies, we consider instead augmented feature vectors, $\tilde{x}_k \in \mathbb{R}^T$, where the first $d + 1$ elements of $\tilde{x}_k$ equal $x_k$ and the rest are determined by the requirement $\tilde{\mathbf{X}}_t = [\tilde{x}_1, \ldots \tilde{x}_t]$ be of rank $t$. With this augmentation, the proof of Theorem 3 in this case follows by the same arguments for the preceding case. This concludes the proof when the components of $X_t$ are continuous random variables.

Finally, if some components of $X_t$ are discrete random variables, we can take the conditional expectation over all possible realizations of the discrete components first and then apply (7.20) for each realization. To illustrate, let $\mathcal{D}$ denote the set of all realizations of the discrete components of $X_t$; e.g., if $X_t \in \mathbb{R}^3$, with $X_t^1 = 1$ almost surely, $X_t^2 = \pm 1/2$ with probability $1/2$ (half male, half female), and $X_t^3$ a continuous random variable, then $\mathcal{D} = \{[1, 1/2], [1, -1/2]\}$. For $d \in \mathcal{D}$, let $X_t^C(d)$ denote the conditioned random variable where the discrete components of $X_t$ are set to the values in $d$. Then, we have

$$\mathbb{E}_{\mu,X}\left\{\mathbb{E}_{\theta}^{\pi}\left[(p_t-\varphi(\theta,X_t))^2\right]\right\}$$
$$=\sum_{d\in\mathcal{D}}\mathbb{P}_X\left\{X_t=X_t^C(d)\right\}\mathbb{E}_{\mu,X^C}\left\{\mathbb{E}_{\theta}^{\pi}\left[(p_t-\varphi(\theta,X_t))^2\mid X_t=X_t^C(d)\right]\right\},$$

where $\mathbb{E}_{\mu,X^C}$ denotes taking expectation over $\mu$ and the reduced feature vector that only contains the continuous components. Applying the multivariate van Trees inequality on $\mathbb{E}_{\mu,X^C}\left\{\mathbb{E}_{\theta}^{\pi}\left[(p_t-\varphi(\theta,X_t))^2\mid X_t=X_t^C(d)\right]\right\}$ for each $d\in\mathcal{D}$, we arrive at the same conclusion as before by following the same proof arguments for the conditional regret

$$\Delta^{\pi,C}(T):=\sup_{\theta\in\Theta}\left\{\sum_{t=1}^{T}\mathbb{E}_{X^C,\theta}^{\pi}\left[-(\beta^{\intercal}X_t)(p_t-\varphi(\theta,X_t))^2\mid X_t=X_t^C(d)\right]\right\}$$

for each $d\in\mathcal{D}$. □

### 7.4.4 Performance of ILQX

Ban and Keskin (2021) prove that the ILQX algorithm described in Sect. 7.4.2, which balances price experimentation with price optimization, is rate-optimal by the following result.

**Theorem 4 (Theorem 3, Ban and Keskin, 2021)** *Let $\pi=$ ILQX$(m_1,m_2,\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\ldots)$ with $\lambda_{t+1}=\tilde{c}t^{1/4}\sqrt{\log d+\log t}$ for all $t$, and $\tilde{c}$ is a positive constant independent of $s$, $d$, $T$. Then, there exists a finite and positive constant $\tilde{C}$ such that*

$$\Delta_{\theta}^{\pi}(T)\leq\tilde{C}s\sqrt{T}(\log d+\log T)\ \text{ for all }\theta\in\Theta\text{ and }T\geq 2.$$

Theorem 4 shows that the lasso-based ILQX policy achieves the lowest possible growth rate of regret presented in Theorem 3 (up to logarithmic terms) and is therefore *first-order optimal*. In addition, Theorem 4 makes the effect of the dimensions ($s$ and $d$) explicit; the sparsity dimension has a linear scaling effect on the regret upper bound, whereas the input dimension $d$ has a logarithmic effect on it. Theorem 4 also dictates how the regularization parameter should be chosen over time, up to a constant factor $\tilde{c}$. In practice, one would need to experiment with different values of $\tilde{c}$ through cross-validation, as it affects the finite-sample performance of the algorithm.

The proof of Theorem 4 relies on the following lemma, which characterizes the convergence rate for the squared norm of the estimation error under ILQX$(m_1,m_2,\boldsymbol{\lambda})$.

**Lemma 3 (Lemma 3, Ban and Keskin, 2021)** *Let $\pi=$ ILQX$(m_1,m_2,\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\ldots)$ with $\lambda_{t+1}=\tilde{c}t^{1/4}\sqrt{\log d+\log t}$ for all $t$, and $\tilde{c}$ is a positive*

*constant. Then, there exist finite and positive constants* $\kappa_3$, $\rho_3$, *and* $t_1$ *such that*

$$\mathbb{P}^{\pi}_{X,\theta}\left\{\|\hat{\theta}^{(\text{lasso})}_{t+1}(\lambda_{t+1}) - \theta\|^2 \le \frac{\rho_3\, s(\log d + \log t)}{\sqrt{t}}\right\} \ge 1 - \frac{\kappa_3\, s(\log d + \log t)}{\sqrt{t}} \tag{7.24}$$

*for all* $\theta \in \Theta$ *and* $t \ge t_1$, *where* $\|\cdot\|$ *denotes the Euclidean norm.*

The proof of Lemma 3 is very long, so we refer the interested reader to Appendix EC.3 of Ban and Keskin (2021) for full details.

***Proof (of Theorem 4)*** Fix $\pi = \mathsf{ILQX}(m_1, m_2, \boldsymbol{\lambda})$. For $p \in [\ell, u]$, $\theta \in \Theta$, and $x \in \mathcal{X}$, consider the Taylor series expansion of $r(p, \theta, x)$ around the revenue-maximizing price, $\varphi(\theta, x)$, noting that there exists a price $\tilde{p}$ between $p$ and $\varphi(\theta, x)$ such that

$$\begin{aligned} r(p, \theta, x) = r\big(\varphi(\theta, x), \theta, x\big) + \tfrac{\partial}{\partial p} r\big(\varphi(\theta, x), \theta, x\big)\big(p - \varphi(\theta, x)\big) \\ + \tfrac{1}{2}\tfrac{\partial^2}{\partial p^2} r\big(\tilde{p}, \theta, x\big)\big(p - \varphi(\theta, x)\big)^2. \end{aligned} \tag{7.25}$$

Because $\frac{\partial}{\partial p} r\big(\varphi(\theta, x), \theta, x\big) = 0$ for all $\theta \in \Theta$ and $x \in \mathcal{X}$, (7.25) implies that

$$r^*(\theta, x) - r(p, \theta, x) = r\big(\varphi(\theta, x), \theta, x\big) - r(p, \theta, x) \le C_3\big(\varphi(\theta, x) - p\big)^2 \tag{7.26}$$

for all $\theta \in \Theta$ and $x \in \mathcal{X}$, where $C_3 = \max\{\frac{1}{2}\frac{\partial^2}{\partial p^2} r\big(p, \theta, x\big) : p \in [\ell, u], \theta \in \Theta, x \in \mathcal{X}\}$. We deduce from (7.26) that, given $\theta \in \Theta$,

$$\begin{aligned} \Delta^{\pi}_{\theta}(T) &= \mathbb{E}_X\left\{\mathbb{E}^{\pi}_{\theta}\left\{\sum_{t=1}^{T}\big[r^*(\theta, X_t) - r(p_t, \theta, X_t)\big] \,\middle|\, \boldsymbol{X}_T\right\}\right\} \\ &\le \sum_{t=1}^{T} \mathbb{E}^{\pi}_{X,\theta}\big\{C_3\big[\varphi(\theta, X_t) - p_t\big]^2\big\} \end{aligned} \tag{7.27}$$

for $T \ge 2$, where $\mathbb{E}^{\pi}_{X,\theta}\{\cdot\} = \mathbb{E}_X\{\mathbb{E}^{\pi}_{\theta}\{\cdot \mid \boldsymbol{X}_T\}\}$.

Now,

$$\begin{aligned} \sum_{t=1}^{T} \mathbb{E}^{\pi}_{X,\theta}\big\{C_3\big[\varphi(\theta, X_t) - p_t\big]^2\big\} &= \sum_{t=1}^{T} \mathbb{E}^{\pi}_{X,\theta}\big\{C_3\big[\varphi(\theta, X_t) - p_t\big]^2 \mathbb{I}\{t \in M\}\big\} \\ &\quad + \sum_{t=1}^{T} \mathbb{E}^{\pi}_{X,\theta}\big\{C_3\big[\varphi(\theta, X_t) - p_t\big]^2 \mathbb{I}\{t \notin M\}\big\} \end{aligned} \tag{7.28}$$

for $T \geq 2$. With regard to the first term on the right-hand side of (7.28), note that $\sum_{t=1}^{T} \chi_t = \sum_{t=1}^{T} \mathbb{I}\{t \in M\} \leq 2\sqrt{T}$ under $\pi = \mathsf{ILQX}(m_1, m_2, \boldsymbol{\lambda})$. Thus,

$$\sum_{t=1}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{C_3\left[\varphi(\theta, X_t) - p_t\right]^2 \mathbb{I}\{t \in M\}\right\} \leq C_4\sqrt{T} \tag{7.29}$$

for $T \geq 2$, where $C_4 = 2C_3(u - \ell)^2$. With regard to the second term on the right-hand side of (7.28),

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{C_3\left[\varphi(\theta, X_t) - p_t\right]^2 \mathbb{I}\{t \notin M\}\right\} \\
&\qquad \overset{(a)}{=} \sum_{t=2}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{C_3\left[\varphi(\theta, X_t) - \varphi\big(\hat{\vartheta}_t^{(\mathrm{lasso})}(\lambda_t), X_t\big)\right]^2 \mathbb{I}\{t \notin M\}\right\} \\
&\qquad \leq C_0 \sum_{t=2}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{\left[\varphi(\theta, X_t) - \varphi\big(\hat{\vartheta}_t^{(\mathrm{lasso})}(\lambda_t), X_t\big)\right]^2 \mathbb{I}\{t \notin M\}\right\} \\
&\qquad \overset{(b)}{\leq} C_0 K_0 \sum_{t=2}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{\|\theta - \hat{\vartheta}_t^{(\mathrm{lasso})}(\lambda_t)\|^2 \mathbb{I}\{t \notin M\}\right\} \\
&\qquad \overset{(c)}{\leq} C_0 K_0 \sum_{t=2}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{\|\theta - \hat{\vartheta}_t^{(\mathrm{lasso})}(\lambda_t)\|^2\right\}
\end{aligned} \tag{7.30}$$

for $T \geq 2$, where $K_0 = \max\{\|\nabla_\theta \varphi(\theta, x)\|^2 : \theta \in \Theta, x \in \mathcal{X}\}$; (a) follows because $1 \in M$ and $p_t = \varphi\big(\hat{\vartheta}_t^{(\mathrm{lasso})}, X_t\big)$ for $t \notin M$, under $\pi = \mathsf{ILQX}(m_1, m_2, \boldsymbol{\lambda})$; (b) follows by the mean value theorem; and (c) follows because $\mathbb{I}\{t \notin M\} \leq 1$. Furthermore,

$$\begin{aligned}
&C_0 K_0 \sum_{t=2}^{T} \mathbb{E}_{X,\theta}^{\pi}\left\{\|\theta - \hat{\vartheta}_t^{(\mathrm{lasso})}(\lambda_t)\|^2\right\} \\
&\qquad \leq C_5\Big(t_1 d_\Theta^2 + \sum_{t=t_1}^{T-1} \mathbb{P}_{X,\theta}^{\pi}\{\mathcal{A}_t^c\}\, d_\Theta^2 + \sum_{t=t_1}^{T-1} \mathbb{E}_{X,\theta}^{\pi}\left\{\|\theta - \hat{\vartheta}_{t+1}^{(\mathrm{lasso})}(\lambda_{t+1})\|^2 \mathbb{I}\{\mathcal{A}_t\}\right\}\Big)
\end{aligned} \tag{7.31}$$

for $T \geq 2$, where

$$\mathcal{A}_t = \left\{\|\theta - \hat{\theta}_{t+1}^{(\mathrm{lasso})}(\lambda_{t+1})\|^2 \leq \frac{\rho_3 s(\log d + \log t)}{\sqrt{t}}\right\},$$

$C_5 = C_3 \max\{\|\nabla_\theta \varphi(\theta, x)\|^2 : \theta \in \Theta, x \in \mathcal{X}\}$, $d_\Theta = \max\{\|\vartheta - \tilde{\vartheta}\| : \vartheta, \tilde{\vartheta} \in \Theta\}$, and $\mathbb{P}^\pi_{X,\theta}\{\cdot\}$ is the probability measure associated with $\mathbb{E}^\pi_{X,\theta}\{\cdot\}$. Lemma 3 implies that $\mathbb{P}^\pi_{X,\theta}\{\mathcal{A}_t^c\} \leq \frac{\kappa_3 s(\log d + \log t)}{\sqrt{t}}$ for $t \geq t_1$, from which we deduce

$$\sum_{t=t_1}^{T-1} \mathbb{P}^\pi_{X,\theta}\{\mathcal{A}_t^c\} \leq 2\kappa_3 s\sqrt{T}(\log d + \log T).$$

We thus arrive at

$$\sum_{t=1}^{T} \mathbb{E}^\pi_{X,\theta}\left\{C_3\left[\varphi(\theta, X_t) - p_t\right]^2 \mathbb{I}\{t \notin M\}\right\} \leq C_6 s\sqrt{T}(\log d + \log T), \tag{7.32}$$

for $T \geq 2$, where $C_6 = C_5(t_1 d_\Theta^2 + 2\kappa_3 d_\Theta^2 + 4\rho_3)$.

Putting everything together, we have the desired result

$$\Delta^\pi_\theta(T) \leq \tilde{C} s\sqrt{T}(\log d + \log T)$$

for $T \geq 2$, where $\tilde{C} = C_4 + C_6$. □

### 7.4.5 Discussion

As there are several competing works on dynamic pricing with high-dimensional data in the recent literature, let us clarify the differences between them. Qing and Bayati (2016) assume the demand follows a linear function of the prices and features and applies a myopic policy based on least-square estimations which achieves a regret of $O(\log(T))$. Javanmard and Nazerzadeh (2019) consider dynamic pricing with (product) features under a binary choice model and construct a regularized maximum likelihood policy which achieves a regret of $O(s \log(d) \log(T))$. Cohen et al. (2020) study dynamic pricing of differentiated products on a homogeneous customer base, where the market value of each product is a linear function of the feature vector. The authors assume that the feature vectors are selected adversarially and introduce an ellipsoid-based algorithm which obtains a regret of $O(d^2 \log(T/d))$. Ban and Keskin (2021) generalize the linear model of Qing and Bayati (2016) to include a feature-dependent price sensitivity term and to allow for nonlinear transformations of the underlying linear model.

Apart from the differences in the demand model, Qing and Bayati (2016), Javanmard and Nazerzadeh (2019), and Cohen et al. (2020) can achieve a logarithmic regret because the demand feedback is assumed to be a deterministic function of some unknown parameter. This contrasts with the square root regret of Ban and Keskin (2021), where the error term in the demand model is assumed to follow a

sub-Gaussian Martingale difference sequence, not a specific parametric distribution. See Kleinberg and Leighton (2003) for a discussion of this distinction and lower bounds in both settings.

## 7.5 Directions for Future Research

In this chapter, we reviewed recent theoretical developments in using high-dimensional customer and/or product information data in pricing. In particular, we focused on the static pricing study of Chen et al. (2022) and the dynamic pricing study of Ban and Keskin (2021).

Future research can extend the growing number of recent works in the following directions. First, consideration of objectives other than pure revenue optimization would be more appropriate for many businesses. Examples include consideration of long-term customer satisfaction, or simply, customer growth (perhaps even at the expense of revenue, which is the case for ambitious start-ups). For such objectives, the problem would be cast as offering personalized discounts, rather than prices, although mathematically speaking, charging personalized prices is equivalent to offering personalized discounts. Further elaboration on different objectives is to consider various business constraints; so far, all papers that analyze the performance of pricing with high-dimensional data only consider a simple interval constraint on the price. This can lead to interesting yet challenging problems, especially if the constraints also depend on high-dimensional data.

Second, there is much scope for investigating novel solution methods, comparing across different estimation paradigms (non-parametric, parametric, semi-parametric, and Bayesian) and gaining deeper understanding of their advantages and disadvantages. There is also room for better understanding the theoretical performance of some of these methods beyond pure empirical comparisons. Ultimately, the goal would be to generate a library of knowledge in aiding businesses with their price decisions in a variety of situations.

Finally, an important yet understudied area is in legal and ethical concerns of pricing with high-dimensional data (Gerlick and Liozu, 2020). The main concerns identified are data privacy and fairness issues. At the time of writing, Chen et al. (2020) and Lei et al. (2020) are two works known to us that construct and analyze privacy-preserving pricing algorithms; other works are sure to follow.

## References

Ban, G. Y., & Keskin, N. B. (2021). Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity. *Management Science, 67*(9), 5549–5568.

Ban, G. Y., & Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, *67*(1), 90–108.

Ban, G. Y., Gallien, J., & Mersereau, A. J. (2019). Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing & Service Operations Management*, *21*(4), 798–815.

Bastani, H., & Bayati, M. (2020). Online decision making with high-dimensional covariates. *Operations Research*, *68*(1), 276–294.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Chen, X., Simchi-Levi, D., & Wang, Y. (2020). Privacy-preserving dynamic personalized pricing with demand learning. Available at SSRN 3700474.

Chen, X., Owen, Z., Pixton, C. & Simchi-Levi, D. (2022). A statistical learning approach to personalization in revenue management. *Management Science, 68*(3), 1923–1937.

Cohen, M. C., Lobel, I., & Paes Leme, R. (2020). Feature-based dynamic pricing. *Management Science*, *66*(11), 4921–4943.

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, *18*(1), 69–88.

Gerlick, J. A., & Liozu, S. M. (2020). Ethical and legal considerations of artificial intelligence and algorithmic decision-making in personalized pricing. *Journal of Revenue and Pricing Management*, *19*, 85–98.

Gill, R. D., & Levit, B. Y. (1995). Applications of the van trees inequality: a Bayesian Cramér-Rao bound. *Bernoulli*, *1*(1/2), 59–79.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Javanmard, A., & Nazerzadeh, H. (2019). Dynamic pricing in high-dimensions. *The Journal of Machine Learning Research*, *20*(9), 1–49.

Kleinberg, R., & Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science* (pp. 594–605). IEEE.

Lei, Y. M., Miao, S., & Momot, R. (2020). *Privacy-preserving personalized revenue management*. Available at SSRN 3704446.

Mandl, C., & Minner, S. (2020) Data-Driven Optimization for Commodity Procurement Under Price Uncertainty. *Manufacturing & Service Operations Management*. https://pubsonline.informs.org/doi/abs/10.1287/msom.2020.0890.

Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, *135*(3), 370–384.

Qian, S., & Bayati, M. (2016). *Dynamic pricing with demand covariates*. Available at SSRN 2765257.

Qu, H., Ryzhov, I. O., Fu, M. C., Bergerson, E., Kurka, M., & Kopacek, L. (2020). Learning demand curves in b2b pricing: A new framework and case study. *Production and Operations Management*, *29*(5), 1287–1306.

Train, K. E. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge University Press.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint* (Vol. 48). Cambridge University Press.

# Part III
# Assortment Optimization

# Chapter 8
# Nonparametric Estimation of Choice Models


**Srikanth Jagabathula and Ashwin Venkataraman**

## 8.1 Introduction

Firms rely on demand predictions to make critical operational decisions. For example, firms need to know how customers respond to price changes in order to optimize the prices it charges. Traditionally, operational decision models relied on what is known as the "independent" demand model. As its name implies, an independent demand model assumes that the demand observed for a product is *independent* of the availability or characteristics, such as price, of other products. That is, the model ignores any cross-product cannibalization effects. Ignoring cross-product effects is hard to justify when products are close substitutes of each other; for example, products belonging to the same product category (e.g., different brands of toothpaste), different fare classes of an airline itinerary, different transportation modes (e.g., car, bus, train, etc.) are all close substitutes of each other. In such cases, ignoring the cross-product effects lead to biased demand estimates, especially when product prices and availability change over time. To deal with such cross-product effects, choice-based demand models have gained in popularity over the last couple of decades.

In the most general form, a choice model specifies the probability that a customer purchases a product from a given subset, or *offer set*, of products. If there are $N$ products, then the model specifies choice probabilities for each of the $2^N$ subsets. Because the model is intractable in such a general form, existing literature

S. Jagabathula
Stern School of Business, New York University, New York, NY, USA
e-mail: sjagabat@stern.nyu.edu

A. Venkataraman (✉)
Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA
e-mail: ashwin.venkataraman@utdallas.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_8

has studied various model sub-classes with varying degrees of tractability. The most studied sub-class, by far, is the random utility maximization (RUM) class of models (McFadden, 1981). These models specify a joint distribution over product utilities and assume that each customer samples a utility vector from the underlying joint distribution and purchases the available product with the maximum utility. A special case of the RUM class that has received the most attention in the literature is the multinomial logit (MNL) model (see, e.g., Ben-Akiva et al., 1985; Train, 2009). Other special cases include the nested logit model, the $d$-level nested logit or the tree logit model (Li et al., 2015), the generalized extreme value (GEV) model, the mixed logit model, etc. These special cases differ in the assumptions they impose on the structure of the joint utility distribution. We refer the reader to Train's book (Train, 2009) and the overview by Gallego and Topaloglu (2019) for a detailed introduction to these and related choice models studied in the operations literature.

In this book chapter, we discuss recent developments in the literature on estimating the RUM class of models from observed sales transaction data. Sales transaction data provide historical choice observations: the product chosen and the other products on offer when the choice was made. These data are regularly collected by firms through their point-of-sale (POS) and inventory systems. Our focus will be specifically on nonparametric estimation techniques, which differ from the traditional, and more prevalent, parametric model estimation techniques. In the context of choice models, parametric models restrict the joint utility distribution to belong to a parametrized class of distributions. This additional structure lends them tractability, and the model parameters are typically estimated using standard model fitting techniques, such as the maximum likelihood estimation (MLE) technique. While parametric restrictions lend tractability, they typically also result in *model misspecification*, which occurs when the imposed restrictions exclude the ground-truth data generating distribution. Model misspecification leads to biased parameter estimates and inaccurate demand predictions. To alleviate this issue, nonparametric techniques do not restrict the joint utility distribution and allow it to be described by any member of the RUM model class. They then use sophisticated mathematical techniques to search for the model that has the best fit to the observed data. Nonparametric techniques generally work best when the volume of data is "large," which has increasingly been the case in the recent past because of firms' ability to collect highly fine-grained data.

We focus our discussion on broadly two types of techniques. The first technique deals with the so-called rank-based choice model (Mahajan and Van Ryzin, 2001). In this model, each product is treated as a bundle of features (e.g., color, weight, size, price, etc.), which remain fixed and do not vary. The model is fit on transaction data in which only the offer sets vary, and the trained model is used to predict demand for a heretofore unseen offer set. The model can accommodate varying product features by treating each product variant (e.g., the same product but with different prices) as a separate product. This modeling approach is ideal when product feature representations are not readily available (e.g., when purchases are driven by hedonistic features, such as taste, feel, etc.; see, for instance, Hoyer and

Ridgway, 1984; Kahn and Lehmann, 1991) and firms want to predict demand for various offer sets of their *existing* products, for which sufficient observed data exist. For example, airlines have existing transactions on customer bookings, which contain information on the purchase fare class and the corresponding offered fare classes for a set of customers. The airline wants to use this data to predict the expected demand for each combination of offered fare classes in order to determine the optimal collection of fare classes to open. Similarly, retailers (both online and offline) want to optimize the offered assortments of existing products to customers. One limitation of the rank-based model is that it cannot extrapolate demand to *new* products or new variants of existing products.

The second technique we discuss addresses the inability of the rank-based choice model to extrapolate demand. It deals with what we call the *nonparametric mixture of closed logit model* (NPMXCL model), which was considered in Jagabathula et al. (2020b).[1] This model assumes that all products have consistent feature representations and specifies a flexible functional form relating product features to choice probabilities. When trained on existing transaction data with "sufficient" variation in features, the model can extrapolate demand to heretofore unseen products or product variants. This model subsumes the rank-based model as a special case and is ideally suited for estimating price elasticities, optimizing discount levels and promotion mix, and determining the cannibalization effects of introducing new products.

Both techniques above formulate the estimation problem as a large-scale constrained convex optimization problem and build on recent developments within the machine learning (ML) literature to propose efficient algorithms for model estimation. We also discuss some of the theoretical guarantees that can be established for these methods.

The rest of the chapter is organized as follows. We first present an overview of the setup, notation, and the data model. We then discuss the model assumptions and the details of the corresponding estimation techniques for the rank-based model and the NPMXCL model. We then briefly review other nonparametric choice models proposed in the literature and conclude with some thoughts on future directions in nonparametric choice modeling.

**Notation** We first summarize notation that is common to the rest of the chapter. For any positive integer $m$, we let $[m]$ denote the set $\{1, 2, \ldots, m\}$, $\mathbf{0}_m$ denote the all-zeros vector in $\mathbb{R}^m$, and $\Delta_m$ denote the unit $m$-simplex in $\mathbb{R}^{m+1}$. Vectors are denoted by lower-case bold letters such as $\boldsymbol{x}$, $\boldsymbol{g}$, etc. For any multivariate function $h(\cdot)$ on the Euclidean space, $\nabla h(\cdot)$ denotes the gradient of $h(\cdot)$, i.e., the vector of partial derivatives with respect to each of the input variables. We let $\|\boldsymbol{x}\|$ denote the $L^2$-norm of any vector $\boldsymbol{x}$ in the Euclidean space. When we write $\boldsymbol{x}_1 > \boldsymbol{x}_2$ for vectors $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, we mean that each element of $\boldsymbol{x}_1$ is greater than the corresponding element in $\boldsymbol{x}_2$. For any set $A$, $|A|$ denotes its cardinality. Finally, $\langle \cdot, \cdot \rangle$ denotes the standard inner product in the Euclidean space.

[1] However, they did not introduce this nomenclature.

## 8.2 General Setup

We consider the setting of a firm whose offerings belong to a universe $[N] = \{1, 2, \ldots, N\}$ of $N$ products. The firm collects choice data over a collection $\mathcal{M}$ of offer sets, where each offer set is a subset of the product universe $[N]$ offered to the customers. For each subset $S \in \mathcal{M}$, we let $y_{i,S} \in [0, 1]$ denote the observed fraction of customers who purchased product $i$ when $S$ was offered. Typically, the customer can leave without making a purchase, which is represented by a special product called the no-purchase or outside option. In our development below, the no-purchase option can be treated as any other product and consequently, we suppose that the product universe $[N]$ and the offer sets already include the no-purchase option. Note that we are implicitly assuming here that the firm can keep track of customers who visited with an intent to purchase but did not make a purchase. This can be done to a certain extent in the online e-commerce settings, but in the offline settings, the no-purchase observations are typically censored. We do not explicitly deal with this issue in this book chapter and suppose that the demand has been uncensored using other means.[2] We represent the observed data as the vector $\boldsymbol{y}_{\mathcal{M}} = (y_{i,S} : i \in S, S \in \mathcal{M})$. Let $M \stackrel{\text{def}}{=} \sum_{S \in \mathcal{M}} |S|$ denote the total number of choice observations, so that $\boldsymbol{y}_{\mathcal{M}} \in [0, 1]^M$.

As mentioned earlier, a choice model specifies the probability that a customer purchases a product from a given offer set. For the collection $\mathcal{M}$, we represent the collection of choice probabilities under a given model as the vector $\boldsymbol{g}_{\mathcal{M}} = (g_{i,S} : i \in S, S \in \mathcal{M})$ where $g_{i,S} \in [0, 1]$ is the probability of choosing product $i$ from offer set $S$ specified by the choice model. Estimating a model typically involves finding the model parameters that best fit the observed data, where the model misfit is measured using a *loss function*. More specifically, we measure the degree of model misfit using a non-negative loss function $\boldsymbol{g}_{\mathcal{M}} \mapsto \mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}})$ that measures the "distance" between the observed choice fractions $\boldsymbol{y}_{\mathcal{M}}$ and model predicted choice probabilities $\boldsymbol{g}_{\mathcal{M}}$. We consider loss functions $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \cdot)$ that are (strictly) convex in the second argument and have the property that $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = 0$ if and only if $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$. Letting $\mathcal{G}$ denote the set of all choice probability vectors (for the observed offer set collection) that can be generated by the choice model family of interest, we solve the following estimation problem:

$$\min_{\boldsymbol{g}_{\mathcal{M}} \in \mathcal{G}} \mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}). \qquad \text{(GENERAL ESTIMATION PROBLEM)}$$

[2] There are numerous papers that explicitly account for the demand censoring issue while estimating the choice model; see, for instance, Haensel and Koole (2011), Newman et al. (2014), and Abdallah and Vulcano (2020).

The following are two commonly used loss functions:

*Example 1 (Log-Likelihood/Kullback–Leibler (KL) Divergence Loss Function)* This loss function is defined as follows:

$$\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = -\sum_{S\in\mathcal{M}} M_S \sum_{i\in S} y_{i,S} \log\left(g_{i,S}/y_{i,S}\right),$$

where the weight $M_S > 0$ associated with offer set $S \in \mathcal{M}$ is equal to the number of customers who were offered the assortment $S$. Note that if $y_{i,S} = 0$ for some $(i, S)$ pair, then the corresponding term is dropped from the loss objective. It can be verified that this loss function is non-negative since it is a weighted sum (with non-negative weights) of individual KL-divergence terms $-\sum_{i\in S} y_{i,S} \log(g_{i,S}/y_{i,S})$ between the distributions $\left(y_{i,S} : i \in S\right)$ and $\left(g_{i,S} : i \in S\right)$ for each $S \in \mathcal{M}$, which are always non-negative. For the same reason, we also have that $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = 0$ if and only if $y_{i,S} = g_{i,S}$ for all $i \in S$ and $S \in \mathcal{M}$. The loss function is strictly convex in the second argument, provided that the observed fractions for all choice observations are strictly positive, i.e., $\boldsymbol{y}_{\mathcal{M}} > \boldsymbol{0}_M$. This follows from the strict concavity of the logarithm function. Because the terms involving the observed choice fractions $(y_{i,S} : i \in S, S \in \mathcal{M})$ are constants for the optimization problem, it can be shown that minimizing the KL-divergence loss function is equivalent to maximizing the log-likelihood. Therefore, employing this loss function in the GENERAL ESTIMATION PROBLEM results in the maximum likelihood estimate (MLE).

*Example 2 (Squared Norm Loss Function)* This loss function is defined as

$$\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = \left\|\boldsymbol{y}_{\mathcal{M}} - \boldsymbol{g}_{\mathcal{M}}\right\|^2.$$

It is easy to see that the squared norm loss function is non-negative and takes a value of 0 if and only if $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$. Further, it is strictly convex in $\boldsymbol{g}_{\mathcal{M}}$ for any fixed $\boldsymbol{y}_{\mathcal{M}}$.

Having introduced the general setup for the estimation problem, we now discuss in more detail two choice model families, the rank-based model and the NPMXCL model.

## 8.3 Estimating the Rank-Based Model

The rank-based choice model is the most general representation of the RUM class of models. Recall that a RUM model assumes that customers sample a utility for each of the products and choose the available product with the highest utility value. For finitely many products, it is clear that as far as the customer's choice is concerned, the actual utility values do not matter—only the preference ordering induced by the sampled utilities matter; see, for instance, Block and Marschak

(1960) and Mas-Colell et al. (1995). The rank-based choice model recognizes this and models the preferences of each customer as a ranking or preference ordering of the products. The preferences of a population of customers are, as a result, modeled as a probability distribution over rankings.

The rank-based choice model has origins in the classical preference and utility theory in economics and psychology (Block and Marschak, 1960; Manski, 1977; Falmagne, 1978; Barberá and Pattanaik, 1986). Most of the work in this area has focused on establishing theoretical properties of the model. For instance, Falmagne (1978) shows that a system of choice probabilities defined over all possible offer sets is consistent with a rank-based model if and only if all the Block–Marschak polynomials are non-negative; see also Barberá and Pattanaik (1986). McFadden (2005) provides additional necessary and sufficient conditions in the form of systems of linear inequalities, and shows how the different conditions relate to one another. For much of its history, the rank-based choice model has mostly served as a theoretical construct because estimating it from choice data is a significant computational challenge. Therefore, the literature on choice modeling has largely focused on specific parametric models, which impose additional structure on the utility distributions to trade off the restrictiveness of the models with the computational tractability of estimating them. Farias et al. (2013) was one of the first papers within the operations literature to tackle the computational challenge of estimating the rank-based model from choice data. They used ideas in linear programming to propose tractable techniques to predict revenues for new offer sets. Subsequent work (van Ryzin and Vulcano, 2015, 2017; Jagabathula and Rusmevichientong, 2017) further built on this paper to make the rank-based model operationally tractable, some of which has focused on estimating the model and solving the subsequent operational decision, such as the assortment or the pricing decision.

Before discussing the estimation of the rank-based model, we formally define the model. Let $\mathscr{P}$ denote the set of all permutations (or linear preference orders) of the $N$ products, so that $|\mathscr{P}| = N!$ ($N$ factorial). Each element $\sigma \in \mathscr{P}$ is a ranking of the $N$ products, and for all $i \in [N]$, we let $\sigma(i)$ denote the *rank* of product $i$. We assume that if $\sigma(i) < \sigma(j)$, then product $i$ is preferred over product $j$ in the ranking $\sigma$. Given any offer set $S$, a customer chooses the product that is most preferred under her ranking $\sigma$. Let $\mathbb{1}[\sigma, i, S]$ denote the indicator variable that takes a value of 1 if and only if product $i$ is the most preferred product in $S$ under $\sigma$; that is, $\mathbb{1}[\sigma, i, S] = 1$ if and only if $\sigma(i) < \sigma(j)$ for all $j \in S$, $j \neq i$. The choice behavior of the customer population is then modeled as a probability distribution $\lambda : \mathscr{P} \to [0, 1]$ over the permutations with $\lambda(\sigma)$ denoting the probability that a customer uses the ranking $\sigma$ when making a purchase. Because $\lambda$ is a probability distribution, we have that $\lambda(\sigma) \geq 0$ for all $\sigma \in \mathscr{P}$ and $\sum_{\sigma \in \mathscr{P}} \lambda(\sigma) = 1$.

Given any distribution over rankings $\lambda$, the vector of choice probabilities for the offer set collection $\mathcal{M}$ under the rank-based model is given by:

$$\boldsymbol{g}_{\mathcal{M}}(\lambda) = (g_{i,S}(\lambda) \colon i \in S,\ S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(\lambda) = \sum_{\sigma \in \mathscr{P}} \mathbb{1}[\sigma, i, S] \cdot \lambda(\sigma). \tag{8.1}$$

The set of all such probability vectors consistent with a rank-based model is denoted by $\mathcal{G}(\mathscr{P})$:

$$\mathcal{G}(\mathscr{P}) = \left\{ \boldsymbol{g}_M(\lambda) \mid \lambda : \mathscr{P} \to [0,1], \sum_{\sigma \in \mathscr{P}} \lambda(\sigma) = 1 \right\}. \tag{8.2}$$

The estimation problem for the rank-based model can then be formulated by plugging in $\mathcal{G} = \mathcal{G}(\mathscr{P})$ in the GENERAL ESTIMATION PROBLEM. However, solving the problem in this form poses some difficulties. This is because the loss function depends on the distribution $\lambda$ only through the predicted choice probability vector $\boldsymbol{g}_M(\lambda)$, and, therefore, the underlying distribution is not directly identifiable in general. In fact, Sher et al. (2011) showed that if $N \geq 4$, there are multiple distributions over rankings that are consistent with *any* given collection of choice probabilities. The idea is that the choice probabilities impose $O(2^N)$ degrees of freedom (corresponding to all the subsets of $[N]$) whereas the space of distributions has $O(N!) = O(2^{N \log N})$ degrees of freedom.

To see this fact more explicitly, we consider an alternate representation of $\mathcal{G}(\mathscr{P})$. For each $\sigma \in \mathscr{P}$, let $\boldsymbol{f}(\sigma) \in \{0,1\}^M$ be the vector of indicators that determine whether product $i$ is chosen from offer set $S$ under ranking $\sigma$:

$$\boldsymbol{f}(\sigma) = (\mathbb{1}[\sigma, i, S] \colon \; S \in \mathcal{M}, i \in S), \tag{8.3}$$

and let $\mathcal{F}(\mathscr{P}) \stackrel{\text{def}}{=} \{\boldsymbol{f}(\sigma) \colon \sigma \in \mathscr{P}\}$ denote the set of all such indicator vectors. Now, consider the convex hull of the set $\mathcal{F}(\mathscr{P})$, which we denote as $\text{conv}(\mathcal{F}(\mathscr{P}))$, defined as:

$$\text{conv}\left(\mathcal{F}(\mathscr{P})\right) = \left\{ \sum_{\boldsymbol{f} \in \mathcal{F}(\mathscr{P})} \alpha_{\boldsymbol{f}} \boldsymbol{f} \colon \alpha_{\boldsymbol{f}} \geq 0 \; \forall \boldsymbol{f} \in \mathcal{F}(\mathscr{P}), \sum_{\boldsymbol{f} \in \mathcal{F}(\mathscr{P})} \alpha_{\boldsymbol{f}} = 1 \right\}.$$

Then, using the above equations it can be verified that $\mathcal{G}(\mathscr{P}) = \text{conv}(\mathcal{F}(\mathscr{P}))$. This shows that $\mathcal{G}(\mathscr{P})$ is a convex polytope in $\mathbb{R}^M$. While $\mathcal{G}(\mathscr{P})$ as defined in (8.2) appears to have a dependence on $N!$ variables, in practice the number of extreme points of $\mathcal{G}(\mathscr{P}) = \text{conv}(\mathcal{F}(\mathscr{P}))$ can be (significantly) smaller than $N!$ ($N$ factorial). This is because two different rankings $\sigma \neq \sigma'$ may result in the same vector of indicators $\boldsymbol{f}(\sigma) = \boldsymbol{f}(\sigma')$ as in the following example:

*Example 3 (Complexity of $\mathcal{G}(\mathscr{P})$ Under Market Shares Data)* Suppose that the firm collects only market shares data, so that the offer set collection $\mathcal{M} = \{[N]\}$. In this case $M = N$ and it follows that each $\boldsymbol{f}(\sigma) \in \{0,1\}^N$ with $\boldsymbol{f}(\sigma_1) = \boldsymbol{f}(\sigma_2)$ for any two rankings $\sigma_1, \sigma_2$ in which the top-ranked product is the same. Consequently, $|\mathcal{F}(\mathscr{P})| = N \ll N!$ ($N$ factorial). Moreover, it can be verified that the number of extreme points of $\text{conv}(\mathcal{F}(\mathscr{P}))$ is, in fact, $N$.

More generally, the number of extreme points of $\text{conv}(\mathcal{F}(\mathscr{P}))$, which is at most $|\mathcal{F}(\mathscr{P})|$, depends on the variation amongst offer sets in $\mathcal{M}$. Therefore, $\text{conv}(\mathcal{F}(\mathscr{P}))$ is a more succinct representation of $\mathcal{G}(\mathscr{P})$.

With the above development, the GENERAL ESTIMATION PROBLEM for the rank-based model takes the form:

$$\min_{\boldsymbol{g} \in \text{conv}(\mathcal{F}(\mathscr{P}))} \mathsf{loss}(\boldsymbol{g}), \quad \text{(RANK-BASED MODEL ESTIMATION PROBLEM)}$$

where we drop the explicit dependence of the set collection $\mathcal{M}$ on the predicted choice probabilities, and the observed choice fractions $\boldsymbol{y}_{\mathcal{M}}$ on the loss function for notational convenience. Since the constraint set is a convex polytope and the objective function is convex, the RANK-BASED MODEL ESTIMATION PROBLEM is a constrained convex program. In theory, it can be solved using standard methods for convex optimization. The challenge, however, is two-fold: (a) the constraint polytope may not have an efficient description and (b) decomposing a candidate solution $\boldsymbol{g}$ into the corresponding proportions $\boldsymbol{\alpha}$ (and, therefore, the underlying distribution $\lambda$) is itself a hard problem. Note that the distribution is required so that out-of-sample choice predictions can be made. To address these issues, Jagabathula and Rusmevichientong (2019) (henceforth JR) used the conditional gradient algorithm, which, as we will see shortly, transforms the convex optimization problem into solving a series of linear optimization problems. But first, we show that the RANK-BASED MODEL ESTIMATION PROBLEM has a unique optimal solution:

**Theorem 1 (Unique Optimal Solution)** *For any strictly convex loss function* $\mathsf{loss}(\cdot)$ *over the domain* $\text{conv}(\mathcal{F}(\mathscr{P}))$*, the* RANK-BASED MODEL ESTIMATION PROBLEM *has a unique optimal solution.*

***Proof*** We prove this result by contradiction. Suppose, if possible, there exist two optimal solutions $\boldsymbol{g}_1^* \neq \boldsymbol{g}_2^*$ and let $\mathsf{loss}^* = \mathsf{loss}(\boldsymbol{g}_1^*) = \mathsf{loss}(\boldsymbol{g}_2^*)$. By strict convexity of $\mathsf{loss}(\cdot)$, it follows that for any $\delta \in (0, 1)$:

$$\begin{aligned} \mathsf{loss}(\delta \boldsymbol{g}_1^* + (1-\delta)\boldsymbol{g}_2^*) &< \delta \cdot \mathsf{loss}(\boldsymbol{g}_1^*) + (1-\delta) \cdot \mathsf{loss}(\boldsymbol{g}_2^*) \\ &= \delta \cdot \mathsf{loss}^* + (1-\delta) \cdot \mathsf{loss}^* = \mathsf{loss}^*. \end{aligned}$$

Since, by definition, $\text{conv}(\mathcal{F}(\mathscr{P}))$ is convex, it follows that $\delta \boldsymbol{g}_1^* + (1-\delta)\boldsymbol{g}_2^* \in \text{conv}(\mathcal{F}(\mathscr{P}))$ is a feasible solution to the RANK-BASED MODEL ESTIMATION PROBLEM. But this contradicts the assumption that $\mathsf{loss}^*$ is the optimal objective and, therefore, the optimal solution must be unique. □

### 8.3.1 Estimation via the Conditional Gradient Algorithm

As mentioned above, JR proposed to solve the RANK-BASED MODEL ESTIMATION PROBLEM using the conditional gradient algorithm. We begin with some background on the algorithm and then discuss its application for estimating the rank-based choice model.

**Background** The conditional gradient (hereafter CG) algorithm (aka Frank–Wolfe) algorithm (Clarkson, 2010; Jaggi, 2013) is an iterative method for solving optimization problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{D}} h(\boldsymbol{x}), \tag{8.4}$$

where $h(\cdot)$ is a differentiable convex function and $\mathcal{D}$ is a compact convex region in the Euclidean space. It is in fact a generalization of the original algorithm proposed by Frank and Wolfe (1956), who considered solving quadratic programming problems with linear constraints. Starting from an arbitrary feasible point $\boldsymbol{x}^{(0)} \in \mathcal{D}$, in each iteration $k \geq 1$, the algorithm finds a *descent direction* $\boldsymbol{d}^{(k)}$ such that $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{d}^{(k)} \right\rangle < 0$ and takes a suitable step in that direction. The algorithm computes such a descent direction by optimizing the linear approximation of $h(\cdot)$ at the current iterate $\boldsymbol{x}^{(k-1)}$ over the feasible domain $\mathcal{D}$. That is, it solves the following problem:

$$\boldsymbol{v}^{(k)} \in \arg\min_{\boldsymbol{v}\in\mathcal{D}} \; h(\boldsymbol{x}^{(k-1)}) + \left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v} - \boldsymbol{x}^{(k-1)} \right\rangle. \quad \text{(FRANK–WOLFE STEP)}$$

Because the objective function in the FRANK–WOLFE STEP is linear in $\boldsymbol{v}$, the optimal solution $\boldsymbol{v}^{(k)}$ is an extreme point of $\mathcal{D}$. Having found the extreme point $\boldsymbol{v}^{(k)}$, the algorithm updates the solution by taking a step along the direction $\boldsymbol{d}^{(k)} \stackrel{\text{def}}{=} \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)}$obtaining $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \gamma^{(k)} \cdot \boldsymbol{d}^{(k)}$ for some step size $\gamma^{(k)} \in [0, 1]$. Since $\boldsymbol{d}^{(k)}$ is a descent direction, it can be shown that for a suitable choice of $\gamma^{(k)}$, we have $h(\boldsymbol{x}^{(k)}) < h(\boldsymbol{x}^{(k-1)})$ so that moving in the direction of $\boldsymbol{v}^{(k)}$ ensures an improving solution; see, e.g., Nocedal and Wright (2006).[3] In the classical Frank–Wolfe algorithm, the step size was fixed to $\gamma^{(k)} = 2/(k+2)$. A standard alternative is to do a line-search for the optimal step size in each iteration to obtain

$$\gamma^{(k)} \in \arg\min_{\gamma\in[0,1]} \; h\left(\boldsymbol{x}^{(k-1)} + \gamma \cdot \boldsymbol{d}^{(k)}\right).$$

[3] This is true as long as $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)} \right\rangle < 0$. If $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)} \right\rangle \geq 0$, then the convexity of $h(\cdot)$ implies that $h(\boldsymbol{x}) \geq h(\boldsymbol{x}^{(k-1)})$ for all $\boldsymbol{x} \in \mathcal{D}$ and consequently, $\boldsymbol{x}^{(k-1)}$ is an optimal solution.

Note that the new iterate $\boldsymbol{x}^{(k)}$ remains feasible; this follows because $\boldsymbol{x}^{(k)}$ is a convex combination of $\boldsymbol{x}^{(k-1)}$ and $\boldsymbol{v}^{(k)}$ and $\mathcal{D}$ is convex. Such feasibility of new iterates is the main benefit of the CG algorithm compared to other classical algorithms such as gradient descent, which may take infeasible steps that are then projected back onto the feasible region; such projection steps are usually computationally expensive. Another feature of the algorithm is that the solution at any iteration $k$ is a convex combination of the initial solution $\boldsymbol{x}^{(0)}$ and the extreme points $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(k)}$.

The CG algorithm is particularly attractive when solving the FRANK–WOLFE STEP is "easy"—for instance, if $\mathcal{D}$ is a polyhedron, it reduces to an LP. The CG algorithm has generated tremendous interest in the ML community for solving large-scale convex optimization problems in the recent past because of its "projection-free" property and ability to deal with structured constraint sets. The interested reader is referred to Jaggi's excellent thesis (Jaggi, 2011) for a more thorough development of the algorithm along with example applications.

We now apply the CG algorithm to solve the RANK-BASED MODEL ESTIMATION PROBLEM. This problem is exactly in the form (8.4) above with $h(\cdot) = \mathsf{loss}(\cdot)$ and $\mathcal{D} = \text{conv}(\mathcal{F}(\mathscr{P}))$. We initialize the algorithm by selecting an initial set of rankings $\mathscr{P}^{(0)} \subseteq \mathscr{P}$ and proportions $\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathscr{P}^{(0)}|-1}$, and setting $\boldsymbol{g}^{(0)} = \sum_{\sigma \in \mathscr{P}^{(0)}} \alpha_\sigma^{(0)} \boldsymbol{f}(\sigma)$, which by definition belongs to $\text{conv}(\mathcal{F}(\mathscr{P}))$.[4] However, we need to ensure that the initial loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are both bounded; this aspect is discussed in more detail in Sect. 8.3.1.3 below. Then, in each iteration $k \geq 1$, the FRANK–WOLFE STEP is of the form:

$$\min_{\boldsymbol{v} \in \text{conv}(\mathcal{F}(\mathscr{P}))} \mathsf{loss}(\boldsymbol{g}^{(k-1)}) + \left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle. \tag{8.5}$$

As mentioned earlier, the optimal solution to the above subproblem occurs at an extreme point of the feasible set $\text{conv}(\mathcal{F}(\mathscr{P}))$. Because this set is the convex hull of the vectors in $\mathcal{F}(\mathscr{P})$, the set of extreme points must be a subset of $\mathcal{F}(\mathscr{P})$. Consequently, problem (8.5) is equivalent to the following:

$$\min_{\boldsymbol{v} \in \mathcal{F}(\mathscr{P})} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle \equiv \min_{\sigma \in \mathscr{P}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\sigma) - \boldsymbol{g}^{(k-1)} \right\rangle, \tag{8.6}$$

where the equivalence follows from the definition of $\mathcal{F}(\mathscr{P})$. Let $\sigma^{(k)} \in \mathscr{P}$ denote an optimal solution to (8.6); we discuss how to solve it in more detail in Sect. 8.3.1.1 below. This means that the CG algorithm is iteratively adding rankings $\sigma^{(1)}, \sigma^{(2)}, \ldots$ to the support of the distribution. Consequently, we term subproblem (8.6) as the SUPPORT FINDING STEP.

As mentioned above, the standard variant of the CG algorithm does a line-search to compute the optimal step size, which results in maximum improvement in the objective value. An alternative is the "fully corrective" Frank–Wolfe (FCFW)

[4] We abuse notation and denote $\alpha_{\boldsymbol{f}(\sigma)}$ as $\alpha_\sigma$ for any $\sigma \in \mathscr{P}$ in the remainder of this section.

**Algorithm 1** CG algorithm for solving the RANK-BASED MODEL ESTIMATION PROBLEM

**Initialize:** $k \leftarrow 0$; $\mathscr{P}^{(0)} \subseteq \mathscr{P}$; $\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathscr{P}^{(0)}|-1}$; $\boldsymbol{g}^{(0)} = \sum_{\sigma \in \mathscr{P}^{(0)}} \alpha_\sigma^{(0)} \boldsymbol{f}(\sigma)$ s.t. $\mathsf{loss}(\boldsymbol{g}^{(0)})$, $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded
**while** stopping condition is not met **do**
  $k \leftarrow k + 1$
  Compute $\sigma^{(k)} \in \arg\min_{\sigma \in \mathscr{P}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\sigma) - \boldsymbol{g}^{(k-1)} \right\rangle$ (SUPPORT FINDING STEP)
  Update support of rankings $\mathscr{P}^{(k)} \leftarrow \mathscr{P}^{(k-1)} \cup \left\{\sigma^{(k)}\right\}$
  Compute $\boldsymbol{\alpha}^{(k)} \in \arg\min_{\boldsymbol{\alpha} \in \Delta_{|\mathscr{P}^{(k)}|-1}} \mathsf{loss}\left(\sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma \boldsymbol{f}(\sigma)\right)$ (PROPORTIONS UPDATE STEP)
  Update support of rankings $\mathscr{P}^{(k)} \leftarrow \left\{\sigma \in \mathscr{P}^{(k)} : \alpha_\sigma^{(k)} > 0\right\}$
  Update $\boldsymbol{g}^{(k)} \leftarrow \sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma^{(k)} \boldsymbol{f}(\sigma)$
**end while**
**Output:** rankings $\mathscr{P}^{(k)}$ and proportions $(\alpha_\sigma^{(k)} : \sigma \in \mathscr{P}^{(k)})$

variant (Shalev-Shwartz et al., 2010), which after finding the extreme point $\boldsymbol{v}^{(k)}$ in the FRANK–WOLFE STEP, re-optimizes the objective function over the convex hull $\mathrm{conv}(\{\boldsymbol{x}^{(0)}, \boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}\})$ of the initial solution and extreme points found so far. When applied to our context, the algorithm computes weights $\boldsymbol{\alpha}^{(k)} = (\alpha_\sigma^{(k)} : \sigma \in \mathscr{P}^{(k)})$ that minimize the loss function $\mathsf{loss}(\cdot)$ over the set $\mathrm{conv}\left(\left\{\boldsymbol{f}(\sigma) : \sigma \in \mathscr{P}^{(k)}\right\}\right)$, where $\mathscr{P}^{(k)}$ is the set of rankings recovered up to iteration $k$ (see the notation in Algorithm 1). It then obtains the next iterate as $\boldsymbol{g}^{(k)} := \sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma^{(k)} \boldsymbol{f}(\sigma)$. The weights $\boldsymbol{\alpha}^{(k)}$ represent the proportions of each ranking and consequently, we call this the PROPORTIONS UPDATE STEP. The fully corrective variant of the CG algorithm makes more progress (in terms of the improvement in the objective value) in each iteration than the line-search variant and is, therefore, most suited when the FRANK–WOLFE STEP is hard to solve. The entire procedure is summarized in Algorithm 1.

*Remark* We note that van Ryzin and Vulcano (2015) proposed a *market discovery algorithm* for obtaining the MLE of the rank-based choice model using a column generation procedure. Though the authors derived their algorithm using duality arguments, it can be verified that their procedure is identical to the one obtained from solving the RANK-BASED MODEL ESTIMATION PROBLEM with the KL-divergence loss function using the CG algorithm.

Next, we discuss the details of how to solve each of the SUPPORT FINDING and PROPORTIONS UPDATE STEPS.

#### 8.3.1.1 Solving the SUPPORT FINDING STEP

Noting that $\boldsymbol{f}(\sigma) = (\mathbb{1}[\sigma, i, S] : S \in \mathcal{M}, i \in S)$, the SUPPORT FINDING STEP can be written as follows:

$$\min_{\sigma \in \mathscr{P}} \sum_{S \in \mathcal{M}} \sum_{i \in S} \left( \nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)}) \right)_{i,S} \cdot \mathbb{1}[\sigma, i, S]. \tag{8.7}$$

This problem requires us to find a ranking with the minimum "cost," which is referred to as the rank aggregation problem in the ranking literature (Dwork et al., 2001) and is known to be NP-hard; see, for instance, van Ryzin and Vulcano (2015) and Jagabathula and Rusmevichientong (2019). In practice, subproblem (8.7) does not need to be solved to optimality and any feasible solution that generates a descent direction is sufficient to ensure an improving solution in Algorithm 1. Below we discuss a few different approaches that can be used to obtain an approximate solution.

**Mixed Integer Program (MIP) Formulation** van Ryzin and Vulcano (2015, Section 4.3.2) formulated a special case of the rank aggregation subproblem (8.7), which they referred to as the "Type Discovery Subproblem," as an MIP. In particular, they considered the case of individual purchase transactions where a single transaction is observed for each offer set. The same formulation extends to the aggregated data setting, which we present below.[5]

To simplify the formulation, we let $\mu_{i,S} = \left( \nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)}) \right)_{i,S}$. We encode the ranking $\sigma$ using binary decision variables $b_{ij} \in \{0, 1\}$ for all $i, j \in [N], i \neq j$, defined so that $b_{ij} = 1$ if and only if product $i$ is preferred to product $j$, i.e., $\sigma(i) < \sigma(j)$. Further, we let $w_{i,S} = \mathbb{1}[\sigma, i, S]$ and denote the collection of decision variables as $\boldsymbol{b} = (b_{ij} : i, j \in [N], i \neq j)$, and $\boldsymbol{w} = (w_{i,S} : S \in \mathcal{M}, i \in S)$. Then, subproblem (8.7) is equivalent to the following MIP:

$$\min_{\boldsymbol{b}, \boldsymbol{w}} \quad \sum_{S \in \mathcal{M}} \sum_{i \in S} \mu_{i,S} \cdot w_{i,S} \tag{8.8a}$$

$$\text{s.t. } b_{ij} + b_{ji} = 1 \quad \forall\, i, j \in [N], i < j \tag{8.8b}$$

$$b_{ij} + b_{jl} + b_{li} \leq 2 \quad \forall\, i, j, l \in [N], i \neq j \neq l \tag{8.8c}$$

$$w_{j,S} \leq b_{ji} \quad \forall S \in \mathcal{M}, \forall i, j \in S, i \neq j \tag{8.8d}$$

$$\sum_{j \in S} w_{j,S} = 1 \quad \forall S \in \mathcal{M} \tag{8.8e}$$

$$b_{ij} \in \{0, 1\} \forall\, i, j \in [N], i \neq j \tag{8.8f}$$

$$w_{i,S} \in \{0, 1\} \forall\, S \in \mathcal{M}, i \in S. \tag{8.8g}$$

The constraint (8.8b) ensures that either product $i$ is preferred to product $j$ or $j$ is preferred to $i$ in the ranking. The second constraint (8.8c) enforces transitivity amongst any three products in the ranking: if product $i$ is preferred to $j$ and $j$ is

[5] Mišić (2016) also proposed a similar formulation for estimating the rank-based choice model with an $L^1$-norm loss function using a column generation approach.

preferred to $l$, then $i$ must be preferred to $l$. The third constraint (8.8d) encodes the consistency of the indicator variables $\mathbb{1}[\sigma, i, S]$; in particular, if $w_{j,S} = 1$, then it means that product $j$ is the most preferred product from offer set $S$. This implies that we must have $b_{ji} = 1$ for all $i \in S \setminus \{j\}$, i.e., $j$ is preferred over all other products in $S$. The fourth constraint (8.8e) ensures that only one of the indicator variables $\mathbb{1}[\sigma, i, S]$ is non-zero for each offer set $S$. The objective function (8.8a) is exactly the objective in (8.7). The formulation has $O(N^2 + M)$ binary variables, and $O(N^3 + N^2\,|\mathcal{M}|)$ constraints. Again, note that MIP (8.8) does not need to be solved to optimality, all we need is a feasible solution that generates a descent direction. Given any feasible solution $(\boldsymbol{b}, \boldsymbol{w})$, the corresponding ranking $\sigma$ can be computed by setting $\sigma(i) = 1 + \sum_{j \neq i} b_{ji}$ for all $i \in [N]$.

**Leverage Structure in Observed Offer Set Collection** Though the rank aggregation subproblem (8.7) is NP-hard in general, JR showed that if the observed offer set collection $\mathcal{M}$ possesses certain structures, it can be solved efficiently. The structure is captured via a *choice graph* over the observed offer sets: each offer set is a vertex and the edges capture relationships amongst the most preferred products (under any ranking) in the different offer sets. They show that subproblem (8.7), which they refer to as the RANK AGGREGATION LP, can be formulated as a DP or LP over the choice graph with linear or polynomial complexity (in $N$ and $|\mathcal{M}|$) for offer set collections that commonly arise in retail and revenue management settings. See Section 3 in JR for more details.

**Local Search Heuristic** A simple method to find an approximate solution to (8.7) is the local search heuristic that was proposed in Mišić (2016) and Jagabathula and Rusmevichientong (2017). This heuristic starts with a randomly chosen ranking and then tries to find a better solution by evaluating all "neighboring" rankings obtained by swapping the positions of any two products. The procedure is repeated until no neighboring ranking yields a smaller objective value for (8.7), resulting in a locally optimal solution $\hat{\sigma}$. If $\hat{\sigma}$ does not produce an improving solution in Algorithm 1, which can be verified by checking if $\boldsymbol{f}(\hat{\sigma}) - \boldsymbol{g}^{(k-1)}$ is a descent direction, i.e., $\left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\hat{\sigma}) - \boldsymbol{g}^{(k-1)} \right\rangle < 0$, then we redo the search starting from a different ranking, until we exhaust a limit on the number of tries.

#### 8.3.1.2 Solving the PROPORTIONS UPDATE STEP

When compared to the SUPPORT FINDING STEP, THE PROPORTIONS UPDATE STEP is easier to solve because the corresponding subproblem is itself a convex program over the unit simplex $\Delta_{|\mathscr{P}^{(k)}|-1}$. It can be solved via the "away steps" variant of the CG algorithm described in Sect. 8.4.1.2, which promotes recovery of sparse distributions. Note that in line 7 in Algorithm 1, we drop the rankings with zero probability mass from the support, decreasing the support size and resulting in a sparser distribution. Another approach to solving the PROPORTIONS UPDATE STEP is to use the expectation-maximization (EM) algorithm proposed by van Ryzin and Vulcano (2017), which was utilized by the same authors in their market discovery

algorithm (van Ryzin and Vulcano, 2015) for estimating the rank-based choice model. An appealing feature of this approach is that the M-step involves closed-form updates for the proportions $\boldsymbol{\alpha}$ and, therefore, is simple to implement.

#### 8.3.1.3 Initialization and Stopping Criterion

Line 1 in Algorithm 1 specifies that the initial collection of rankings $\mathscr{P}^{(0)}$ should be chosen such that the loss function and its gradient are bounded. In particular, for the KL-divergence loss function, choosing $\mathscr{P}^{(0)} = \left\{\sigma^{(0)}\right\}$ (and $\boldsymbol{\alpha}^{(0)} = (1)$) is not possible since this results in $g_{i,S}^{(0)} = 0$ for any $(i, S)$ where $\mathbb{1}[\sigma^{(0)}, i, S] = 0$, making the initial loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ unbounded. van Ryzin and Vulcano (2015) initialized their method with $N$ rankings, with each product $i \in [N]$ being the top-ranked product in exactly one ranking.[6] This ensures that $\boldsymbol{g}^{(0)} > \mathbf{0}_M$ so that both $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and the gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded. Jagabathula and Rusmevichientong (2017) considered an alternative approach where they start with a 'sales ranking' in which products are ranked according to their aggregate sales (across all offer sets), with higher sales products being more preferred in the ranking. Then, they obtain $N$ rankings by modifying the sales ranking: ranking $i$ is obtained by moving product $i$ to the top-rank while the remaining products are shifted down in the ranking. Again, this initialization ensures that $\boldsymbol{g}^{(0)} > \mathbf{0}_M$.

Depending on the end goal, different stopping conditions may be used to terminate the algorithm. If the objective is to get the best possible fit to the data, then ideally we would like to run the algorithm until we are "close" to the optimal solution $\boldsymbol{g}^*$ of the RANK-BASED MODEL ESTIMATION PROBLEM. If the SUPPORT FINDING STEP can be solved optimally in each iteration, then its solution can be used to construct an upper bound on the *optimality gap* of the current solution $\boldsymbol{g}^{(k)}$, defined as $\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*)$; see Jaggi (2011) for details. Consequently, we can choose to terminate the algorithm when $\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*) \leq \varepsilon$ for some small $\varepsilon > 0$. An alternative approach is to stop when the absolute (or relative) change in the loss function objective is smaller than some pre-defined threshold. On the other hand, if the objective is to achieve good predictive performance out-of-sample, then the above approach may not work well as the final support may have a large number of rankings and thus overfit to the observed choice data. In such cases, standard information-theoretic measures proposed in the mixture modeling literature (McLachlan and Peel, 2004) such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc. that penalize overly complex mixture models or ML techniques such as cross-validation can be used for model selection. The approach used in Jagabathula et al. (2020b) was to limit the number of iterations of the algorithm based on an upper bound on the support size of rankings that one

[6] The remaining products in each ranking can be chosen arbitrarily.

is interested in finding. This idea is inspired by the *early stopping* rule in the ML literature (Yao et al., 2007; Prechelt, 2012).

### 8.3.2 Convergence Guarantee for the Estimation Algorithm

We can establish a convergence rate guarantee for the iterates $\boldsymbol{g}^{(k)}$ generated by Algorithm 1. Since the guarantee is identical to the case of the NPMXCL model discussed below, we do not repeat it here and refer the reader to Sect. 8.4.2 for the formal result. However, an interesting question is whether the special structure of the polytope $\text{conv}(\mathcal{F}(\mathscr{P}))$ can be leveraged to come up with stronger convergence rates. Of course, the result does not address recovery of the underlying distribution over rankings since it is not identifiable in general, as discussed earlier. To identify the distribution, additional constraints need to be imposed. One such approach was taken by Farias et al. (2013) where the goal was to find a distribution over rankings compatible with observed transaction data that produces the worst-case revenue for a given fixed assortment. The authors showed that such a distribution is approximately the *sparsest* rank-based choice model that explains the observed data.

## 8.4 Estimating the Nonparametric Mixture of Closed Logit (NPMXCL) Model

The rank-based model does not have the ability to extrapolate demand to new products or newer variants of existing products. One approach to address this issue was considered in Jagabathula and Rusmevichientong (2017). These authors extended the rank-based model to accommodate products with varying prices by adding a random *consideration set* layer on top of the rank-based model. Their consideration set model assumes that customers sample a threshold parameter and consider for purchase only those products who prices are less than the sampled threshold. From among the considered products, the customers then choose according to a rank-based choice model. In this model, the consideration set layer captures the impact of price changes on preferences and the rank-based model layer captures the impact of assortment changes on the preferences. The authors proposed an EM method to estimate the parameters of this generalized model and also showed how to use this model to jointly optimize product prices and the assortment offered to customers. However, this approach does not directly extend to capturing the variation in other product features.

For that, Jagabathula et al. (2020b) (henceforth JSV) generalize the rankings in the rank-based model to have a more flexible functional form that can incorporate product features. More formally, suppose that each product is represented by a

$D$-dimensional feature vector in some feature space $\mathcal{Z} \subseteq \mathbb{R}^D$. Example features include the price, brand, color, size, weight, etc. We let $z_{iS}$ denote the feature vector of product $i$ in offer set $S$, allowing product features (such as prices) to change over time/location with each offer set. If one of the products is the no-purchase option, then its feature vector is set to $\mathbf{0}_D$ in all offer sets.[7] For any offer set $S$, let $\boldsymbol{Z}_S = (z_{iS} : i \in S)$. Then we denote the collection of all observed feature vectors as $\boldsymbol{Z}_{\mathcal{M}} = (\boldsymbol{Z}_S : S \in \mathcal{M})$.

The population preferences are modeled as a distribution over *customer types*, defined as follows. We first consider the standard multinomial logit (MNL) types, whose choice behavior is governed by the MNL model. In particular, given a parameter (or "taste") vector $\boldsymbol{\beta} \in \mathbb{R}^D$, the MNL model specifies that a customer purchases product $i$ from offer set $S$ with probability

$$f_{i,S}(\boldsymbol{\beta}; \boldsymbol{Z}_S) = \frac{\exp(\boldsymbol{\beta}^\top z_{iS})}{\sum_{j \in S} \exp(\boldsymbol{\beta}^\top z_{jS})}, \quad \text{(MNL CHOICE PROBABILITY FUNCTION)}$$

where we have made the dependence on the set of feature vectors $\boldsymbol{Z}_S$ explicit. The taste vector $\boldsymbol{\beta}$ captures the "value" that a customer places on each of the product features in deciding which product to purchase. Each standard logit type is represented using the vector $\boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) \in (0, 1)^M$ which specifies the choice probabilities for the observed offer set collection:

$$\boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) = (f_{i,S}(\boldsymbol{\beta}; \boldsymbol{Z}_S) : i \in S, S \in \mathcal{M}). \tag{8.9}$$

Denote the set of all standard logit types as $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}}) \stackrel{\text{def}}{=} \left\{ \boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) : \boldsymbol{\beta} \in \mathbb{R}^D \right\}$. A key limitation of standard logit types is that they always assign a non-zero purchase probability to every product in every offer set. As a result, they cannot capture rank-based preferences, which allow for zero probabilities of purchase. To address this limitation, JSV allow the customer types to be also described by what they call *boundary* logit types, which include types on the "boundary" of the set $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})$. Formally, these types arise when the parameter vector $\boldsymbol{\beta}$ becomes unbounded, as we see below. Including the boundary types results in a model that is a distribution over the *closed logit types* $\overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})}$, which is the closure of the set $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})$ in $\mathbb{R}^M$; we consider closure with respect to the standard Euclidean topology on $\mathbb{R}^M$.

The following lemma establishes that the closed logit types contain rankings as special cases, showcasing that the rank-based choice model is subsumed by this model.

**Lemma 1** *For any offer set collection $\mathcal{M}$, there exists a feature specification $\boldsymbol{Z}_{\mathcal{M}}$ such that $\mathcal{F}(\mathscr{P}) \subset \overline{\mathcal{F}_{MNL}(\boldsymbol{Z}_{\mathcal{M}})}$.*

[7] In this case, the feature vector for other products would typically include a constant feature 1 to allow for general no-purchase market shares.

***Proof*** Recall that $\mathcal{F}(\mathscr{P}) = \{\boldsymbol{f}(\sigma)\colon \sigma \in \mathscr{P}\}$, where $\boldsymbol{f}(\sigma)$ is defined in (8.3). Suppose that the feature representation of each product $j \in [N]$ is set to the one-hot encoded vector, so that, $z_{jS} = \boldsymbol{e}_j$ for all offer sets $S$, where $\boldsymbol{e}_j \in \mathbb{R}^N$ is a vector of all zeros except 1 at the $j^{th}$ position. Note that the number of features $D = N$ in this case. Letting $\boldsymbol{E}_S = (\boldsymbol{e}_j\colon j \in S)$ and $\boldsymbol{E}_{\mathcal{M}} = (\boldsymbol{E}_S\colon S \in \mathcal{M})$, we will show that $\boldsymbol{f}(\sigma) \in \overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{E}_{\mathcal{M}})}$ for all $\sigma \in \mathscr{P}$.

Given any ranking $\sigma$, define $\boldsymbol{\beta}_\sigma \stackrel{\text{def}}{=} (-\sigma(1), -\sigma(2), \cdots, -\sigma(N))$ and consider the sequence of standard logit types $\boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}})$ for each $r \in \mathbb{N}$. Using the MNL CHOICE PROBABILITY FUNCTION, it follows that for any $S \in \mathcal{M}$ and any $i \in S$:

$$\begin{aligned}
\lim_{r\to\infty} f_{i,S}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_S) &= \lim_{r\to\infty} \frac{\exp\left(r \cdot (\boldsymbol{\beta}_\sigma^\top \boldsymbol{e}_i)\right)}{\sum_{j\in S} \exp\left(r \cdot (\boldsymbol{\beta}_\sigma^\top \boldsymbol{e}_j)\right)} \\
&= \lim_{r\to\infty} \frac{\exp\left(-r \cdot \sigma(i)\right)}{\sum_{j\in S} \exp\left(-r \cdot \sigma(j)\right)} \\
&= \lim_{r\to\infty} \frac{1}{1 + \sum_{j\in S\setminus\{i\}} \exp\left(r \cdot (\sigma(i) - \sigma(j))\right)} \\
&= \mathbb{1}[\sigma, i, S],
\end{aligned}$$

where the last equality follows from the definition of $\mathbb{1}[\sigma, i, S]$. Letting $\lim_{r\to\infty} \boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}}) \stackrel{\text{def}}{=} \left(\lim_{r\to\infty} f_{i,S}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_S)\colon i \in S, S \in \mathcal{M}\right)$, it follows that $\lim_{r\to\infty} \boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}}) = \boldsymbol{f}(\sigma)$. Since the closure of a set contains all limit points, $\boldsymbol{f}(\sigma) \in \overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{E}_{\mathcal{M}})}$ and the claim follows. □

In the remainder of the section, we leave the dependence of the closed logit types on the observed feature vectors implicit and use $f_{i,S}(\boldsymbol{\beta})$ and $\boldsymbol{f}(\boldsymbol{\beta})$, respectively, to denote the choice probability under an MNL model and a standard logit type, and $\overline{\mathcal{F}_{\text{MNL}}}$ to denote the set of closed logit types. We also use $\mathcal{B}_{\text{MNL}} \stackrel{\text{def}}{=} \overline{\mathcal{F}_{\text{MNL}}} \setminus \mathcal{F}_{\text{MNL}}$ to denote the set of boundary logit types. Further, because the parameter vector $\boldsymbol{\beta}$ for a boundary logit type is not well-defined, we refer to a general customer type in $\overline{\mathcal{F}_{\text{MNL}}}$ simply as $\boldsymbol{f} = (f_{i,S}\colon i \in S, S \in \mathcal{M})$.

Now, as mentioned above, the population is described by a distribution over the customer types $\overline{\mathcal{F}_{\text{MNL}}}$. Let $\mathcal{Q} \stackrel{\text{def}}{=} \left\{Q\colon Q \text{ is a distribution over } \overline{\mathcal{F}_{\text{MNL}}}\right\}$ denote the space of all distributions over $\overline{\mathcal{F}_{\text{MNL}}}$.[8] Given any distribution $Q \in \mathcal{Q}$, the vector of choice probabilities for the offer set collection $\mathcal{M}$ is given by:

$$\boldsymbol{g}_{\mathcal{M}}(Q) = (g_{i,S}(Q)\colon i \in S, S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(Q) = \int_{\overline{\mathcal{F}_{\text{MNL}}}} f_{i,S}\, \mathrm{d}Q(\boldsymbol{f}). \tag{8.10}$$

[8] Our development here is closely related to that in JSV but with slight differences.

Then, defining $\mathcal{G}(\mathcal{Q}) \stackrel{\text{def}}{=} \{\boldsymbol{g}_{\mathcal{M}}(Q) : Q \in \mathcal{Q}\}$, the goal is to solve the GENERAL ESTIMATION PROBLEM with $\mathcal{G} = \mathcal{G}(\mathcal{Q})$. Unlike the rank-based model, however, where the distribution $\lambda$ was over the finite set of permutations $\mathscr{P}$, the distribution $Q$ is now defined over an infinite set of customer types $\overline{\mathcal{F}_{\text{MNL}}}$, and consequently it is more challenging to describe the constraint set $\mathcal{G}(\mathcal{Q})$. Despite this, JSV showed that $\mathcal{G}(\mathcal{Q})$ does permit an alternative representation that is easier to handle. For instance, suppose that $Q$ is a discrete distribution with finite support. Then, it is easy to see that $\boldsymbol{g}_{\mathcal{M}}(Q)$ must belong to the convex hull of the set $\overline{\mathcal{F}_{\text{MNL}}}$, defined as:

$$\text{conv}(\overline{\mathcal{F}_{\text{MNL}}}) = \left\{ \sum_{\boldsymbol{f} \in F} \alpha_{\boldsymbol{f}} \boldsymbol{f} \ : \ F \subset \overline{\mathcal{F}_{\text{MNL}}} \text{ is finite and } \sum_{\boldsymbol{f} \in F} \alpha_{\boldsymbol{f}} = 1, \alpha_{\boldsymbol{f}} \geq 0 \ \ \forall \boldsymbol{f} \in F \right\}.$$

More generally, since $\overline{\mathcal{F}_{\text{MNL}}}$ is a compact subset of $\mathbb{R}^M$ (it is closed by definition and bounded since each $\boldsymbol{f} \in [0,1]^M$), it follows from existing results (see, e.g., Lindsay, 1983) that the set $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ contains vectors $\boldsymbol{g}_{\mathcal{M}}(Q)$ generated by *any* distribution $Q$ over $\overline{\mathcal{F}_{\text{MNL}}}$, so in fact $\mathcal{G}(\mathcal{Q}) = \text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$. This is the reason we term this model the *nonparametric* mixture of closed logit (NPMXCL) model, since it does not impose any parametric assumptions on the mixing distribution $Q$.

With the above development, the GENERAL ESTIMATION PROBLEM for the NPMXCL model takes the form:

$$\min_{\boldsymbol{g} \in \text{conv}(\overline{\mathcal{F}_{\text{MNL}}})} \ \mathsf{loss}(\boldsymbol{g}), \qquad \text{(NPMXCL MODEL ESTIMATION PROBLEM)}$$

where again we drop the explicit dependence of the predicted choice probability vector $\boldsymbol{g}_{\mathcal{M}}$ on the offer set collection $\mathcal{M}$, and of the loss function on $\boldsymbol{y}_{\mathcal{M}}$. It can be verified that the NPMXCL MODEL ESTIMATION PROBLEM is a convex program with a compact constraint set; see Lemma 1 in JSV. Moreover, the strict convexity of the loss function again ensures that the NPMXCL MODEL ESTIMATION PROBLEM has a unique optimal solution (the proof is identical to that of Theorem 1 earlier).

**Relation to the Mixed Logit Models** The mixture of logit or mixed logit model (Hensher and Greene, 2003; Train, 2009) assumes that customer preferences are modeled as a distribution over standard logit types, that is, as a distribution over $\mathcal{F}_{\text{MNL}}$.[9] This model is designed to capture heterogeneity in customer preferences and also to overcome the restrictive independence of irrelevant alternatives (IIA) property of the MNL model (Luce, 1959) to allow for complex substitution patterns. In fact, McFadden and Train (2000) showed that any model in the RUM class can

[9] Technically, the distribution is modeled over the parameter vector $\boldsymbol{\beta}$ as opposed to its "type" representation $\boldsymbol{f}(\boldsymbol{\beta})$.

be approximated to arbitrary degree of accuracy by a mixed logit model with an appropriate specification for the product features and the mixing distribution.

While the mixed logit model is stated in this general form, it is rarely estimated as such. Traditionally, for purposes of tractability, the mixing distribution is restricted to belong to some parametric family $\mathcal{Q}(\Theta)$ of distributions defined over parameter space $\Theta$ such that $\mathcal{Q}(\Theta) \stackrel{\text{def}}{=} \{Q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ and $Q_{\boldsymbol{\theta}}$ is the mixing distribution over the MNL taste vector $\boldsymbol{\beta}$ corresponding to the parameter vector $\boldsymbol{\theta} \in \Theta$. Analogous to (8.10), the predicted choice probability vector $\boldsymbol{g}_{\mathcal{M}}(Q_{\boldsymbol{\theta}})$ corresponding to mixing distribution $Q_{\boldsymbol{\theta}}$ is given by:

$$\boldsymbol{g}_{\mathcal{M}}(Q_{\boldsymbol{\theta}}) = (g_{i,S}(Q_{\boldsymbol{\theta}}) : i \in S, S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(Q_{\boldsymbol{\theta}}) = \int_{\mathbb{R}^D} f_{i,S}(\boldsymbol{\beta}) \, \mathrm{d} \, Q_{\boldsymbol{\theta}}(\boldsymbol{\beta}). \tag{8.11}$$

The best fitting distribution from the family $\mathcal{Q}(\Theta)$ is then obtained by solving the following MLE problem:[10]

$$\max_{\boldsymbol{\theta} \in \Theta} \sum_{S \in \mathcal{M}} M_S \sum_{i \in S} y_{i,S} \log \left( g_{i,S}(Q_{\boldsymbol{\theta}}) \right). \tag{8.12}$$

Different assumptions for the family $\mathcal{Q}(\Theta)$ lead to different mixed logit models.

The most standard assumption is that the mixing distribution follows a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, parametrized by $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix of the distribution. The resulting model is referred to as the random parameters logit (RPL) model (Train, 2009). Under the RPL model, computing the choice probabilities in (8.11) requires the evaluation of an integral, which is often approximated through a Monte Carlo simulation. This results in a maximum simulated likelihood estimator (MSLE). Since the log-likelihood objective is typically non-convex in the parameters $\boldsymbol{\theta}$, gradient-based optimization routines are used to reach a local optimal solution. Often, additional structure is imposed on the covariance matrix (such as a diagonal matrix) to reduce the dimensionality of the parameter space. The interested reader is referred to Chapters 8 and 9 in Train (2009) for an overview of such estimation procedures.

The other common assumption is that the mixing distribution has a finite support of size $K$. The distribution is then parametrized by $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_K, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$, where $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$ denotes the support of the distribution and $(\alpha_1, \ldots, \alpha_K)$ denotes the corresponding mixture proportions, so that $\sum_{k \in [K]} \alpha_k = 1$ and $\alpha_k \geq 0$ for all $k \in [K]$. The resulting model is referred to as the latent class MNL (LC-MNL) model (Bhat, 1997; Boxall and Adamowicz, 2002; Greene and Hensher, 2003). In this case, the predicted choice probabilities in (8.11) simplify to $g_{i,S}(Q_{\boldsymbol{\theta}}) = \sum_{k=1}^{K} \alpha_k f_{i,S}(\boldsymbol{\beta}_k)$. However, direct optimization of the log-likelihood objective is challenging since it is non-convex in the parameters $\boldsymbol{\theta}$ and further, the

[10] This is equivalent to minimizing the KL-divergence loss function and is the standard choice when estimating the mixed logit model.

number of parameters scales with the number of mixture components: for a $K$ class LC-MNL model, we need to estimate $K \cdot D + K - 1$ parameters. Consequently, the EM algorithm is employed to solve the MLE problem, which reduces the original problem into iteratively fitting $K$ MNL models on weighted transformations of the observed sales fractions $\boldsymbol{y}_{\mathcal{M}}$. We refer the reader to Chapter 14 in Train (2009) for a detailed description of the EM algorithm for estimating LC-MNL models.

The NPMXCL model differs from the traditional mixed logit model in two key ways: it allows (a) individual customer types to be boundary logit types, as opposed to only standard logit types, and (b) the mixing distribution to be an arbitrary distribution. By allowing for boundary logit types, it subsumes the rank-based model (as shown in Lemma 1 above). In addition, by allowing for arbitrary mixing distributions, it mitigates the *model misspecification* issue. Both the RPL and the LC-MNL models are susceptible to model misspecification, which occurs when the ground-truth mixing distribution is not contained in the search space $Q(\Theta)$. Model misspecification can result in biased estimates for the parameters (Train, 2008) as well as poor goodness-of-fit (Fox et al., 2011). These issues are mitigated by the NPMXCL model.

### 8.4.1 Estimation via the Conditional Gradient Algorithm

We now discuss how to estimate the model parameters from observed choice data. The development of this section closely follows that of the rank-based model above. Since the NPMXCL MODEL ESTIMATION PROBLEM is a constrained convex program, in theory, we can use any standard method for convex optimization to solve it. However, similar to estimating the rank-based model earlier, there are two challenges: (a) the constraint region $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ lacks an efficient description; and (b) decomposing any candidate solution $\boldsymbol{g}$ into the underlying mixing distribution $Q$, which is needed so that out-of-sample predictions can be made, is a hard problem. In particular, note that $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ may *not* be a convex polytope as it could have infinitely many extreme points. JSV showed that the conditional gradient (CG) algorithm is again the ideal candidate to address both of these challenges.

As in the case of the rank-based model, we start with a distribution on an initial set of types $\mathcal{F}^{(0)} \subseteq \overline{\mathcal{F}_{\text{MNL}}}$ such that both the loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\, \mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded (see the discussion in Sect. 8.4.1.3 below). Then, using analogous arguments as in Sect. 8.3.1, the FRANK–WOLFE STEP in iteration $k \geq 1$ can be shown to be of the form:

$$\min_{\boldsymbol{v} \in \overline{\mathcal{F}_{\text{MNL}}}} \left\langle \nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle. \tag{8.13}$$

Let $\boldsymbol{f}^{(k)}$ denote an optimal solution to (8.13); we discuss how to solve it in Sect. 8.4.1.1 below. Again, we observe that the CG algorithm is iteratively adding customer types $\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots$ to the support of the mixing distribution. As before,

we use the FCFW variant that re-optimizes the loss objective over the support of the customer types recovered so far to promote recovery of sparser mixing distributions. Algorithm 2 summarizes the estimation procedure.

**Algorithm 2** CG algorithm for solving the NPMXCL MODEL ESTIMATION PROBLEM

1: **Initialize:** $k \leftarrow 0$; $\mathcal{F}^{(0)} \subseteq \overline{\mathcal{F}_{\text{MNL}}}$; $\boldsymbol{\alpha}^{(0)} \in \Delta_{\left|\mathcal{F}^{(0)}\right|-1}$; $\boldsymbol{g}^{(0)} = \sum_{\boldsymbol{f}\in\mathcal{F}^{(0)}} \alpha_{\boldsymbol{f}}^{(0)} \boldsymbol{f}$ s.t. $\mathsf{loss}(\boldsymbol{g}^{(0)})$, $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded
2: **while** stopping condition is not met **do**
3: $k \leftarrow k+1$
4: Compute $\boldsymbol{f}^{(k)} \in \arg\min_{\boldsymbol{v}\in\overline{\mathcal{F}_{\text{MNL}}}} \left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle$ (SUPPORT FINDING STEP)
5: Update support of types $\mathcal{F}^{(k)} \leftarrow \mathcal{F}^{(k-1)} \cup \left\{\boldsymbol{f}^{(k)}\right\}$
6: Compute $\boldsymbol{\alpha}^{(k)} \in \arg\min_{\boldsymbol{\alpha}\in\Delta_{\left|\mathcal{F}^{(k)}\right|-1}} \mathsf{loss}\left(\sum_{\boldsymbol{f}\in\mathcal{F}^{(k)}} \alpha_{\boldsymbol{f}} \boldsymbol{f}\right)$ (PROPORTIONS UPDATE STEP)
7: Update support of types $\mathcal{F}^{(k)} \leftarrow \left\{\boldsymbol{f} \in \mathcal{F}^{(k)} : \alpha_{\boldsymbol{f}}^{(k)} > 0\right\}$
8: Update $\boldsymbol{g}^{(k)} \leftarrow \sum_{\boldsymbol{f}\in\mathcal{F}^{(k)}} \alpha_{\boldsymbol{f}}^{(k)} \boldsymbol{f}$
9: **end while**
10: **Output:** customer types $\mathcal{F}^{(k)}$ and proportions $(\alpha_{\boldsymbol{f}}^{(k)} : \boldsymbol{f} \in \mathcal{F}^{(k)})$

Below, we discuss how to solve the SUPPORT FINDING STEP and PROPORTIONS UPDATE STEP in more detail.

#### 8.4.1.1 Solving the SUPPORT FINDING STEP

Recall that $\mathcal{F}_{\text{MNL}} = \left\{\boldsymbol{f}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{R}^D\right\}$ and $\boldsymbol{f}(\boldsymbol{\beta}) = (f_{i,S}(\boldsymbol{\beta}) : S \in \mathcal{M}, i \in S)$. By plugging in the MNL CHOICE PROBABILITY FUNCTION and ignoring constant terms, it follows that:

$$\begin{aligned} &\min_{\boldsymbol{v}\in\mathcal{F}_{\text{MNL}}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle \\ &\equiv \min_{\boldsymbol{\beta}\in\mathbb{R}^D} \sum_{S\in\mathcal{M}} \sum_{i\in S} \left(\nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)})\right)_{i,S} \cdot \frac{\exp\left(\boldsymbol{\beta}^\top \boldsymbol{z}_{iS}\right)}{\sum_{j\in S} \exp\left(\boldsymbol{\beta}^\top \boldsymbol{z}_{jS}\right)}. \end{aligned} \tag{8.14}$$

The optimal solution to the above problem may be unbounded. Such unbounded solutions are instances of the boundary logit types $\mathcal{B}_{\text{MNL}} = \overline{\mathcal{F}_{\text{MNL}}} \setminus \mathcal{F}_{\text{MNL}}$, as we show in Sect. 8.4.3 below.

Even if the optimal solution is bounded, finding it may be intractable because the objective in (8.14) is non-convex in the parameter $\boldsymbol{\beta}$ (see Online Appendix D in JSV). However, in practice, we only need to find a feasible descent direction to ensure an improving solution in Algorithm 2 and, therefore, general-purpose nonlinear program solvers can be employed to obtain approximate solutions to (8.14).

JSV reported favorable performance of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Nocedal and Wright, 2006, Section 6.1) in generating improving solutions, although other methods could also be explored.

#### 8.4.1.2 Solving the PROPORTIONS UPDATE STEP

As in the case of the rank-based model, the PROPORTIONS UPDATE STEP is a convex program over the unit simplex $\Delta_{\left|\mathcal{F}^{(k)}\right|-1}$. While in principle any method can be used to solve it, a particular variant of the CG algorithm is ideally suited. This variant (Guélat and Marcotte, 1986) compares two opposing steps to update the estimate in each iteration: the FRANK–WOLFE STEP that finds a descent direction, and an "away" step that reduces probability mass—possibly to zero—from a previously found extreme point (one amongst $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}$) or the initial solution $\boldsymbol{x}^{(0)}$. Observe that the FRANK–WOLFE STEP can be solved exactly for the PROPORTIONS UPDATE STEP by searching over the extreme points of the unit simplex $\Delta_{\left|\mathcal{F}^{(k)}\right|-1}$. The next iterate $\boldsymbol{x}^{(k)}$ is determined by the step (Frank–Wolfe or away) that results in larger improvement in the objective value; see Krishnan et al. (2015, Appendix B.1) for the precise description of this variant. The presence of away steps implies that the algorithm can 'drop' customer types, i.e., assign zero probability mass to, found in previous iterations from the support of the mixing distribution, resulting in sparser solutions. This is implemented in line 7 of Algorithm 2.

#### 8.4.1.3 Initialization and Stopping Criterion

Algorithm 2 can be initialized with any $\boldsymbol{g}^{(0)} \in \overline{\mathcal{F}_{\text{MNL}}}$ such that both the initial loss $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded. For instance, we could choose $\mathcal{F}^{(0)} = \left\{\boldsymbol{f}(\boldsymbol{\beta}_{\text{MNL}})\right\}$ and $\boldsymbol{\alpha}^{(0)} = (1)$, resulting in $\boldsymbol{g}^{(0)} = \boldsymbol{f}(\boldsymbol{\beta}_{\text{MNL}})$; where $\boldsymbol{\beta}_{\text{MNL}}$ is the parameter estimate obtained by fitting an MNL model to the data. The MNL log-likelihood objective is globally concave in $\boldsymbol{\beta}$ and there exist efficient algorithms (Hunter, 2004; Jagabathula and Venkataraman, 2020) that exhibit fast convergence in practice. Another option is to fit an LC-MNL model with a "small" number of classes using the EM algorithm.

The same stopping criterion listed in Sect. 8.3.1.3 can also be adopted for Algorithm 2.

### *8.4.2 Convergence Guarantee for the Estimation Algorithm*

JSV established a sublinear convergence guarantee for Algorithm 2. We state here a simplified version of their result (ignoring the derived constants) and the interested reader is referred to Section 5.1 in JSV for the precise guarantee:

**Theorem 2 (Sublinear Convergence of the CG Algorithm)** *For both loss functions defined in Sect. 8.2, the iterates generated by Algorithm 2 satisfy:*

$$\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*) = O\left(\frac{1}{k}\right) \qquad \text{for all } k \geq \bar{K},$$

*where* $\boldsymbol{g}^*$ *is an optimal solution to the* NPMXCL MODEL ESTIMATION PROBLEM *and* $\bar{K} \geq 1$ *is some index.*

***Proof*** For the detailed proof, please see Online Appendix A.2 in JSV; here we provide a sketch of the proof. Jaggi (2013) showed that the CG algorithm converges at an $O(1/k)$ rate if the (non-negative) *curvature constant* is bounded from above. The curvature constant is bounded if the constraint set is bounded and the hessian of the objective function is bounded from above. For the NPMXCL MODEL ESTIMATION PROBLEM, the domain $\mathrm{conv}(\overline{\mathcal{F}_{\mathrm{MNL}}}) \subseteq [0, 1]^M$ is bounded. For the squared norm loss function (Example 2 in Sect. 8.2), the hessian is also bounded from above and so the convergence guarantee follows from existing results. However, it can be verified that the hessian of the KL-divergence loss function (Example 1 in Sect. 8.2) becomes unbounded close to the boundary of the domain $\mathrm{conv}(\overline{\mathcal{F}_{\mathrm{MNL}}})$, i.e., when $\boldsymbol{g}$ has entries that are close to 0, and thus, the existing guarantee does not apply. JSV showed that each iterate $\boldsymbol{g}^{(k)}$ generated by Algorithm 2 has entries that are bounded from below by a data-dependent constant $\xi_{\min} > 0$. In other words, the iterates do not get too close to the boundary of the domain and they exploit this fact to establish the $O(1/k)$ convergence rate for the KL-divergence loss function as well, with the constant scaling in $1/\xi_{\min}^2$. □

While the above result establishes convergence of $\mathsf{loss}(\boldsymbol{g}^{(k)})$ to the optimal objective $\mathsf{loss}(\boldsymbol{g}^*)$, it does not say anything regarding convergence to the true mixing distribution from which the data was generated. Without additional assumptions, establishing convergence to the ground-truth mixing distribution is challenging since $\boldsymbol{g}^*$ can be decomposed into many underlying distributions in general. JSV showed through a simulation study that Algorithm 2 does recover good approximations to different ground-truth mixing distributions when there is sufficient variation in the observed choice data. Identifying conditions under which the CG algorithm recovers the ground-truth mixing distribution is an interesting direction for future work.

To gain further insights, JSV also analyzed the support of the mixing distribution recovered by the CG algorithm, which is determined by the structure of the optimal solutions to the SUPPORT FINDING STEP. We discuss this next.

### 8.4.3 Characterizing the Choice Behavior of Closed Logit Types

As alluded to earlier, the optimal solution to the SUPPORT FINDING STEP can either be a standard logit type or a boundary logit type. Standard logit types are

characterized by their corresponding taste parameters $\boldsymbol{\beta}$, which can be used to make out-of-sample predictions on new offer sets. However, it is not immediately clear how to think of boundary logit types, since by definition there exists no parameter $\boldsymbol{\beta}$ that can describe such types. To address this issue, JSV provided the following concise characterization of boundary logit types (see Online Appendix A.3 in JSV for the proof):

**Theorem 3 (Characterization of Boundary Logit Types)** *Any boundary logit type $\boldsymbol{f} \in \mathcal{B}_{MNL}$ satisfies $f_{i,S} = 0$ for at least one $(i, S)$ pair in the observed offer set collection $\mathcal{M}$. Moreover, we can find parameters $\boldsymbol{\beta}_0, \boldsymbol{\omega} \in \mathbb{R}^D$ such that, for each $S \in \mathcal{M}$ and all $i \in S$ (with $r \in \mathbb{N}$ below):*

$$f_{i,S} = \lim_{r \to \infty} \frac{\exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_{iS}\right)}{\sum_{j \in S} \exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_{jS}\right)}.$$

The result establishes that boundary logit types assign zero probability to at least one data point in the observed offer set collection $\mathcal{M}$, compared to standard logit types that assign non-zero probabilities to all observations. Moreover, boundary logit types arise as a result of limiting MNL models, obtained as the parameter vector $\boldsymbol{\beta}$ is pushed to infinity. In particular, for any boundary logit type $\boldsymbol{f}$, there exist parameters $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$ such that $\boldsymbol{f} = \lim_{r \to \infty} \boldsymbol{f}(\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})$, where recall that $\boldsymbol{f}(\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})$ corresponds to a standard logit type with parameter vector $\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega}$. Thus, unlike standard logit types that are described by a single parameter vector, boundary types are characterized by a pair of parameters. In fact, boundary logit types can be considered as natural generalizations of rankings to capture the impact of changing product features, as we show next.

The above characterization reveals a preference ordering over the products induced by the parameter vector $\boldsymbol{\omega}$, that determines which product is chosen from a given offer set. For ease of exposition, suppose that product features do not vary with the offer set, so that we can write $\boldsymbol{z}_j$ instead of $\boldsymbol{z}_{jS}$ for the feature vector of product $j$ in each offer set $S$. The preference order is determined by product utilities $u_j \stackrel{\text{def}}{=} \boldsymbol{\omega}^\top \boldsymbol{z}_j$ for each product $j \in [N]$. In particular, the utilities induce a preference order $\succsim$ among the products such that $j \succsim j'$, read as "product $j$ is weakly preferred over product $j'$," if and only if $u_j \geq u_{j'}$. The relation $\succsim$ is in general a weak (or partial) ordering and *not* a strict (or complete) ordering because utilities of two products may be equal. Consequently, we write $j \succ j'$ if $u_j > u_{j'}$ and $j \sim j'$ if $u_j = u_{j'}$. Note that such a preference order differs from a ranking in two ways: (a) it can be a partial ordering, and (b) the ordering depends on the values of the product features.

Similar to rankings, it can be shown that when offered any set $S \subseteq [N]$, boundary logit types choose only amongst the most preferred products in $S$, determined according to the preference order $\succsim$. To see that, let $C(S)$ denote the set of most preferred products in $S$, so that for all $j \in C(S)$, we have $j \sim \ell$ if $\ell \in C(S)$ and $j \succ \ell$ if $\ell \in S \setminus C(S)$. Let $u^* \stackrel{\text{def}}{=} \max\left\{u_j : j \in S\right\}$ denote the maximum utility

among the products in $S$. From the definition of $\succsim$, it follows that $u^* = u_j$ for all $j \in C(S)$ and $u^* > u_j$ for all $j \in S \setminus C(S)$. Note it is possible that $C(S) = S$ in case all the product utilities are equal. Now to determine which products will be chosen from $S$, we first multiply the numerator and denominator of the choice probabilities defined in Theorem 3 by $e^{-r \cdot u^*}$. Then, it follows that for any $j \in S$:

$$\begin{aligned} &\frac{\exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top z_j\right)}{\sum_{\ell \in S} \exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top z_\ell\right)} \\ &= \frac{e^{-r \cdot (u^* - u_j)} \cdot \exp(\boldsymbol{\beta}_0^\top z_j)}{\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top z_\ell) + \sum_{\ell \in S \setminus C(S)} e^{-r \cdot (u^* - u_\ell)} \cdot \exp(\boldsymbol{\beta}_0^\top z_\ell)}, \end{aligned} \tag{8.15}$$

where we plugged in $\boldsymbol{\omega}^\top z_\ell = u_\ell$ for each $\ell \in S$. Taking the limit $r \to \infty$, it follows that each of the terms $e^{-r \cdot (u^* - u_\ell)}$, $\ell \in S \setminus C(S)$, goes to zero since $u_\ell < u^*$. As a result, the denominator in (8.15) converges to $\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top z_\ell)$. On the other hand, the numerator converges to $\exp(\boldsymbol{\beta}_0^\top z_j)$ if $j \in C(S)$ and 0 if $j \in S \setminus C(S)$. Combining the two, we obtain the following choice probability prediction for any product $j$ in offer set $S$ from Theorem 3:

$$f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) = \begin{cases} \exp(\boldsymbol{\beta}_0^\top z_j) / \left(\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top z_\ell)\right), & \text{if } j \in C(S) \text{ and} \\ 0, & \text{if } j \in S \setminus C(S), \end{cases}$$

where we abuse notation and let $f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega})$ denote the probability of choosing product $j$ from offer set $S$ under the boundary logit type described by $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$. This implies that only products that are within $C(S)$ are considered for purchase. Algorithm 3 outlines the above procedure for the general case.

Note the contrasting roles of the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\beta}_0$ in defining the choice probabilities for a boundary logit type. The parameter vector $\boldsymbol{\omega}$ (through the preference ordering $\succsim$ it induces) determines the *consideration set* $C(S)$—the subset of products that the customer considers for purchase—whereas the parameter vector

**Algorithm 3** Predicting choice probabilities for boundary logit type described by parameters $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$

1: **Input:** Offer set $S$ with product features $z_{jS} \in \mathbb{R}^D$ for each $j \in S$
2: Compute utilities $u_j = \boldsymbol{\omega}^\top z_{jS}$ for each $j \in S$.
3: Form consideration set $C(S) = \left\{ j \in S \mid u_j = \max_{\ell \in S} u_\ell \right\}$
4: For any $j \notin C(S)$, $f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) \leftarrow 0$
5: For any $j \in C(S)$,

$$f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) \leftarrow \frac{\exp\left(\boldsymbol{\beta}_0^\top z_{jS}\right)}{\sum_{\ell \in C(S)} \exp\left(\boldsymbol{\beta}_0^\top z_{\ell S}\right)}$$

6: **Output:** Choice probabilities $\left(f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) : j \in S\right)$

$\boldsymbol{\beta}_0$ determines the choice probabilities from within the consideration set, governed by an MNL model. In particular, the parameter vector $\boldsymbol{\omega}$ dictates how a product's features impact its inclusion into the consideration set. For instance, suppose that product $j$ with utility $u_j < u^*$ is not in consideration currently, where recall that $u^*$ is the maximum utility of a product in offer set $S$. Further, suppose one of the features is price and the corresponding coefficient in parameter vector $\boldsymbol{\omega}$ is $\omega_p < 0$. Then, product $j$ will enter into consideration only if its price is sufficiently reduced so that its resulting utility is at least $u^*$ (assuming all other features are held constant). In other words, the price should be dropped by at least $\frac{u^* - u_j}{-\omega_p}$ to ensure consideration of product $j$. Such a dependence cannot be modeled via rankings since they do not capture the impact of changing product features on the choice probabilities. Consequently, boundary logit types can be viewed as generalizations of rankings that account for more nuanced dependence of the choice behavior on the product features.

The choice behavior of boundary logit types is consistent with prior literature, which establishes that customers often consider a subset of the products on offer before making the choice; see, e.g., Hauser (2014), Jagabathula and Rusmevichientong (2017), and Aouad et al. (2020b). For further insights, we refer the reader to Section 5.3 in JSV where the authors analyze the consideration sets of the boundary logit types recovered by the CG algorithm.

## 8.5 Other Nonparametric Choice Models

There is growing interest in developing nonparametric methods to estimate choice models, and our discussion above has but scratched the surface. In this section, we briefly discuss other nonparametric choice models that have received attention in the operations literature.

**Choice Model Trees** Aouad et al. (2020a) propose *choice model trees*, a novel choice model which leverages a decision tree to segment the customer population based on observable characteristics like demographics and prior purchase history, and then fits an MNL model for each segment, where the segments correspond to the leaf/terminal nodes in the tree. The tree splits are recursively chosen to maximize the log-likelihood of the observed choice data, which is obtained by summing over the log-likelihoods for each leaf node. Their approach can be viewed as a nonparametric variant of the LC-MNL model introduced in Sect. 8.4, since the decision tree splits can be used to capture flexible mappings from customer characteristics to segments. Moreover, choice model trees assign each customer to exactly one segment, unlike the classical LC-MNL model that outputs a probabilistic assignment over the different segments. The authors show that their proposed model outperforms natural benchmarks in predictive accuracy, while also providing an interpretable segmentation of the population.

**Nonparametric Tree Choice Model** Paul et al. (2018) propose a general tree choice model where the customer demand is modeled via a rooted (undirected) binary tree in which each node corresponds to a product, and the set of all possible customer types is characterized by the set of all linear paths—paths that move either progressively toward or away from the root node—in the tree. Since each path can be viewed as a preference ordering of the products appearing on the path, their model can be viewed as a special case of the rank-based choice model as it considers only a subset of all possible rankings.[11] Their model generalizes the one proposed in Honhon et al. (2012), which only considered paths that start or end at the root node. To estimate the model, Paul et al. (2018) propose a greedy heuristic that incrementally adds nodes to the existing tree with the goal of maximizing the number of customer types that is consistent with the observed choice data, and prevents overfitting by controlling the depth of the tree. Having estimated the tree and, therefore, the set of customer types, they solve the MLE problem for estimating the distribution $\lambda$ over these types (recall the notation in Sect. 8.3). Since the log-likelihood objective is concave in the ranking probabilities $\lambda(\sigma)$ and the number of customer types is $O(N^2)$, the MLE problem can be solved efficiently using standard non-linear solvers. They also propose tractable algorithms for several assortment and pricing problems under the proposed choice model.

**Mixture of Mallows Model** One limitation of the rank-based choice model is that it assigns zero probability to any choice that is not consistent with any of the rankings in its support. This can be problematic since typically sparse models are chosen that have "small" support sizes. One remedy to this is the NPMXCL model of Jagabathula et al. (2020b) that we discussed above. An alternative approach was recently proposed by Désir et al. (2021), who consider a smoothed generalization of (sparse) rank-based models by assuming that the underlying probability distribution over rankings is specified as a mixture of Mallows models, with the number of mixture components equal to the support size of the rank-based model. The Mallows model (Mallows, 1957) assumes that consumer preferences are concentrated around a central ranking $\tau$ and the probability of sampling a ranking $\sigma$ different from $\tau$ falls exponentially with the Kendall-Tau distance $d(\sigma, \tau)$, defined as the number of pairwise disagreements between $\sigma$ and $\tau$. In other words, the Mallows model creates a smoothing property around the central ranking $\tau$. Therefore, the mixture of Mallows model provides a natural generalization of the rank-based choice model, assigning a non-zero probability to every possible choice. Désir et al. (2021) propose an EM algorithm to estimate the mixture of Mallows model, where the M-step involves solving a MIP. Moreover, they propose several practical approaches for solving the assortment optimization problem and show that Mallows-based smoothing can improve both the prediction as well as decision accuracy compared to the rank-based model.

[11] The rank-based model can allow for the number of products in a ranking to be strictly smaller than the size of the product universe, in which case the customer selects the no-purchase option if none of the products in the ranking is part of the offer set.

**DAG-Based Choice Model** The existing work on choice-based demand models in the operations literature has largely focused on using aggregate sales transaction data for estimation, and this has been the focus of our discussion in this book chapter as well. However, with the increasing availability of individual-level transaction data (also referred to as *panel* data), there is an opportunity to capture and estimate individual preferences. One of the first steps in this direction was taken by Jagabathula and Vulcano (2018) who introduced a nonparametric choice model in which each customer is characterized by a directed acyclic graph (DAG) representing a partial order among products in a category. A directed edge from node $i$ to node $j$ in the DAG indicates that the customer prefers the product corresponding to node $i$ over the product corresponding to node $j$. The DAG captures the fact that customer preferences are acyclic or *transitive*. Unlike a full preference ordering, a DAG specifies pairwise preferences for only a subset of product pairs; therefore, it represents a partial order. When visiting the store, the customer samples a full preference ordering (ranking) consistent with her DAG according to a pre-specified distribution, forms a consideration set and then purchases the most preferred product (according to the sampled ranking) amongst the ones she considers. The authors provide a procedure to construct the DAG for each customer based on her store visits, and they define several behavioral models to form consideration sets. Then, they estimate the distribution over rankings that best explains the observed purchasing patterns of the customers. Using real-world panel data on grocery store visits, the authors show that their proposed approach provides more accurate and fine-grained predictions for individual purchase behavior compared to state-of-the-art benchmark methods. Recently, Jagabathula et al. (2020a) consider a refinement of this choice model with the objective of designing personalized promotions.

**Models Beyond the RUM Class** Our discussion has focused primarily on the RUM model class as it has been the de-facto choice model in the operations and revenue management literature for the past two decades. However, the recent work of Jagabathula and Rusmevichientong (2019) on the *limit of stochastic rationality* (LoR) provides evidence for the need to go beyond the RUM class. Recall from Sect. 8.2 that the global minimum of the loss function is achieved when $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$, resulting in zero loss and a perfect fit to the observed choice data. However, this may not be achievable if the observed choice data is inconsistent with the RUM model, so that $\boldsymbol{y}_{\mathcal{M}} \notin \mathcal{G}$. Using a case study on grocery stores sales transaction data, Jagabathula and Rusmevichientong (2019) showed that the *rationality loss*, which they define as the best fit achievable using a model in the RUM class, i.e., $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}^*)$ where $\boldsymbol{g}^*$ is the optimal solution to the RANK-BASED MODEL ESTIMATION PROBLEM, can be high for many product categories, suggesting the need for more sophisticated choice models. In their paper, the authors show that fitting a latent class generalized attraction model (LC-GAM) (Gallego et al., 2015), a parametric choice model that lies outside the RUM class, can help to breach the LoR for many categories. Since then, there has been significant progress in developing nonparametric models that extend the RUM class: the generalized stochastic preference (GSP) choice model (Berbeglia, 2018), the decision forest

choice model (Chen and Mišić, 2019) and the binary choice forest model (Chen et al., 2019) to name a few. This is an emerging research area and we expect a lot more work in this space.

## 8.6 Concluding Thoughts

Developing nonparametric methods for estimating choice models is an active area of research, with substantial interest both from academics and practitioners. With the availability of large volumes of increasingly granular data and corresponding access to flexible large-scale computing, nonparametric methods are not only possible but also necessary for attaining a high degree of prediction accuracy. We expect firms to continue to invest in implementing these methods to improve automated decision making.

We note that while the focus of this chapter has been on estimating choice models, there is a parallel stream of literature on using these models to solve operational decision problems of interest to firms; see Strauss et al. (2018) for a recent review. Two decision problems that have received significant attention within the literature are the assortment and the price optimization problems. In these decision problems, the firm wants to find the assortment (or offer set) and prices to offer to its customers, respectively, to maximize expected revenue or profit. Because of the cross-product cannibalization effects, firms must use choice models to solve these decision problems. Finding the optimal assortment or prices is significantly more difficult in nonparametric choice models because of the lack of exploitable structure. Existing literature has taken the approach of proposing efficient algorithms, sometimes using recent developments in solving mixed integer programs (MIPs), to approximate the optimal solution, see, e.g., Rusmevichientong et al. (2014), Jagabathula and Rusmevichientong (2017), Paul et al. (2018), Bertsimas and Mišić (2019), Aouad et al. (2020b), and Désir et al. (2021). We expect this parallel development to continue for the newer (and often, more general) choice models being proposed in the literature.

The design of general methods to effectively estimate large-scale choice models is taking place within the larger context of broader developments in artificial intelligence (AI) and machine learning (ML). The areas of AI/ML and operations research (OR) overlap significantly especially when it comes to model estimation. There is a healthy cross-pollination of ideas across these two communities (for example, the conditional gradient algorithm, which is a classical OR algorithm for solving quadratic programs, has recently gained in popularity in the ML community), and we expect this cross-pollination to push more of the model developments. As an example, consider that the methods discussed in this book chapter focused on generalizing the distributions over individual customer types. Each customer type in the NPMXCL model can be made more complex by allowing product utility values to depend on the features in a non-linear fashion. Linear specification is most common, partly driven by tractability reasons and partly by behavioral reasons (as

model parameters could then be conceived as marginal utilities, see, e.g., Ben-Akiva et al., 1985). Misspecified utility functions result in biased parameter estimates and low predictive accuracy. Popular ML approaches (such as random forests, neural networks, etc.) are well-suited for this purpose as they can learn highly non-linear representations of the utility, without imposing any a priori structures. Recent work has taken this approach in the context of transportation mode choices (see Han et al., 2020; Sifringer et al., 2020 and the references therein), and we expect this to be a fruitful future direction to pursue.

In addition, ML techniques can leverage unstructured data sources such as text, image, and video to construct feature representations, which can then be plugged into the utility specification along with other observed features such as price. Leveraging such sources is especially important in the context of online retail and e-commerce, where signals such as the image quality of the product, the (textual) reviews posted by prior customers, etc. are critical indicators of customer choice; see Liu et al. (2019, 2020) for some recent work using such types of data. We believe this is an exciting direction for the field and look forward to reading papers within this theme.

## References

Abdallah, T., & Vulcano, G. (2020). Demand estimation under the multinomial logit model from sales transaction data. *Manufacturing & Service Operations Management, 23*, 1005–1331.

Aouad, A., Elmachtoub, A. N., Ferreira, K. J., & McNellis, R. (2020a). Market segmentation trees. arXiv:1906.01174.

Aouad, A., Farias, V., & Levi, R. (2020b). Assortment optimization under consider-then-choose choice models. *Management Science, 67*, 3321–3984.

Barberá, S., & Pattanaik, P. K. (1986). Falmagne and the rationalizability of stochastic choices in terms of random orderings. *Econometrica: Journal of the Econometric Society, 54*, 707–715.

Ben-Akiva, M. E., Lerman, S. R., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and application to travel demand* (vol. 9). Cambridge: MIT Press.

Berbeglia, G. (2018). The generalized stochastic preference choice model. Available at SSRN 3136227.

Bertsimas, D., & Mišić, V. V. (2019). Exact first-choice product line optimization. *Operations Research, 67*(3), 651–670.

Bhat, C. R. (1997). An endogenous segmentation mode choice model with an application to intercity travel. *Transportation Science, 31*(1), 34–48.

Block, H. D., & Marschak, J. (1960). Random orderings and stochastic theories of responses. *Contributions to Probability and Statistics, 2*, 97–132.

Boxall, P. C., & Adamowicz, W. L. (2002). Understanding heterogeneous preferences in random utility models: A latent class approach. *Environmental and Resource Economics, 23*(4), 421–446.

Chen, N., Gallego, G., & Tang, Z. (2019). The use of binary choice forests to model and estimate discrete choices. Available at SSRN 3430886.

Chen, Y. C., & Mišić, V. (2019). Decision forest: A nonparametric approach to modeling irrational choice. Available at SSRN 3376273.

Clarkson, K. L. (2010). Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms, 6*(4), 63.

Désir, A., Goyal, V., Jagabathula, S., & Segev, D. (2021). Mallows-smoothed distribution over rankings approach for modeling choice. *Operations Research, 69*, 1015–1348.

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 613–622). New York: ACM.

Falmagne, J. C. (1978). A representation theorem for finite random scale systems. *Journal of Mathematical Psychology, 18*(1), 52–72.

Farias, V. F., Jagabathula, S., & Shah, D. (2013). A nonparametric approach to modeling choice with limited data. *Management Science, 59*(2), 305–322.

Fox, J. T., il Kim, K., Ryan, S. P., & Bajari, P. (2011). A simple estimator for the distribution of random coefficients. *Quantitative Economics, 2*(3), 381–418.

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly, 3*(1–2), 95–110.

Gallego, G., Ratliff, R., & Shebalov, S. (2015). A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research, 63*(1), 212–232.

Gallego, G., & Topaloglu, H. (2019). Introduction to choice modeling. In *Revenue management and pricing analytics* (pp. 109–128). Berlin: Springer.

Greene, W. H., & Hensher, D. A. (2003). A latent class model for discrete choice analysis: Contrasts with mixed logit. *Transportation Research Part B: Methodological, 37*(8), 681–698.

Guélat, J., & Marcotte, P. (1986). Some comments on wolfe's 'away step'. *Mathematical Programming, 35*(1), 110–119.

Haensel, A., & Koole, G. (2011). Estimating unconstrained demand rate functions using customer choice sets. *Journal of Revenue and Pricing Management, 10*(5), 438–454.

Han, Y., Zegras, C., Pereira, F. C., & Ben-Akiva, M. (2020). A neural-embedded choice model: Tastenet-mnl modeling taste heterogeneity with flexibility and interpretability. arXiv:200200922.

Hauser, J. R. (2014). Consideration-set heuristics. *Journal of Business Research, 67*(8), 1688–1699.

Hensher, D. A., & Greene, W. H. (2003). The mixed logit model: The state of practice. *Transportation, 30*(2), 133–176.

Honhon, D., Jonnalagedda, S., & Pan, X. A. (2012). Optimal algorithms for assortment selection under ranking-based consumer choice models. *Manufacturing & Service Operations Management, 14*(2), 279–289.

Hoyer, W. D., & Ridgway, N. M. (1984). Variety seeking as an explanation for exploratory purchase behavior: A theoretical model. In T. C. Kinnear (Ed.), *NA - Advances in consumer research* (vol. 11, pp. 114–119). Provo: ACR North American Advances.

Hunter, D. R. (2004). MM algorithms for generalized bradley-terry models. *Annals of Statistics, 32*, 384–406.

Jagabathula, S., & Rusmevichientong, P. (2017). A nonparametric joint assortment and price choice model. *Management Science, 63*(9), 3128–3145.

Jagabathula, S., Mitrofanov, D., & Vulcano, G. (2020a). Personalized retail promotions through a dag-based representation of customer preferences. *Operations Research, 70*, 641–1291.

Jagabathula, S., & Rusmevichientong, P. (2019). The limit of rationality in choice modeling: Formulation, computation, and implications. *Management Science, 65*(5), 2196–2215.

Jagabathula, S., Subramanian, L., & Venkataraman, A. (2020b). A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science, 66*(8), 3635–3656.

Jagabathula, S., & Venkataraman, A. (2020). An MM algorithm for estimating the MNL model with product features. Available at SSRN: https://ssrncom/abstract=3733971

Jagabathula, S., & Vulcano, G. (2018). A partial-order-based model to estimate individual preferences using panel data. *Management Science, 64*(4), 1609–1628.

Jaggi, M. (2011). *Sparse convex optimization methods for machine learning*. Ph.D. Thesis, ETH Zürich.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 427–435).

Kahn, B. E., & Lehmann, D. R. (1991). Modeling choice among assortments. *Journal of Retailing, 67*(3), 274–300.

Krishnan, R. G., Lacoste-Julien, S., & Sontag, D. (2015). Barrier frank-wolfe for marginal inference. In *Advances in Neural Information Processing Systems* (vol. 28, pp. 532–540)

Li, G., Rusmevichientong, P., & Topaloglu, H. (2015). The d-level nested logit model: Assortment and price optimization problems. *Operations Research, 63*(2), 325–342.

Lindsay, B. G. (1983). The geometry of mixture likelihoods: A general theory. *The Annals of Statistics, 11*, 86–94.

Liu, L., Dzyabura, D., & Mizik, N. (2020). Visual listening in: Extracting brand image portrayed on social media. *Marketing Science, 39*(4), 669–686.

Liu, X., Lee, D., & Srinivasan, K. (2019). Large-scale cross-category analysis of consumer review content on sales conversion leveraging deep learning. *Journal of Marketing Research, 56*(6), 918–943.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical analysis*. New York: Wiley.

Mahajan, S., & Van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research, 49*(3), 334–351.

Mallows, C. L. (1957). Non-null ranking models. I. *Biometrika, 44*(1–2), 114–130.

Manski, C. F. (1977). The structure of random utility models. *Theory and Decision, 8*(3), 229–254.

Mas-Colell, A., Whinston, M. D., Green. J. R. (1995). *Microeconomic theory* (vol 1). New York: Oxford University Press.

McFadden, D. (1981). Econometric models of probabilistic choice. In: *Structural analysis of discrete data with econometric applications* (pp. 198–272). Cambridge: MIT Press.

McFadden, D., & Train, K. (2000). Mixed MNL models for discrete response. *Journal of Applied Econometrics, 15*, 447–470.

McFadden, D. L. (2005). Revealed stochastic preference: A synthesis. *Economic Theory, 26*(2), 245–264.

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. Hoboken: Wiley.

Mišić, V. V. (2016). *Data, models and decisions for large-scale stochastic optimization problems*. Ph. D. Thesis, Massachusetts Institute of Technology, chapter 4: Data-driven Assortment Optimization.

Newman, J. P., Ferguson, M. E., Garrow, L. A., & Jacobs, T. L. (2014). Estimation of choice-based models using sales data from a single firm. *Manufacturing & Service Operations Management, 16*(2), 184–197.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd edn.). Berlin: Springer.

Paul, A., Feldman, J., & Davis, J. M. (2018). Assortment optimization and pricing under a nonparametric tree choice model. *Manufacturing & Service Operations Management, 20*(3), 550–565.

Prechelt, L. (2012). Early stopping—but when? In *Neural networks: Tricks of the trade* (pp. 53–67), Berlin: Springer.

Rusmevichientong, P., Shmoys, D., Tong, C., & Topaloglu, H. (2014). Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management, 23*(11), 2023–2039.

Shalev-Shwartz, S., Srebro, N., & Zhang, T. (2010). Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization, 20*(6), 2807–2832.

Sher, I., Fox, J. T., il Kim, K., & Bajari, P. (2011). Partial identification of heterogeneity in preference orderings over discrete choices. Tech. Rep., National Bureau of Economic Research.

Sifringer, B., Lurkin, V., & Alahi, A. (2020). Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological, 140*, 236–261.

Strauss, A. K., Klein, R., & Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research, 271*(2), 375–387.

Train, K. E. (2008). EM algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling, 1*(1), 40–69.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.

van Ryzin, G., & Vulcano, G. (2015). A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science, 61*(2), 281–300.

van Ryzin, G., & Vulcano, G. (2017). An expectation-maximization method to estimate a rank-based choice model of demand. *Operations Research, 65*(2), 396–407.

Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation, 26*(2), 289–315.

# Chapter 9
# The MNL-Bandit Problem


**Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi**

## 9.1 Introduction

One fundamental problem in revenue management that arises in many settings including retail and display-based advertising is assortment optimization. Here, the focus is on understanding how consumers select from a large number of substitutable items and identifying the optimal offer set to maximize revenues. Typically, for tractability, we assume a model that captures consumer preferences and focus on computing the optimal offer set. However, model selection and estimating the parameters is a challenging problem. In many e-commerce settings such as fast fashion retail, products have short selling seasons. Therefore, the data on consumer choices is either limited or nonexistent. The retailer needs to learn consumer preferences by offering different assortments and observing purchase decisions, but short selling seasons limit the extent of experimentation. There is a natural trade-off in these settings, where the retailer needs to learn consumer preferences and also maximizes cumulative revenues simultaneously. Finding the right balance between exploration and exploitation is a challenge. This chapter focuses on designing tractable robust algorithms for managing this trade-off in

S. Agrawal (✉) · V. Goyal
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA
e-mail: sa3305@columbia.edu; vgoyal@ieor.columbia.edu

V. Avadhanula
Facebook, Menlo Park, CA, USA

A. Zeevi
Decision, Risk and Operations, Columbia Business School, New York, NY, USA
e-mail: assaf@gsb.columbia.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_9

sequential decision-making under uncertainty for assortment optimization, which is a key component in many revenue management applications.

**Organization** We first provide an overview of assortment planning and the multinomial logit model (MNL), which is the most popular predictive model for this application domain in Sect. 9.2. In Sect. 9.3, we introduce the "MNL-Bandit problem" (term first coined in Agrawal et al. (2019)) that formulates the problem of dynamic assortment optimization and learning under the MNL choice model. In Sect. 9.4, we discuss approaches based on the principle of optimism under uncertainty from Agrawal et al. (2016) that bridges the aforementioned gap between theory and practice. In Sect. 9.5, discuss the Thompson Sampling (TS)-based approach from Agrawal et al. (2017) with similar theoretical guarantees. This approach motivated by the growing popularity of TS approaches in practice due to their attractive empirical performance. In Sect. 9.6, we discuss fundamental limits on the performance of any dynamic learning algorithm for the MNL-Bandit problem which establishes that the algorithms discussed in this chapter are near-optimal. We conclude in Sect. 9.7 with some discussion on recent progress on the extensions of MNL-Bandit problem to settings involving contextual features and a large number of products.

## 9.2 Choice Modeling and Assortment Optimization

In many settings, a decision-maker is faced with the problem of identifying an optimal mix of items from a large feasible set. For example, an online retailer needs to select a subset (assortment) of products to display to its shoppers. Due to substitution effects, the demand for an individual product is influenced by other products in the assortment presented to the shopper. In display-based online advertising, a publisher needs to select a set of advertisements to display to its users, and due to competition between the ads, the click rates for an individual ad depends on the assortment of ads displayed. A movie recommendation system like the one used by Netflix or Amazon must determine a small subset of items to suggest to its users from a large pool of similar alternatives, and the user response may depend on the overall attractiveness of the recommended set. Furthermore, in all these settings, different items may be valued differently from the decision-maker's perspective. Therefore, the assortment of items offered to users has significant impact on revenues. In order to identify the ideal set to offer, the decision-maker must understand the substitution patterns of users.

Choice models capture these substitution effects among items by specifying the probability with which a user selects an item from an offered set of items. More specifically, let $\mathcal{N} = \{1, \cdots, N\}$ be the set of all available items for the decision-maker to choose from. For any subset $S \subset \mathcal{N}$ and any item $i \in S$, a choice model describes the probability of a random consumer preferring item $i$ in the set $S$ as

$$\pi(i, S) = \Pr(\text{customer selects item } i \text{ from offer set } S).$$

We refer to $\pi(i, S)$ as choice probabilities. Using these choice probabilities, one can compute the expected revenue associated with an offer set as the weighted sum of revenues of items in the offer set and the choice probabilities. Specifically, if the value (revenue) associated with item $i \in \mathcal{N}$ is given by $r_i$, then the expected revenue $R(S)$ of any assortment $S \subset \mathcal{N}$ can be written as

$$R(S) = \sum_{i \in S} r_i \cdot \pi(i, S).$$

Then, the decision-maker can identify an optimal set by computing the set with highest expected revenue, resulting in an optimization problem commonly referred to as the *assortment optimization problem* and formulated as

$$\max_{S \subseteq \mathcal{N}} R(S). \tag{9.1}$$

More generally, assortment optimization problems also allow for constraints that arise in practice, e.g., budget for inventory, product purchasing, display capacity, etc.

A fundamental problem in assortment planning is (choice) model selection. There is a trade-off between working with models that have greater predictive power vs. simple models that allow greater tractability. Given a large number of alternatives, estimating choice probabilities from transactional data is a highly nontrivial task. As an extreme case, one may consider a choice model that makes no structural assumptions on the choice probabilities $\pi(i, S)$ and therefore can represent any customer choice behavior. Learning and optimizing under such a choice model would require estimating $2^N$ parameters and solving an intractable combinatorial optimization problem. The trade-offs between the representation power and the tractability of a choice model are an important consideration for the decision-maker in its deployment, particularly in settings where one needs to constantly estimate and optimize the model.

The *Multinomial Logit Model (MNL)*, owing primarily to its tractability, is one of the most widely used choice models for assortment selection problems. Recently, large-scale field experiments by Alibaba Feldman et al. (2021) have demonstrated the efficacy of the MNL model in boosting revenues. In this chapter, we use the MNL choice model to model customer preferences and develop efficient approaches that learn the model while simultaneously optimizing revenue.

Under the MNL model, the probability that a consumer purchases product $i$ when offered an assortment $S \subset \{1, \ldots, N\}$ is given by $\pi_{\mathsf{MNL}}(i, S) = \frac{v_i}{v_0 + \sum_{j \in S} v_j}$, where $v_i$ is the *attraction parameter* for product $i$ in the MNL model. Without loss of generality, we can assume that $v_0 = 1$, and therefore, the choice probabilities can be reformulated as

$$\pi_{\mathsf{MNL}}(i, S) = \frac{v_i}{1 + \sum_{j \in S} v_j}, \tag{9.2}$$

and the expected revenue for any assortment $S$ is given by

$$\mathbb{R}(S, \boldsymbol{v}) = \sum_{i \in S} r_i \frac{v_i}{1 + \sum_{j \in S} v_j}. \tag{9.3}$$

From the choice probabilities, we can see that the ratio of choice probabilities of two items, $\pi_{\mathsf{MNL}}(i, S)$ and $\pi_{\mathsf{MNL}}(j, S)$, is independent of the offer set $S$. This property is known as the independent of irrelevant attributes (IIA) property (Ben-Akiva and Lerman, 1985) and is a limitation of the MNL model. Other random utility-based choice models like Nested Logit (NL) (Williams, 1977) and Mixed Logit model (mMNL) (McFadden and Train, 2000) generalize the MNL model and are not restricted by the IIA property. However, estimation of these models and the corresponding assortment planning problems involved are often intractable highlighting the challenges involved in model selection. See Désir et al. (2021) for further discussion on tractability of choice models. The closed-form expression of the choice probabilities makes the MNL model extremely tractable from estimation and optimization point of view (see Talluri and Van Ryzin (2004).) The tractability of the model in decision-making is the primary reason MNL has been extensively used in practice (Greene, 2003; Ben-Akiva and Lerman, 1985; Train, 2009).

Traditionally, assortment decisions are made at the start of the selling period based on a choice model that has been estimated from historical data; see (Kok and Fisher, 2007) for a detailed review. In many business applications such as fast fashion and online retail, new products can be introduced or removed from the offered assortments in a fairly frictionless manner, and the selling horizon for a particular product can be short. Therefore, the traditional approach of first estimating the choice model and then using a static assortment based on the estimates is not practical in such settings. Rather, it is essential to experiment with different assortments to learn consumer preferences, while simultaneously attempting to maximize immediate revenues. Suitable balancing of this exploration–exploitation trade-off is the focus of the remainder of this chapter.

## 9.3 Dynamic Learning in Assortment Selection

As alluded to above, many instances of assortment optimization problems commence with very limited or even no a priori information about consumer preferences. Traditionally, due to production considerations, retailers used to forecast the uncertain demand before the selling season starts and decide on an optimal assortment to be held throughout. There are a growing number of industries like fast fashion and online display advertising where demand trends change constantly and new products (or advertisements) can be introduced (or removed) from offered

assortments in a fairly frictionless manner. In such situations, it is possible to experiment by offering different assortments and observing resulting purchases. Of course, gathering more information on consumer choice in this manner reduces the time remaining to exploit the said information.

Motivated by aforementioned applications, let us consider a stylized dynamic optimization problem that captures some salient features of the above application domain. The goal is to develop an exploration–exploitation policy that balances between gaining new information for learning the model and exploiting past information for optimizing revenue. In particular, consider a constrained assortment selection problem under the multinomial logit (MNL) model with $N$ substitutable products and a "no purchase" option. The objective is to design a policy that adaptively selects a sequence of history-dependent assortments $(S_1, S_2, \ldots, S_T) \in \mathcal{S}^T$ so as to maximize the cumulative expected revenue,

$$\mathbb{E}\left(\sum_{t=1}^{T} R(S_t, \mathbf{v})\right), \tag{9.4}$$

where $R(S, \mathbf{v})$ is the revenue corresponding to assortment $S$ as defined as in (9.3). We measure the performance of a decision-making policy via its *regret*. The objective then is to design a policy that approximately minimizes the *regret* defined as

$$\mathsf{Reg}(T, \boldsymbol{v}) = \sum_{t=1}^{T} R(S^*, \boldsymbol{v}) - \mathbb{E}[R(S_t, \mathbf{v})], \tag{MNL-Bandit}$$

where $S^* = \underset{S \in \mathcal{S}}{\operatorname{argmax}} \; R(S, \mathbf{v})$, with $\mathcal{S}$ being the set of feasible assortments. This exploration–exploitation problem, which is referred to as the **MNL-Bandit** problem, is the focus of this chapter.

**Constraints Over Assortment Selection** The literature considers several naturally arising constraints over the assortments that the retailer can offer. The simplest form of constraints is cardinality constraints, i.e., an upper bound on the number of products that can be offered in the assortment. Other more general constraints include partition matroid constraints (where the products are partitioned into segments and the retailer can select at most a specified number of products from each segment) and joint display and assortment constraints (where the retailer needs to decide both the assortment and the display segment of each product in the assortment and there is an upper bound on the number of products in each display segment). More generally, consider the set of totally unimodular (TU) constraints on the assortments. Let $\boldsymbol{x}(S) \in \{0, 1\}^N$ be the incidence vector for assortment $S \subseteq \{1, \ldots, N\}$, i.e., $x_i(S) = 1$ if product $i \in S$ and 0 otherwise. The approaches discussed here extend to constraints of the form

$$\mathcal{S} = \{S \subseteq \{1, \ldots, N\} \mid A\, \boldsymbol{x}(S) \leq \boldsymbol{b},\ \mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}\}, \tag{9.5}$$

where $\boldsymbol{A}$ is a totally unimodular matrix and $\boldsymbol{b}$ is integral (i.e., each component of the vector $\boldsymbol{b}$ is an integer). The totally unimodular constraints model a rich class of practical assortment planning problems including the examples discussed above. We refer the reader to Davis et al. (2013) for a detailed discussion on assortment and pricing optimization problems that can be formulated under the TU constraints.

**Algorithmic Approaches** Some initial works that consider the problem of minimizing regret under the MNL choice model include (Rusmevichientong et al., 2010; Sauré and Zeevi, 2013). Both these works present an "explore first and exploit later" approach. In particular, a selected set of assortments are explored until parameters can be estimated to a desired accuracy, and then the optimal assortment corresponding to the estimated parameters is offered for the remaining selling horizon. More specifically, when the expected revenue difference between the optimal and next best assortments is $\Delta$, existing approaches uniformly explore all the products for $O(\log T/\Delta)$ time periods and use the obtained data to estimate the optimal assortment. The exploration period that depends on the knowledge of the revenue gap, $\Delta$, is to ensure that the algorithm can identify the optimal assortment with "high probability." Following this approach, (Sauré and Zeevi, 2013) show an asymptotic $O(N \log T/\Delta)$ regret bound, while (Rusmevichientong et al., 2010) establish an $O(N^2 \log^2 T/\Delta)$ regret bound; recall $N$ is the number of products and $T$ is the time horizon. However, as highlighted above, their algorithm relies crucially on a priori knowledge of the revenue gap, $\Delta$, which is not readily available in practice. In Sect. 9.4.4, we will highlight via numerical simulations how lack of this knowledge can result in settings where these algorithms perform quite poorly. In the remainder of the chapter, we focus on approaches that simultaneously explore and exploit demand information. Specifically, we discuss a UCB (upper confidence bound)-based approach from Agrawal et al. (2016, 2019) and a Thompson Sampling-based approach from Agrawal et al. (2017). An advantage of these adaptive approaches is that they do not require any a priori knowledge or assumptions, and their performance is in some sense best possible (matches the worst-case lower bound), thereby, making these approaches more universal in its scope.

## 9.4 A UCB Approach for the MNL-Bandit

In this section, we discuss an algorithm from Agrawal et al. (2016, 2019) that adapts the popular upper confidence bounds (UCBs) approach to the MNL-Bandit problem. After presenting the details of the algorithm, in Sect. 9.4.2, we present the regret analysis that shows that this algorithm achieves a worst-case regret bound of $O(\sqrt{NT \log NT})$ under a mild assumption, namely that the no purchase

is the most "frequent" outcome. In Sect. 9.4.3, we also present the instance-dependent regret bounds that show that for "well separated" instances, the regret of the policy is bounded by $O\left(\min\left(N^2 \log NT/\Delta, \sqrt{NT \log NT}\right)\right)$, where $\Delta$ is the "separability" parameter discussed in the previous section. This is comparable to the regret bounds, $O\left(N \log T/\Delta\right)$ and $O\left(N^2 \log^2 T/\Delta\right)$, established in Sauré and Zeevi (2013) and Rusmevichientong et al. (2010), respectively, even though the policy does not require any prior information on $\Delta$ unlike the aforementioned work. Finally, in Sect. 9.4.4, we present a computational study from Avadhanula (2019) that highlights several salient features of the UCB-based policy. In particular, the study tests the performance of the proposed algorithm over instances with varying degrees of separability between optimal and suboptimal solutions and observe that the performance is bounded irrespective of the "separability parameter." In contrast, the approach of Sauré and Zeevi (2013) "breaks down" and results in linear regret for some values of the "separability parameter."

**Challenges and Overview**

A key difficulty in applying standard multi-armed bandit techniques to this problem is that the response observed on offering a product $i$ is *not* independent of other products in assortment $S$. Therefore, the $N$ products cannot be directly treated as $N$ independent arms. The algorithm presented here utilizes the specific properties of the dependence structure in MNL model to obtain an efficient algorithm with order $\sqrt{NT}$ regret.

The algorithm is based on a nontrivial extension of the UCB algorithm in Auer et al. (2002), which is predicated on Lai and Robbins (1985). It uses the past observations to maintain increasingly accurate upper confidence bounds for the MNL parameters $\{v_i, i = 1, \ldots, N\}$ and also uses these to (implicitly) maintain an estimate of expected revenue $R(S, \boldsymbol{v})$ for every feasible assortment $S$. In every round, the algorithm picks the assortment $S$ with the highest optimistic revenue. There are two main challenges in implementing this scheme. First, the customer response to being offered an assortment $S$ depends on the entire set $S$ and does not directly provide an unbiased sample of demand for a product $i \in S$. In order to obtain unbiased estimates of $v_i$ for all $i \in S$, we offer a set $S$ multiple times: specifically, it is offered repeatedly until a no purchase occurs. We show that proceeding in this manner, the average number of times a product $i$ is purchased provides an unbiased estimate of the parameter $v_i$. The second difficulty is the computational complexity of maintaining and optimizing revenue estimates for each of the exponentially many assortments. To this end, we use the structure of the MNL model and define our revenue estimates such that the assortment with maximum estimated revenue can be efficiently found by solving a simple optimization problem. This optimization problem turns out to be a static assortment optimization problem with upper confidence bounds for $v_i$'s as the MNL parameters, for which efficient solution methods are available.

### 9.4.1 Algorithmic Details

The algorithm divides the time horizon into epochs, where in each epoch we offer an assortment repeatedly until a no purchase outcome occurs. Specifically, in each epoch $\ell$, we offer an assortment $S_\ell$ repeatedly. Let $\mathcal{E}_\ell$ denote the set of consecutive time steps in epoch $\ell$. $\mathcal{E}_\ell$ contains all time steps after the end of epoch $\ell - 1$, until a no purchase happens in response to offering $S_\ell$, including the time step at which no purchase happens. The length of an epoch $|\mathcal{E}_\ell|$ conditioned on $S_\ell$ is a geometric random variable with success probability defined as the probability of no purchase in $S_\ell$. The total number of epochs $L$ in time $T$ is implicitly defined as the minimum number for which $\sum_{\ell=1}^{L} |\mathcal{E}_\ell| \geq T$.

At the end of every epoch $\ell$, we update our estimates for the parameters of MNL, which are used in epoch $\ell + 1$ to choose assortment $S_{\ell+1}$. For any time step $t \in \mathcal{E}_\ell$, let $c_t$ denote the consumer's response to $S_\ell$, i.e., $c_t = i$ if the consumer purchased product $i \in S_\ell$, and 0 if no purchase happened. We define $\hat{v}_{i,\ell}$ as the number of times a product $i$ is purchased in epoch $\ell$,

$$\hat{v}_{i,\ell} := \sum_{t \in \mathcal{E}_\ell} \mathbb{1}(c_t = i). \tag{9.6}$$

For every product $i$ and epoch $\ell \leq L$, we keep track of the set of epochs before $\ell$ that offered an assortment containing product $i$ and the number of such epochs. We denote the set of epochs by $\mathcal{T}_i(\ell)$ and the number of epochs by $T_i(\ell)$; that is,

$$\mathcal{T}_i(\ell) = \{\tau \leq \ell \,|\, i \in S_\tau\}, \quad T_i(\ell) = |\mathcal{T}_i(\ell)|. \tag{9.7}$$

We compute $\bar{v}_{i,\ell}$ as the number of times product $i$ was purchased per epoch,

$$\bar{v}_{i,\ell} = \frac{1}{T_i(\ell)} \sum_{\tau \in \mathcal{T}_i(\ell)} \hat{v}_{i,\tau}. \tag{9.8}$$

We show that for all $i \in S_\ell$, $\hat{v}_{i,\ell}$ and $\bar{v}_{i,\ell}$ are unbiased estimators of the MNL parameter $v_i$ (see Corollary 6). Using these estimates, we compute the upper confidence bounds, $v_{i,\ell}^{\mathsf{UCB}}$, for $v_i$ as

$$v_{i,\ell}^{\mathsf{UCB}} := \bar{v}_{i,\ell} + \sqrt{\bar{v}_{i,\ell} \frac{48 \log(\sqrt{N}\ell + 1)}{T_i(\ell)}} + \frac{48 \log(\sqrt{N}\ell + 1)}{T_i(\ell)}. \tag{9.9}$$

We establish that $v_{i,\ell}^{\mathsf{UCB}}$ is an upper confidence bound on the true parameter $v_i$, i.e., $v_{i,\ell}^{\mathsf{UCB}} \geq v_i$, for all $i, \ell$ with high probability (see Lemma 1). The role of the upper confidence bounds is akin to their role in hypothesis testing; they ensure that the likelihood of identifying the parameter value is sufficiently large. We then offer the optimistic assortment in the next epoch, based on the previous updates as follows:

$$S_{\ell+1} := \underset{S\in\mathcal{S}}{\text{argmax}} \max \left\{ R(S, \hat{\boldsymbol{v}}) : \hat{v}_i \le v_{i,\ell}^{\mathsf{UCB}} \right\}, \tag{9.10}$$

where $R(S, \hat{\boldsymbol{v}})$ is as defined in (9.3). We later show that the above optimization problem is equivalent to the following optimization problem:

$$S_{\ell+1} := \underset{S\in\mathcal{S}}{\text{argmax}}\ \tilde{R}_{\ell+1}(S), \tag{9.11}$$

where $\tilde{R}_{\ell+1}(S)$ is defined as

$$\tilde{R}_{\ell+1}(S) := \frac{\sum_{i\in S} r_i v_{i,\ell}^{\mathsf{UCB}}}{1+\sum_{j\in S} v_{j,\ell}^{\mathsf{UCB}}}. \tag{9.12}$$

We summarize the precise steps of this UCB-based algorithm in Algorithm 1.

Finally, we may remark on the computational complexity of implementing (9.10). The optimization problem (9.10) is formulated as a static assortment optimization problem under the MNL model with TU constraints, with model parameters being $v_{i,\ell}^{\mathsf{UCB}}$, $i = 1, \ldots, N$ (see (9.11)). There are efficient polynomial time algorithms to solve the static assortment optimization problem under

**Algorithm 1** Exploration–Exploitation algorithm for MNL-Bandit

**Initialization:** $v_{i,0}^{\mathsf{UCB}} = 1$ for all $i = 1, \ldots, N$
$t = 1$ ; $\ell = 1$ keeps track of the time steps and total number of epochs, respectively
**while** $t < T$ **do**
  Compute $S_\ell = \underset{S\in\mathcal{S}}{\text{argmax}}\ \tilde{R}_\ell(S) = \dfrac{\sum_{i\in S} r_i v_{i,\ell-1}^{\mathsf{UCB}}}{1+\sum_{j\in S} v_{j,\ell-1}^{\mathsf{UCB}}}$
  Offer assortment $S_\ell$, observe the purchasing decision, $c_t$ of the consumer
  **if** $c_t = 0$ **then**
    compute $\hat{v}_{i,\ell} = \sum_{t\in\mathcal{E}_\ell} \mathbb{1}(c_t = i)$, no. of consumers who preferred $i$ in epoch $\ell$, for all $i \in S_\ell$
    update $\mathcal{T}_i(\ell) = \{\tau \le \ell \,|\, i \in S_\ell\}$, $T_i(\ell) = |\mathcal{T}_i(\ell)|$, no. of epochs until $\ell$ that offered product $i$
    update $\bar{v}_{i,\ell} = \dfrac{1}{T_i(\ell)} \sum_{\tau\in\mathcal{T}_i(\ell)} \hat{v}_{i,\tau}$, sample mean of the estimates
    update $v_{i,\ell}^{\mathsf{UCB}} = \bar{v}_{i,\ell} + \sqrt{\bar{v}_{i,\ell} \dfrac{48\log(\sqrt{N}\ell+1)}{T_i(\ell)}} + \dfrac{48\log(\sqrt{N}\ell+1)}{T_i(\ell)}$; $\ell = \ell + 1$
  **else**
    $\mathcal{E}_\ell = \mathcal{E}_\ell \cup t$, time indices corresponding to epoch $\ell$
  **end if**
  $t = t + 1$
**end while**

MNL model with known parameters (see Avadhanula et al. 2016; Davis et al. 2013; Rusmevichientong et al. 2010). We will now briefly comment on how Algorithm 1 is different from the existing approaches of Sauré and Zeevi (2013) and Rusmevichientong et al. (2010) and also why other standard "bandit techniques" are not applicable to the MNL-Bandit problem.

*Remark 1 (Universality)* Note that Algorithm 1 does not require any prior knowledge/information about the problem parameters $\boldsymbol{v}$ (other than the assumption $v_i \leq v_0$, refer to Avadhanula (2019) for discussion on designing algorithms for settings when $v_i > v_0$). This is in contrast with the approaches of Sauré and Zeevi (2013) and Rusmevichientong et al. (2010), which require the knowledge of the "separation gap," namely, the difference between the expected revenues of the optimal assortment and the second best assortment. Assuming knowledge of this "separation gap," both these existing approaches explore a predetermined set of assortments to estimate the MNL parameters within a desired accuracy, such that the optimal assortment corresponding to the estimated parameters is the (true) optimal assortment with high probability. This forced exploration of predetermined assortments is avoided in Algorithm 1, which offers assortments adaptively, based on the current observed choices. The confidence regions derived for the parameters $\boldsymbol{v}$ and the subsequent assortment selection ensure that Algorithm 1 judiciously maintains the balance between exploration and exploitation that is central to the MNL-Bandit problem.

*Remark 2 (Estimation Approach)* Because the MNL-Bandit problem is parameterized with parameter vector ($\boldsymbol{v}$), a natural approach is to build on standard estimation approaches like maximum likelihood (MLE), where the estimates are obtained by optimizing a loss function. However, the confidence regions for estimates resulting from such approaches are either asymptotic and are not necessarily valid for finite time with high probability or typically depend on true parameters, which are not known a priori. For example, finite time confidence regions associated with maximum likelihood estimates require the knowledge of $\sup_{\boldsymbol{v}\in\mathcal{V}} I(\boldsymbol{v})$ (see Borovkov 1984), where $I$ is the Fisher information of the MNL choice model and $\mathcal{V}$ is the set of feasible parameters (that is not known a priori). Note that using $I(\boldsymbol{v}^{\mathsf{MLE}})$ instead of $\sup_{\boldsymbol{v}\in\mathcal{V}} I(\boldsymbol{v})$ for constructing confidence intervals would only lead to asymptotic guarantees and not finite sample guarantees. In contrast, in Algorithm 1, the estimation problem is resolved by a sampling method designed to give us unbiased estimates of the model parameters. The confidence bounds of these estimates and the algorithm do not depend on the underlying model parameters. Moreover, our sampling method allows us to compute the confidence regions by simple and efficient "book keeping" and avoids computational issues that are typically associated with standard estimation schemes such as MLE. Furthermore, the confidence regions associated with the unbiased estimates also facilitate a tractable way to compute the optimistic assortment (see (9.10), (9.11), and Step 4 of Algorithm 1), which is less accessible for the MLE estimate.

### 9.4.2 Min–Max Regret Bounds

For the regret analysis, we make the following assumptions.

**Assumption 1**

1. *The MNL parameter corresponding to any product* $i \in \{1, \ldots, N\}$ *satisfies* $v_i \leq v_0 = 1$.
2. *The family of assortments* $\mathcal{S}$ *is such that* $S \in \mathcal{S}$ *and* $Q \subseteq S$ *implies that* $Q \in \mathcal{S}$.

The first assumption is equivalent to the "no purchase option" being the most likely outcome. We note that this holds in many realistic settings, in particular, in online retailing and online display-based advertising. The second assumption implies that removing a product from a feasible assortment preserves feasibility. This holds for most constraints arising in practice including cardinality and matroid constraints more generally. We would like to note that the first assumption is made for ease of presentation of the key results and is not central to deriving bounds on the regret. The main result is the following upper bound on the regret of the policy stated in Algorithm 1.

**Theorem 1 (Performance Bounds for Algorithm 1)** *For any instance* $\boldsymbol{v} = (v_0, \ldots, v_N)$ *of the MNL-Bandit problem with* $N$ *products,* $r_i \in [0, 1]$, *and Assumption 1, the regret of the policy given by Algorithm 1 at any time* $T$ *is bounded as*

$$Reg_\pi(T, \boldsymbol{v}) \leq C_1 \sqrt{NT \log NT} + C_2 N \log^2 NT,$$

*where* $C_1$ *and* $C_2$ *are absolute constants (independent of problem parameters).*

**Proof Outline**
In this section, we briefly discuss an outline of different steps involved in proving Theorem 1. We refer the interested readers to Agrawal et al. (2019) and Avadhanula (2019) for detailed proofs.

**Confidence Intervals** The first step in the regret analysis is to prove the following two properties of the estimates $v_{i,\ell}^{UCB}$ computed as in (9.9) for each product $i$. Specifically, that $v_i$ is bounded by $v_{i,\ell}^{\mathsf{UCB}}$ with high probability and that as a product is offered an increasing number of times, the estimates $v_{i,\ell}^{\mathsf{UCB}}$ converge to the true value with high probability. Specifically, we have the following result.

**Lemma 1** *For every* $\ell = 1, \cdots, L$, *we have:*

1. $v_{i,\ell}^{UCB} \geq v_i$ *with probability at least* $1 - \frac{6}{N\ell}$ *for all* $i = 1, \ldots, N$.
2. *There exist constants* $C_1$ *and* $C_2$ *such that*

$$v_{i,\ell}^{UCB} - v_i \leq C_1 \sqrt{\frac{v_i \log(\sqrt{N}\ell + 1)}{T_i(\ell)}} + C_2 \frac{\log(\sqrt{N}\ell + 1)}{T_i(\ell)},$$

*with probability at least* $1 - \frac{7}{N\ell}$.

Intuitively, these properties establish $v_{i,\ell}^{UCB}$ as upper confidence bounds converging to actual parameters $v_i$, akin to the upper confidence bounds used in the UCB algorithm for MAB in Auer et al. (2002). These properties follow from an observation that is conceptually equivalent to the IIA (independence of irrelevant alternatives) property of MNL and shows that in each epoch $\tau$, $\hat{v}_{i,\tau}$ (the number of purchases of product $i$) provides independent unbiased estimates of $v_i$. Intuitively, $\hat{v}_{i,\tau}$ is the ratio of probabilities of purchasing product $i$ to preferring product 0 (no purchase), which is independent of $S_\tau$. This also explains why we choose to offer $S_\tau$ repeatedly until no purchase occurs. Given these unbiased i.i.d. estimates from every epoch $\tau$ before $\ell$, we apply a multiplicative Chernoff–Hoeffding bound to prove concentration of $\bar{v}_{i,\ell}$.

**Validity of the Optimistic Assortment** The product demand estimates $v_{i,\ell-1}^{\mathsf{UCB}}$ were used in (9.12) to define expected revenue estimates $\tilde{R}_\ell(S)$ for every set $S$. In the beginning of every epoch $\ell$, Algorithm 1 computes the optimistic assortment as $S_\ell = \arg\max_S \tilde{R}_\ell(S)$ and then offers $S_\ell$ repeatedly until no purchase happens. The next step in the regret analysis is to leverage the fact that $v_{i,\ell}^{\mathsf{UCB}}$ is an upper confidence bound on $v_i$ to prove similar, though slightly weaker, properties for the estimates $\tilde{R}_\ell(S)$. First, we note that estimated revenue is an upper confidence bound on the optimal revenue, i.e., $R(S^*, \boldsymbol{v})$ is bounded by $\tilde{R}_\ell(S_\ell)$ with high probability. The proof for these properties involves careful use of the structure of MNL model to show that the value of $\tilde{R}_\ell(S_\ell)$ is equal to the highest expected revenue achievable by any feasible assortment, among all instances of the problem with parameters in the range $[0, v_i^{\mathsf{UCB}}]$, $i = 1, \ldots, n$. Since the actual parameters lie in this range with high probability, we have that $\tilde{R}_\ell(S_\ell)$ is at least $R(S^*, \boldsymbol{v})$ with high probability. In particular, we have the following result.

**Lemma 2** *Suppose $S^* \in \mathcal{S}$ is the assortment with highest expected revenue, and Algorithm 1 offers $S_\ell \in \mathcal{S}$ in each epoch $\ell$. Then, for every epoch $\ell$, we have*

$$\tilde{R}_\ell(S_\ell) \geq \tilde{R}_\ell(S^*) \geq R(S^*, \boldsymbol{v}) \text{ with probability at least } 1 - \frac{6}{\ell}.$$

**Bounding the Regret** The final part of the analysis is to bound the regret in each epoch. First, we use the fact that $\tilde{R}_\ell(S_\ell)$ is an upper bound on $R(S^*, \boldsymbol{v})$ to bound the loss due to offering the assortment $S_\ell$. In particular, we show that the loss is bounded by the difference between the "optimistic" revenue estimate, $\tilde{R}_\ell(S_\ell)$, and the actual expected revenue, $R(S_\ell)$. We then prove a Lipschitz property of the expected revenue function to bound the difference between these estimates in terms of errors in individual product estimates $|v_{i,\ell}^{\mathsf{UCB}} - v_i|$. Finally, we leverage the structure of the MNL model and the properties of $v_{i,\ell}^{UCB}$ to bound the regret in each epoch. Lemma 3 provides the precise statements of above properties.

**Lemma 3** *If $r_i \in [0, 1]$, there exist constants $C_1$ and $C_2$ such that for every $\ell = 1, \cdots, L$, we have*

$$(1 + \sum_{j \in S_\ell} v_j)(\tilde{R}_\ell(S_\ell) - R(S_\ell, \boldsymbol{v})) \leq C_1 \sqrt{\frac{v_i \log(\sqrt{N}\ell + 1)}{|\mathcal{T}_i(\ell)|}} + C_2 \frac{\log(\sqrt{N}\ell + 1)}{|\mathcal{T}_i(\ell)|},$$

*with probability at least* $1 - \frac{13}{\ell}$.

### 9.4.3 Improved Regret Bounds for "Well Separated" Instances

In this section, we consider the problem instances that are "well separated" and present an improved logarithmic regret bound. More specifically, we present an $O(\log T)$ regret bound for Algorithm 1 for instances that are "well separated." In Sect. 9.4.2, we established worst-case regret bounds for Algorithm 1 that hold for all problem instances satisfying Assumption 1. While the algorithm ensures that the exploration–exploitation trade-off is balanced at all times, we demonstrate that it quickly converges to the optimal solution for the problem instances that are "well separated," leading to even better regret bounds. More specifically, we consider problem instances where the optimal assortment and "second best" assortment are sufficiently "separated" and derive an $O(\log T)$ regret bound that depends on the parameters of the instance. Note that, unlike the regret bound derived in Sect. 9.4.2 that holds for all problem instances satisfying Assumption 1, the bound we derive here only holds for instances having certain separation between the revenues corresponding to optimal and second best assortments. In particular, let $\Delta(\boldsymbol{v})$ denote the difference between the expected revenues of the optimal and second best assortment, i.e.,

$$\Delta(\mathbf{v}) = \min_{\{S \in \mathcal{S} | R(S, \boldsymbol{v}) \neq R(S^*, \boldsymbol{v})\}} \{R(S^*, \boldsymbol{v}) - R(S)\}. \tag{9.13}$$

We have the following result.

**Theorem 2 (Performance Bounds for Algorithm 1 in "Well Separated" Case)** *For any instance $\boldsymbol{v} = (v_0, \ldots, v_N)$ of the MNL-Bandit problem with $N$ products, $r_i \in [0, 1]$, and Assumption 1, the regret of the policy given by Algorithm 1 at any time $T$ is bounded as*

$$\mathit{Reg}(T, \boldsymbol{v}) \leq B_1 \left( \frac{N^2 \log T}{\Delta(\boldsymbol{v})} \right) + B_2,$$

*where $B_1$ and $B_2$ are absolute constants.*

**Proof Outline** We provide a proof outline here. We refer the interested readers to Avadhanula (2019) for a detailed proof. In this setting, we analyze the regret

by separately considering the epochs that satisfy certain desirable properties and the ones that do not. Specifically, we denote epoch $\ell$ as a "good" epoch if the parameters $v_{i,\ell}^{\mathsf{UCB}}$ satisfy the following property:

$$0 \leq v_{i,\ell}^{\mathsf{UCB}} - v_i \leq C_1 \sqrt{\frac{v_i \log(\sqrt{N}\ell + 1)}{T_i(\ell)}} + C_2 \frac{\log(\sqrt{N}\ell + 1)}{T_i(\ell)},$$

and we call it a "bad" epoch otherwise, where $C_1$ and $C_2$ are constants as defined in Lemma 1. Note that every epoch $\ell$ is a good epoch with high probability $(1 - \frac{13}{\ell})$, and we show that regret due to "bad" epochs is bounded by a constant (see Lemma 1). Therefore, we focus on "good" epochs and show that there exists a constant $\tau$, such that after each product has been offered in at least $\tau$ "good" epochs, Algorithm 1 finds the optimal assortment. Based on this result, we can then bound the total number of "good" epochs in which a suboptimal assortment can be offered by our algorithm. Specifically, let

$$\tau = \frac{4NC \log NT}{\Delta^2(\boldsymbol{v})}, \tag{9.14}$$

where $C = \max\{C_1^2, C_2\}$. Then, we have the following result.

**Lemma 4** *Let $\ell$ be a "good" epoch and $S_\ell$ be the assortment offered by Algorithm 1 in epoch $\ell$. If every product in assortment $S_\ell$ is offered in at least $\tau$ "good epochs," i.e., $T_i(\ell) \geq \tau$ for all $i$, then we have $R(S_\ell, \boldsymbol{v}) = R(S^*, \boldsymbol{v})$.*

The next step in the analysis is to show that Algorithm 1 will offer a small number of suboptimal assortments in "good" epochs. More specifically, we have the following result:

**Lemma 5** *Algorithm 1 cannot offer suboptimal assortments in more than $N\tau$ "good" epochs.*

It should be noted that the bound obtained in Theorem 2 is similar in magnitude to the regret bounds obtained by Sauré and Zeevi (2013) and is strictly better than the regret bound $O(N^2 \log^2 T)$ established by Rusmevichientong et al. (2010). Moreover, the algorithm does not require the knowledge of $\Delta(\boldsymbol{v})$, unlike the aforementioned papers that build on a conservative estimate of $\Delta(\boldsymbol{v})$ to implement their proposed policies.

### *9.4.4 Computational Study*

In this section, we present insights from numerical experiments in Avadhanula (2019) that test the empirical performance of our policy and highlight some of its salient features. We study the performance of Algorithm 1 from the perspective of

robustness with respect to the "separability parameter" of the underlying instance. In particular, we consider varying levels of separation between the revenues corresponding to the optimal assortment and the second best assortment and perform a regret analysis numerically. We contrast the performance of Algorithm 1 with the approach in Sauré and Zeevi (2013) for different levels of separation. We observe that when the separation between the revenues corresponding to optimal assortment and second best assortment is sufficiently small, the approach in Sauré and Zeevi (2013) breaks down, i.e., incurs linear regret, while the regret of Algorithm 1 only grows sub-linearly with respect to the selling horizon.

#### 9.4.4.1 Robustness of Algorithm 1

Here, we present a study that examines the robustness of Algorithm 1 with respect to the instance separability. We consider a parametric instance (see (9.15)), where the separation between the revenues of the optimal assortment and the next best assortment is specified by the parameter $\epsilon$ and compare the performance of Algorithm 1 for different values of $\epsilon$.

**Experimental Setup** We consider the parametric MNL setting with $N = 10$, $K = 4$, $r_i = 1$ for all $i$, and utility parameters $v_0 = 1$ and for $i = 1, \ldots, N$,

$$v_i = \begin{cases} 0.25 + \epsilon, & \text{if } i \in \{1, 2, 9, 10\} \\ 0.25, & \text{else}, \end{cases} \tag{9.15}$$

where $0 < \epsilon < 0.25$, specifies the difference between revenues corresponding to the optimal assortment and the next best assortment. Note that this problem has a unique optimal assortment $\{1, 2, 9, 10\}$ with an expected revenue of $1 + 4\epsilon/2 + 4\epsilon$ and the next best assortment has revenue of $1 + 3\epsilon/2 + 3\epsilon$. We consider four different values for $\epsilon$, $\epsilon = \{0.05, 0.1, 0.15, 0.25\}$, where higher value of $\epsilon$ corresponds to larger separation and hence an "easier" problem instance.

**Results** Figure 9.1 summarizes the performance of Algorithm 1 for different values of $\epsilon$. The results are based on running 100 independent simulations, and the standard errors are within 2%. Note that the performance of Algorithm 1 is consistent across different values of $\epsilon$, with a regret that exhibits sub-linear growth. Observe that as the value of $\epsilon$ increases, the regret of Algorithm 1 decreases. While not immediately obvious from Fig. 9.1, the regret behavior is fundamentally different in the case of "small" $\epsilon$ and "large" $\epsilon$. To see this, in Fig. 9.2, we focus on the regret for $\epsilon = 0.05$ and $\epsilon = 0.25$ and fit to $\log T$ and $\sqrt{T}$, respectively. (The parameters of these functions are obtained via simple linear regression of the regret vs $\log T$ and $\sqrt{T}$, respectively). It can be observed that the regret is roughly logarithmic when $\epsilon = 0.25$ and in contrast roughly behaves like $\sqrt{T}$ when $\epsilon = 0.05$. This illustrates the theory developed in Sect. 9.4.3, where we showed that the regret grows logarithmically with time, if the optimal assortment and the next best assortment are "well separated," while the worst-case regret scales as $\sqrt{T}$.

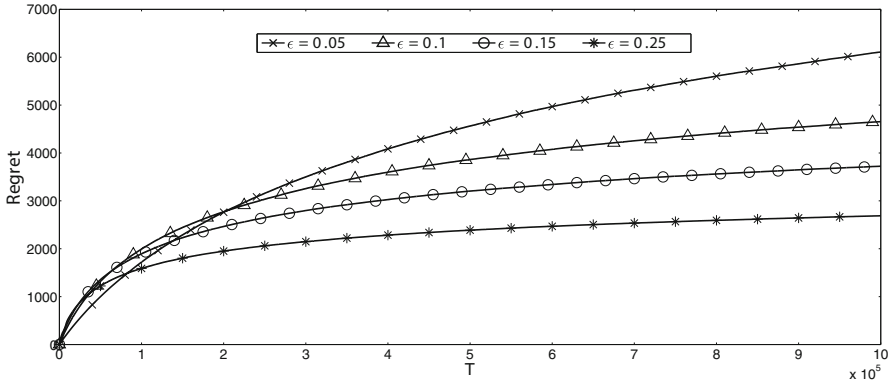


**Fig. 9.1** Performance of Algorithm 1 measured as the regret on the parametric instance (9.15). The graphs illustrate the dependence of the regret on $T$ for "separation gaps" $\epsilon = 0.05, 0.1, 0.15,$ and 0.25, respectively

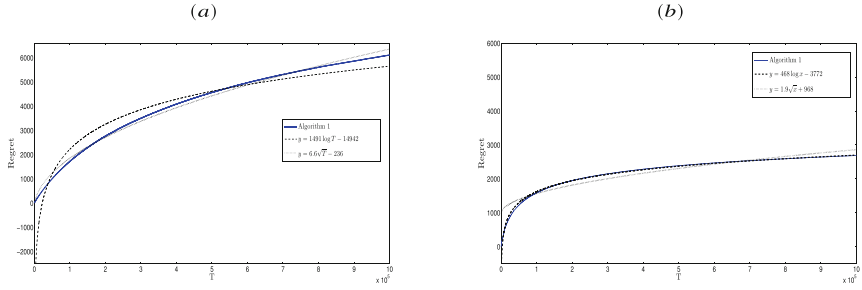


**Fig. 9.2** Best fit for the regret of Algorithm 1 on the parametric instance (9.15). The graphs (**a**) and (**b**) illustrate the dependence of the regret on $T$ for "separation gaps" $\epsilon = 0.05$ and 0.25, respectively. The best $y = \beta_1 \log T + \beta_0$ fit and the best $y = \beta_1 \sqrt{T} + \beta_0$ fit are superimposed on the regret curve

### 9.4.4.2 Comparison with Existing Approaches

In this section, we present a computational study comparing the performance of our algorithm to that of Sauré and Zeevi (2013). To be implemented, their approach requires certain a priori information of a "separability parameter"; roughly speaking, measuring the degree to which the optimal and next best assortments are distinct from a revenue standpoint. More specifically, their algorithm follows an *explore-then-exploit* approach, where every product is offered for a minimum duration of time that is determined by an estimate of said "separability parameter." After this mandatory exploration phase, the parameters of the choice model are estimated based on the past observations, and the optimal assortment corresponding to the estimated parameters is offered for the subsequent consumers. If the optimal assortment and the next best assortment are "well separated," then the offered assortment

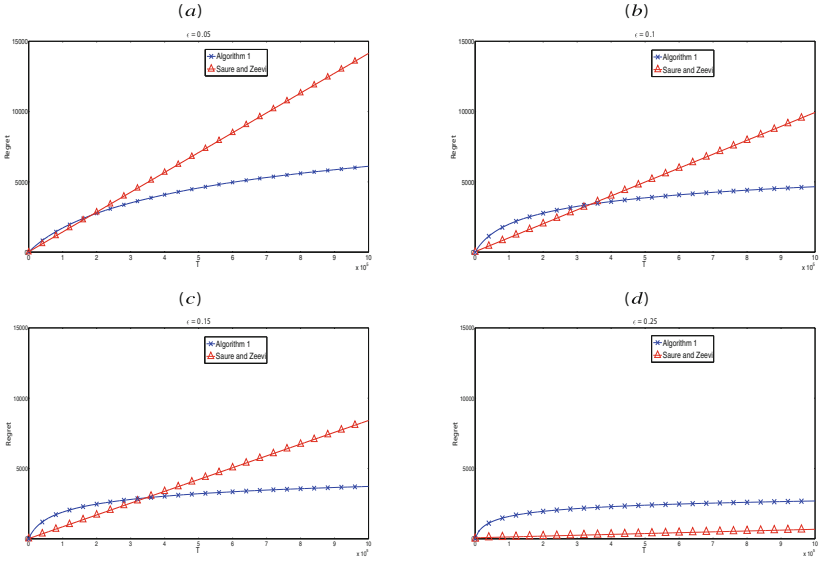


**Fig. 9.3** Comparison with the algorithm of Sauré and Zeevi (2013). The graphs (**a**), (**b**), (**c**), and (**d**) compare the performance of Algorithm 1 to that of Sauré and Zeevi (2013) on problem instance (9.15), for $\epsilon = 0.05, 0.1, 0.15,$ and 0.25 respectively

is optimal with high probability, otherwise, the algorithm could potentially incur linear regret. Therefore, the knowledge of this "separability parameter" is crucial. For our comparison, we consider the exploration period suggested by Sauré and Zeevi (2013) and compare it with the performance of Algorithm 1 for different values of separation ($\epsilon$). We will see that for any given exploration period, there is an instance where the approach in Sauré and Zeevi (2013) "breaks down" or in other words incurs linear regret, while the regret of Algorithm 1 grows sub-linearly ($O(\sqrt{T})$, more precisely) for all values of $\epsilon$ as asserted in Theorem 1.

**Experimental Setup and Results** We consider the parametric MNL setting as described in (9.15) and for each value of $\epsilon \in \{0.05, 0.1, 0.15, 0.25\}$. Since the implementation of the policy in Sauré and Zeevi (2013) requires knowledge of the selling horizon and minimum exploration period a priori, we take the exploration period to be $20 \log T$ as suggested in Sauré and Zeevi (2013) and the selling horizon $T = 10^6$. Figure 9.3 compares the regret of Algorithm 1 with that of Sauré and Zeevi (2013). The results are based on running 100 independent simulations with standard error of 2%. We observe that the regret for Sauré and Zeevi (2013) is better than the regret of Algorithm 1 when $\epsilon = 0.25$ but is worse for other values of $\epsilon$. This can be attributed to the fact that for the assumed exploration period, their algorithm fails to identify the optimal assortment within the exploration phase with sufficient probability and hence incurs a linear regret for $\epsilon = 0.05, 0.1,$ and 0.15. Specifically, among the 100 simulations we tested, the algorithm in Sauré and Zeevi

(2013) identified the optimal assortment for only 7%, 40%, 61%, and 97% cases, when $\epsilon = 0.05, 0.1, 0.15,$ and $0.25$, respectively. This highlights the sensitivity to the "separability parameter" and the importance of having a reasonable estimate for the exploration period. Needless to say, such information is typically not available in practice. In contrast, the performance of Algorithm 1 is consistent across different values of $\epsilon$, insofar as the regret grows in a sub-linear fashion in all cases.

## 9.5 Thompson Sampling for the MNL-Bandit

Motivated by the attractive empirical properties, in this section, we focus on a Thompson Sampling (TS)-based approach to the MNL-Bandit problem, first presented in Agrawal et al. (2017). In Sect. 9.5.1, we present the details of TS-based policy. In particular, we describe how to leverage the sampling technique introduced in Chap. 9.4 and design a prior distribution on the parameters of the MNL model such that the posterior update under the MNL-bandit feedback is tractable. In Sect. 9.5.4, we prove that the proposed algorithm achieves an $\tilde{O}(\sqrt{NT}\log TK)$ regret upper bound. Here, we also highlight the key ingredient of the TS-based approach, a two-moment approximation of the posterior, and the ability to judicially correlate samples, which is done by embedding the two-moment approximation in a normal family. Section 9.5.5 demonstrates the empirical efficiency of our algorithm design.

### *9.5.1 Algorithm*

In this section, we describe the posterior sampling (aka Thompson Sampling)-based algorithm for the MNL-Bandit problem. The basic structure of Thompson Sampling involves maintaining a posterior on the unknown problem parameters, which is updated every time new feedback is obtained. At the beginning of every round, a sample set of parameters is generated from the current posterior distribution, and the algorithm selects the best offer set according to these sample parameters. In the MNL-Bandit problem, there is one unknown parameter $v_i$ associated with each item. To adapt the TS algorithm for this problem, we would need to maintain a joint posterior for $(v_1, \ldots, v_N)$. However, updating such a joint posterior is nontrivial since the feedback observed in every round is a choice sampled from the multinomial distribution. This depends on the subset $S$ offered in that round. In particular, even if we initialize with an independent prior from a popular analytical family such as multivariate Gaussian, the posterior distribution after observing the MNL choice feedback will have a complex description. As a first step in addressing this challenge, we attempt to design a Thompson Sampling algorithm

with independent priors. In particular, we leverage a sampling technique introduced in Sect. 9.4 that allows us to decouple individual parameters from the MNL choice feedback and provide unbiased estimates of these parameters. We can utilize these unbiased estimates to efficiently maintain independent conjugate Beta priors for the parameters $v_i$ for each $i$. We present the details in Algorithm 1 below.

### *9.5.2 A TS Algorithm with Independent Beta Priors*

Here, we present the first version of the Thompson sampling algorithm, which will serve as an important building block for the main algorithm in Sect. 9.5.3. In this version, we maintain a Beta posterior distribution for each item $i = 1, \ldots, N$, which is updated as we observe users' choice of items from the offered subsets. A key challenge here is to choose priors that can be efficiently updated on observing user choice feedback, to obtain increasingly accurate estimates of parameters $\{v_i\}$. To address this, we use the sampling technique introduced in the previous section to decouple estimates of individual parameters from the complex MNL feedback. The idea is to offer a set $S$ multiple times; in particular, a chosen set $S$ is offered repeatedly until the "outside option" is picked (in the online advertising application discussed earlier, this corresponds to displaying the same subset of ads repeatedly until we observe a user who does not click on any of the displayed ads). Proceeding in this manner, due to the structure of the MNL model, the average number of times an item $i$ is selected provides an unbiased estimate of parameter $v_i$. Moreover, the number of times an item $i$ is selected is also independent of the displayed set and is a geometric distribution with success probability $1/(1 + v_i)$ and mean $v_i$. This observation is used as the basis for the epoch-based algorithmic structure and the choice of prior/posterior, as a conjugate to this geometric distribution.

**Epoch-Based Offerings** Similar to the UCB approach, the algorithm proceeds in epochs $\ell = 1, 2, \ldots$ An epoch is a group of consecutive time steps, where a set $S_\ell$ is offered repeatedly until the outside option is picked in response to offering $S_\ell$. The set $S_\ell$ to be offered in epoch $\ell$ is picked at the beginning of the epoch based on the sampled parameters from the current posterior distribution; the construction of these posteriors and choice of $S_\ell$ is described in the next paragraph. We denote the group of time steps in an epoch as $\mathcal{E}_\ell$, which includes the time step at which an outside option was preferred. The following lemmas provide important building blocks for our construction. Refer to Avadhanula (2019) for detailed proofs.

**Lemma 6 (Unbiased Estimate)** *Let $\tilde{v}_{i,\ell}$ be the number of times an item $i \in S_\ell$ is picked when the set $S_\ell$ is offered repeatedly until the outside option is picked. Then, for any $\ell$ and $i$, $\tilde{v}_{i,\ell}$ are i.i.d. geometric random variables with success probability $\frac{1}{1+v_i}$ and expected value $v_i$.*

**Lemma 7 (Conjugate Priors)** *For any $\alpha > 3, \beta > 0$, and $Y_{\alpha,\beta} \sim \mathsf{Beta}(\alpha, \beta)$, let $X_{\alpha,\beta} = \frac{1}{Y_{\alpha,\beta}-1}$ and $f_{\alpha,\beta}$ denote the probability distribution of random variable*

*$X_{\alpha,\beta}$. If the prior distribution of $v_i$ is $f_{\alpha,\beta}$, then after observing $\tilde{v}_{i,\ell}$, a geometric random variable with success probability $\frac{1}{v_i+1}$, the posterior distribution of $v_i$ is given by*

$$\mathbb{P}\left(v_i \middle| \tilde{v}_{i,\ell} = m\right) = f_{\alpha+1,\beta+m}(v_i).$$

**Construction of Conjugate Prior/Posterior** From Lemma 6, we have that for any epoch $\ell$ and for any item $i \in S_\ell$, the estimate $\tilde{v}_{i,\ell}$, the number of picks of item $i$ in epoch $\ell$ is geometrically distributed with success probability $1/(1+v_i)$. Therefore, if we use the distribution of $1/\mathsf{Beta}(1,1) - 1$ as the initial prior for $v_i$, and then, in the beginning of epoch $\ell$, from Lemma 7, we have that the posterior is distributed as $\frac{1}{\mathsf{Beta}(n_i(\ell), V_i(\ell))} - 1$, with $n_i(\ell)$ being the number of epochs the item $i$ has been offered before epoch $\ell$ (as part of an assortment) and $V_i(\ell)$ being the number of times it was picked by the user.

**Selection of Subset to be Offered** To choose the subset to be offered in epoch $\ell$, the algorithm samples a set of parameters $\mu_1(\ell), \ldots, \mu_N(\ell)$ independently from the current posteriors and finds the set that maximizes the expected revenue as per the sampled parameters. In particular, the set $S_\ell$ to be offered in epoch $\ell$ is chosen as

$$S_\ell := \operatorname*{argmax}_{|S|\le K} R(S, \boldsymbol{\mu}(\ell)). \tag{9.16}$$

The details of the above procedure are provided in Algorithm 2.

**Algorithm 2** A TS algorithm for MNL-Bandit with Independent Beta priors

**Initialization:** For each item $i = 1, \cdots, N$, $V_i = 1, n_i = 1$.

$t = 1$, keeps track of the time steps

$\ell = 1$, keeps count of total number of epochs

**while** $t \le T$ **do**

(a) (*Posterior Sampling*) For each item $i = 1, \cdots, N$, sample $\theta_i(\ell)$ from the $\mathsf{Beta}(n_i, V_i)$ and compute $\mu_i(\ell) = \frac{1}{\theta_i(\ell)} - 1$

(b) (*Subset Selection*) Compute $S_\ell = \operatorname*{argmax}_{|S|\le K} R(S, \boldsymbol{\mu}(\ell)) = \frac{\sum_{i\in S} r_i \mu_i(\ell)}{1+\sum_{j\in S} \mu_j(\ell)}$

(c) (*Epoch-based offering*)
**repeat**

Offer the set $S_\ell$, and observe the user choice $c_t$;

Update $\mathcal{E}_\ell = \mathcal{E}_\ell \cup t$, time indices corresponding to epoch $\ell$; $t = t + 1$

**until** $c_t = 0$ ot $t = T$

(d) (*Posterior update*)

For each item $i \in S_\ell$, compute $\tilde{v}_{i,\ell} = \sum_{t\in\mathcal{E}_\ell} \mathbb{I}(c_t = i)$, number of picks of item $i$ in epoch $\ell$.

Update $V_i = V_i + \tilde{v}_{i,\ell}$, $n_i = n_i + 1$, $\ell = \ell + 1$.

**end while**

Algorithm 2 presents some unique challenges for theoretical analysis. A worst-case regret analysis of Thompson Sampling-based algorithms for MAB typically relies on showing that the best arm is optimistic at least once every few steps, in the sense that the parameter sampled from the posterior is better than the true parameter. Due to the combinatorial nature of our problem, such a proof approach requires showing that every few steps, all the $K$ items in the optimal offer set have sampled parameters that are better than their true counterparts. However, Algorithm 2 samples the posterior distribution for each parameter *independently* in each round. This makes the probability of being optimistic exponentially small in $K$. In Sect. 9.5.3, we modify Algorithm 2 to address these challenges and in a manner amenable to theoretical analysis.

### 9.5.3 A TS Algorithm with Posterior Approximation and Correlated Sampling

In this section, we present a variant of TS with correlated sampling that achieves provably near-optimal regret bounds. We address the challenge associated with the combinatorial nature of the MNL-Bandit by employing *correlated sampling* across items. To implement correlated sampling, we find it useful to approximate the Beta posterior distribution by a Gaussian distribution with approximately the same mean and variance as the former, what was referred to in the introduction as a two-moment approximation. This allows us to generate correlated samples from the $N$ Gaussian distributions as linear transforms of a single standard Gaussian random variable. Under such correlated sampling, we can guarantee that the probability that all $K$ optimal items are simultaneously optimistic is constant, as opposed to being exponentially small (in $K$) in the case of independent sampling. However, such correlated sampling reduces the overall variance of the maximum of $N$ samples severely, thus inhibiting exploration. We "boost" the variance by taking $K$ samples instead of a single sample of the standard Gaussian. The resulting variant of Thompson Sampling, therefore, has three main modifications: posterior approximation through a Gaussian distribution, correlated sampling, and taking multiple samples (for "variance boosting"). We elaborate on each of these changes below.

**Posterior Approximation** First, we present the following result that helps us in approximating the posterior.

**Lemma 8 (Moments of the Posterior Distribution)** *If $X$ is a random variable distributed as $\mathsf{Beta}(\alpha, \beta)$, then*

$$\mathbb{E}\left(\frac{1}{X}-1\right)=\frac{\beta}{\alpha-1}, \quad \textit{and} \quad \mathsf{Var}\left(\frac{1}{X}-1\right) \;=\frac{\frac{\beta}{\alpha-1}\left(\frac{\beta}{\alpha-1}+1\right)}{\alpha-2}.$$

We approximate the posterior distributions used in Algorithm 2 for each MNL parameter $v_i$, by a Gaussian distribution with approximately the same mean and variance given in Lemma 8. In particular, let

$$\hat{v}_i(\ell) := \frac{V_i(\ell)}{n_i(\ell)}, \quad \hat{\sigma}_i(\ell) := \sqrt{\frac{50\hat{v}_i(\ell)(\hat{v}_i(\ell)+1)}{n_i(\ell)}} + 75\frac{\sqrt{\log TK}}{n_i(\ell)}, \quad \ell = 1, 2, \ldots \tag{9.17}$$

where $n_i(\ell)$ is the number of epochs item $i$ has been offered before epoch $\ell$, and $V_i(\ell)$ being the number of times it was picked by the user. We will use $\mathcal{N}\left(\hat{v}_i(\ell), \hat{\sigma}_i^2(\ell)\right)$ as the posterior distribution for item $i$ in the beginning of epoch $\ell$. The Gaussian approximation of the posterior facilitates efficient correlated sampling from posteriors that plays a key role in avoiding the theoretical challenges in analyzing Algorithm 2.

**Correlated Sampling** Given the posterior approximation by Gaussian distributions, we correlate the samples by using a common standard normal variable and constructing our posterior samples as an appropriate transform of this common standard normal. More specifically, in the beginning of an epoch $\ell$, we generate a sample from the standard normal distribution, $\theta \sim \mathcal{N}(0, 1)$, and the posterior sample for item $i$ is generated as $\hat{v}_i(\ell) + \theta\hat{\sigma}_i(\ell)$. Intuitively, this allows us to generate sample parameters for $i = 1, \ldots, N$ that are either simultaneously large or simultaneously small, thereby, boosting the probability that the sample parameters for *all* the $K$ items in the best offered set are optimistic (i.e., the sampled parameter values are higher than the true parameter values).

**Multiple ($K$) Samples** The correlated sampling decreases the joint variance of the sample set. More specifically, if $\theta_i$ were sampled independently from the standard normal distribution for every $i$, then for any epoch $\ell$, we have that

$$\mathsf{Var}\left(\max_{i=1,\cdots,N}\left\{\hat{v}_i(\ell) + \theta\hat{\sigma}_i(\ell)\right\}\right) \leq \mathsf{Var}\left(\max_{i=1,\cdots,N}\left\{\hat{v}_i(\ell) + \theta_i\hat{\sigma}_i(\ell)\right\}\right).$$

In order to boost this joint variance and ensure sufficient exploration, we modify the procedure to generate multiple sets of samples. In particular, in the beginning of an epoch $\ell$, we now generate $K$ independent samples from the standard normal distribution, $\theta^{(j)} \sim \mathcal{N}(0, 1)$, $j = 1, \ldots, K$. And then for each $j$, a sample parameter set is generated as

$$\mu_i^{(j)}(\ell) := \hat{v}_i(\ell) + \theta^{(j)}\hat{\sigma}_i(\ell), \quad i = 1, \ldots, N.$$

Then, we use the largest valued samples

$$\mu_i(\ell) := \max_{j=1,\cdots,K} \mu_i^{(j)}(\ell), \forall i,$$

**Algorithm 3** TS algorithm with Gaussian approximation and correlated sampling

Input parameters: $\alpha = 50$, $\beta = 75$
Initialization: $t = 0$, $\ell = 0$, $n_i = 0$ for all $i = 1, \cdots, N$.
**for** each item, $i = 1, \cdots, N$ **do**

Offer item $i$ to users until the user selects the "outside option". Let $\tilde{v}_{i,1}$ be the number of times item $i$ was offered. Update: $V_i = \tilde{v}_{i,1} - 1$, $t = t + \tilde{v}_{i,1}$, $\ell = \ell + 1$ and $n_i = n_i + 1$.

**end for**
**while** $t \leq T$ **do**

(a) (*Correlated Sampling*) **for** $j = 1, \cdots, K$

Sample $\theta^{(j)}(\ell)$ from the distribution $\mathcal{N}(0, 1)$ and let $\theta_{\mathsf{max}}(\ell) = \max_{j=1,\cdots,K} \theta^{(j)}(\ell)$; update $\hat{v}_i = \frac{V_i}{n_i}$.

For each item $i \leq N$, compute $\mu_i^{(j)}(\ell) = \hat{v}_i + \theta_{\mathsf{max}}(\ell) \cdot \left( \sqrt{\frac{\alpha \hat{v}_i (\hat{v}_i + 1)}{n_i}} + \frac{\beta \sqrt{\log TK}}{n_i} \right)$.

**end**

(b) *(Subset selection)* Same as step (b) of Algorithm 2.
(c) *(Epoch-based offering)* Same as step (c) of Algorithm 2.
(d) *(Posterior update)* Same as step (d) of Algorithm 2.

**end while**

to decide the assortment to offer in epoch $\ell$,

$$S_\ell := \arg\max_{S \in \mathcal{S}} \{R(S, \boldsymbol{\mu}(\ell))\}.$$

We describe the algorithmic details formally in Algorithm 3.

Intuitively, the second-moment approximation provided by Gaussian distribution and the multiple samples taken in Algorithm 3 may make the posterior converge slowly and increase exploration. However, the correlated sampling may compensate for these effects by reducing the variance of the maximum of $N$ samples and therefore reducing the overall exploration. In Sect. 9.5.5, we illustrate some of these insights through numerical simulations. Here, correlated sampling is observed to provide significant improvements as compared to independent sampling and while posterior approximation by Gaussian distribution has little impact.

### *9.5.4 Regret Analysis*

The following bound on the regret of Algorithm 3 was proven in Agrawal et al. (2017).

**Theorem 3** *For any instance $\boldsymbol{v} = (v_0, \cdots, v_N)$ of the MNL-Bandit problem with $N$ products, $r_i \in [0, 1]$, and satisfying Assumption 1, the regret of Algorithm 3 in time $T$ is bounded as*

$$\mathit{Reg}(T, \boldsymbol{v}) \leq C_1 \sqrt{NT} \log TK + C_2 N \log^2 TK,$$

*where $C_1$ and $C_2$ are absolute constants (independent of problem parameters).*

**Proof Outline**

We provide a proof sketch for Theorem 3. We break down the expression for total regret

$$\mathsf{Reg}(T, \boldsymbol{v}) := \mathbb{E}\left[\sum_{t=1}^{T} R(S^*, \boldsymbol{v}) - R(S_t, \boldsymbol{v})\right],$$

into regret per epoch, and rewrite it as follows:

$$\mathsf{Reg}(\mathsf{T}, \boldsymbol{v}) = \underbrace{\mathbb{E}\left[\sum_{\ell=1}^{L} |\mathcal{E}_\ell| \left(R(S^*, \boldsymbol{v}) - R(S_\ell, \boldsymbol{\mu}(\ell))\right)\right]}_{\mathsf{Reg}_1(\mathsf{T}, \boldsymbol{v})}$$
$$+ \underbrace{\mathbb{E}\left[\sum_{\ell=1}^{L} |\mathcal{E}_\ell| \left(R(S_\ell, \boldsymbol{\mu}(\ell)) - R(S_\ell, \boldsymbol{v})\right)\right]}_{\mathsf{Reg}_2(\mathsf{T}, \boldsymbol{v})},$$

where $|\mathcal{E}_\ell|$ is the number of periods in epoch $\ell$, and $S_\ell$ is the set repeatedly offered by our algorithm in epoch $\ell$. We bound the two terms: $\mathsf{Reg}_1(T, \boldsymbol{v})$ and $\mathsf{Reg}_2(T, \boldsymbol{v})$ separately.

Since $S_\ell$ is chosen as the optimal set for the MNL instance with parameters $\boldsymbol{\mu}(\ell)$, the first term $\mathsf{Reg}_1(T, \boldsymbol{v})$ is essentially the difference between the optimal revenue of the true instance and the optimal revenue of the sampled instance. This term contributes no regret if the revenues corresponding to the sampled instances are optimistic, i.e., if $R(S_\ell, \boldsymbol{\mu}(\ell)) \geq R(S^*, \boldsymbol{v})$. Unlike optimism under uncertainty approaches such as UCB, this property is not directly ensured by the Thompson Sampling-based algorithm. To bound this term, we utilize the anti-concentration properties of the posterior, as well as the dependence between samples for different items. In particular, we use these properties to prove that at least one of the $K$ sampled instances is optimistic "often enough."

The second term $\mathsf{Reg}_2(T, \boldsymbol{v})$ captures the difference in reward from the offered set $S_\ell$ when evaluated on sampled parameters in comparison to the true parameters. We bound this by utilizing the concentration properties of the posterior distributions.

It involves showing that for the sets that are played often, the posterior will converge quickly so that revenue on the sampled parameters will be close to that on the true parameters.

In what follows, we elaborate on the anti-concentration properties of the posterior distribution required to prove Theorem 3.

**Anti-Concentration of the Posterior Distribution** The last and important component of our analysis is showing that revenues corresponding to the sampled instances are not optimistic, i.e., if $R(S_\ell, \boldsymbol{\mu}(\ell)) < R(S^*, \boldsymbol{v})$ only in a "small number" of epochs. We utilize the anti-concentration properties of the posterior to prove that one of the $K$ sampled instances corresponds to higher expected revenue. We then leverage this result to argue that the $\mathsf{Reg}_1(T, \mathbf{v})$ is small.

We will refer to an *epoch $\ell$ as optimistic* if the expected revenue of the optimal set corresponding to the sampled parameters is higher than the expected revenue of the optimal set corresponding to true parameters, i.e., $R(S^*, \boldsymbol{\mu}(\ell)) \geq R(S^*, \boldsymbol{v})$. Any epoch that is not optimistic is referred to as a *non-optimistic epoch.* Since $S_\ell$ is an optimal set for the sampled parameters, we have $R(S_\ell, \boldsymbol{\mu}(\ell)) \geq R(S^*, \boldsymbol{\mu}(\ell))$. Hence, for any optimistic epoch $\ell$, the difference between the expected revenue of the offer set corresponding to sampled parameters $R(S_\ell, \boldsymbol{\mu}(\ell))$ and the optimal revenue $R(S^*, \boldsymbol{v})$ is bounded by zero. This suggests that as the number of optimistic epochs increases, the term $\mathsf{Reg}_1(T, \boldsymbol{v})$ decreases.

The central technical component of our analysis is showing that the regret over non-optimistic epochs is "small." More specifically, we prove that there are only a "small" number of non-optimistic epochs. From the restricted monotonicity property of the optimal revenue (see Lemma 2), we have that an epoch $\ell$ is optimistic if every sampled parameter, $\mu_i(\ell)$, is at least as high as the true parameter $v_i$ for every item $i$ in the optimal set $S^*$. Recall that each posterior sample, $\mu_i^{(j)}(\ell)$, is generated from a Gaussian distribution, whose mean concentrates around the true parameter $v_i$. We can use this observation to conclude that any sampled parameter will be greater than the true parameter with constant probability, i.e., $\mu_i^{(j)}(\ell) \geq v_i$. However, to show that an epoch is optimistic, we need to show that sampled parameters for *all* the items in $S^*$ are larger than the true parameters. This is where the correlated sampling feature of our algorithm plays a key role. We use the dependence structure between samples for different items in the optimal set and variance boosting (by a factor of $K$) to prove an upper bound of roughly $1/K$ on the number of consecutive epochs between two optimistic epochs. More specifically, we have the following result.

**Lemma 9 (Spacing of Optimistic Epochs)** *Let $\mathcal{E}^{An}(\tau)$ denote the set of consecutive epochs between an optimistic epoch $\tau$ and the subsequent optimistic epoch $\tau'$. For any $p \in [1, 2]$, we have*

$$\mathbb{E}\left[\left|\mathcal{E}^{An}(\tau)\right|^p\right] \leq \left(\frac{e^{12}}{K} + 30^{1/p}\right)^p.$$

### 9.5.5 Empirical Study

In this section, we test the various design components of the Thompson Sampling-based approach through numerical simulations. The aim is to isolate and understand the effect of individual features of our algorithm like Beta posteriors vs. Gaussian approximation, independent sampling vs. correlated sampling, and single sample vs. multiple samples, on the practical performance.

We simulate an instance of the MNL-Bandit problem with $N = 1000$, $K = 10$, and $T = 2 \times 10^5$, when the MNL parameters $\{v_i\}_{i=1,\ldots,N}$ are generated randomly from Unif$[0, 1]$. And, we compute the average regret based on 50 independent simulations over the randomly generated instance. In Fig. 9.4, we report the performance of the following different variants of TS:

- $(i)$ Algorithm 2: Thompson Sampling with independent Beta priors, as described in Algorithm 2.
- $(ii)$ $\mathsf{TS}_{\mathsf{IID\ Gauss}}$: Algorithm 2 with Gaussian posterior approximation and independent sampling. More specifically, for each epoch $\ell$ and for each item $i$, we sample a Gaussian random variable independently with the mean and variance equal to the mean and variance of the Beta prior in Algorithm 2 (see Lemma 9.17).
- $(iii)$ $\mathsf{TS}_{\mathsf{Gauss\ Corr}}$: Algorithm 3 with Gaussian posterior approximation and correlated sampling. In particular, for every epoch $\ell$, we sample a standard normal random variable. Then, for each item $i$, we obtain a corresponding sample by multiplying and adding the preceding sample with the standard deviation and mean of the Beta prior in Algorithm 2 (see Step (a) in Algorithm 3). We use the values $\alpha = \beta = 1$ for this variant of Thompson Sampling.
- $(iv)$ Algorithm 3: Algorithm 1 with Gaussian posterior approximation with correlated sampling and boosting by using multiple $(K)$ samples. This is essentially the version with all the features of Algorithm 3. We use the values $\alpha = \beta = 1$ for this variant of Thompson Sampling.

For comparison, we also present the performance of UCB approach discussed in the previous section. The performance of all the variants of TS is observed to be better than the UCB approach in our experiments, which is consistent with the other empirical evidence in the literature.

Figure 9.4 shows the performance of the TS variants. Among the TS variants, the performance of Algorithm 2, i.e., Thompson Sampling with independent Beta priors is similar to $\mathsf{TS}_{\mathsf{IID\ Gauss}}$, the version with independent Gaussian (approximate) posteriors, indicating that the effect of posterior approximation is minor. The performance of $\mathsf{TS}_{\mathsf{Gauss\ Corr}}$, where we generate correlated samples from the Gaussian distributions, is significantly better than the other variants of the algorithm. This is consistent with our remark earlier that to adapt the Thompson sampling approach of the classical MAB problem to our setting, ideally, we would like to maintain a joint prior over the parameters $\{v_i\}_{i=1,\ldots,N}$ and update it to a joint posterior using the Bandit feedback. However, since this can be quite challenging,

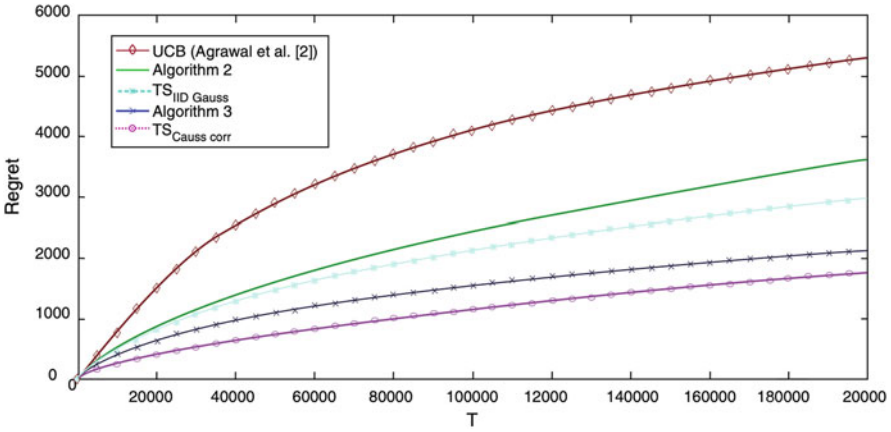


**Fig. 9.4** Regret growth with $T$ for various heuristics on a randomly generated MNL-Bandit instance with $N = 1000$, $K = 10$

and intractable in general, we use independent priors over the parameters. The superior performance of $\mathsf{TS}_{\text{Gauss Corr}}$ demonstrates the potential benefits of considering a joint (correlated) prior/posterior in settings with a combinatorial structure. Finally, we observe that the performance of Algorithm 3, where an additional "variance boosting" is provided through $K$ independent samples, is worse than $\mathsf{TS}_{\text{Gauss Corr}}$. Note that while "variance boosting" facilitates theoretical analysis, it also results in a longer exploration period explaining the observed degradation of performance in comparison to the TS variant without "variance boosting." However, Algorithm 3 performs significantly better than the independent Beta posterior version Algorithm 2. Therefore, significant improvements in performance due to the correlated sampling feature of Algorithm 3 compensate for the slight deterioration caused by boosting.

## 9.6 Lower Bound for the MNL-Bandit

In this section, we present the fundamental theoretical limits that any policy must incur a regret of $\Omega(\sqrt{NT})$. More precisely, (Chen and Wang, 2017) established the following result.

**Theorem 4 (Lower Bound on Achievable Performance (Chen and Wang, 2017))** *There exists a (randomized) instance of the MNL-Bandit problem with* $v_0 \geq v_i$ , $i = 1, \ldots, N$, *such that for any* $N$ *and* $K$, *and any policy* $\pi$ *that offers assortment* $\mathcal{S}_t^\pi$, $|S_t^\pi| \leq K$ *at time* $t$, *we have for all* $T \geq N$ *that*

$$\mathit{Reg}(T, \boldsymbol{v}) := \mathbb{E}_{\pi}\left(\sum_{t=1}^{T} R(S^*, \boldsymbol{v}) - R(S_t^{\pi}, \boldsymbol{v})\right) \geq C\sqrt{NT},$$

*where $S^*$ is (at-most) $K$-cardinality assortment with maximum expected revenue, and $C$ is an absolute constant.*

Theorem 4 is proved by a reduction to a parametric multi-armed bandit (MAB) problem, for which a lower bound is known. We refer the interested readers to Chen and Wang (2017) for a detailed proof. Note that Theorem 4 establishes that Algorithms 1 and 3 achieve near-optimal performance without any a priori knowledge of problem parameters. Furthermore, these algorithms are adaptive in the sense that their performance is near-optimal in the "well separated" case.

## 9.7 Conclusions and Recent Progress

In this chapter, we studied the dynamic assortment selection problem under the widely used multinomial logit (MNL) choice model. Formulating the problem as a parametric multi-arm bandit problem, we discussed algorithmic approaches that learn the parameters of the choice model while simultaneously maximizing the cumulative revenue. We focused on UCB and Thompson Sampling-based algorithms that are universally applicable, and whose performance (as measured by the regret) is provably nearly optimal.

However, the approaches presented here only considered the settings where every product has its own utility parameter and has to be estimated separately. Such approaches can handle only a (small) finite number of products. Many real application settings involve a large number of products essentially described by a small of features, via what is often referred to as a factor model. Recently, several works (Chen et al., 2019, 2020, 2021; Cheung and Simchi-Levi, 2017; Saha and Gopalan, 2019; Feng et al., 2018; Miao and Chao, 2021, 2019; Oh and Iyengar, 2021, 2019) have considered extensions of the approaches presented here to those more complex settings.

The works of Chen et al. (2020); Miao and Chao (2019); Oh and Iyengar (2021) consider the more general contextual variant of the MNL-Bandit problem. These papers build upon (Agrawal et al., 2016, 2019) to develop UCB-based approaches and establish worst-case regret bounds of $\tilde{O}(d\sqrt{T})$, where $d$ is the dimension of contexts, with some additional dependencies on certain problem parameters.

The works of Cheung and Simchi-Levi (2017); Miao and Chao (2021); Oh and Iyengar (2019) developed Thompson Sampling-based approaches for contextual variations of the MNL-Bandit problem. These works achieve a Bayesian regret bound of $\tilde{O}(d\sqrt{T})$ that are dependent on problem parameters. Feng et al. (2018) and Saha and Gopalan (2019) consider the best arm identification variant of the MNL-Bandit problem, where the focus is only on exploration to identify the best $K$

items. Chen et al. (2019) consider a variant of the MNL-Bandit where feedback from a small fraction of users is not consistent with the MNL choice model. They present a near-optimal algorithm with a worst-case regret bound of $\tilde{O}(\epsilon K^2 T + \sqrt{NKT})$, where $\epsilon$ is the fraction of users for whom the feedback is corrupted.

**Disclaimer** This work was done when Vashist (one of the authors) was at Columbia University.

## References


Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2016). A near-optimal exploration-exploitation approach for assortment selection. In *Proceedings of the 2016 ACM conference on economics and computation* (pp. 599–600).

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2017). Thompson sampling for the MNL-bandit. In *Conference on learning theory* (pp. 76–78). PMLR.

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2019). MNL-Bandit: A dynamic learning approach to assortment selection. *Operations Research, 67*(5), 1453–1485.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning, 47*(2), 235–256.

Avadhanula, V. (2019). *The MNL-Bandit problem: Theory and applications*. New York: Columbia University.

Avadhanula, V., Bhandari, J., Goyal, V., & Zeevi, A. (2016). On the tightness of an lP relaxation for rational optimization and its applications. *Operations Research Letters, 44*(5), 612–617.

Borovkov, AA. (1984). Mathematical statistics. (estimation of parameters, testing of hypotheses).

Ben-Akiva, M., & Lerman, S. (1985). *Discrete choice analysis: Theory and application to travel demand*. MIT Press, Cambridge.

Chen, X., & Wang, Y. (2017). A note on tight lower bound for MNL-bandit assortment selection models. arXiv preprint arXiv:170906192.

Chen, X., Krishnamurthy, A., & Wang, Y. (2019). Robust dynamic assortment optimization in the presence of outlier customers. arXiv preprint arXiv:191004183.

Chen, X., Wang, Y., & Zhou, Y. (2020). Dynamic assortment optimization with changing contextual information. *Journal of Machine Learning Research, 21*, 216–221.

Chen, X., Shi, C., Wang, Y., & Zhou, Y. (2021). Dynamic assortment planning under nested logit models. *Production and Operations Management, 30*(1), 85–102.

Cheung, W., & Simchi-Levi, D. (2017). Thompson sampling for online personalized assortment optimization problems with multinomial logit choice models. Available at SSRN 3075658.

Davis, J., Gallego, G., & Topaloglu, H. (2013). Assortment planning under the multinomial logit model with totally unimodular constraint structures. New York: Cornell University. Technical Report.

Désir, A., Goyal, V., & Zhang, J. (2021). Capacitated assortment optimization: Hardness and approximation. *Operations Research, 70*(2), 893–904.

Feldman, J., Zhang, D., Liu, X., & Zhang, N. (2021). Customer choice models versus machine learning: Finding optimal product displays on Alibaba. *Operations Research, 70*(1), 309–328.

Feng, Y., Caldentey, R., & Ryan, C. (2018). Robust learning of consumer preferences. Available at SSRN 3215614.

Greene, W. H. (2003). *Econometric analysis* (5th ed.). Prentice Hall.

Kok, A. G., & Fisher, M. L. (2007). Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research, 55*(6), 1001–1021.

Lai, T., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6*(1), 4–22.

McFadden, D., & Train, K. (2000). Mixed MNL models for discrete response. *Journal of Applied Econometrics, 15*(5), 447–470.

Miao, S., & Chao, X. (2019). Fast algorithms for online personalized assortment optimization in a big data regime. Available at SSRN 3432574.

Miao, S., & Chao, X. (2021). Dynamic joint assortment and pricing optimization with demand learning. *Manufacturing and Service Operations Management, 23*(2), 525–545.

Oh, M., & Iyengar, G. (2019). Thompson sampling for multinomial logit contextual bandits. *Advances in Neural Information Processing Systems, 32*, 3151–3161.

Oh, M., & Iyengar, G. (2021). Multinomial logit contextual bandits: Provable optimality and practicality. In *Proceedings of the AAAI conference on artificial intelligence* (vol 35, pp 9205–9213).

Rusmevichientong, P., Shen, Z. J. M., & Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research, 58*(6), 1666–1680.

Saha, A., & Gopalan, A. (2019). Regret minimisation in multinomial logit bandits. arXiv preprint arXiv:190300543v1.

Sauré, D, & Zeevi, A. (2013). Optimal dynamic assortment planning with demand learning. *Manufacturing and Service Operations Management, 15*(3), 387–404.

Talluri, K., & Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science, 50*(1), 15–33.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge Books.

Williams, H. (1977). On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and Planning A, 9*(3), 285–344.

# Chapter 10
# Dynamic Assortment Optimization: Beyond MNL Model


**Yining Wang and Yuan Zhou**

## 10.1 Overview

Dynamic assortment optimization with demand learning is a fundamental question in online data-driven revenue management research. It captures the two usually conflicting tasks in revenue management: the *learning* or *estimation* of consumers' demand behaviors, and the *efficient optimization* of assortments for maximized expected revenue.

Mathematically, the dynamic assortment optimization with demand learning question is usually formulated as follows. The retailer has in stock $N$ substitutable products and needs to offer assortments $S_1, \cdots, S_T \subseteq [N]$ to $T$ sequentially arriving customers. Since the products are substitutable, the customer arriving at time $t$ will purchase at most one product $i_t \in S_t$ (for which the retailer gains a profit $r_{i_t} > 0$) or leave without making any purchase (denoted as $i_t = 0$, for which the retailer gains nothing). The retailer needs to learn or estimate consumers' discrete choice probabilities $\mathbb{P}(\cdot|S)$, while at the same time aims at maximizing his/her expected revenue $R(S_t) = \mathbb{E}[r_{i_t}|S_t] = \sum_{i \in S_t} r_i \mathbb{P}(i|S_t)$.

To make the learning and optimization problems feasible, it is clear that structures and assumptions need to be imposed on the (family) of unknown choice models $\mathbb{P}(\cdot|S)$. In the work of Rusmevichientong et al. (2010) as well as many more follow-up results (see the previous chapter for more details), it is assumed that $\mathbb{P}(\cdot|S)$ admits the form of the *multinomial logit (MNL)* choice model:

Y. Wang (✉)
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA
e-mail: yxw220006@utdallas.edu

Y. Zhou
Yau Mathematical Sciences Center, Tsinghua University, Beijing, China
e-mail: yuan-zhou@tsinghua.edu.cn



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_10

$$\mathbb{P}(i|S) = \frac{v_i}{v_0 + \sum_{j \in S} v_j}, \qquad \forall i \in S \cup \{0\}, \tag{10.1}$$

where $v_0, v_1, \cdots, v_N > 0$ are unknown mean utility parameters.

While the MNL choice model has classical econometrical motivations (McFadden, 1973) and has amenable estimation and optimization properties, such models also exhibit several limitations and disadvantages. Below we list several well-known limitations of the MNL model in Eq. (10.1) in the context of dynamic assortment optimization:

1. It can be shown that Eq. (10.1) corresponds to consumers' randomized utilities distributed as centered, homoscedastic Gumbel random variables (see Example 1 in Sect. 10.2). Needless to say, such a model could be mis-specified in practice when consumers' utilities are not distributed as extreme-value distributions, and it is valuable to study more general distributions of stochastic utility parameters.
2. The MNL model has the property that the consumers' preferences between two products are independent from other product choices (i.e., $\mathbb{P}[i|S]/\mathbb{P}[j|S] = v_i/v_j$ is constant for *all* assortments $S$ consisting of $i, j$), known as the *independence of irrelevant alternatives (IIA)* property. The IIA property is, however, frequently violated in application scenarios (Train, 2009), calling for more sophisticated discrete choice models.
3. The MNL model in Eq. (10.1) in the context of dynamic assortment optimization essentially assumes that the $T$ sequentially arriving customers are homogeneous with shared mean utility parameters, and the products' popularity remains stationary too. In reality, however, it is very common that customer's preferences and products' popularity are non-stationary and change with time, usually together with identifiable *features*. Therefore, extending the MNL model to the non-stationary setting is of great theoretical and practical importance.

In the rest of this chapter we give an overview of existing literature on dynamic assortment optimization with demand learning beyond the MNL choice model that partially addresses the above-mentioned limitations from different perspectives. In Sect. 10.2 we present results that are applicable to general utility distributions beyond extreme-value distributions (Sauré and Zeevi, 2013). In Sect. 10.3 we present results for the *nested* multinomial logit choice model, which alleviates concerns over the IIA property of MNL (Chen et al., 2021). In Sect. 10.4 we present results on dynamic assortment optimization with non-stationary demand/choice models, such as with contextual consumer features (Cheung and Simchi-Levi, 2017; Miao and Chao, 2019; Oh and Iyengar, 2019; Chen et al., 2020). Finally, in Sect. 10.5 we conclude the chapter by discussing interesting future directions under the general theme of dynamic assortment planning beyond MNL model.

## 10.2 General Utility Distributions

It is a common practice in econometrics theory to derive discrete choice models from consumers' randomized utilities. Suppose consumers' utility for product $i$ is randomly distributed as $u_i = \mu_i + \xi_i$, where $\mu_i$ is a certain fixed mean utility parameter and $\xi_1, \cdots, \xi_N \overset{i.i.d}{\sim} F$ are i.i.d. centered random variables. Given assortment $S \subseteq [N]$, the customer would purchase product $i \in S$ with the largest $u_i$ value or leave without purchasing any product if $\max_{i \in S} u_i \leq u_0$ with $u_0 = \mu_0 + \xi_0 = \xi_0$ as $\mu_0 = 0$. Clearly, the mean utility vector $\mu = (\mu_1, \cdots, \mu_N)$ corresponds to the popularity of different substitutable products, and the probabilistic distribution $F$ dictates the form of the discrete choice model $\mathbb{P}(\cdot|S)$.

*Example 1* Suppose $F$ is the standard Gumbel distribution (i.e., $F(t) = \Pr[\xi_i \leq t] = e^{-e^{-t}}$ for all $t \in \mathbb{R}$). Then $\mathbb{P}(i|S, \mu) = v_i/(v_0 + \sum_{j \in S} v_j)$, where $v_i = e^{\mu_i}$.

The purpose of this section is to study dynamic assortment optimization with demand learning when the underlying distribution $F$ is not necessarily the Gumbel distribution.

### *10.2.1 Model Formulation and Assumptions*

With the mean utility vector $\mu \in \mathbb{R}^N$ and the utility distribution $F$ fixed, the discrete choice model $\mathbb{P}(\cdot|S)$ takes the form of

$$\mathbb{P}(i|S, \mu) = \int_{-\infty}^{\infty} \left[ \prod_{j \in S \cup \{0\} \setminus \{i\}} F(x - \mu_j) \right] \mathrm{d}F(x - \mu_i), \qquad \forall i \in S \cup \{0\}, \tag{10.2}$$

where $F(\cdot)$ is the CDF of the centered distribution $F$.

It is assumed that the retailer has full knowledge of the utility distribution $F(\cdot)$ and the profit margin parameters $\{r_i\}_{i=1}^N$ but does not know the mean utility vector $\mu$. At time $t$ a potential customer comes, the retailer offers an assortment $S_t \subseteq [N]$ subject to the capacity constraint $|S_t| \leq K \leq N$, and observes a random purchase activity $i_t \sim \mathbb{P}(\cdot|S_t, \mu)$ realized from Eq. (10.2). A retailer's policy $\pi$ is *admissible* if at every time period $t$, the (potentially random) assortment $S_t$ is measurable with respect to the filtration of $\{S_\tau, i_\tau\}_{\tau < t}$ from previous time periods. Let $\mathcal{P}$ denote the class of all admissible policies. The objective is to design an admissible policy $\pi \in \mathcal{P}$ that achieves a high expected cumulative revenue

$$J^\pi(T, \mu) := \mathbb{E}^\pi \left[ \sum_{t=1}^{T} R(S_t, \mu) \right] \qquad \text{where} \quad R(S, \mu) := \sum_{i \in S} r_i \mathbb{P}(i|S, \mu).$$

To understand and analyze the performance $J^\pi(T, \mu)$ in a relative manner, it is instructive to compare $J^\pi(T, \mu)$ with the expected revenue of a simpler, clairvoyant optimal policy. Let $S^*$ be the "optimal assortment" defined as

$$S^* := \arg\max_{|S| \le K} R(S, \mu).$$

In the rest of the section we will also write $S^* = S^*(\mu)$ to emphasize that $S^*$ depends on the mean utility vector $\mu \in \mathbb{R}^N$. The optimal reward benchmark $J^*(T, \mu)$ is defined as

$$J^*(T, \mu) := T \times R(S^*(\mu), \mu).$$

Clearly, $J^\pi(T, \mu) \le J^*(T, \mu)$ for all admissible policy $\pi \in \mathcal{P}$. It is thus beneficial to study the *competitive ratio* between $J^\pi(T, \mu)$ and $J^*(T, \mu)$, defined as

$$\mathcal{R}^\pi(T, \mu) := 1 - \frac{J^\pi(T, \mu)}{J^*(T, \mu)}. \tag{10.3}$$

The competitive ratio $\mathcal{R}^\pi(T, \mu)$ is always between 0 and 1, and the larger $\mathcal{R}^\pi(T, \mu)$, the better $\pi$ performs.

To ensure identifiability, throughout this section we impose the following assumption on the model $F$ and the unknown mean utility vector $\mu$.

**Assumption (identifiability of general choice models)**

For any vector $\rho \in \mathbb{R}_+^N$ such that $\sum_{i=1}^N \rho_i < 1$, there exists a unique vector $\eta(\rho) \in \mathbb{R}^N$ such that $\mathbb{P}(i|[N], \eta(\rho)) = \rho_i$ for all $i \in [N]$. Additionally, $\mathbb{P}(i|[N], \cdot)$ is Lipschitz continuous for all $i$, and $[\eta(\cdot)]_i$ is Lipschitz continuous in the neighborhood of $\rho$, when $\rho_i > 0$.

Intuitively, the above assumption asserts that for any marginal distribution $\mathbb{P}(\cdot|[N]) \equiv \rho$, there exists a unique parameterization $\eta(\rho)$ that delivers such a marginal distribution, with the parameterization map $\eta(\cdot)$ further satisfying certain Lipschitz continuity conditions. This ensures that the underlying mean utility parameter $\mu = \eta(\rho)$ is identifiable and estimable from empirical observations of consumers' purchase decisions.

### *10.2.2 Algorithm Design*

The work of Sauré and Zeevi (2013) proposed two policies, one simpler and the other more sophisticated but with better regret guarantees. We will introduce both policies here and explain their design motivations.

**Algorithm 1** $\pi_1 = \pi_1(\kappa_1, T, K)$: separate exploration and exploitation

*Exploration*. Offer each assortment $A_j \in \mathcal{A}$ to $\lceil \kappa_1 \ln T \rceil$ customers;
*Estimation*. Compute estimates $\widehat{\mu}$ of $\mu$ using maximum-likelihood estimation;
*Exploitation*. Offer $S^*(\widehat{\mu})$ for the rest of the customers.

Because of the capacity constraint $|S_t| \leq K \leq N$, the designed policy could not offer all products at once in a single assortment. Therefore, the entire set of $N$ products is partitioned into $\lceil N/K \rceil$ "test assortments" as

$$\mathcal{A} = \{A_1, \cdots, A_{\lceil N/K \rceil}\} \qquad \text{where} \quad A_j = \{(j-1)K + 1, \cdots, \min(jK, N)\}.$$

Algorithm 1 gives the pseudocode of the first policy.

At a higher level, Algorithm 1 uses the strategy of *separated* exploration and exploitation, by first exploring all test assortments $A_1, \cdots, A_{\lceil N/K \rceil} \in \mathcal{A}$ each for $\lceil \kappa_1 \ln T \rceil$ times to obtain mean utility estimate $\widehat{\mu}$, and then committing to (exploiting) the near-optimal assortment $S^*(\widehat{\mu})$ calculated using the utility estimates $\widehat{\mu}$. The estimate $\widehat{\mu}$ could be obtained via the classical maximum-likelihood estimation (MLE) approach, see, e.g., Daganzo (2014, pp. 118). The algorithm parameter $\kappa_1 > 0$ characterizes the length of the exploration phase and needs to be set appropriately: too small $\kappa_1$ results in insufficient exploration and subsequently inaccurate utility estimate $\widehat{\mu}$ and worse exploitation assortment $S^*(\widehat{\mu})$, while, on the other hand, a $\kappa_1$ value too large would lead to large regret from the exploration phase. The next section gives detailed theoretical and practical choices of the $\kappa_1$ value in Algorithm $\pi_1(\kappa_1, T, K)$.

The first policy $\pi_1$ completely separates exploration and exploitation of assortments, which is less ideal. It is possible to design a more refined policy that jointly combine exploration and exploitation phases, which also attains lower overall regret. To introduce the refined policy we need to define some notations. Define

$$\overline{\mathcal{N}}(\mu) := \{j \in [N] : \exists \gamma \in \mathbb{R}^N, \gamma_i = \mu_i \forall i \in S^*(\mu), \text{ such that } j \in S^*(\gamma)\} \tag{10.4}$$

as the set of *potentially optimal* products under $\mu$. Intuitively, $j \in [N]$ is potentially optimal if it is possible to change the mean utility parameters of products not belonging to $S^*(\mu)$ so that $j$ becomes optimal. Clearly, $S^*(\mu) \in \overline{\mathcal{N}}(\mu)$ by definition but $\overline{\mathcal{N}}(\mu)$ could contain products other than $S^*(\mu)$. One can similarly define

$$\underline{\mathcal{N}}(\mu) := [N] \backslash \overline{\mathcal{N}}(\mu) \tag{10.5}$$

as the set of *strictly sub-optimal* products. The design of the second improved policy $\pi_2$ is motivated from the following observation:

**Proposition 1** *For any $\mu \in \mathbb{R}^N$, there exists $\omega(\mu) \leq R(S^*(\mu), \mu)$ such that $\underline{\mathcal{N}}(\mu) = \{i \in [N] : r_i < \omega(\mu)\}$.*

**Algorithm 2** $\pi_2 = \pi_2(\kappa_2, \omega(\cdot), T, K)$: joint exploration and exploitation

*Initialization*. Offer each $B_j \in \mathcal{B}$ to a single customer.
**for** each remaining customer $t$ **do**
  Compute estimate $\widehat{\mu}_t = (\widehat{\mu}_{t1}, \cdots, \widehat{\mu}_{tN})$ and $\omega_t = \omega(\widehat{\mu}_t)$;
  Compute $\mathcal{B}_t = \{B_j \in \mathcal{B} : \max\{r_i : i \in B_j\} \geq \omega_t\}$;
  **if** there exists $B_j \in \mathcal{B}_t$ that has been offered to fewer than $\kappa_2 \ln t$ customers **then**
    Offer assortment $B_j$ to customer $t$;
  **else**
    Offer assortment $S^*(\widehat{\mu}_t)$ to customer $t$;
  **end if**
**end for**

It is in general a difficult question to calculate or analyze the threshold function $\omega(\mu)$. In some special cases, however, $\omega(\cdot)$ takes a simpler form. For example, if $F(\cdot)$ is the standard Gumbel distribution (corresponding to the MNL model), then setting $\omega(\mu) := R(S^*(\mu), \mu)$ would satisfy Proposition 1 (Sauré and Zeevi, 2013, Sec. 5.3).

Algorithm 2 gives a complete pseudocode description of the improved policy $\pi_2 = \pi_2(\kappa_2, \omega(\cdot), T, K)$. Note that this policy uses a different test assortment structure from $\pi_1$, defined as

$$\mathcal{B} = \{B_1, \cdots, B_{\lceil N/K \rceil}\} \qquad \text{where } B_j = \{i_{(j-1)K+1}, \cdots, i_{\min(jK,T)}\},$$

where $i_\ell$ is the product with the $\ell$th largest value of $r_i$.

The second policy $\pi_2$ jointly combines exploration and exploitation, by adaptively removing assortments containing strictly sub-optimal products. More specifically, the policy maintains "active" assortment subsets $\mathcal{B}_t$, which could be much smaller than the entire set of test assortment $\mathcal{B}$ if many products have low profit margins $r_i$ which are removed by the test $\max\{r_i : i \in B_j\} \geq \omega_t$. This has the potential of greatly lowering the cumulative regret of the policy, as we shall see in more detail in the next section.

### *10.2.3 Theoretical Analysis*

In the first part of the theoretical analysis we state and discuss (asymptotic) regret upper bounds of the two policies $\pi_1$ and $\pi_2$. We first state the regret upper bound of the first policy $\pi_1$.

**Regret upper bound of policy $\pi_1$**

**Theorem 1** *For any $\mu \in \mathbb{R}^N$, there exists a constant $C_1 < \infty$ independent of $N$ and $T$, such that if policy $\pi_1 = \pi_1(\kappa_1, T, K)$ is executed with parameter $\kappa_1 \geq C_1$, then*

$$\limsup_{T\to\infty} \frac{\mathcal{R}^{\pi_1}(T,\mu)}{\ln T} \le \frac{\kappa_1 N}{K},$$

*where* $\mathcal{R}^{\pi}(T,\mu)$ *is defined in Eq. (10.3).*

While the above result does not hint on how $\kappa_1 \ge C_1$ could be chosen, in practice it suffices to use asymptotically larger exploration phases (e.g., $|\mathcal{A}|\kappa_1(\ln T)^{1+a}$ instead of $|\mathcal{A}|\kappa_1 \ln T$ for some small $a > 0$), to achieve $\limsup_{T\to\infty} \frac{\mathcal{R}^{\pi_1}(T,\mu)}{\ln^{1+a} T} \le \frac{\kappa_1 N}{K}$.

We next state the regret upper bound for the improved policy $\pi_2$.

**Regret upper bound of policy $\pi_2$**

**Theorem 2** *For any* $\mu \in \mathbb{R}^N$*, there exists a constant* $C_2 < \infty$ *independent of* $N$ *and* $T$*, such that if policy* $\pi_2 = \pi_2(\kappa_2, \omega, T, K)$ *is executed with parameter* $\kappa_2 \ge C_2$ *and* $\omega(\cdot)$ *satisfying Proposition 1, then*

$$\limsup_{T\to\infty} \frac{\mathcal{R}^{\pi_2}(T,\mu)}{\ln T} \le \frac{\kappa_2 |\overline{\mathcal{N}}(\mu)|}{K}.$$

Note that $|\overline{\mathcal{N}}(\mu)| \le N$ always holds since $\overline{\mathcal{N}}(\mu) \subseteq [N]$. This implies that the regret of $\pi_2$ is asymptotically lower than $\pi_1$, especially in the case when the majority of the products have low profit margins and are, therefore, strictly sub-optimal as defined in Eqs. (10.4) and (10.5). This shows the advantage of joint exploration and exploitation.

Finally, we give an information-theoretical lower bound on the fundamental limit of regret attainable by *any* good policies. To formally state the lower bound we need some notations. We say an admissible policy $\pi \in \mathcal{P}$ is *consistent* if for all $a > 0$ and $\mu \in \mathbb{R}^N$ it holds that

$$\lim_{T\to\infty} \frac{\mathcal{R}^{\pi}(T,\mu)}{T^a} = 0.$$

We also define the set of potentially optimal products *unilateral* utility changes as

$$\widetilde{\mathcal{N}}(\mu) := \{i \in [N] : \exists \gamma = (\mu_1, \cdots, \mu_{i-1}, v, \mu_{i+1}, \cdots, \mu_N) \text{ for } v \in \mathbb{R} \text{ such that } i \in S^*(\gamma)\}.$$

Comparing the definition of $\widetilde{\mathcal{N}}(\mu)$ with one of the $\overline{\mathcal{N}}(\mu)$ in Eq. (10.4), $i \in \widetilde{\mathcal{N}}(\mu)\backslash S^*(\mu)$ is only allowed to change the utility parameter of product $i$, which is much more restrictive than $i \in \overline{\mathcal{N}}(\mu)$ which is allowed to change the utility parameter of any product not in $S^*(\mu)$. Hence, it holds by definition that

$\widetilde{\mathcal{N}}(\mu)\backslash S^*(\mu) \subseteq \overline{\mathcal{N}}(\mu)$. The following regret lower bound can then be established, by using change-of-measure tools from the seminal work of Lai and Robbins (1985).

**Regret lower bound of all consistent policies**

**Theorem 3** *Let $\mathcal{P}^c \subseteq \mathcal{P}$ denote the class of all consistent admissible policies. Then for $\mu \in \mathbb{R}^N$, there exists a constant $C_3 > 0$ independent of $N$ and $T$, such that*

$$\inf_{\pi \in \mathcal{P}^c} \liminf_{T\to\infty} \frac{\mathcal{R}^\pi(T,\mu)}{\ln T} \geq \frac{C_3|\widetilde{\mathcal{N}}(\mu)\backslash S^*(\mu)|}{K}.$$

### *10.2.4 Bibliographic Notes and Discussion of Future Directions*

The majority of the results in this section was developed in the work of Sauré and Zeevi (2013), which is also built upon the earlier work of Rusmevichientong et al. (2010) focused exclusively on MNL choice models. Sauré and Zeevi (2013) also include the design and analysis of a policy for the special case of the MNL choice model, with more practical algorithmic designs and improved theoretical results. The general model for assortment selection (or combinatorial bandit) has also been studied in the computer science literature, most often under a much more informative *semi-bandit* feedback model when unbiased utility observations of *all* products in an offered assortment are available (Chen et al., 2013, 2016). Such feedback models are less relevant in online revenue management questions.

It is worth pointing out that both the works of Sauré and Zeevi (2013); Rusmevichientong et al. (2010) adopt the *pointwise* asymptotic analytical framework, in the sense that all analytical constants ($C_1$, $C_2$, $C_3$ in the previous section) depend on the underlying mean utility vector $\mu \in \mathbb{R}^N$. This is in contrast to the *minimax* asymptotic analytical framework adopted in many recent works (Agrawal et al., 2019, 2017; Chen and Wang, 2018; Chen et al., 2018, 2021), which generally exhibit $\widetilde{O}(\sqrt{NT})$ type regret. This motivates the following research question for future studies:

**? Minimax regret under general choice models**

For a general choice model induced by a distribution $F$, can we design an admissible policy $\pi$ such that, for a reasonable compact $\Theta \subseteq \mathbb{R}^N$ and a finite exponent $a < \infty$, there exists a constant $C < \infty$ such that

$$\limsup_{T\to\infty} \sup_{\mu\in\Theta} \frac{\mathcal{R}^\pi(T,\mu)}{\sqrt{NT}\ln^a(NT)} \leq C?$$

The $\sqrt{NT}\ln^a(NT)$ asymptotic regret rate is motivated by the results from Agrawal et al. (2019). A positive answer to the above question, however, is likely to require new ideas and insights from both the works of Sauré and Zeevi (2013) and Agrawal et al. (2019), as explore-then-commit type policies are in general sub-optimal in the minimax sense (Bubeck and Cesa-Bianchi, 2012), and epoch-based approaches adopted in Agrawal et al. (2019, 2017) are unlikely to succeed when the underlying choice model is not MNL. Specifically, it is of great interest to see whether the $C$ constant in the above question contains any polynomial $K$ factors when the underlying choice model no longer satisfies the IIA property.

## 10.3 Nested Logit Models

The nested logit model is another popular form to generalize the MNL model. It models a customer's choice in a hierarchical way: a customer first selects a category of products and iteratively proceeds to select sub-categories under the current category (or sub-category), until the current category (or sub-category) only contains products and a product is selected. The categories, sub-categories, and products form a tree structure, where the leaf nodes correspond to products and the internal nodes correspond to categories and sub-categories, which are also known as the *nests*. The nested logit model is considered as "the most widely used member of the GEV (generalized extreme value) family" and "has been applied by many researchers in a variety of situations" (see Chapter 4 from Train (2009)). The model also relaxes the IIA assumption on alternatives in different nests and thus provides a richer set of substitution patterns. In this section, we will detail the recent research progress on the dynamic assortment planning question under the two-level nested logit model, where the depth of the corresponding tree structure is 2. We will discuss both algorithmic results and lower bounds. Due to the complicated structure of the nested logit models, the problem on general nested model remains widely open and will be discussed in Sect. 10.3.5.

### *10.3.1 Model Formulation and Assumptions*

In a two-level nested logit model, the customer first selects a nest among the $M$ nests, and then chooses a product in the selected nest. We use $[M] = \{1, 2, \cdots, M\}$ to label the $M$ nests. For each nest $i \in [M]$, we label the products in nest $i$ by $[N_i] = \{1, 2, \cdots, N_i\}$. Each product $j \in [N_i]$ is associated with a *known* revenue parameter $r_{ij}$ and an *unknown* mean utility parameter $v_{ij}$. We assume each nest has an equal number of products, i.e., $N_1 = \cdots = N_M = N$. Further, let $\{\gamma_i\}_{i\in[M]} \subseteq [0, 1]$ be a collection of *unknown* correlation parameters for different nests. Each

parameter $\gamma_i$ is a measure of the degree of independence among the products in nest $i$: a larger value of $\gamma_i$ indicates less correlation.

At each time period $t \in \{1, 2, \cdots, T\}$, the retailer offers the arriving customer an assortment $S_i^{(t)} \in \mathbb{S}_i = 2^{[N]}$ for every nest $i \in [M]$, conveniently denoted as $\mathbf{S}^{(t)} = (S_1^{(t)}, \cdots, S_M^{(t)})$. The retailer then observes a nest-level purchase option $i_t \in [M] \cup \{0\}$. If $i_t \in [M]$, a product $j_t \in [N]$ is purchased within the nest $i_t$. On the other hand, $i_t = 0$ means no purchase occurs at time $t$. The probabilistic model for the purchasing option $(i_t, j_t)$ can be formulated as below:

$$\Pr\left[i_t = i|\mathbf{S}^{(t)}\right] = \frac{V_i(S_i^{(t)})^{\gamma_i}}{V_0 + \sum_{i'=1}^{M} V_{i'}(S_{i'}^{(t)})^{\gamma_{i'}}}, \qquad \forall i \in [M] \cup \{0\}, \tag{10.6}$$

$$\Pr\left[j_t = j|i_t = i, \mathbf{S}^{(t)}\right] = \frac{v_{ij}}{\sum_{j' \in S_i^{(t)}} v_{ij'}}, \qquad \forall j \in S_i^{(t)}, \tag{10.7}$$

where $V_0 = 1$ and $V_i(S_i^{(t)}) = \sum_{j \in S_i^{(t)}} v_{ij}$. Note that when $\gamma_i = 1$ for all $i \in [M]$, the nested logit model reduces to the standard MNL model.

The retailer then collects revenue $r_{i_t, j_t}$ provided that $i_t \neq 0$. The expected revenue $R(\mathbf{S}^{(t)})$ given the assortment combination $\mathbf{S}^{(t)}$ can then be written as

$$\begin{aligned} R(\mathbf{S}^{(t)}) &= \sum_{i=1}^{M} \Pr\left[i_t = i|\mathbf{S}^{(t)}\right] \sum_{j \in S_i^{(t)}} r_{ij} \Pr\left[j_t = j|i_t = i, \mathbf{S}^{(t)}\right] \\ &= \frac{\sum_{i=1}^{M} R_i(S_i^{(t)}) V_i(S_i^{(t)})^{\gamma_i}}{1 + \sum_{i=1}^{M} V_i(S_i^{(t)})^{\gamma_i}}, \quad \text{where} \quad R_i(S_i^{(t)}) = \frac{\sum_{j \in S_i^{(t)}} r_{ij} v_{ij}}{\sum_{j \in S_i^{(t)}} v_{ij}}. \end{aligned} \tag{10.8}$$

Let $\boldsymbol{\psi} = \{r_{ij}, v_{ij}, \gamma_i\}_{i,j=1}^{M,N}$ denote all model parameters. We shall also write $R(\mathbf{S}, \boldsymbol{\psi})$ when we would like to emphasize that the expected revenue of an assortment combination $\mathbf{S}$ depends on the underlying model parameter $\boldsymbol{\psi}$. The objective of the seller is to design an admissible policy $\pi \in \mathcal{P}$ so as to minimize *expected (accumulated) regret*, defined as

$$\mathcal{R}^{\pi}(T, \boldsymbol{\psi}) := \sum_{t=1}^{T} \max_{\mathbf{S} \in \mathbb{S}} R(\mathbf{S}, \boldsymbol{\psi}) - \mathbb{E}^{\pi}\left[R(\mathbf{S}^{(t)}, \boldsymbol{\pi})\right]. \tag{10.9}$$

It is easy to verify that $\mathcal{R}^{\pi}(T, \boldsymbol{\pi})$ is always non-negative, and the smaller the regret, the better the performance of the policy $\pi$ is.

Throughout this section, we make the following boundedness assumptions on revenue and utility parameters:

**Boundedness assumptions on model parameters**

1. The revenue parameters satisfy $0 \le r_{ij} \le 1$ for all $i \in [M]$ and $j \in [N]$.
2. The utility parameters satisfy $0 < v_{ij} \le C_V$ for all $i \in [M]$ and $j \in [N]$ with some constant $C_V \ge 1$.

Note that both assumptions can be regarded as without loss of generality as the parameter values could be normalized.

### *10.3.2 Assortment Space Reductions*

For nested logit models, the complete assortment selection space (a.k.a. action space) $\mathbb{S} = \mathbb{S}_1 \times \mathbb{S}_2 \times \cdots \times \mathbb{S}_M$ is extremely large, consisting of an exponential number of candidate assortment selections (on the order of $(2^N)^M$). Existing bandit learning approaches treating each assortment set in $\mathbb{S}$ independently would easily incur a regret also exponentially large. To address this challenge, the work of Chen et al. (2021) proposed to leverage the structure of optimal $\mathbf{S}$ to reduce the number of candidate assortment sets in $\mathbb{S}$, which will be detailed as follows.

To introduce the structural property of the optimal $\mathbf{S}$, we consider the *level sets* $\mathcal{L}_i(\theta_i) := \{j \in [N] : r_{ij} \ge \theta_i\}$ for each nest $i$. In other words, $\mathcal{L}_i(\theta_i)$ is the set of products in nest $i$ with revenue larger than or equal to a given threshold $\theta_i \ge 0$. Define $\mathbb{P}_i := \{\mathcal{L}_i(\theta_i) : \theta_i \ge 0\} \subseteq \mathbb{S}_i$ to be all the possible level sets of $\mathbb{S}_i$ and let

$$\mathbb{P} := \mathbb{P}_1 \times \mathbb{P}_2 \times \cdots \times \mathbb{P}_M \subseteq \mathbb{S}. \tag{10.10}$$

The following lemma formally states the structural property of the optimal $\mathbf{S}$. It shows that one can restrict the assortment selections to $\mathbb{P}$ without loss of any optimality in terms of expected revenue.

**Lemma 1 (Davis et al. (2014); Li et al. (2015))** *There exists level set threshold parameters* $(\theta_1^*, \ldots, \theta_M^*)$ *and* $\mathbf{S}^* = (\mathcal{L}_1(\theta_1^*), \cdots, \mathcal{L}_M(\theta_M^*)) \in \mathbb{P}$ *such that* $R(\mathbf{S}^*, \boldsymbol{\psi}) = \max_{\mathbf{S} \in \mathbb{S}} R(\mathbf{S}, \boldsymbol{\psi})$.

The lemma shows that the optimal assortments are "revenue-ordered" within each nest. Compared to the original action space $\mathbb{S}$, the reduced "level set" space $\mathbb{P}$ is much smaller, with each $\mathbb{P}_i$ consisting of $N$ instead of $2^N$ candidate assortments.

With Lemma 1, an assortment combination $\mathbf{S} = (S_1, \cdots, S_M) \in \mathbb{P}$ can then be parameterized without loss of optimality by a vector $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_M) \in ([0, 1] \cup \{\infty\})^M$, such that $\mathbf{S}(\boldsymbol{\theta}) = (\mathcal{L}_1(\theta_1), \cdots, \mathcal{L}_M(\theta_M))$. Note that $\mathcal{L}_i(\infty) = \emptyset$ indicates the empty set for nest $i$. Denote $\mathcal{K}_i = [0, 1] \cup \{\infty\}$, and for any $i \in [M]$, $\theta_i \in \mathcal{K}_i$ define

$$u_{i,\theta_i} := V_i(\mathcal{L}_i(\theta_i))^{\gamma_i} \quad \text{and} \quad \phi_{i,\theta_i} := R_i(\mathcal{L}_i(\theta_i)), \tag{10.11}$$

where $V_i(\cdot)$ and $R_i(\cdot)$ are nest-level utility parameter and expected revenue associated with the level set $\mathcal{L}_i(\theta_i)$ (see definitions of $V_i$ and $R_i$ in Eqs. (10.6) and (10.8), respectively). By the boundedness assumptions, it is easy to verify that $\phi_{i,\theta_i} \in [0, 1]$ and $u_{i,\theta_i} \in [0, (NC_V)^{\gamma_i}] \subseteq [0, NC_V]$ for all $i \in [M]$ and $\theta_i \in \mathcal{K}_i$. Furthermore, because each nest consists of at most $N$ products, the sets $\mathcal{K}_i$ can be made *finite* by considering only levels $\theta_i$ corresponding to revenue parameters of the $N$ products. Finally, using elementary algebra, the expected revenue $R(\mathbf{S}(\boldsymbol{\theta}), \boldsymbol{\psi})$ can be written as

$$R(\mathbf{S}(\boldsymbol{\theta}), \boldsymbol{\psi}) = \frac{\sum_{i=1}^{M} \phi_{i,\theta_i} u_{i,\theta_i}}{1 + \sum_{i=1}^{M} u_{i,\theta_i}} =: R'(\boldsymbol{\theta}, \boldsymbol{\psi}).$$

Consequently, the question of learning and optimizing $\max_{\mathbf{S}\in\mathbb{S}} R(\mathbf{S}, \boldsymbol{\psi})$ can be reduced to learning and optimizing $\max_{\boldsymbol{\theta}\in\mathcal{K}_1\times\cdots\times\mathcal{K}_M} R'(\boldsymbol{\theta}, \boldsymbol{\psi})$, which is much easier and simpler both statistically and computationally. We will introduce a UCB-based policy and its analysis to accomplish precisely the question of learning and maximizing $R'(\boldsymbol{\theta}, \boldsymbol{\psi})$ in the next section.

### *10.3.3 Algorithm Design and Regret Analysis*

We now introduce the dynamic planning policy for the two-level nested logit model using the upper confidence bound (UCB) approach. The policy was proposed in Chen et al. (2021) and leverages the level set space reduction results introduced in the previous subsection. The detailed pseudocode of the policy is given in Algorithm 3. The policy is titled $\pi^{N-UCB}$ in this section with $N$ standing for "Nested."

The high-level idea behind Algorithm 3 is as follows: for every nest $i$ and level set $\theta \in \mathcal{K}_i$, a pair of upper confidence estimates $\overline{\phi}_{i,\theta}$ and $\overline{u}_{i,\theta}$ are constructed and maintained, respectively, estimating following the nest-level revenue and utility parameters $\phi_{i,\theta}$ and $u_{i,\theta}$ defined in Eq. (10.11). For every potential customer, an optimal assortment combination based on current (upper) parameter estimates $\overline{\phi}_{i,\theta}, \overline{u}_{i,\theta}$ is computed, which is then offered to the customers repetitively until a no-purchase action occurs. All the time steps during this repetitive offering period constitute an *epoch*, and the step of time steps in the $\tau$-th epoch is denoted by $\mathcal{E}_\tau$. After an epoch has finished, the parameter estimates $\overline{\phi}_{i,\theta}, \overline{u}_{i,\theta}$ are updated for all assortments provided in each nest, and the dynamic assortment planning procedure continues until a total of $T$ customers are served.

Below we give more detailed explanations for the key lines in Algorithm 3. First, in the assortment combination $\boldsymbol{\theta}^\tau = \boldsymbol{\theta}$ for the current epoch $\tau$ is computed at Line 3. We note that the optimization task at this line is an instance of the fractional programming problems (Megiddo, 1978) and can be solved efficiently by a binary search method. For more details about the binary search algorithm, interested

**Algorithm 3** Policy $\pi^{N-UCB}(\mathcal{K}_1, \cdots, \mathcal{K}_M, C_V, T)$ for nested logit model

1: Initialization: $\tau = 1$, $\{\mathcal{E}_\tau\}_{\tau=1}^{\infty} = \emptyset$, $t = 1$; for every $i \in [M]$ and $\theta \in \mathcal{K}_i$, set $\mathcal{T}(i,\theta) = \emptyset$, $T(i,\theta) = 0$, $\widehat{\phi}_{i,\theta} = \overline{\phi}_{i,\theta} = 1$, $\widehat{u}_{i,\theta} = \overline{u}_{i,\theta} = U$; for all $i \in [M]$ and $\theta \in \mathcal{K}_i$ corresponding to the empty assortment (i.e., $\mathcal{L}_i(\theta) = \emptyset$), set $\overline{\phi}_{i,\theta} = \phi_{i,\theta} = \overline{u}_{i,\theta} = u_{i,\theta} = 0$;
2: **while** $t \le T$ **do**
3: Find $\widehat{\boldsymbol{\theta}}^{(\tau)} = \widehat{\boldsymbol{\theta}} \leftarrow \arg\max_{\boldsymbol{\theta} \in \mathcal{K}_1 \times \cdots \times \mathcal{K}_M} \overline{R}'(\boldsymbol{\theta})$, where $\overline{R}'(\boldsymbol{\theta}) = \frac{\sum_{i=1}^{M} \overline{\phi}_{i,\theta_i} \overline{u}_{i,\theta_i}}{1+\sum_{i=1}^{M} \overline{u}_{i,\theta_i}}$;
4: **repeat**
5: Pick $\boldsymbol{\theta}^{(t)} = \widehat{\boldsymbol{\theta}}$ and observe $i_t, r_t$;
6: Update $\mathcal{E}_\tau \leftarrow \mathcal{E}_\tau \cup \{t\}$, $t \leftarrow t+1$;
7: **until** $i_{t-1} = 0$ or $t > T$;
8: **for** each $i \in [M]$ with $\mathcal{L}_i(\widehat{\theta}_i) \neq \emptyset$ **do**
9: Compute $\widehat{n}_{i,\tau} = \sum_{t' \in \mathcal{E}_\tau} \mathbf{1}\{i_{t'} = i\}$ and $\widehat{r}_{i,\tau} = \sum_{t' \in \mathcal{E}_\tau} r_{t'} \mathbf{1}\{i_{t'} = i\}$;
10: Let $\theta = \widehat{\theta}_i$ (for notational simplicity);
11: Update: $\mathcal{T}(i,\theta) \leftarrow \mathcal{T}(i,\theta) \cup \{\tau\}$, $T(i,\theta) \leftarrow T(i,\theta) + 1$;
12: Update the utility and mean revenue estimates and their associated confidence bounds:

$$\widehat{u}_{i,\theta} = \tfrac{1}{T(i,\theta)} \textstyle\sum_{\tau' \in \mathcal{T}(i,\theta)} \widehat{n}_{i,\tau'}, \quad \widehat{\phi}_{i,\theta} = \frac{\sum_{\tau' \in \mathcal{T}(i,\theta)} \widehat{r}_{i,\tau'}}{\sum_{\tau' \in \mathcal{T}(i,\theta)} \widehat{n}_{i,\tau'}};$$

13: **if** $T(i,\theta) \ge 96 \ln(2MTK)$ **then**
14: $\overline{u}_{i,\theta} = \min\{U, \widehat{u}_{i,\theta} + \sqrt{\frac{96 \max(\widehat{u}_{i,\theta}, \widehat{u}_{i,\theta}^2) \ln(2MTK)}{T(i,\theta)}} + \frac{144 \ln(2MTK)}{T(i,\theta)}\}$,
15: $\overline{\phi}_{i,\theta} = \min\{1, \widehat{\phi}_{i,\theta} + \sqrt{\frac{\ln(2MTK)}{T(i,\theta)\widehat{u}_{i,\theta}}}\}$;
16: **else**
17: $\overline{u}_{i,\theta} = U$, $\overline{\phi}_{i,\theta} = 1$;
18: **end if**
19: **end for**
20: $\tau \leftarrow \tau + 1$;
21: **end while**

readers may refer to Chen et al. (2021); similar approach was also introduced in Rusmevichientong et al. (2010) for the dynamic assortment optimization under the MNL model.

At Line 4–7 in Algorithm 3, the same assortment combination $\boldsymbol{\theta}$ is offered until the no-purchase action is observed (or the time horizon has reached). And during this iteration, the $\tau$-th epoch $\mathcal{E}_\tau$ is constructed. We further explain a few additional notations: we use $\mathcal{T}(i,\theta)$ to denote the indices of epochs in which $\theta \in \mathcal{K}_i$ is supplied in nest $i$; and use $T(i,\theta) = |\mathcal{T}(i,\theta)|$ to denote the cardinality of $\mathcal{T}(i,\theta)$. We also use $\widehat{n}_{i,\tau}$ to denote the number of iterations in the epoch $\tau$ (i.e., $\mathcal{E}_\tau$) in which a product in nest $i$ is purchased; and use $\widehat{r}_{i,\tau}$ to denote the total revenue collected for all iterations in $\mathcal{E}_\tau$ in which a product in nest $i$ is purchased.

We remark that the epoch-based strategy (i.e., offering the same assortment until no purchase is observed) in Algorithm 3 was first introduced by Agrawal et al. (2019) for the dynamic assortment planning problem under the MNL model. Such an epoch-based strategy is motivated by the observation that the observations $\widehat{n}_{i,\tau}$ and $\widehat{r}_{i,\tau}$ are *unbiased* statistics of certain model parameters, or more specifically $\mathbb{E}[\widehat{n}_{i,\tau}] = u_{i,\widehat{\theta}_i}$ and $\mathbb{E}[\widehat{r}_{i,\tau} | \widehat{n}_{i,\tau}] = \widehat{n}_{i,\tau} \phi_{i,\widehat{\theta}_i}$ (see, e.g., Chen et al. (2021, Lemma

2)), which enables construction of upper confidence bounds using concentration inequalities as the observations $\{\widehat{n}_{i,\tau}, \widehat{r}_{i,\tau}\}$ are unbiased and independent across epochs.

Below we state the main regret theorem for Algorithm 3.

**Regret upper bound of policy $\pi^{N-UCB}$**

**Theorem 4** *Suppose policy $\pi = \pi^{N-UCB}$ is executed with $\mathcal{K}_i = \{r_{ij} : j \in [N_i]\}$. Then it holds that*

$$\sup_{\boldsymbol{\psi} \in \Psi} \mathcal{R}^{\pi}(T, \boldsymbol{\psi}) \leq O(\sqrt{MKT \log(MKT)} + MKU \log^2(MKT)), \tag{10.12}$$

*where $\Psi$ is the set of all model parameters satisfying all stated assumptions, $K = \max_i |\mathcal{K}_i|$ and $U = \max_{i \in [M]} \max_{\theta \in \mathcal{K}_i} u_{i,\theta}$.*

As a corollary, with $K = |\mathcal{K}_i| = N + 1$ (for any $i \in [M]$) and $U \leq NC_V$, the regret upper bound in Theorem 4 can be simplified to

$$\begin{aligned} \mathcal{R}^{\pi}(T, \boldsymbol{\psi}) &\leq O(\sqrt{MNT \log(MNT)} + MN^2 C_V \log^2(MNT)) \\ &= \widetilde{O}(\sqrt{MNT} + MN^2). \end{aligned} \tag{10.13}$$

On the above regret upper bound, we remark that in online and bandit learning literature, the time horizon $T$ is usually considered to be the dominating term asymptotically. Therefore, when $T > M$ and the number of items per nest $N$ is small as compared to $T$, the dominating term in Eq. (10.13) is $\widetilde{O}(\sqrt{MNT})$. This matches the lower bound result $\Omega(\sqrt{MT})$ in Theorem 5 in the next section within a factor of $\sqrt{N}$.

### *10.3.4 Regret Lower Bound*

It is possible to establish a regret *lower bound* showing that dependency on the number of nests $M$ is necessary. Below we state a lower bound on the regret of any dynamic assortment planning policy under nested Logit models, proved in Chen et al. (2021, Theorem 2).

**Regret lower bound of any policy**

**Theorem 5** *Suppose the number of nests $M$ is divisible by 4 and $\gamma_1 = \cdots = \gamma_M = 0.5$. Assume also that the parameter boundedness assumptions hold. Then there exists a numerical constant $C_0 > 0$ such that for any admissible policy $\pi \in \mathcal{P}$,*

$$\sup_{\boldsymbol{\psi} \in \Psi} \mathcal{R}^{\pi}(T, \boldsymbol{\psi}) \geq C_0 \sqrt{MT}.$$

We note that the condition that $M$ is divisible by 4 is only a technical condition and does not affect the main message delivered in Theorem 5, which shows necessary dependency on $M$ asymptotically when $M$ is large. The proof of the above lower bound result involves careful construction of two types (categories) of nests that result in an exponential number of possible nest configurations, yielding a lower bound that scales polynomially with $M$. Interested readers should refer to Chen et al. (2021, Sec. 4) for details and complete proofs.

Comparing Theorem 5 with the regret upper bound $\widetilde{O}(\sqrt{MNT} + MN^2)$ established in the previous section, we notice that when $T$ (time horizon) is large compared to $M$ (the number of nests), both regret bounds have an $O(\sqrt{M})$ dependency on $M$. This suggests that the policy $\pi^{N-UCB}$ and its regret analysis deliver *optimal* dependency of regret on the number of nests $M$ in a dynamic nested assortment planning problem.

However, when comparing the upper and lower bounds, we also notice that there is a gap of $\sqrt{N}$ factor. It was conjectured in Chen et al. (2021) that the upper bound analysis for $\pi^{N-UCB}$ with an additional $O(\sqrt{N})$ factor is in fact tight. Actually, because $\pi^{N-UCB}$ treats each "level set" assortments (within each nest) as standalone estimation units, it is intuitive to see that the regret that $\pi^{N-UCB}$ incurs has to scale polynomially with $N$. Furthermore, it was also conjectured in Chen et al. (2021) that *any* possible dynamic strategy for nested logit models has to suffer at least an $\Omega(\sqrt{N})$ term in regret bound.

### *10.3.5 Bibliographic Notes and Discussion of Future Directions*

Most of the learning-while-doing results in this section were developed in the work of Chen et al. (2021), inspired by the epoch-based exploration strategies originated from Agrawal et al. (2019). The work of Chen et al. (2021) also discussed a discretization heuristic that attains lower regret when each nest consists of a large number of available products. Some structural results for the optimal solution in a nested Logit choice model were proved in Davis et al. (2014) and Li et al. (2015).

We remark that in the original nested Logit choice model (Davis et al., 2014), it is allowed that $\gamma_i > 1$ and furthermore there is a no-purchase option *within each nest*. In this section, we assumed $\gamma_i \leq 1$ because it is the setting in which the full-information combinatorial optimization problem is easy to solve, which is the foundation of the theoretical regret analysis. Indeed, when $\gamma_i$ exceeds one, it is proved in Davis et al. (2014) that the combinatorial optimization question (when

all parameters are known) is NP-hard, and only approximation algorithms can be developed.

For future directions, an intriguing question is to close the gap between the $\widetilde{O}(\sqrt{NMT})$ regret upper bound and the $\Omega(MT)$ lower bound for the two-level nested logit model. As discussed in the lower bound subsection, it is conjectured that the lower bound may be improved for $N \gg M$. However, the tight dependence on $N$ remains a mystery.

It is also worthwhile to study the dynamic assortment planning problem for the general $d$-level nested logit model. While the static optimization problem was well studied in Li et al. (2015), little is known about the regret bounds in the dynamic learning setting. Answering the following question may be the first step to reveal the tight regret for the general $d$-level nested logit model.

**? Diminishing average regret for the general $d$-level nested logit model**

Suppose there are $N$ products and $M$ nests in a $d$-level nested logit model.[1] We further suppose that the revenue parameter parameters of products are known to the seller, while the utility parameters of the products and the correlation parameters of the nests are unknown. Is there a policy $\pi$ such that the regret at time horizon $T$ is at most $\text{poly}(N, M, d, \ln T) \times T^c$, where $c$ is a constant strictly less than 1? Furthermore, is it possible to achieve $c = 0.5$?

A positive answer to the above question means that we are able to achieve a diminishing average regret (a.k.a., no regret) for the $d$-level nested logit model, and the next step would be to pin down the optimal value for $c$, as well as the optimal dependence on $N$, $M$, and $d$.

## 10.4 MNL Model with Contextual Features

In the conventional setup of dynamic assortment optimization with demand learning, it is usually assumed that the retailer offers assortments to a large number of potential customers during a selling season and the pool of customers *share* the same preference/choice model which allows the retailer to learn or estimate the customers' preferences. In reality, however, it is rarely the case that consumers' preferences are homogeneous. Instead, different customers with different personal profiles such as gender, age, geographical location, and past purchase activities typically display different product preferences or purchasing behaviors.

[1] In other words, there are $N$ leaves and $(M-1)$ internal nodes in the corresponding tree structure.

In this section we overview existing works on dynamic assortment optimization with demand learning when the retailer has access to *consumer features*, which enables modeling of heterogeneous consumer preferences of substitutable products.

### 10.4.1 Model Formulation and Assumptions

The retailer has $N$ substitutable products and offers assortments $S_1, \cdots, S_T \subseteq [N]$ subject to the capacity constraint $|S_t| \leq K \leq N$ for each of $T$ sequentially arriving customers. At the beginning of time period $t$, a potential customer arrives and reveals his/her *feature vector* $x_t \in \mathbb{R}^d$ to the retailer. The feature vector consists of personal information of the arriving customer such as his/her gender, age, geographical location, credit worthiness, and past purchase activities. The retailer then offers an assortment $S_t \subseteq [N]$, $|S_t| \leq K$ to the incoming customer and observes a randomized purchase activity $i_t \in S_t \cup \{0\}$. The purchase activity $i_t$ is governed by a *personalized* or *contextualized* MNL choice model, as

$$\mathbb{P}(i|S_t, x_t, \boldsymbol{\theta}) = \frac{e^{x_t^\top \theta_i}}{1 + \sum_{j \in S_t} e^{x_t^\top \theta_j}}, \qquad \forall i \in S_t \cup \{0\}, \tag{10.14}$$

with unknown contextual models $\theta_1, \cdots, \theta_N \in \mathbb{R}^d$, and the understanding that $\theta_0 = 0$, abbreviated as $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)$. The involvement of consumer feature vector $x_t$ in Eq. (10.14) clearly implies the heterogeneity of the choice models from different consumer pools.

A policy $\pi$ is *admissible* if at every time period $t$, the distribution of the offered assortment $S_t$ is supported on $\{S \subseteq [N] : |S| \leq K\}$ and is measurable with respect to the filtration of $\{S_\tau, i_\tau, x_\tau\}_{\tau<t} \cup \{x_t\}$. We use $\mathcal{P}$ to denote the class of all admissible policies. The consumer features $x_1, \cdots, x_N \in \mathbb{R}^d$ are generated from an unknown underlying process $P_\mathcal{X}$ such that $x_t$ is measurable with respect to the filtration of $\{S_\tau, i_\tau, x_\tau\}_{\tau<t}$. For any admissible policy $\pi \in \mathcal{P}$ and unknown $\boldsymbol{\theta}$, its *regret* is defined as

$$\mathcal{R}^\pi(T, \boldsymbol{\theta}) := \mathbb{E}^\pi \left[ \sum_{t=1}^{T} \max_{|S| \leq K} R(S|x_t, \boldsymbol{\theta}) - R(S_t|x_t, \boldsymbol{\theta}) \right], \tag{10.15}$$

where $x_1, \cdots, x_T \sim P_\mathcal{X}$ and $R(S|x, \boldsymbol{\theta}) := \sum_{i \in S} r_i \mathbb{P}(i|S, x, \boldsymbol{\theta})$ is the expected revenue of assortment $S$ with consumer feature $x$. Here $\{r_i\}_{i=1}^N$ are the profit margin parameters of each product which are assumed to be known to the retailer a priori. Clearly, $\mathcal{R}^\pi(T, \boldsymbol{\theta})$ is always non-negative and it is the objective to design admissible policy $\pi \in \mathcal{P}$ that minimizes $\mathcal{R}^\pi(T, \boldsymbol{\theta})$ as much as possible.

*Remark 1* In some existing works (Chen et al., 2020; Oh and Iyengar, 2019) the model is formulated slightly differently, with the retailer having access to *product*

*features* $\{x_{t,i}\}_{t,i=1}^{T,N}$ and the mean utility of product $i$ is $\langle x_{t,i}, \theta \rangle$ for a single, unknown model $\theta \in \mathbb{R}^d$. This "product feature" model encapsulates the above "consumer feature" model as a special case. To see this, let $\{x_t\}_{t=1}^{T} \subseteq \mathbb{R}^d$ and $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N) \in \mathbb{R}^{dN}$ be a problem instance in the consumer feature model. Define $x_{t,i} := (0, \cdots, 0, x_t, 0, \cdots, 0) \in \mathbb{R}^{dN}$ for each $i \in [N]$. Then it is easy to verify that $\langle x_t, \theta_i \rangle \equiv \langle x_{t,i}, \boldsymbol{\theta} \rangle$.

Throughout this section we make the following assumptions. Note that some individual algorithms/policies designed require additional assumptions or conditions, which will be stated later when we introduce such algorithms.

**Assumptions for contextual assortment with consumer features**

1. There exists a constant $L < \infty$ such that $0 \le r_{ti} \le 1$, $\|\theta_i\|_2 \le L$ and $\|x_t\|_2 \le L$ with probability 1 for all $i \in [N]$ and $t \in [T]$;
2. There exists a constant $\Upsilon > 0$ such that for all $S \subseteq [N]$, $|S| \le K$ and $x, \boldsymbol{\theta} = \{\theta_i\}_{i \in S} \subseteq \mathbb{R}^d$ with $\|x\|_2$, $\|\theta_i\|_2 \le L$, it holds that $\min_{i \in S \cup \{0\}} \mathbb{P}(i|S, x, \boldsymbol{\theta}) \ge \Upsilon$.

### *10.4.2 Algorithm Design: Thompson Sampling*

*Thompson sampling* (Thompson, 1933) is a generic algorithmic idea valuable to sequential decision making within an underlying Bayesian framework. In a Bayesian treatment, the unknown model parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)$ are sampled from a known *prior* distribution $\Phi_0$, which captures the a priori belief/knowledge of the retailer about the model uncertainty. The *posterior* distribution $\Phi_t$ of the model parameters $\boldsymbol{\theta}$ conditioned on the observations $\mathcal{H}_t := \{S_\tau, i_\tau, x_\tau\}_{\tau<t}$ can then be computed using the Bayes rule:

$$\Phi_t(\boldsymbol{\theta}|\mathcal{H}_t) \propto \Phi_0(\boldsymbol{\theta}) \times \prod_{\tau<t} \mathbb{P}(i_t|S_t, x_t, \boldsymbol{\theta}). \tag{10.16}$$

The Thompson sampling principle would then *sample* a parameter estimate $\boldsymbol{\theta}_t$ at time period $t$ from its posterior $\Phi_t(\cdot|\mathcal{H}_t)$, and then make assortment optimization decisions based on $\boldsymbol{\theta}_t$. This results in a careful tradeoff between *exploration* and *exploitation*, with the exploitation achieved by taking into account the collected data $\mathcal{H}_t$ and the exploration from the inherent uncertainty of $\boldsymbol{\theta}$ from the prior distribution. With a model parameter $\boldsymbol{\theta}_t$ randomly sampled, the assortment $S_t$ can be computed by maximizing the expected revenue with respect to $\boldsymbol{\theta}_t$. A complete pseudocode description is given in Algorithm 4.

Before presenting the theoretical properties of the Thompson sampling policy, it is important to remark on computational strategies of certain steps in Algorithm 4 that are seemingly computationally intractable. Step 4 is a combinatorial opti-

**Algorithm 4** Thompson sampling policy $\pi = \pi^{TS}(\Phi_0, T, K)$

1: **for** $t = 1, 2, \cdots, T$ **do**
2: Observe feature vector $x_t \in \mathbb{R}^d$ of the incoming customer;
3: Sample $\boldsymbol{\theta}_t \sim \Phi_t(\cdot|\mathcal{H}_t)$ with $\Phi_t$ defined in Eq. (10.16);
4: Offer assortment $S_t = \arg\max_{|S|\le K} \sum_{i\in S} r_i \mathbb{P}(i|S, x_t, \boldsymbol{\theta}_t)$;
5: **end for**

mization question involving approximately $\binom{N}{K}$ assortment choices. While directly solving the optimization is computationally intractable, the problem can be solved efficiently by using fractional programming techniques. See Rusmevichientong et al. (2010); Davis et al. (2013) for details and (Megiddo, 1978) for the general fractional programming principle.

Step 3 of Algorithm 4, on the other hand, requires more efforts to mitigate its computational burden. Because the MNL choice model lacks conjugate priors and each observation point $(S_\tau, i_\tau, x_\tau)$ involves multiple products, the posterior distribution $\Phi_t(\cdot|\mathcal{H}_t)$ is unlikely to be decomposable, making exact computation and sampling from $\Phi_t(\cdot|\mathcal{H}_t)$ intractable. It is proposed in Cheung and Simchi-Levi (2017) to use Metropolis-Hasting Markov-chain Monte Carlo (MH-MCMC) (Andrieu et al., 2003) to approximately sample from the complex joint distribution $\Phi_t(\cdot|\mathcal{H}_t)$. For implementation details we refer the readers to Cheung and Simchi-Levi (2017, Appendix D).

The performance of the Thompson sampling policy $\pi^{TS}$, assuming the sampling step $\boldsymbol{\theta}_t \sim \Phi_t(\cdot|\mathcal{H}_t)$ is executed exactly, can be analyzed by the following result from Cheung and Simchi-Levi (2017, Theorem 3.3).

**Bayes regret upper bound of $\pi^{TS}$**

**Theorem 6** *Fix an arbitrary prior distribution $\Phi_0$. There exists a universal numerical constant $C < \infty$ such that for sufficiently large $T$,*

$$\mathbb{E}_{\boldsymbol{\theta}\sim\Phi_1}\left[\mathcal{R}^{\pi^{TS}}(T, \boldsymbol{\theta})\right] \le C \times (\Upsilon^{-1}\sqrt{d} + L^2)N\sqrt{dKT\ln^2(NT)}.$$

The above theorem upper bounds the *average* regret of the Thompson sampling policy $\pi^{TS}$ over $\boldsymbol{\theta}$ distributed from the known prior distribution $\Phi_1$, also known as the *Bayes regret*. Such Bayes regret guarantees are in general weaker than worst-case minimax regret guarantees, except when $\Phi_1$ is taken to be the least favorable prior which is in general difficult to identify. The proof of the theorem draws machinery from the work of Russo and Van Roy (2014) that upper bounds the Bayes regret of Thompson sampling using a sequence of confidence interval sums, and self-normalized empirical process arguments that are the foundations of existing works on generalized linear bandit (Filippi et al., 2010; Li et al., 2017). Interested readers are referred to Cheung and Simchi-Levi (2017, Sec. 4.2) for proof ideas and details.

**Algorithm 5** Optimistic sampling policy $\pi = \pi^{OTS}(M, \alpha, \Phi_0, T, K)$

**for** $t = 1, 2, \cdots, T$ **do**
  Observe feature vector $x_t \in \mathbb{R}^d$ of the incoming customer;
  Compute the MAP estimate $\widehat{\boldsymbol{\theta}}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} \Phi_t(\theta|\mathcal{H}_t)$;
  **for** each product $i = 1, 2, \cdots, N$ **do**
    Compute $V_{t,i} = I + \sum_{\tau<t: i\in S_\tau} x_\tau x_\tau^\top$;
    Obtain $M$ i.i.d. samples $\theta_{t,i}^{(1)}, \cdots, \theta_{t,i}^{(M)} \sim \mathcal{N}(\widehat{\theta}_{t,i}, \alpha^2 V_{t,i}^{-1})$;
    Compute $\widehat{u}_{t,i} = \max_{1\le\ell\le M} x_t^\top \theta_{t,i}^{(\ell)}$;
  **end for**
  Offer assortment $S_t = \arg\max_{|S|\le K} \sum_{i\in S} r_i \mathbb{P}(i|S, \widehat{\boldsymbol{u}}_t)$, where $\mathbb{P}(i|S, \widehat{\boldsymbol{u}}_t) = \frac{e^{\widehat{u}_{t,i}}}{1+\sum_{j\in S} e^{\widehat{u}_{t,j}}}$;
**end for**

The upper bound of the $\pi^{TS}$ policy in Theorem 6 is only applicable to the Bayes regret. It is argued in Oh and Iyengar (2019) that the worst-case regret of the $\pi^{TS}$ policy is likely to scale exponentially with assortment capacity $K$, rendering the policy less useful when the worst-case regret is of interest. To address this issue, the work of Oh and Iyengar (2019) proposed an *optimistic sampling* variant of the Thompson sampling policy when the prior distribution of $\boldsymbol{\theta}$ is the standard Normal distribution. More specifically, let

$$\boldsymbol{\theta} \sim \Phi_0: \qquad \theta_1, \cdots, \theta_N \overset{i.i.d.}{\sim} \mathcal{N}(0, \lambda I_{d\times d}).$$

The optimistic Thompson sampling policy is described in Algorithm 5.

Comparing with the Thompson sampling policy $\pi^{TS}$, the major difference is that in $\pi^{OTS}$, instead of directly sampling $\boldsymbol{\theta}_t$ from the posterior distribution $\Phi_t(\cdot|\mathcal{H}_t)$, we use a multi-variate Gaussian distribution $\mathcal{N}(\widehat{\boldsymbol{\theta}}_t, \alpha^2 V_t^{-1})$ centered around the maximum a posteriori (MAP) estimate $\widehat{\boldsymbol{\theta}}_t$ to approximate the true posterior distribution. This not only has the advantage of making the posterior sampling step computationally tractable but also enables the policy to have worst-case regret guarantees, as shown below.

**Worst-case regret upper bound of $\pi^{OTS}$**

**Theorem 7** *Suppose the optimistic sampling policy $\pi = \pi^{OTS}(M, \alpha, \Phi_0, T, K)$ is executed with $M \gtrsim 1 + \ln(K)$, $\alpha = \Upsilon^{-1}(\sqrt{dN\ln(1+NT)} + \sqrt{\lambda})$ and $\Phi_0 = \mathcal{N}(0, \lambda I)$ with $\lambda \ge 1$. Then there exists a universal numerical constant $C < \infty$ and a positive constant $a > 0$ such that for sufficiently large $T$,*

$$\sup_{\boldsymbol{\theta}\in\Theta} \mathcal{R}^\pi(T, \boldsymbol{\theta}) \le C(\Upsilon^{-1}\lambda L \ln(NT))^a \times (dN)^{3/2}\sqrt{T}.$$

Comparing Theorem 7 with Theorem 6, we notice that the $\pi^{OTS}$ policy enjoys $\widetilde{O}(\sqrt{T})$ worst-case regret but has worse regret dependency on $N$ and $d$ (i.e., $\widetilde{O}((dN)^{3/2})$ instead of $\widetilde{O}(dN)$). This is likely due to the Gaussian approximation error of the posterior distributions.

### *10.4.3 Algorithm Design: Upper Confidence Bounds*

In this section we introduce frequentist algorithms built upon upper confidence bounds, which easily enjoy worst-case regret guarantees that only scale squarely with the number of products $N$. The key difference of these types of algorithms from the Thompson sampling algorithm introduced from the previous section is that, at time $t$ with history $\mathcal{H}_t = \{i_\tau, S_\tau, x_\tau\}_{\tau<t}$, instead of calculating and sampling from the posterior distribution of $\boldsymbol{\theta}$, one obtains the maximum-likelihood estimate

$$\widehat{\boldsymbol{\theta}}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} \sum_{\tau<t} \ln \mathbb{P}(i_t | S_t, x_t, \boldsymbol{\theta}), \tag{10.17}$$

where $\Theta = \{\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N) \subseteq \mathbb{R}^d : \|\theta_i\|_2 \leq L, \forall i\}$. With the MLE $\widehat{\boldsymbol{\theta}}_t$, assortment $S_t$ for customer arriving at time $t$ is calculated using upper confidence estimates of the expected revenue based on incoming customer's preferences. A complete pseudocode description is given in Algorithm 6.

Note that, apart from the difference of the use of MLE in $\pi^{UCB}$ instead of a posteriori sampling in $\pi^{TS}$, another significant difference is that the policy $\pi^{UCB}$ needs to construct *upper confidence* estimates of customers' utility parameters $\overline{v}_{t,i}$

**Algorithm 6** $\pi = \pi^{UCB}(\alpha, T, K)$ for MNL with contextual features

1: Offer an arbitrary assortment to the first customer;
2: **for** $t = 2, 3, \cdots, T$ **do**
3: Observe feature vector $x_t \in \mathbb{R}^d$ of the incoming customer;
4: Compute the MLE $\widehat{\boldsymbol{\theta}}_t$ using Eq. (10.17);
5: For each product $i \in [N]$ calculate upper confidence estimate $\overline{v}_{t,i}$ as

$$\overline{v}_{t,i} = e^{x_t^\top \widehat{\boldsymbol{\theta}}_t} + \alpha\sqrt{x_t^\top \overline{V}_{t,i}^{-1} x_t}$$

where $\overline{V}_{t,i} = I_{d\times d} + \sum_{\tau<t: i\in S_\tau} \frac{x_\tau x_\tau^\top}{|S_\tau|}$;
6: Offer assortment $S_t = \arg\max_{|S|\leq K} R(S|x_t, \{\overline{v}_{t,i}\})$ where

$$R(S|x_t, \{\overline{v}_{t,i}\}) = \frac{\sum_{i\in S} r_i \overline{v}_{t,i}}{1 + \sum_{i\in S} \overline{v}_{t,i}};$$

7: **end for**

before the assortment maximizing step. This is because the MLE $\widehat{\boldsymbol{\theta}}_t$ alone does *not* offer exploration, and, therefore, remaining uncertainty of $\boldsymbol{\theta}_t$ must be manually calculated and enforced through the upper confidence bound approach.

Unlike the Thompson sampling policy $\pi^{TS}$ in the previous section, the UCB policy $\pi^{UCB}$ is fully computationally efficient, because the MLE formulation in Eq. (10.17) amounts to convex optimization and can be solved using standard first-order methods, and the combinatorial optimization step of calculating $S_t$ can be efficiently solved using fractional programming techniques (Rusmevichientong et al., 2010; Davis et al., 2013; Megiddo, 1978).

The following result upper bounds the *worst-case* regret of the UCB policy, provided that the algorithm parameter $\alpha$ is appropriately chosen.

**Worst-case regret upper bound of $\pi^{UCB}$**

**Theorem 8** *Suppose the UCB policy $\pi = \pi^{UCB}(\alpha, T, K)$ is executed with parameter $\alpha \gtrsim K\sqrt{dN\ln^2(NT)}$. There exists a universal numerical constant $C < \infty$ such that, for sufficiently large $T$,*

$$\sup_{\boldsymbol{\theta}\in\Theta} \mathcal{R}^{\pi^{UCB}}(T, \boldsymbol{\theta}) \leq C \times \Upsilon^{-2.5} LdNK\sqrt{T\ln^2(NT)}.$$

Comparing the above result for the UCB policy with the one for the Thompson sampling policy, the UCB policy has two major advantages: that it is completely computationally efficient in contrast to $\pi^{TS}$ that needs MCMC approximate computing strategies, and the UCB policy has regret upper bound that uniformly holds for all bounded model parameters $\boldsymbol{\theta}$ while Thompson sampling only has Bayes regret upper bounds. On the other hand, the UCB policy has worse regret bounds in terms of dependency on $\Upsilon^{-1}$ and $K$ when compared to Theorem 6.

The work of Chen et al. (2020) shows that it is possible to further improve the dependency on $K$ in Theorem 8, by using more complicated upper confidence bound structures. We name the improved policy $\pi^{MLE-UCB}$, with a pseudocode description given in Algorithm 7. Comparing $\pi^{MLE-UCB}$ with the previous UCB policy $\pi^{UCB}$, we note that the major difference is in constructing the upper confidence bounds: in $\pi^{UCB}$ the UCBs of $v_{t,i} = e^{x_t^\top \theta_i}$ are first constructed and then used in the $S_t$ optimization problem; in contrast, the $\pi^{MLE-UCB}$ policy constructs UCB for the entire expected revenue function $R(S|x_t, \boldsymbol{\theta})$ and is, therefore, more statistically efficient in exploiting the uncertainty structures in the underlying parameter estimates $\{\widehat{\boldsymbol{\theta}}_t\}$.

The $S_t = \arg\max_{|S|\leq K} \overline{R}_t(S)$ step in Algorithm 7 is computationally challenging because the upper confidence bound components destroy the MNL choice model structure, making fractional programming techniques not applicable. It is, however, possible to design computationally tractable approximation algorithms that approxi-

**Algorithm 7** $\pi = \pi^{MLE-UCB}(T_0, \rho_0, \alpha, T, K)$ for MNL with contextual features

For the first $T_0$ customers, offer $S_t$ consisting of $K$ products sampled uniformly at random from $\{1, 2, \cdots, N\}$, and record purchasing actions $\{i_t\}_{t \le T_0}$;
Compute *pilot* estimator $\boldsymbol{\theta}^p = \arg\max_{\boldsymbol{\theta} \in \Theta} \sum_{t \le T_0} \ln \mathbb{P}(i_t | S_t, x_t, \boldsymbol{\theta})$;
**for** $t = T_0 + 1, T_0 + 2, \cdots, T$ **do**
    Observe the feature vector $x_t \in \mathbb{R}^d$ for the incoming customer;
    Let $z_{t,i} \in \mathbb{R}^{dN}$ be defined as $z_{t,i} = (0, \cdots, 0, x_t, 0, \cdots, 0)$ for all $i \in [N]$;
    Compute local MLE $\widehat{\boldsymbol{\theta}}_t = \arg\max_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^p\|_2 \le \rho_0} \sum_{\tau < t} \ln \mathbb{P}(i_\tau | S_\tau, x_\tau, \boldsymbol{\theta})$;
    Compute $S_t = \arg\max_{|S| \le K} \overline{R}_t(S)$, where

- $\overline{R}_t(S) = \sum_{i \in S} r_i \mathbb{P}(i | S, x_t, \widehat{\boldsymbol{\theta}}_t) + \min\{1, \alpha \sqrt{\|\widehat{I}_t^{-1/2}(\widehat{\boldsymbol{\theta}}_t) \widehat{M}_t(\widehat{\boldsymbol{\theta}}_t | S) \widehat{I}_t^{-1/2}(\widehat{\boldsymbol{\theta}}_t)\|_{\text{op}}}\}$;
- $\widehat{M}_t(\boldsymbol{\theta} | S) = \sum_{i \in S} \mathbb{P}(i | S, x_t, \boldsymbol{\theta}) z_{t,i} z_{t,i}^\top - (\sum_{i \in S} \mathbb{P}(i | S, x_t, \boldsymbol{\theta}) z_{t,i})(\sum_{i \in S} \mathbb{P}(i | S, x_t, \boldsymbol{\theta}) z_{t,i})^\top$;
- $\widehat{I}_t(\boldsymbol{\theta}) = \sum_{\tau < t} \widehat{M}_\tau(\boldsymbol{\theta} | S_\tau)$;

    Offer assortment $S_t$ to the incoming customer and record purchasing action $i_t \in S_t \cup \{0\}$;
**end for**

mate the optimal solution of the combinatorial optimization problem well. See Chen et al. (2020, Sec. 5) for details.

The $\pi^{MLE-UCB}$ policy, unlike the other policies introduced in this section, also requires a "warm-up" phase that offers random assortments to the first few customers in order to obtain a good "pilot" estimator $\theta^p$. For the purpose of this procedure, the policy requires an additional non-degeneracy assumption imposed on the generating process of the context vectors $x_t$, as shown below:

- $\{x_t\}_{t=1}^T$ are i.i.d. generated from an unknown underlying distribution $\mu$ supported on $\{x \in \mathbb{R}^d : \|x\|_2 \le L\}$ with density $\mu$ satisfying $\lambda_{\min}(\mathbb{E}_\mu[(x-a)(x-a)^\top]) \ge \lambda_0 > 0$ for all $a \in \mathbb{R}^d$, $\|a\|_2 \le L$.

With the above non-degeneracy assumption, together with the two assumptions imposed in Sect. 10.4.1, the worst-case regret of policy $\pi^{MLE-UCB}$ can be upper bounded as follows.

**Worst-case regret upper bound of $\pi^{MLE-UCB}$**

**Theorem 9** *Suppose policy $\pi = \pi^{MLE-UCB}(T_0, \rho_0, \alpha, T, K)$ is executed with $T_0 = \lceil \sqrt{T} \rceil$, $\rho_0 = T^{-1/8}$ and $\alpha = \sqrt{dN \ln(TK)}$. Then there exists a universal numerical constant $C < \infty$ and some positive number $a > 0$ such that, for sufficiently large $T$,*

$$\sup_{\boldsymbol{\theta} \in \Theta} \mathcal{R}^\pi(T, \boldsymbol{\theta}) \le C(\Upsilon^{-1} L)^a \times dN \sqrt{T} \ln(\lambda_0^{-1} TK).$$

Comparing the regret upper bound in Theorem 9 with the one in Theorem 8 that applies to the $\pi^{MLE}$ policy, we observe that Theorem 9 saves an $O(K)$ factor, because $\pi^{MLE-UCB}$ uses more sophisticated upper confidence bound constructions for entire expected revenues of assortments.

### *10.4.4 Lower Bounds*

We present two versions of information-theoretical lower bounds to the contextual assortment optimization problem with demand learning and consumer features.

The first lower bound is obtained by reducing the contextual assortment optimization problem to standard assortment optimization and learning with the MNL model. Consider the problem setting of $d = 1$, $x_t \equiv 1$ and $\theta_i = v_i \in [0, 1]$. Then $\mathbb{P}(i|S_t, x_t, \boldsymbol{\theta}) = \frac{e^{v_i}}{1+\sum_{j\in S_t} v_j}$ reduces to the standard MNL choice model without contextual vectors. The result of Chen and Wang (2018) subsequently yields the following lower bound.

**Dimension-independent lower bound for contextual MNL**

For $T \geq N$, and $N \geq K/4$, there is an $\Omega(\sqrt{NT})$ lower bound for the worst-case regret of any admissible policy.

The above lower bound does not involve the feature vector dimension $d$, which is sub-optimal when feature vectors are long and involve many factors. To address this issue, the work of Miao and Chao (2019, Theorem 2) establishes the following dimension-independent lower bound for contextual MNL optimization with demand learning.

**Dimension-dependent lower bound for contextual MNL**

For $L = 1$, $d \geq 4$, $N \geq dK$ and $T \geq Nd/144$, there is an $\Omega(\sqrt{dNT}/K)$ lower bound for the worst-case regret of any admissible policy.

The above lower bound result clearly involves the dimension factor $d$, showing that the worst-case regret of any admissible policy should grow as the policy deals with longer feature vectors (i.e., larger $d$ values). On the other hand, the lower bound has an undesirable $1/K$ factor due to technical limitations.

### 10.4.5 Bibliographic Notes and Discussion of Future Directions

The majority results in this section are developed in the works of Cheung and Simchi-Levi (2017); Miao and Chao (2019); Chen et al. (2020); Oh and Iyengar (2019). The work of Cheung and Simchi-Levi (2017) also studied the more flexible setting in which the candidate subsets of assortments are personalized as well. The work of Miao and Chao (2019) provides several important improvements and extension to the $\pi^{UCB}$ policy, including a more computationally efficient algorithm based on Newton step updates and random projection methods for high-dimensional and sparse consumer features. The tools of linearly or generalized linearly parameterized bandits are essential in the development of the results in this section (Li et al., 2017; Filippi et al., 2010; Rusmevichientong and Tsitsiklis, 2010; Abbasi-Yadkori, 2011).

There are two obvious questions regarding the optimality of the regret upper bounds[2] in the problem of contextual MNL with consumer features. The first question concerns the optimality gap in terms of dependency of $L$, the upper bound on the $\ell_2$ norm of $\{\theta_i\}$. Because of the exponentiating operator, the $\Upsilon$ constant defined in the assumption typically scales as $e^{-R}$, meaning that both regret upper bounds in Theorems 6 and 8 would scale exponentially with respect to $L$. It is an interesting question whether it is possible to design a policy whose regret scales *polynomially* with $R$ and other problem parameters, as formulated below:

**? Question on polynomial-$L$ regret**

Suppose $L, N, K, d \to \infty$ as $T \to \infty$ at a reasonable asymptotic scale. Is it possible to design a policy class $\pi = \{\pi_T\}_T$ such that, for some constant $a \geq 0$,

$$\limsup_{T\to\infty} \sup_{\|\theta_i\|_2 \leq L} \frac{\mathcal{R}^{\pi_T}(T, \boldsymbol{\theta})}{\sqrt{T}(dKNL\ln(NT))^a} < \infty?$$

*Remark 2* Under the MNL model, the above question essentially asks whether it is possible to design a policy with regret scaling polynomially with respect to $\ln(1/\Upsilon)$.

A positive answer to the above question requires careful exploration strategies so that the Fisher's information on explored assortments will not be too degenerated. It is an open question at the time of the writing of this chapter.

[2] In terms of either the worst-case regret or the Bayes regret. We shall adopt the worst-case regret formulation here as it is more popular.

The second question concerns the optimality of the dependency of $N$ and $K$ parameters. The previous section shows an $\Omega(\sqrt{NT})$ lower bound of the contextual assortment optimization problem with consumer features. It is clear that there is an $O(\sqrt{N})$ term and potentially many polynomial $K$ terms between the established upper and lower bounds. It is an open question how the gap is to be closed, especially the gap on the number of products $N$ as $N$ is usually very large for online retailers.

**? Question on optimality gap of $N$ and $K$**

Fix $d$, $L$, and $\Upsilon$ and suppose $N, K \to \infty$ as $T \to \infty$ at a reasonable asymptotic scale. Is it possible to design a policy class $\pi = \{\pi_T\}_T$ such that, for some constants $a, b \geq 0$, it holds that

$$\limsup_{T\to\infty} \sup_{\boldsymbol{\theta}\in\Theta} \frac{\mathcal{R}^{\pi_T}(T, \boldsymbol{\theta})}{\sqrt{NT}K^a \ln^b(NT)} < \infty?$$

Furthermore, what is the smallest possible constant $a$ in the above inequality?

To achieve the improved $O(\sqrt{N})$ term it is likely that SupLinUCB type algorithms and analysis need to be applied (Li et al., 2017; Chu et al., 2011; Auer, 2002). To identify the $K^a$ dependency it is likely that much more insights into the upper confidence structure need to be gained to achieve tight $K$ dependency.

## 10.5 Conclusion

In this chapter we give an overview of the majority body of works on learning-while-doing in the dynamic assortment optimization question when the underlying demand or discrete choice model is governed by ones beyond the classical multinomial Logit (MNL) model. The choice models studied in the literature including general choice models with independent utility distributions beyond the Gumbel distribution, the nested Logit choice model that incorporates certain "nest" structures in consumers' decision making process, and contextual MNL models with contextual or feature vectors available for each customer or product. We have also mentioned several detailed open questions and future research directions based on the existing literature that would further complete the study of the mentioned general choice models beyond the MNL.

To conclude this section we mention two potential directions to further extend this line of research, from a higher level of view.

1. From a **modeling** perspective, what other types of discrete choice models beyond the MNL could be studied and analyzed under the learning-while-doing framework? A notable example is the *mixed* logit models (McFadden and

Train, 2000), whose parameter estimation properties have been explored in Train (2008); Jagabathula et al. (2020b). More interesting models are those that model the consumers' discrete choices as DAGs (Jagabathula et al., 2020a) or Markov chains Feldman and Topaloglu (2017), extends significantly beyond all choice models studied in this section that model consumers' utility as random variables that are independent among substitutable products;

2. From a **methodological** perspective, what other algorithmic frameworks could be useful in study dynamic assortment selection with demand learning, apart from the Thompson sampling and Upper-Confidence-Bound strategies that are used in the majority of existing works. For example, it is interesting to explore *first-order optimization methods* such as stochastic/estimating gradient descent based potentially on continuous relaxation of discrete choice optimization problems (Davis et al., 2013), or EXP-family methods based on online mirror descent (Bubeck and Cesa-Bianchi, 2012; Auer et al., 1995) that could potentially be applied to dynamic assortment problems with non-stationary or even adversarial choice models.

**Acknowledgments** We would like to thank the editors for their invitation and helpful guidelines on the writing of this chapter. We would also like to thank Sentao Miao for his suggestions that greatly helped the writing of Sect. 10.4.

## References

Abbasi-Yadkori, Y., Pál, D., & Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems (NeurIPS)* (pp. 2312–2320).

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2017). Thompson sampling for the MNL-bandit. In *Proceedings of the 30th Conference on Learning Theory (COLT)* (pp. 76–78). PMLR

Agrawal, S., Avadhanula, V., Goyal, V., & Zeevi, A. (2019). MNL-bandit: A dynamic learning approach to assortment selection. *Operations Research, 67*(5), 1453–1485.

Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning, 50*(1), 5–43.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research, 3*(Nov), 397–422.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science (FOCS)* (pp. 322–331). New York: IEEE.

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning, 5*(1), 1–122.

Chen, W., Wang, Y., & Yuan, Y. (2013). Combinatorial multi-armed bandit: General framework and application. In *Proceedings of the 30th International Conference on Machine Learning (ICML)* (pp. 151–159).

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., & Lu, P. (2016). Combinatorial multi-armed bandit with general reward functions. In *Proceedings of the 30th Conference on Advances in Neural Information Processing Systems (NeurIPS)*

Chen, X., & Wang, Y. (2018). A note on a tight lower bound for capacitated MNL-bandit assortment selection models. *Operations Research Letters, 46*(5), 534–537.

Chen, X., Wang, Y., & Zhou, Y. (2018). An optimal policy for dynamic assortment planning under uncapacitated multinomial logit models. *Mathematics of Operations Research* (in press). arXiv preprint arXiv:1805.04785.

Chen, X., Wang, Y., & Zhou, Y. (2020). Dynamic assortment optimization with changing contextual information. *Journal of Machine Learning Research, 21*(216), 1–44.

Chen, X., Shi, C., Wang, Y., & Zhou, Y. (2021). Dynamic assortment planning under nested logit models. *Production and Operations Management, 30*(1), 85–102.

Cheung, W. C., & Simchi-Levi, D. (2017). Thompson sampling for online personalized assortment optimization problems with multinomial logit choice models. Available at SSRN 3075658.

Chu, W., Li, L., Reyzin, L., & Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)* (pp. 208–214). JMLR Workshop and Conference Proceedings.

Daganzo, C. (2014). *Multinomial probit: The theory and its application to demand forecasting*. Amsterdam: Elsevier.

Davis, J., Gallego, G., & Topaloglu, H. (2013). Assortment planning under the multinomial logit model with totally unimodular constraint structures. Work in Progress.

Davis, J. M., Gallego, G., Topaloglu, H. (2014). Assortment optimization under variants of the nested logit model. *Operations Research, 62*(2), 250–273.

Feldman, J. B., & Topaloglu, H. (2017). Revenue management under the Markov chain choice model. *Operations Research, 65*(5), 1322–1342.

Filippi, S., Cappe, O., Garivier, A., & Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In *Proceedings of the 24th conference on advances in neural information processing systems (NeruIPS)* (pp. 586–594).

Jagabathula, S., Mitrofanov, D., & Vulcano, G. (2020a). Personalized retail promotions through a DAG-based representation of customer preferences. Available at SSRN 3258700.

Jagabathula, S., Subramanian, L., & Venkataraman, A. (2020b). A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science, 66*(8), 3635–3656.

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6*(1), 4–22.

Li, G., Rusmevichientong, P., & Topaloglu, H. (2015). The d-level nested logit model: Assortment and price optimization problems. *Operations Research, 63*(2), 325–342.

Li, L., Lu, Y., & Zhou, D. (2017). Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)* (pp. 2071–2080). PMLR.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics* (pp. 105–142)

McFadden, D., Train, K. (2000). Mixed MNL models for discrete response. *Journal of Applied Econometrics, 15*(5), 447–470.

Megiddo, N. (1978). Combinatorial optimization with rational objective functions. In *Proceedings of the annual ACM symposium on Theory of computing (STOC)*

Miao, S. & Chao, X. (2019). Fast algorithms for online personalized assortment optimization in a big data regime. Available at SSRN 3432574.

Oh, M. h., & Iyengar, G. (2019). Thompson sampling for multinomial logit contextual bandits. In *Proceedings of the 33rd conference on advances of neural information processing systems (NeurIPS)* (pp. 3145–3155).

Rusmevichientong, P., & Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research, 35*(2), 395–411.

Rusmevichientong, P., Shen, Z. J. M., & Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research, 58*(6), 1666–1680.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243.

Sauré, D., & Zeevi, A. (2013). Optimal dynamic assortment planning with demand learning. *Manufacturing and Service Operations Management, 15*(3), 387–404.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3/4), 285–294.

Train, K. E. (2008). EM algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling, 1*(1), 40–69.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.

# Part IV
# Inventory Optimization

# Chapter 11
# Inventory Control with Censored Demand


**Xiangyu Gao and Huanan Zhang**

## 11.1 Introduction

Inventory control problems have long been one of the most important topics in operations and supply chain management since the seminal study on the newsvendor problem. It captures some of the fundamental trade-offs when matching demand with supply. Firms typically need to make inventory control decisions over time, where the revenue and cost depend on the current system states and realizations of random demand and other uncertainties. Most existing research papers on inventory control problems assume that firms have complete knowledge about the distribution of uncertainties. They have characterized the structures of optimal polices and developed efficient algorithms for a large number of inventory models.

However, the study of joint learning and optimization problems when the demand distribution is not known a priori is relatively new in the field of inventory management. As the business world is rapidly changing and a huge amount of data becomes available, it is paramount for firms to make good use of data and design joint learning and optimization algorithms that can update demand information based on observed data to make better inventory control decisions. In this chapter, we will discuss the recent development in studying *nonparametric* joint learning and optimization algorithms for inventory models which do not involve pricing decisions since joint inventory and pricing models and Bayesian inventory models will be covered in the later chapters.

X. Gao
The Chinese University of Hong Kong, Hong Kong, China
e-mail: xiangyu@cuhk.edu.hk

H. Zhang (✉)
University of Colorado Boulder, Boulder, CO, USA
e-mail: huanan.zhang@colorado.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_11

One of the main challenges in designing efficient joint learning and optimization algorithms for inventory control models is the *censored demand* due to lost-sales. Because of the censored demand, the well-known exploration and exploitation trade-off in most online learning algorithm needs to be balanced. Note that there are research papers studying inventory models where the demand is not known a priori, but demand observations are *uncensored*. In this case, the sample average approximation method is commonly used, see Levi et al. (2007, 2015); Cheung and Simchi-Levi (2019). These papers typically study the offline problem about how many samples of demand data are needed to generate a demand estimation *before* making inventory decisions. Besbes and Muharremoglu (2013) studied a repeated newsvendor problem with censored discrete demand and showed that active exploration is needed. Huh and Rusmevichientong (2009) studied a lost-sales inventory system with zero lead time by proposing a stochastic gradient-descent (SGD) method. They proved that the regret of their SGD algorithm is upper bounded by $O(\sqrt{T})$ through the online convex programming theory. The idea of designing algorithms based on SGD was later adopted in various papers. Shi et al. (2016) proposed an SGD-based algorithm for multiproduct systems under a warehouse capacity constraint. Zhang et al. (2018) designed a cycle-update SGD algorithm for a perishable inventory system. Besides the SGD-type algorithms, (Huh et al., 2011) applied the concept of Kaplan–Meier estimator to develop a data-driven algorithm for a repeated newsvendor problem with censored demand.

Another challenge for joint learning and optimization inventory control problems is *positive lead times*. It is well known that it is difficult to solve an inventory control problem with positive lead times and lost-sales even when the demand distribution is known. The positive lead times result in a high-dimensional state space due to the need to keep track of all pipeline orders. Huh et al. (2009a) was the first to consider the learning algorithms for lost-sales inventory system with positive lead times and proved a regret bound of $O(T^{2/3})$. Later, (Zhang et al., 2020) improved the regret bound of this problem to $\sqrt{T}$. Both of these two papers designed their algorithms based on SGD. Agrawal and Jia (2019) applied a different approach based on interval reduction of stochastic convex bandit problems to the lost-sales system with positive lead times. Due to the convexity of cost function when a base-stock policy is used, they developed a line search method with a confidence interval estimate of costs. Their algorithm's regret bound depends linearly on lead time $L$, which improves the previously best-known result where the dependence on $L$ was exponential.

The third challenge for designing joint learning and optimization algorithms for inventory models is the *high-dimensional decision space*. Note that for inventory models with positive lead times, the state space is high dimensional. However, the action space is still one-dimensional since only base-stock policies are considered (see Huh et al., 2009a; Zhang et al., 2020; Agrawal and Jia, 2019). This enables the use of the line search method. Moreover, the cost function is convex with respect to the decision, which is a key property utilized in SGD-type algorithms. Therefore, it becomes increasingly challenging to consider an inventory problem with multi-dimensional decision space without the convexity property of the cost function.

Yuan et al. (2021) considered an inventory problem with fixed setup cost. Although it is well known that the celebrated $(s, S)$ policy is optimal under full demand distributional information, designing a learning algorithm is not straightforward due to lack of convexity. Combining the ideas from bandit controls and SGD, they developed an algorithm using policy elimination with SGD. Chen and Chao (2019) studied a multiproduct inventory control problems with stock-out substitutions where the substitution behavior is captured by substitution probabilities between each pair of product and demand. They designed an algorithm with multiple cycles. In each cycle, there is an exploration phase consisting of multiple intervals followed by an exploitation phase. Both the primary demand and substitution probabilities are learned and updated during the exploration phase. Gao and Zhang (2021) introduced a learning framework for multiproduct inventory systems with customer choices. They developed two improvements to the UCB-type algorithm to utilize the sales information better.

Table 11.1 provides a partial summary of papers in the Operations Management Literature on joint learning and inventory control with censored demand. In this table, we briefly list the model each paper considered, the primary method its algorithm used, and the order of regret on time horizon $T$. This list is nowhere near exhaustive, especially due to a large number of working papers. We hope that this table can provide a steppingstone for navigating in this fast-developing field. The remaining of this chapter is organized as follows. Section 11.2 establishes the lower bound for inventory learning problems with censored demand. Section 11.3 demonstrates how to deal with censored demand through a perishable inventory

**Table 11.1** Partial summary of literature on joint learning and inventory control with censored demand

| Paper | Model | Method | Regret |
|---|---|---|---|
| Huh and Rusmevichientong (2009) | Lost-sales with zero lead time | SGD | $O(\sqrt{T})$ |
| Shi et al. (2016) | Multi-product with warehouse capacity | SGD | $O(\sqrt{T})$ |
| Zhang et al. (2018) | Perishable inventory | SGD | $O(\sqrt{T})$ |
| Huh et al. (2009a) | Lost-sales with positive lead times | SGD | $O(T^{2/3})$ |
| Zhang et al. (2020) | Lost-sales with positive lead times | SGD | $O(\sqrt{T})$ |
| Agrawal and Jia (2019) | Lost-sales with positive lead times | Line search | $O(L\sqrt{T})$ |
| Huh et al. (2011) | Repeated newsvendor | Kaplan–Meier estimator | N/A |
| Yuan et al. (2021) | Inventory with fixed set up cost | Policy elimination with SGD | $O(\sqrt{T})$ |
| Gao and Zhang (2021) | High-dimensional (substitution) | Tailored UCB | $O(\sqrt{T})$ |
| Chen and Chao (2019) | Inventory with stock-out substitution | Explore than exploit | $O(\sqrt{T})$ |

system. Section 11.4 discusses three different algorithms for the inventory learning model with last-sales and positive lead times. Section 11.5 shows a learning algorithm for a multiproduct inventory model with customer choices where the decision space is high dimensional.

## 11.2 Regret Lower Bound for Inventory Models with Censored Demand

Perhaps the simplest inventory learning problem with censored demand is the repeated newsvendor problem with censored demand considered in Besbes and Muharremoglu (2013), this work first studies the lower bound of the regret rate and then constructs a learning algorithm to match this lower bound. In this section, we focus on the lower bound results in Besbes and Muharremoglu (2013).

### *11.2.1 Model Formulation*

In the classic (one-period) newsvendor problem, the decision-maker is facing a random demand $D$ from a given distribution with cdf $F(\cdot)$ and needs to decide the ordering quantity $x$ at the beginning of the period. The decision-maker faces two types of cost at the end of the period: a per-unit overage cost $h$ and a per-unit shortage cost $p$. The per-unit purchase cost is sunk to 0. The expected cost in this period is given by

$$C(x) = \mathbb{E}[h(x - D)^+ + p(D - x)^+],$$

where $(a)^+$ denotes $\max(0, a)$. The well-known newsvendor optimal order quantity is given by $x_F^* = \min\{x : F(x) \geq \beta\}$, where $\beta = \frac{b}{h+b}$. For the learning problem with censored demand in the repeated newsvendor problem, the decision-maker does not know the distribution $F(\cdot)$ a priori and has to relay on the sales data observed in each period $t$ to make ordering decisions. We denote the ordering quantity in each period as $x_t$ and the demand as $D_t$. The sales data is $\min(x_t, D_t)$. The goal of the learning problem is to minimize the $T$-period regret of the learning algorithm $\pi$, which is defined as

$$\mathcal{R}_T(F) = \sum_{t=1}^{T} \mathbb{E}\left[C_t^\pi\right] - TC(x_F^*).$$

Note that the regret in Besbes and Muharremoglu (2013) is defined as the worst-case regret for any given distribution. We use this simpler definition to be consistent with other sections in this chapter.

### 11.2.2 Strictly Convex and Well-Separated Cases

When the demand distribution satisfies the following conditions, the lower bound of the regret rate is shown to be $\Omega(\log T)$.

**Assumption 1** *Assumptions on demand distribution $F(x)$:*

- $x^*(F) \le M$ *for some* $M > 0$.
- *If demand is continuous,* $F(\cdot)$ *is differentiable, and* $F'(x) \ge \varepsilon$ *for some* $\varepsilon > 0$ *for all* $x \ge 0$.
- *If demand is discrete,* $|F(x) - \beta| > \varepsilon$ *for* $x = x_F^* - 1, x_F^*$.

The first condition ensures a bounded optimal order quantity. For continuous demand, the second condition is about a strictly convex cost function, and the last condition for the discrete demand gives a minimal separation around the optimal condition. Under either continuous or discrete demand distributions, the worst-case regret with censored demand observation is lower bounded by $\Omega(\log T)$. The proof for the continuous case is by the construction of a family of uniform distributed demand between $[\theta, 1]$, where $\theta$ varies between $[0, 1/2]$, and the proof for the discrete case is by constructing two very close Bernoulli distributions, where the optimal order qualities for them are 0 and 1, respectively. One difference between the continuous and discrete demand is that the lower bound $\Omega(\log T)$ is proved under full demand observation for the continuous demand, but under censored demand observation for the discrete demand. Actually, it was shown that when the demand is discrete and satisfies the minimal separation condition, the worst-case regret can be upper bounded by $O(1)$ with full demand observation.

### 11.2.3 Worst-Case Regret Under General Demand Distributions

One may view this repeated newsvendor problem with censored demand as a variant of the Multi-Armed Bandit problem, where the decision-maker polls an arm and observes the one-period profit and also the sales information. Motivated by the lower bound results of the MAB problem, one may anticipate that the similar $\Omega(\sqrt{T})$ rate should also hold under the inventory learning problems. The answer is correct for both the continuous and discrete demand. In Besbes and Muharremoglu (2013), the following example is provided as a counterexample when Assumption 1 does not hold for discrete distribution.

*Example 1* For a $T$-period problem with large enough $T$, consider the following two distributions:

$$F_a(k) = \begin{cases} \beta + 1/\sqrt{T} & \text{if } k = 0, \\ 1 & \text{if } k \ge 1; \end{cases} \quad F_a(k) = \begin{cases} \beta - 1/\sqrt{T} & \text{if } k = 0, \\ 1 & \text{if } k \ge 1. \end{cases}$$

**Table 11.2** Regret rate lower bound for repeated newsvendor problems

| Distribution type | Demand observation | Assumption 1 holds | Assumption 1 does not hold |
|---|---|---|---|
| Continuous | Full | $\Omega(\log T)$ | $\Omega(\sqrt{T})$ |
| | Censored | $\Omega(\log T)$ | $\Omega(\sqrt{T})$ |
| Discrete | Full | $\Omega(1)$ | $\Omega(\sqrt{T})$ |
| | Censored | $\Omega(\log T)$ | $\Omega(\sqrt{T})$ |

In this example, the optimal order quantity for $F_a$ is 0 and 1 for $F_b$. As $T$ increases, the two distributions will be closer to each other. Even with full demand observation, it is not possible to distinguish the two distributions in $T$-periods. The error due to a wrong inventory decision shrinks linearly with $1/\sqrt{T}$. Hence, it can be shown that an $\Omega(\sqrt{T})$ regret rate is inevitable for any policy.

For continuous demand, we can easily transform the above example to a continuous distribution, say by adding a very small uniformly distributed random number to the demand, which is not affecting the structure of the difference between $F_a$ and $F_b$, and this additional small number is also not giving any new information for the learning algorithm. The detailed description can be found in Proposition 1 in Zhang et al. (2020). And the same $\Omega(\sqrt{T})$ lower bound holds under continuous demands. We summarize the lower bound of the regret rate in Table 11.2.

In Table 11.2, we can see that from the lower bound perspective, the demand censoring has some effects but not critical. But the challenge is, with full demand information, one may simply use the empirical data and keep following the empirical optimal policy to converge to the optimal policy. With demand censoring, a good learning algorithm has to carefully balance the learning-and-earning trade-offs to converge to the optimal policy, especially when the underlying inventory system is complicated. In the next section, we will discuss the design and analysis of learning algorithms for perishable inventory systems with censored demand.

## 11.3 Censored Demand Example: Perishable Inventory System

Periodic-review perishable inventory system is an indispensable part of our society. For example, managing meat, vegetable, and frozen products in supermarket/grocery stores, and managing pharmaceuticals and blood products in the healthcare industry all belongs to the perishable inventory system. It is known to be a challenging inventory problem even with full demand distribution information. Early works on perishable inventory system include (Nahmias, 1975; Fries, 1975) that proved that the optimal policy is complex even when the product lifetime is only

2 periods. Recently, (Chen et al., 2014; Li and Yu, 2014) derived some new propriety of the optimal policy using the $L^{\#}$-convexity and multimodularity, respectively.

Another line of researches focus on the development of heuristics for perishable inventory system, including (Nahmias, 1976, 1977a,b; Nandakumar and Morton, 1993; Cooper, 2001; Chao et al., 2015; Zhang et al., 2016; Chao et al., 2018; Zhang et al., 2019). One of the simple efficient heuristics is the base-stock policy, i.e., ignoring the age difference between the on-hand inventory, and only order up to the base-stock level. This is a near-optimal and widely adopted policy. In this section, we focus on how to converge to the best *base-stock* policy without prior demand information using only sales data, as discussed in Zhang et al. (2018).

### *11.3.1 Model Formulation*

In the perishable product system, each product is assumed to have a fixed lifetime of $m \geq 2$ periods. The demand in each period $t$, denoted as $D_t$, is assumed to be an i.i.d. continuous random variable. We focus on base-stock policies with first-in first-out (FIFO) issuing policy. Denote the base-stock level as $S$. The FIFO issuing policy means we always try to use older inventory to meet the demand. The lead time is assumed to be negligible in the problem. To keep track of on-hand products with different remaining lifetime, we use the vector $\mathbf{x_t} = [x_{t,1}, \ldots, x_{t,i}, \ldots, x_{t,m-1}, x_{t,m}]$, where each $x_{t,i}$ represents the total inventory in period $t$ with remaining lifetime $\leq i$. We have $x_{t,i} \leq x_{t,i+i}$ for $i = 1, \ldots, m-1$, and specifically, $x_{t,m-1}$ is the total on-hand inventory at the beginning of period $t$ before ordering, and $x_{t,m-1}$ is the total on-hand inventory at the beginning of period $t$ after ordering, which is $S$ when there is no overshoot from the previous period.

The unmet demand is assumed to be lost and censored. At the end of each period $t$, apart from the typical holding and shortage cost $h(x_{t,m}-D_t)^+ + p(D_t - x_{t,m})^+$, all the product with remaining lifetime 1 will expire if they fail to meet the demand in this period, and each unit would incur a unit outdating cost of $\theta$. The unit purchasing cost is sunk to 0 without loss of generality. We summarize the one-period cost and inventory dynamics as follows:

$$\begin{aligned} C_t &= h(x_{t,m} - D_t)^+ + p(D_t - x_{t,m})^+ + \theta(x_{t,1} - D_t)^+, \\ x_{t+1,j} &= \left(x_{t,j+1} - D_t - (x_{t,1} - D_t)^+\right)^+, \text{ for } 1 \leq j \leq m-1, \\ x_{t+1,m} &= \min(S, x_{t+1,m-1}). \end{aligned}$$

Our goal is to develop a nonparametric learning algorithm that can converge to the (clairvoyant) optimal *base-stock* policy $S^*$, which minimizes the long-run average cost, i.e.,

$$S^* = \inf_S \left\{ \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[ \sum_{t=1}^{T} C_t^{\pi(S)} \right] \right\},$$

where $\pi(S)$ denotes the base-stock policy with base-stock level $S$. Recall that as shown in Table 11.2, the lower bound of the regret for inventory learning problem is $\Omega(\sqrt{T})$, which also applies to the perishable inventory system. The target of the learning algorithm is to achieve the lower bound rate with censored demand information.

### 11.3.2 Challenges and Preliminary Results

Unlike in a typical online learning problem where each period the cost is only a function of the decision, in the perishable inventory system, we can see that the cost $C_t$ is a function of $\mathbf{x_t}$ and $S$, where $\mathbf{x_t}$ is affected by the previous orders. On the other hand, that means all the inventory decisions have long-lasting effects on the cost. We have to consider these effects when adjusting decisions under censored data.

To enable further discussions on stochastic gradient-descent methods, we first need to show the convexity of the total cost with respect to the base-stock level $S$. Unfortunately, it can be shown that the *one-period* cost $C_t^{\pi(S)}$ may not be convex for each sample path. This further confirms that we cannot directly apply stochastic gradient-descent methods to this problem. Luckily, if we consider the *total cost* over a $T$-period problem, the convexity is back, as stated in the following theorem.

**Theorem 1** *For the perishable inventory system operating under a base-stock policy $\pi(S)$, if the system begin with empty inventory, then for any realization of demand $\omega = (d_1, d_2, \ldots)$, the $T$-period total cost is convex in $S$ for any $T \geq 1$.*

The proof of this theorem is based on a linear programming reformulation of the problem and is similar in spirit to the one developed by Janakiraman and Roundy (2004) for the lost-sales inventory system with lead times. As a by-product of this theorem, it can be shown that under the same conditions except for the inventory perishability, the optimal base-stock level for a perishable inventory system is lower than the optimal base-stock level for the nonperishable counterpart.

### 11.3.3 Learning Algorithm Design: Cycle-Update Policy

In this subsection, we introduce the learning algorithm developed in Zhang et al. (2018). The algorithm can be considered as a variant of the stochastic (online) gradient-descent method. Like a typical gradient-descent algorithm, it is assumed that there is a known compact set $[0, \bar{S}]$ that contains the optimal policy $S^*$. As its name suggests, the Cycle-Update Policy updates the inventory decisions in cycles,

**Algorithm** Cycle-Update Policy (CUP)

**Initialization.** Set $\tau_1 = 1$, and the initial base-stock level $S_1$ for period 1 is arbitrarily chosen from $(0, \bar{S})$. Set $x_{1,m-1} = S_1$, and the cycle counter to $k = 1$.
**for** each period $t \geq 2$ **do**
  **if** the starting inventory level $x_{t,m-1} > 0$ (i.e., lost-sales did not occur in period $t-1$) **then**
    Keep the same base-stock level as in period $t-1$, i.e., order up to $S_k$ in period $t$ so that $x_{t,m} = S_k$. Go to the next period.
  **else**
    Set $\tau_{k+1} = t$ at the beginning of a new cycle $k+1$, and update the base-stock level $S_{k+1}$ by

$$S_{k+1} = \mathbf{P}_{[0,\bar{S}]}\big(S_k - \eta_k \nabla_1 G(S_k, (\tau_k, \tau_{k+1}); \omega)\big), \tag{11.1}$$

    where the step size $\eta_k = \gamma/\sqrt{k}$ for some positive constant $\gamma$, and $\nabla_1 G(S_k, (\tau_k, \tau_{k+1}); \omega)$ is a subgradient of the $k$-th cycle cost with respect to $S_k$ (by fixing $\tau_k$ and $\tau_{k+1}$). Order up to $S_{k+1}$ for period $t$ so that $x_{t,m} \geq S_{k+1}$, and set $k := k+1$. Go to the next period. □
  **end if**
**end for**

instead of in each period. The complexity of the problem is mainly due to the inventory carryover and the high-dimensional state space introduced by the lead times. Even we know the total amount of starting inventory, it is not enough, as we also need to know their age distribution. The only exception is when there is a stock-out from the previous periods, and all the inventory is hence cleared. Together with the convexity results, we see that the stock-out event can be considered as a clear start of a new cycle and also preserves the convexity property of the cycle cost. We present the Cycle-Update Policy (CUP) as follows.

The subgradient $\nabla_1 G(S_k, (\tau_k, \tau_{k+1}); \omega)$ can be computed efficiently by only using censored demand data. We refer the interested reader to Zhang et al. (2018) for a detailed description of the computation of the subgradient. We see that the design of the CUP algorithm is not complicated, as basically it updates the base-stock level every time a stock-out ($D_t \geq S_k$) happens. One may think that we can also design cycles based on the event $\{D_t \geq \bar{S}\}$, as all the policies (within $[0, \bar{S}]$) would stock-out and be "refreshed." However, this is not possible as the learning algorithm only observes the sales data.

### *11.3.4 Regret Analysis of CUP Algorithm*

Following the conventional notation of regret, we define the $T$-period regret of CUP, $\mathcal{R}_T^{\mathbf{CUP}}(\omega)$, as

$$\mathcal{R}_T^{\mathbf{CUP}}(\omega) = \mathbb{E}\left[\sum_{t=1}^{T}\left(C_t^{\pi(S_t)}(\omega) - C_t^{\pi(S^*)}(\omega)\right)\right],$$

where $S_t$ is the base-stock level prescribed by the CUP algorithm, and $S^*$ is the (clairvoyant) optimal base-stock level. We present the regret upper bound of CUP in the following theorem.

**Theorem 2** *Under the assumption that there is a known finite number $\bar{S}$ such that $S^* \leq \bar{S}$, and $\mathbb{P}(D_t \geq \bar{S}) > 0$, for each problem instance of the perishable inventory system, the regret of the Cycle-Update Policy (CUP) satisfies*

$$\mathcal{R}_T^{\mathbf{CUP}} \leq K_1 \sqrt{T}, \qquad \text{for all } T \geq 1,$$

*where $K_1$ is a positive constant not affected by the problem instance.*

First, we see that the regret rate is tight, and the assumptions are quite mild. We need to have the compact interval for the CUP algorithm, just like for other SGD methods. The $\mathbb{P}(D_t \geq \bar{S}) > 0$ condition ensures that the CUP algorithm will not be stuck at a very high inventory level.

We discuss the regret analysis of the CUP algorithm as follows.

**A Bridging Policy—Replacement of Old Inventories (ROI)** To prove Theorem 2, we compare and bound the difference between the $k$th ($k = 1, 2, \ldots$) cycle costs of CUP and the clairvoyant optimal policy, OPT. As it is possible that the OPT may not have a fresh start at the beginning of every cycle $k$ of the CUP algorithm, this comparison is not immediate. The key idea of the analysis is to introduce a bridging policy, called the replacement of old inventories (ROI for short), between CUP and the optimal base-stock policy $\pi(S^*)$. For each sample path, similar to the optimal policy, the bridging policy ROI uses $S^*$ as its base-stock level. However, at the beginning of $\tau_k$ ($k = 1, 2, \ldots$), ROI replaces all its inventory units (regardless of their ages) with brand new inventory units with remaining lifetime $m$. Intuitively, the ROI has the same holding and lost-sales cost as the OPT, given that their base-stock levels are the same ($S^*$), and the holding/lost-sales cost is only determined by the base-stock level. For the outdating costs, as the ROI has a fresher inventory, the ROI would have a lower total outdating cost. The next proposition confirms that, for each sample path, the total cost incurred by ROI gives a lower bound on the total cost incurred by the optimal base-stock policy $\pi(S^*)$, and it provides a bridge in comparing the total costs of CUP and the optimal policy.

**Proposition 1** *For each problem instance of the perishable inventory system, given any sample path $\omega = \{d_1, d_2, \ldots\}$ and any $T \geq 1$, the total cost incurred by the bridging policy ROI is less than or equal to the total cost incurred by the optimal base-stock policy $\pi(S^*)$.*

With Proposition 1, when analyzing the regret of CUP, instead of directly comparing with OPT, we can compare with the ROI policy, and that will serve as an upper bound of the regret. The proof of Theorem 2 is based on comparing the cost difference between CUP and ROI. Consider, at the beginning of each cycle $k$, both policies start from zero inventory, and CUP operates at the base-stock level $S_k$, and ROI operates at the base-stock level $S^*$. Within cycle $k$, define the total cost

under base-stock level $S$ (with empty starting inventory) as $G_k(S)$. By Theorem 1, $G_k(S)$ is a convex function. We can adopt the regret analysis of the SGD algorithm to analyze the regret of CUP. One remaining difference is that function $G_k(S)$ is not bounded in every sample path. We need to bound the expectation of $G_k(S)$ (and also the gradient of $G_k(S)$). We refer the interested reader to Zhang et al. (2018) for the detailed proof.

### *11.3.5 Strongly Convex Extension*

As shown in Besbes and Muharremoglu (2013), the lower bound of the inventory learning problem with censored demand is $\Omega(\log T)$, when the demand satisfies some conditions. And for the continuous demand considered in Zhang et al. (2018), the condition is strongly convex. In inventory systems, a sufficient condition to ensure the strongly convexity if the expected cost is a lower bound on the support of the probability density function. We state the following assumption for the strongly convex extension:

**Assumption 2** *There exist three known finite numbers $\bar{S}$, $\underline{S}$, and $\lambda$, such that*

*(i)* $0 \le \underline{S} < \bar{S}$, $\lambda > 0$.

*(ii)* $\underline{S} \le S^* \le \bar{S}$, *and* $\mathbb{P}(D_t \ge \bar{S}) > 0$.

*(iii) The probability density function $f(x)$ of single-period demand $D$ satisfies* $\inf_{x\in[\underline{S},\bar{S}]} f(x) \ge \lambda$.

Under Assumption 2, the long-run average cost $\left\{\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} C_t^{\pi(S)}\right]\right\}$ is strongly convex w.r.t. the base-stock level $S$ with parameter $\lambda(h+p)$ between $[\underline{S}, \bar{S}]$. Under this assumption, the (modified) CUP achieves a logarithmic regret rate, as stated in the following theorem.

**Theorem 3** *For perishable inventory system, we modify CUP as follows:*

1. *Use the projection operator $\mathbf{P}_{[\underline{S},\bar{S}]}$, instead of using $\mathbf{P}_{[0,\bar{S}]}$.*
2. *Change the step size to $\eta_k = \left(\frac{1}{\lambda(h+p)}\right)\frac{1}{k}$, $k = 1, 2, \ldots$.*

*Then, under Assumption 2, there exists some positive constant $K_2$, such that for any $T \ge 1$, the expected cumulative regret of CUP for any problem instance satisfies $\mathcal{R}_T^{\mathbf{CUP}} \le K_2 \log T$.*

We can see that the regret matches the lower bound results stated in Besbes and Muharremoglu (2013) for the case where the demand is continuous and has bounded support.

## 11.4 Lead Times Example: Lost-Sales System with Lead Times

In this section, we discuss the lost-sales inventory system with positive lead times. This is one of the most fundamental inventory systems, and it is known to be a challenging inventory problem, even from the pure optimization perspective. The problem suffers from the well-known curse-of-dimensionality, and the optimal policy is proved to be complex (see Zipkin 2008). A stream of research focused on the design of online learning algorithm for this problem, including (Huh et al., 2009a; Zhang et al., 2020; Agrawal and Jia, 2019). We discuss their learning algorithms in this section.

### *11.4.1 Model Formulation*

We first rigorously define the lost-sales inventory system with positive lead times. Similar to the previous section, we also consider i.i.d. *continuous* random demands, $\{D_1, D_2, \ldots, D_t, \ldots\}$, and censored demand observation. The lead time is denoted as $L$, which means any new order will stay in the pipeline for $L$-periods before arrival. To keep track of all the pipeline orders together with the on-hand inventory, we need to use an $(L+1)$-dimensional vector for the inventory state:

$$\mathbf{x}_t = [q_{t-1}, \ldots, q_{t-L+1}, I_t],$$

where $I_t$ is the on-hand inventory at the beginning of period $t$, and $q_k$ is the order placed in period $t$. Denote $\mathbf{y}_t = [q_t, q_{t-1}, \ldots, q_{t-L+1}, I_t]$ as the inventory after ordering in period $t$.

In every period, the sequence of events in each period $t$ is defined as follows:

1. At the beginning of period $t$, the firm observes the starting inventory vector $\mathbf{x}_t$ and determines the ordering quantity $q_t \geq 0$.
2. Then, the demand is realized as $d_t$. The demand is satisfied using on-hand inventory $I_t$ to the maximum extent, and the firm observes the sales $\min(d_t, I_t)$.
3. The one-period cost is

$$C_t = h(I_t - d_t)^+ + p(d_t - I_t)^+.$$

And the system proceeds to the next period with starting state

$$\mathbf{x}_{t+1} = \left[q_t, \ldots, q_{t-L+2}, I_{t+1} = q_{t-L+1} + (I_t - d_t)^+\right]. \tag{11.2}$$

Later, we also have a learning algorithm that considers the profit maximization equivalent version of the problem. We replace $C_t$ as $Q_t = p\min(d_t, I_t) - h(I_t - d_t)^+$ in this case. Following the conventional assumption, we consider the starting inventory vector is empty, i.e., $\mathbf{x}_1 = \mathbf{0}$. The goal of the learning algorithm is to find an ordering policy, based on the sales information, that can minimize the $T$-period regret against the clairvoyant optimal policy.

### *11.4.2 Base-Stock Policy and Convexity Results*

Due to the complexity of the problem, it is challenging to use the optimal policy as the benchmark. One of the widely adopted policies is the base-stock policy (e.g., see Zipkin 2008). Under a base-stock policy with base-stock level $S$, every period the decision-maker orders $q_t = (S - I_t - \sum_{k=1}^{L-1} q_{t-k})^+$ to bring the total *inventory position*, i.e., the sum of on-hand inventory and pipeline inventory, up to $S$. For lost-sales inventory system with lead times, although the base-stock policy is not optimal, it is shown in Huh et al. (2009b) that it is asymptomatic optimal, when the lost-sales penalty cost goes to infinity. Like the base-stock policy for the perishable inventory system, it was shown in Janakiraman and Roundy (2004) that the "cycle" cost under a base-stock policy is convex for a $T$-period problem.

### *11.4.3 Challenges from Lead Times*

The convexity results remind us of the use of SGD for the perishable problem. We can think of also constructing cycles to follow the gradient direction to update the cycle base-stock levels. However, in the presence of the lead times, this is becoming much more challenging.

Consider we are following this direction and construct cycles with base-stock level $S_k$ for each cycle $k$; then, we need to adjust this base-stock level between cycles. Because ultimately we compare the learning system with the optimal base-stock level $S^*$ that uses $S^*$ from the first period, and in the regret analysis, the policy that uses $S_k$ from the first period till the end of this cycle will naturally be a bridging policy. Denote the inventory vector of policy $S_k$ as $\mathbf{x}_t^{S_k}$ for each period $t$ and the beginning of each cycle $k$ as $t(k)$. If we can "magically" adjust the inventory vector at the beginning of each cycle, then we want to adjust it to $\mathbf{x}_{t(k)}^{S_k}$ at the beginning of cycle $k$, and then using the gradient of the cycle cost to update the base-stock level from $S_k$ to $S_{k+1}$, and so on, and we can easily build an SGD-type algorithm in this way. However, in practice, we cannot adjust the inventory vector freely. We can only control the ordering quantity in each period. One may think that this means we would need at most $L$-periods to adjust the inventory vector to be the same at the $S_k$-system. However, this is not guaranteed with random demands and censored

demand observations. For example, suppose at the beginning of cycle $k$, the learning algorithm has an inventory vector [3, 2, 1] and the $S_k$ policy has an inventory vector [2, 2, 2] (actually, this information would not be available to the learning algorithm, but we assume it for simplicity), and the demand realized to be greater than or equal to 3 for all the future periods for a long time. In the first period, the learning algorithm only observes a sales of 1 and hence will not be able to know that is the sales under system $S_k$. Hence, either ordering 1 or ordering 2 could be wrong. And if the learning algorithm just follows the base-stock policy, 6 in this case, then the learning algorithm system will just be repeating [3, 2, 1], [1, 3, 2], and [2, 1, 3], while the $S_k$-system is always [2, 2, 2], and they will not converge.

For this problem, the lead times give us a high-dimensional inventory state, while we can only work on the last dimension, and let the remaining entries interact with the demand. This also means we have indirect control over the information observed from the sales. In the next two subsections, we will discuss three different methods to overcome these challenges.

### *11.4.4 Gradient Methods*

**A Black-Box Method with Increasing Cycle Length** The first paper that studies this problem is Huh et al. (2009a). Their approach can be considered as a "black-box" approach that mainly uses one property of the inventory system—it will eventually converge over time. Despite the complexity of the inventory system, the expected cost under an inventory system is determined by the stationary distribution of the on-hand inventory under the base-stock level $S$, denoted as $I_\infty(S)$. We can write the expected average cost as $C(I_\infty(S))$. If we can observe the gradient of $C(I_\infty(\cdot))$, then the problem is solved. However, as we would need infinite periods to converge to the stationary on-hand inventory, we cannot observe $C(I_\infty(\cdot))$.

The solution offered by Huh et al. (2009a) is to replace the gradient of $C(I_\infty(\cdot))$ using a single-period cost gradient. The single-period cost gradient can be computed in the following way.

**Theorem 4** *Let $S \geq 0$ be the base-stock level of the inventory system and $\mathbf{x}_1$ be the initial inventory vector. Under the base-stock policy $S$, define $V(S, \mathbf{x}_1)$ as the first time that the total inventory position drops below $S$ under initial inventory $\mathbf{x}_1$. In each period $t$, denote the derivatives of the order quantity as $Q'_t(S)$ and the on-hand inventory as $I'_t(S)$. Define $I'_0(S)$ and $Q'_t(S) = 0$ for $t \geq 0$. Then we have*

- $Q'_t(S) \in \{0, 1\}$ *and* $I'_t(S) \in \{0, 1\}$.
- $Q'_t(S) = \begin{cases} 0 & \text{if } 1 \leq t < V(S, \mathbf{x}_1), \\ 1 & \text{if } t = V(S, \mathbf{x}_1), \\ I'_{t-1} \cdot \mathbb{1}[D_{t-1} \geq I_{t-1}] & \text{if } t > V(S, \mathbf{x}_1). \end{cases}$
- $I'_t(S) = I'_{t-1}(S) \cdot \mathbb{1}[D_{t-1} < I_{t-1}] + Q'_{t-L}(S)$.
- *With probability 1,* $I'_t(S) + \sum_{k=t-L+1}^{t} Q'_k(S) = 1$.

With the above theorem, we can keep track of the $I_t'(S)$ in every period, and hence we can easily compute the gradient of the cost in each period as a function of $S$.

The algorithm in Huh et al. (2009a) requires the decision-maker to know an interval $[\underline{S}, \bar{S}]$ such that $S^* \in [\underline{S}, \bar{S}]$. We briefly introduce the algorithm in Huh et al. (2009a), with some details omitted, as follows:

**Algorithm** Adaptive Algorithm

1: **Initialization.** Initialize the starting base-stock level $S_1$ as any number within $[\underline{S}, \bar{S}]$. Assume the starting inventory vector is empty. The length of cycle $k$, denoted as $T_k$, is defined by $T_k := \lceil \sqrt{k} \rceil$.

2: **for** each cycle $k$ with base-stock level $S_k$, **do**

3: Adopt base-stock level $S_k$ for every period, keep track of the derivative of the on-hand inventory.

4: At the end of the cycle, use the derivative of the last period's cost, $H_k(S_k)$, to update the $S_k$, following

$$S_{k+1} = \mathbf{P}_{[\underline{S}, \bar{S}]}\big(S_k - \frac{c}{\sqrt{k}} \cdot H_k(S_k)\big),$$

where c is a constant determined by problem parameters, $\mathbf{P}_{[\underline{S}, \bar{S}]}(x)$ is the projection function to project $x$ back to the interval $[\underline{S}, \bar{S}]$, and $H_k(S_k)$ is the one-period cost gradient in the last period of the cycle. □

5: **end for**

We can see that the algorithm uses the one-period cost gradient in period $T_k$ as a proxy of the one-period cost gradient in period $\infty$. It can be shown that when $T_k \to \infty$, the one-period cost gradient in period $T_k$ will converge to the one-period cost gradient in period $\infty$ in distribution, under mild conditions. Hence, to ensure convergence, the cycle length $T_k$ has to be increasing. Despite the simplicity of this approach, the downside is the convergence rate. Because the algorithm will update less and less frequently, the worst-case convergence rate is not tight in $T$ but is $\tilde{O}(T^{2/3})$.

**A More Complicated Method with Stable Cycle Length** The second paper that studies this problem (Zhang et al., 2020), closed this gap with a more sophisticated SGD-type algorithm. Unlike (Huh et al., 2009a) that takes a black-box approach of the underlying inventory system, the algorithm (Zhang et al., 2020) takes a closer look at the inventory dynamics and the information structure of the system.

The main idea of the algorithm is to deliberately control the start of each new cycle, instead of using a predetermined cycle length, so that we can control the starting state of the system. The first observation is that, consider the $\underline{S}$-system, i.e., the system with the lowest base-stock level between $[\underline{S}, \bar{S}]$, and another system with $S > \underline{S}$. Focus on the same sample path of demands. Then, it can be easily shown that the inventory vector of the $S$ system will be no less than the inventory vector

of the $\underline{S}$-system, in every entry in every period. This means if the $\underline{S}$-system has no stock-out for a period $t$, then all the other systems with a higher base-stock level will also have no stock-out for this period, and the ordering quantity for period $t+1$ will be $D_t$. Following this path, if the $\underline{S}$-system has no stock-out for $L$ consecutive periods, then all the systems with a higher base-stock level will have the same pipeline inventory vector, and the only difference will be the on-hand inventory. We can set the beginning of each cycle to be when the $\underline{S}$-system has no stock-out for $L$ consecutive periods, denote this period as a "triggering period." This can help us calibrate the computation of the gradient, as one of the challenges of the learning algorithm is to compute (estimate) the gradient of the $S_k$-system using censored demand data, while the inventory vector of the learning system could be different from the $S_k$-system.

The downside of setting the beginning of each cycle to be a triggering period instead of a predetermined cycle length is the requirement of knowledge of the triggering period. To know if the $\underline{S}$-system has no stock-out for each period, the learning algorithm has to be able to "simulate" the $\underline{S}$-system in the back end. In order to simulate the $\underline{S}$-system, the learning algorithm $\pi$ has to ensure that the inventory vector $\mathbf{x}_t^{\pi}$ dominates $\mathbf{x}_t^{\underline{S}}$ in every period. This requires a careful design of the learning algorithm, especially when decreasing the base-stock level between cycles. For example, when we decrease the base-stock level by a large amount, the learning algorithm may not order for several periods, while the $\underline{S}$-system could still be ordering for these periods. To overcome this issue, (Zhang et al., 2020) introduced the idea of "withheld inventory," to slow down the decrease of base-stock levels and ensure the simulation of the $\underline{S}$-system. When some on-hand inventory is marked as withheld, it is not included when the ordering quantity is computed using the base-stock level. The base-stock system pretends the withheld inventory does not exist, and it is only used to meet demand when all the other on-hand inventory have been used to meet demand in each period. Due to the space limit, we omit the details of the withheld inventory.

The second challenge of the learning algorithm is the computation of the gradient information. Unlike (Huh et al., 2009a) that relays on the natural convergence of the system with increasing cycle length, (Zhang et al., 2020) uses a stable cycle length, determined by the triggering period, and in this case, the learning algorithm needs to be able to get a more accurate gradient information with censored demand observation. Note that the learning algorithm's observation is based on $\mathbf{x}_t^{\pi}$, and we need to get the cycle cost gradient of $\mathbf{x}_t^{S_k}$. Recall that at the beginning of cycle $k$, call it period $\tau_k$, by the definition of the triggering period, the inventory vector of the $S_k$-system would be $[d_{\tau_k-1}, \ldots, d_{\tau_k-L+1}, S_k - \sum_{i=\tau_k-L+1}^{\tau_k-1} d_i]$, while the learning system could be $[d_{\tau_k-1} + S_k - S_{k-1}, \ldots, d_{\tau_k-L+1}, S_{k-1} - \sum_{i=\tau_k-L+1}^{\tau_k-1} d_i]$, we can see that when $S_k > S_{k-1}$, the learning system will have a lower on-hand inventory level than the $S_k$-system. In this case, we will not be able to guarantee that we can simulate the $S_k$-system. For example, if the learning system faces stock-out in period $\tau_k$, then we cannot tell the gradient in this period for the $S_k$-system. To overcome

this issue, (Zhang et al., 2020) proposed a two-phase design of a cycle. Still consider the $S_k$-system and the learning system in period $\tau_k$, with starting inventory vector

$$[d_{\tau_k-1}, \ldots, d_{\tau_k-L+1}, S_k - \sum_{i=\tau_k-L+1}^{\tau_k-1} d_i],$$

and

$$[d_{\tau_k-1} + S_k - S_{k-1}, \ldots, d_{\tau_k-L+1}, S_{k-1} - \sum_{i=\tau_k-L+1}^{\tau_k-1} d_i],$$

respectively.

It can be seen that when $S_k > S_{k-1}$, the learning algorithm has more inventory in the pipeline, which is not yet arrived. The trickily part is that even we wait for $L$-periods for those inventory to arrive, we cannot guarantee the two systems will converge. For example, suppose the demand is very high for the next $L$-periods and both systems face stock-out in all the $L$-period, then the inventory vector of both systems will keep shuffling and goes back to the same as in period $\tau_k$ after $L$-periods. Intuitively, high demands will keep the two systems from converging. And we shall think of demands realized to be low, which remind us of the event we defined before: the triggering period. Indeed, if the learning algorithm keeps using base-stock level $S_k$ and waits for another triggering period, the two systems will be the same, and after that, the learning algorithm could use the sales to estimate the gradient in this second part of the cycle and wait for another triggering period to update the base-stock level. This is the two-phase design of a cycle. First, wait for another triggering period to ensure the inventory vector to be the same as the $S_k$-system, and then use the cost gradient from the second triggering period to the last triggering period in the cycle to update the base-stock level, following the gradient direction from $S_k$ to $S_{k+1}$.

As the detailed algorithm, denoted as the Simulated Cycle-Update (SCU) algorithm, description in Zhang et al. (2020) is more than one-page long, we just summarize the main ideas of the algorithm as follows:

- The SCU algorithm maintains the same base-stock level, $S_k$ within each cycle $k$. The length of each cycle is not predetermined but begins with a "triggering period." Each triggering period is defined as when the $\underline{S}$-system, the system that uses base-stock level $\underline{S}$ from period 1, has no stock-out for $L$ consecutive periods.
- In order to know the triggering periods, the SCU algorithm needs to simulate the $\underline{S}$-system in each period $t$. To ensure the inventory vector of the SCU-system dominates the $\underline{S}$-system, the SCU algorithm marks some on-hand inventory as withheld when dropping the base-stock level and gradually releases them to avoid sudden drops of the base-stock level and to sustain the simulation.

- In each cycle, there are two phases. Each phase begins with a triggering period: the first phase is used for the inventory vector to converge to the $S_k$-system, and in the second phase, when the learning system has the same inventory vector (excluding the withheld inventory part) as the $S_k$-system, the algorithm can compute the gradient of the $S_k$-system and use that to update the base-stock level following the gradient-descent method.

**Regret Analysis** To analyze the regret of the SCU algorithm, i.e., the cost difference between the SCU-system and the $S^*$-system, (Zhang et al., 2020) introduced two bridging systems: the $\overline{\text{SCU}}$-system and the $G$-system. Both bridging systems are imaginary systems that are not implementable and are only used for regret analysis purposes. The $\overline{\text{SCU}}$-system has the same inventory vector as the SCU-system in every period, except that the withheld inventory pays no holding cost in the $\overline{\text{SCU}}$-system. So, the gap between the SCU-system and the $\overline{\text{SCU}}$-system contains the holding cost of the withheld inventory, which can be shown to be $O(\sqrt{T})$. The second bridging system, the $G$-system, is defined as the system that uses $S_k$ within each cycle, and the starting inventory in $\tau_k$ is changed to be the same as the $S_k$-system, i.e.,

$$[d_{\tau_k-1}, \ldots, d_{\tau_k-L+1}, S_k - \sum_{i=\tau_k-L+1}^{\tau_k-1} d_i].$$

The gap between the $\overline{\text{SCU}}$-system and the $G$-system is the cost difference in the first phase of each cycle between the learning system and the $G$-system, when $S_k > S_{k-1}$. This part can be shown to be $O(\sqrt{T})$. The remaining gap is the gap between the $G$-system and the optimal $S^*$-system. This part is similar to the gap between the ROI system and the optimal base-stock system in the perishable inventory system in Zhang et al. (2018). As both the $G$-system and the optimal $S^*$-system incur a convex cycle cost within each cycle, the $G$-system updates the base-stock level using correct gradient information (gathered in the SCU-system). This gap is also $O(\sqrt{T})$. We summarize the structure of the regret analysis in Fig. 11.1.

From Fig. 11.1, we can see the challenge introduced by lead times. With lead times, the high-dimensional inventory state can only be affected by the inventory decisions indirectly. And with censored demand observation, there could be challenges in both increasing and decreasing the base-stock level. Zhang et al. (2020) designed a quite complicated algorithm to overcome these issues. However, there is still one drawback of the algorithm—the exponential dependency on $L$ for the regret bound. Because the cycles are constructed with triggering periods and the cycle length is exponential in the lead time $L$, the regret is inevitably exponential in the lead time $L$.

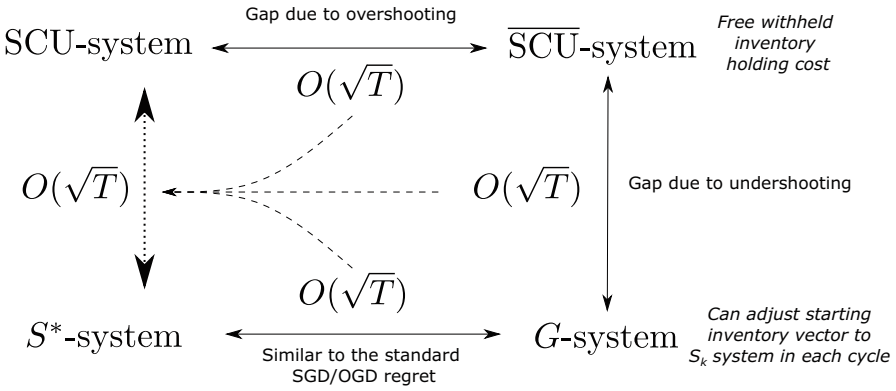


**Fig. 11.1** Illustration of the regret analysis of the SCU algorithm

### *11.4.5 A Ternary Search Method*

Agrawal and Jia (2019) proposed a learning algorithm that can avoid the exponential dependency on $L$ in Zhang et al. (2020). The algorithm takes a different approach than Huh et al. (2009a); Zhang et al. (2020). Instead of utilizing the convexity of the problem, (Agrawal and Jia, 2019) makes use of the fact that the decision space is single-dimensional, and the expected cost is unimodal. These enable the use of ternary search methods.

The challenges from censored demand observation, positive lead times, and demand randomness remain the same when applying ternary search methods. With censored demand observation, the cost is not fully observable. As a common approach in the literature, the paper transformed the cost minimization problem it into the equivalent profit maximization problem (called as pseudo-cost in the paper). Censored demand observation also contributes to the learning-and-earning trade-off. With uncensored demand observation, we can directly use the observed demand data to test the cost under a different policy, but with censored demand information, the learning algorithm in Agrawal and Jia (2019) has to test each point separately.

To overcome the challenge of positive lead times, the major breakthrough in Agrawal and Jia (2019) is to prove that for the lost-sales inventory system with positive lead times under base-stock policies, the cost difference from different initial inventory vectors is bounded by a term that is linear in $L$. More precisely, they have shown the following proposition (originally presented as Lemma 2.5 in the paper):

**Proposition 2** *Consider a base-stock level $S > 0$, planning horizon $T$, and two starting inventory vectors $\mathbf{x}$ and $\mathbf{x}'$ with $\sum \mathbf{x} \leq S$ and $\sum \mathbf{x}' \leq S$. For any given sample path, the $T$-period total cost for the two systems, denoted as $C_T(\mathbf{x}, S)$ and $C_T(\mathbf{x}', S)$, satisfies*

$$\left|C_T(\mathbf{x}, S) - C_T(\mathbf{x}', S)\right| \leq 36 \max(h, p) LS.$$

Note that the gap between the cost of the two systems and the profit of the two system is the same for any sample path. From Proposition 2, we can see that the bias introduced from different initial inventory vectors is bounded by $O(L)$ and is not increasing with $T$. This result removes the necessity of carefully designing the initial state of each cycle and enables an $O(LT)$ regret rate of the learning algorithm.

The last challenge is on the demand randomness. The usual ternary search methods only apply to deterministic problems, as random observations could lead to wrong trimming directions and ultimately lead to a wrong result. Agrawal and Jia (2019) proposed a UCB-/LCB-based method to avoid this with high probability. Recall that the goal is to find the base-stock level $S^*$ with the highest long-run expected profit. Instead of comparing the profit value at two different $S$ values to determine which interval to trim, the learning algorithm in Agrawal and Jia (2019) constructed the UCB and the LCB (with a concentration lemma based on Proposition 2) at each point, and if the UCB of the long-run expected profit at point $S$ is lower than the LCB of the long-run expected profit at point $S$', then we can conclude that the $S$ is dominated by $S$', with a very high probability, and we can trim the intervals accordingly. We present the learning algorithm in Agrawal and Jia (2019) as follows:

**Algorithm** Learning Algorithm in Agrawal and Jia (2019)

**Initialization.** Initialize the algorithm with the initial interval of base-stock levels $[0, \bar{S}]$, the lead time $L$, and the planning horizon $T$. Set $l_1 = 0, r_1 = \bar{S}$.
**for** epochs $k = 1, 2, \ldots,$ **do**
  Let $w_k = r_k - l_k, x_l = l_k + w_k/4, x_c = l_k + w_k/2, x_r = l_k + 3w_k/4$.
  **for** round $i = 1, 2, \ldots,$ **do**
    Let $\gamma_i = 2^{-i}$ and $N = \frac{\log T}{\gamma_i^2}$.
    If the initial inventory position (total inventory) is higher than $x_l$, order nothing and wait until it drops below $x_l$.
    Test the base-stock level $x_l$ for $N$-periods and observe the $N$-period's average profit as $C_l$. Construct the LCB and UCB of $x_l$ as

$$LB^l = C_l - \tfrac{H\gamma}{2} \; UB^l = C_l + \tfrac{H\gamma}{2},$$

    where $H = 576 \max(h, p)(L+1)U$. Repeat the same for $x_c$ and $x_r$ to get the $LB^c$, $UB^c$, $LB^r$, and $UB^r$. If the total planning horizon $T$ has been reached, then stop.
    **if** $\min(UB^l, UB^r) \le \max(LB^l, LB^c, LB^r)$ **then**
      **if** $UB^l \le UB^r$ **then**
        $l_{k+1} = x_l, r_{k+1} = r_k$.
      **else**
        $l_{k+1} = l_k, r_{k+1} = x_r$.
      **end if**
      Goes to next epoch $k+1$. □
    **else**
      Goes to next round $i+1$.
    **end if**
  **end for**
**end for**

We can see that the algorithm gradually shrinks the interval $[l_k, r_k]$ in each epoch, by testing in each round $i$, with an increasing test length until a tie is broken. Note that we present the algorithm in terms of profit maximization instead of the pseudo-cost minimization version as in their paper. The regret of the algorithm is $O(LT)$, and we omit the regret analysis of the learning algorithm. We can see that the search method with the right concentration propriety can also be an efficient method for the single-dimensional learning problem.

## 11.5 High Dimensionality Example: Multiproduct Inventory Model with Customer Choices

We start with a general multiproduct periodic-review inventory model. When a customer arrives and decides what to purchase, her decision can depend on the availability of multiple products. For instance, if one customer intends to buy product A which is out of stock, then she may purchase product B, which is a substitute for A. Or, if another customer plans to purchase both A and C at the same time, but product C is not available, then she may decide not to purchase at all. Notice that in this model, the customers' purchase decisions depend on the availability of more than one product. If customers are heterogeneous in their choice preferences, then customers' arrival sequence will also influence the demand.

Consider $I$ types of products. From period $t = 1, \ldots, T$, the initial inventory level is denoted by $X$. The firm needs to decide the order-up-to level $Y$. Note that $X$ and $Y$ are vectors. In each period, a random number of customers arrive sequentially. Let $N$ denote the total number of customers on a sample path. Each customer $n = 1, \ldots, N$ may choose from the products that are available at the moment she arrives. The vector of inventory levels observed by customer $n$ is $X^n = (x_1^n, \ldots, x_I^n)$. At the beginning of period $t$, the inventory level is $X^1 = Y$. At the end of period $t$, the inventory level is $X^{N+1}$. We use $A^n$ to represent the availability of products faced by the $n$-th customer. Hence, $A^n$ is a vector with binary entries. For all $i = 1, \ldots, I$, $A_i^n = 1$ if $x_i^n > 0$, otherwise $A_i^n = 0$. Denote the type of the $n$-th customer by $U^n$. The purchasing decision of the $n$-th customer is denoted by $d(A^n, U^n) \in \{0, 1\}^I$, which is affected by the customer's type and the inventory availability faced by this customer. Let $\omega = \{U^n : n = 1, \ldots, N\}$ denote the sample path. We assume $\omega$ is a sample from some probability distribution space $(\Omega, \mathcal{F}, P)$ with $P(N < \infty) = 1$. In any period $t$, the firm determines the target inventory level $Y_t \geq X_t$ based on the starting inventory $X_t = X_{t-1}^{N+1}$. The firm's optimization problem can be formulated as a dynamic program. For $t = 1, \ldots, T$, we have

$$G_t^*(X_t) = \max_{Y_t \geq X_t, Y_t \in \mathcal{Y}} \rho(Y_t) + E\left[G_{t+1}^*(X_t^{N+1})\right]. \tag{11.3}$$

In each period, the order-up-to level $Y$ may be restricted by some constraints. For example, the total inventory cannot exceed the warehouse capacity. These restraints

are captured by the constraint set $\mathcal{Y}$. Given inventory level $Y_t$ and demand outcome $\omega$, the sample path profit in each period is $f(Y_t, \omega)$, which includes the total revenue minus the holding cost and possibly some other cost terms. The per-period expected profit is given by

$$\rho(Y_t) = E_\omega \left[ f(Y_t, \omega) \right]. \tag{11.4}$$

The state transition can be computed recursively using $X_t^{n+1} = X_t^n - d(A^n, U^n), n = 1, \ldots, N$. The boundary condition is given by $G_{T+1}^*(\cdot) \equiv 0$.

Since the demand is identically distributed in each period, the optimal policy of this problem is a myopic one, as formally stated in the following proposition. The proof follows Theorem 6.1 of Porteus (2002).

**Proposition 3** *Let $Y^*$ be an optimal solution of $\max_{Y \in \mathcal{Y}} \rho(Y)$. The optimal policy of problem (11.3) uses $Y^*$ as base-stock level for all $t = 1, \ldots, T$.*

If we have full information about demand, we can solve the optimization problem $\max_{Y \in \mathcal{Y}} \rho(Y)$ and apply the optimal base-stock policy $Y^*$ in each period. However, in practice, the decision-maker does not know the distribution of $\omega$ or the functional form of $d(\cdot, \cdot)$ and needs to learn them on the fly. This is the focus of this subsection.

Before we present the algorithms for solving this online inventory control problem, we first point out the two major difficulties. One difficulty is that the demand observations are both *censored* and *partial*. Censored demand observations are due to lost-sales, which is similar to other inventory models discussed in this chapter. Here, we focus on explaining why the demand observations are partial. When we observe a customer's purchasing decision, it is only for one particular inventory state. We cannot observe this customer's purchasing decisions with all possible inventory states. For example, suppose a customer who wants to purchase product 1 may purchase product 2 as a substitute if product 1 is not available. If we have no inventory for product 1 and one unit of inventory for product 2, we can observe that this customer ends up purchasing one unit of product 2. However, we do not know that if we had one unit of inventory for product 1, this customer would purchase product 1 instead. Therefore, we cannot observe this customer's true preferences even if we do not encounter lost-sales.

The other difficulty is the high dimensionality of this problem. Since we do not assume any explicit parametric form for the customer choice model, the purchasing decision function $d(\cdot, \cdot)$ can be extremely complex and difficult to learn. The action space can also be huge, even with a moderate number of products. To see this, suppose we naively apply the multi-armed bandit algorithm and treat each feasible inventory level as one arm, the total number of arms will increase exponentially with the number of products. A naive implementation of the UCB algorithm is given as follows:

**Algorithm**

1: Initialization: Select each policy $Y \in \mathcal{Y}$ once.
2: In each period $t$, use $n_t(Y)$ and $\bar{\rho}_t(Y)$ to construct the optimistic reward for each $Y$:

$$\bar{\rho}_t(Y) + \beta \sqrt{\frac{2 \ln t}{n_t(Y)}},$$

where $\beta$ is an algorithm parameter.
3: Select policy:

$$Y_t \in \arg\max_{Y \in \mathcal{Y}} \left\{ \bar{\rho}_t(Y) + \beta \sqrt{\frac{2 \ln t}{n_t(Y)}} \right\}.$$

4: Update $n_t(Y_t)$ and $\bar{\rho}_t(Y_t)$. Go to Step 2. □

Due to the high dimensionality of $Y$, this naive-UCB algorithm may have an extremely slow convergence rate. Suppose there are four products where the inventory level of each product can be $1, 2, \ldots, 10$. There are $10{,}000$ arms in total, and this naive-UCB algorithm needs to use $10{,}000$ periods to select each arm at least once to initialize.

In order to expedite the learning, we need to make better use of the sales data. However, the censored and partial observation of demand seems to make it very difficult. In the following, we will present two improvement ideas.

**Improvement idea 1:** In any period $t$, after an inventory level $Y$ is implemented and a sample path profit $f(Y, \omega)$ is observed, we identify $f(Y', \omega)$ with $Y' \neq Y$ which can be simulated without any bias. For example, consider a single-product inventory system; once we use the base-stock level $Y = 10$ for one period and observe $f(Y, \omega)$, we can also simulate $f(Y', \omega)$ for all $Y' = 1, \ldots, 9$. In this way, every time we observe a sample path profit with a certain inventory level, we can update the sample path profit associated with other inventory levels as well and thus improve the usage of sales data.

**Improvement idea 2:** When we calculate the expected profit with inventory level $Y$, we gather the information from all the $Y'$ that is "close enough" to $Y$ as an approximation. As we have more information, we can gradually shrink the distance between $Y$ and $Y'$. For example, consider a two-product system, the sequence of sales data under policy $Y = (10, 10)$ might be used to approximately estimate the profit under policy $Y' = (9, 11)$.

Figure 11.2 illustrates how these two improvements fit into the learning algorithm. Generally speaking, the implementation of a UCB algorithm consists of four steps. The two steps below are the inventory dynamics, while the two steps above are how the algorithm stores and uses the data. The first idea focuses on improving the information obtained from each demand observation. We make use of the simulated system to provide us unbiased estimations. Hence, all the data stored in the "database" of the UCB algorithm are always unbiased. The second idea focuses on improving how to use the unbiased data stored in the database. In particular, for

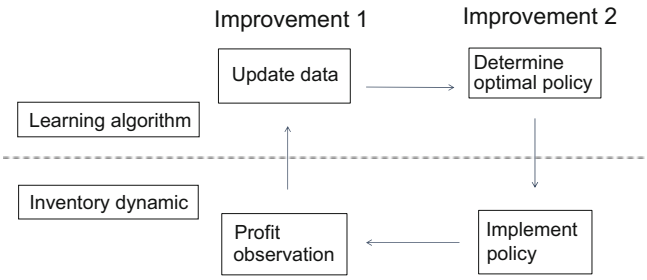


**Fig. 11.2** How Improvements 1 and 2 fit into the learning algorithm

each inventory level, we take the data from all inventory levels that are close by and combine them together to construct the UCB term. These two improvement ideas can be integrated together to improve the efficiency of the UCB algorithm. However, the caveat is that we need to keep the *independence* of random samples to ensure that the UCB algorithm remains valid. This is because the first improvement idea introduces multiple data points based on one particular sample path. We need to make sure that when we calculate the expected profit associated with any inventory level, we cannot use multiple observations generated from the same sample paths. In the following, we will demonstrate how to integrate these two ideas effectively while maintaining independent samples with an inventory substitution model.

### *11.5.1 Inventory Substitution*

In this subsection, we consider a specific customer choice model where a customer can purchase a substitute product if the product she initially wants is not available. This is often called a customer-driven, stock-out-based inventory substitution model (see Mahajan and Van Ryzin, 2001). We use $[I]$ to denote the set of product $\{1, \ldots, I\}$. The per-unit price for these products is $p_1, \ldots, p_i, \ldots, p_I$, respectively. Demands arrive at the beginning of each period $t$. If any product's demand cannot be satisfied, the customer can choose to buy a different product as a substitute.

We adopt a very general choice model to capture this stock-out-based substitution behavior: the ranking-based choice rule (see Mahajan and Van Ryzin, 2001; Honhon et al., 2010, 2012; Honhon and Seshadri, 2013, among others). Each customer $n$ has a customer type $U^n = (U_0^n, U_1^n, \ldots, U_I^n)$, where each $U_i^n$ is the utility assigned to product $i$, and $U_0^n$ is the utility of no purchase. Customer $n$ makes the purchase decision $d(A^n, U^n)$ according to her utility vector $U^n$ and the availability of products. Specifically, $d(A^n, U^n) = 0$ if $U_0^n \geq U_i^n$ for all $i = 1, \ldots, I$. Otherwise $d_i(A^n, U^n) = 1$ for $i = \arg\max_{i \geq 1, A_i^n = 1} U_i^n$, and $d_i(A^n, U^n) = 0$ for all other $i$. This choice rule means that if a customer has a personal rank for all products. When she arrives, she will first check whether her favorite product is available. If it is available, she will purchase this product; otherwise, she will continue to see

whether her second favorite product is available. She will continue this process until she purchases a product to leave without purchasing. Many existing customer choice models, such as the locational choice model or the multi-nomial logit model, can be viewed as special cases of this ranking-based choice rule.

After a customer has made a purchase decision, the firm observes the sale. Note that the firm cannot observe anything if the customer does not purchase at all. The firm can only observe the customer's final purchase decision, but not the substitution thinking process in the customer's mind. Let $u_i(Y, \omega)$ denote the number of products $i$ sold on the sample path $\omega$ given initial inventory levels $Y$. Define a vector $u(Y, \omega) = (u_1(Y, \omega), \ldots, u_I(Y, \omega))$. Let $\xi_i^n(X, \omega)$ denote the total sales of product $i$ up to customer $n$ given the inventory level observed by customer $n$ is $X$ and the demand sample path is $\omega$. We have $\xi_i^0 = 0$, $X^1 = Y$. The sales and inventory dynamics can be computed as follows:

$$\xi_i^n(X^n, \omega) = d_i(A^n, U^n) + \xi_i^{n-1}(X^{n-1}, \omega),$$
$$X^{n+1} = X^n - d(A^n, U^n).$$

The total sales of each product are $u_i(Y, \omega) = \xi_i^N(Y, \omega), \ \forall\ i \in [I]$. The sample path profit $f(Y, \omega)$ is given by

$$f(Y, \omega) = p^\intercal u(Y, \omega) - h^\intercal (Y - u(Y, \omega)),$$

where $p = (p_1, \ldots, p_I)^\intercal$ is the unit price vector and $h = (h_1, \ldots, h_I)^\intercal$ is the unit-holding cost vector. The firm wants to maximize the total expected profit $\rho(Y) = E[f(Y, \omega)]$.

The optimization of the multiproduct inventory system with ranking-based substitutions has been studied in the literature. As our focus is on the learning algorithm for this model, we refer the interested reader to Mahajan and van Ryzin (2001); Honhon et al. (2010, 2012); Honhon and Seshadri (2013); Chen and Gong (2018) for detailed discussions on the optimization of this model. One important point worth mentioning is that, as shown by Theorem 2 in Mahajan and Van Ryzin (2001), the profit function is not component-wise concave. The lack of concavity renders tools such as the stochastic gradient approach powerless for the online learning version of this model. Therefore, we turn to use the UCB-type learning algorithm by treating each possible order-up-to inventory level as an arm. For this purpose, assume that the order-up-to level for each product $Y_i$ is chosen from a finite set of admissible levels $\{y_1, \ldots, y_k, \ldots, y_K\}$, where $y_K$ is the highest feasible level. Let $\mathcal{Y}$ denote this set of all admissible arms. Therefore, there are $K^I$ arms in the feasible set $\mathcal{Y}$. The high dimensionality of this problem leads to an exponentially large number of arms. This will make a naive implementation of the UCB algorithm extremely slow to converge. In the following, we present the improved-UCB algorithm.

**Algorithm** Improved-UCB for Inventory Substitution

1: Input: $\gamma$
2: Initialization: $t = 1$, $t_\ell = 0$ (starting period of episode $\ell$)
3: **for** episodes $\ell = 1, 2, \ldots,$ **do**
4: $t_\ell \leftarrow t$, $\gamma_{t_\ell} = \frac{\gamma}{\sqrt{t_\ell}}$.
5: (**Improvement 2**) Let the *virtual counter* $\hat{n}_{i,t}(Y) = \sum_{Y': Y'_i = Y_i,\ \|Y'-Y\|_1 \le \gamma_{t_\ell}} n_{i,t}(Y')$ for all $i \in [I]$, $y \in \mathcal{Y}$.
The corresponding estimator is

$$\hat{\rho}_{i,t_\ell}(Y) = \frac{\sum_{Y': Y'_i = Y_i, |Y'-Y|_1 \le \gamma_{t_\ell}} n_{i,t}(Y') \cdot \rho_{i,t}(Y')}{\hat{n}_{i,t_\ell}(Y)}. \tag{11.5}$$

6: The UCB is given by $U_{t_\ell}(Y) = \sum_{i\in[I]}[\hat{\rho}_{i,t_\ell}(Y) + \hat{c}_{i,t_\ell}(Y)]$, where $\hat{c}_{i,t_\ell}(Y) = y_K \cdot (\bar{p} + \bar{h}) \cdot \sqrt{\frac{2 \ln t_\ell}{\hat{n}_{i,t_\ell}(Y)}}$ for all $Y \in \mathcal{Y}$, $i \in [I]$, $\bar{p} = \max_{i\in[I]} p_i$, $\bar{h} = \max_{i\in[I]} h_i$. Pick $Y_\ell = \arg\max U_{t_\ell}(Y)$. This order-up-to level $Y_\ell$ will be used in the whole episode $\ell$.
7: **while** $n_t(Y) \le 2n_{t_\ell}(Y)$ for all $Y \in \mathcal{Y}$ **do**
8: Apply the target order-up-to level $Y_\ell$. If this is not feasible, then use $Y = \min\{Y_\ell, X_t\}$ as the target order-up-to level.
9: $t \leftarrow t + 1$
10: Update the relevant counter $n_t(Y)$.
11: (**Improvement 1**) Update the counters $n_{i,t}(Y)$ for all $i \in [I]$. Update the estimator $\rho_{i,t}(Y)$ using the observed sample path profit in this period.
Update the counters $n_{i,t}(Y')$ for all $i \in [I]$ where $Y'_i < Y_i, Y'_{-i} = Y_{-i}$. Update the estimator $\rho_{i,t}(Y')$ using the simulated sample path profit.
12: **end while**
13: **end for**

The algorithm proceeds in episodes of increasing length. Within each episode $\ell$, the target order-up-to level is the same, denoted by $Y_\ell$. For each possible inventory level $Y \in \mathcal{Y}$, we use $n_t(Y)$ to count the number of times this inventory level $Y$ is chosen before time $t$. An episode is ended if at least one counter gets doubled. There are two reasons why we use episodes instead of simply updating the target inventory level in every period. One reason is that new target order-up-to inventory levels may not be reachable if the current inventory level is higher than it. Using the same target inventory level in the entire episode ensures that the target order-up-to level is reached for the majority of the time. The second reason is that the computational time can be reduced since we do not update the policy too frequently without a significant performance loss. Within each episode, steps 4 to 6 are about generating the UCB and finding the new order-up-to level for the episode, while steps 8–11 are for updating relevant counters and estimators after observing the sample path profit.

We will firstly discuss steps 8–11. Step 8 sets the inventory level to the target order-up-to level $Y_\ell$ unless it is lower than the current inventory level, in which case the inventory level remains unchanged. Step 11 is a critical step that utilizes Improvement Idea 1. First of all, we keep records of the counters, sample path profits, and expected profits *for each product* $i \in [I]$, with a subscript $i$ in the corresponding notation. In each time period $t$, if an order-up-to level

$Y = (Y_1, \ldots, Y_I)$ is applied, we can observe the sample path profit for each product $i \in [I]$, denoted by $f_i(Y, \omega)$. Then, obviously we can update the counter $n_{i,t}(Y)$ and the estimator $\rho_{i,t}(Y)$ using the sample path profit $f_i(Y, \omega)$. Moreover, we will also update counters $n_{i,t+1}(Y')$ and estimators $\rho_{i,t+1}(Y')$, where $Y'_i < Y_i, Y'_{-i} = Y_{-i}$ using the *simulated sample path profit*, denoted by $\hat{f}_i(Y'; Y, \omega)$. Notice that the simulated sample path profit $\hat{f}_i(Y'; Y, \omega)$ is *unbiased* in the sense that $\hat{f}_i(Y'; Y, \omega) = f_i(Y', \omega)$. This is because before product $i$ runs out of stock, all customers' choices when the order-up-to level is $Y'$ are the same as those when the order-up-to level is $Y$.

For instance, suppose that $Y = (4, 3)$, which means that the inventory level is 4 for products 1 and 3 for product 2. Then, given some demand sample path $\omega$, we can observe the sample path profit $f_1(Y, \omega)$ and $f_2(Y, \omega)$. We can also obtain the unbiased simulated sample path profit $\hat{f}_1(Y'; Y, \omega)$ for $Y' = (3, 3)$. Since $\hat{f}_1(Y'; Y, \omega) = f_1(Y', \omega)$, we can use this simulated sample path profit to update the profit estimator $\rho_1(Y')$. In other words, after setting the inventory level to (4, 3) and observing sales, we can have a better estimation of profits generated from product 1 if the inventory level was (3, 3) as well. This helps us extract more information from the sales data.

Step 4 sets a parameter $\gamma_{t_\ell}$ used in Step 5, which is another critical step utilizing Improvement Idea 2. To generate the estimator as inputs for the UCB in period $t$, for each product $i$ the algorithm includes all observations associated with $n_{i,t}(Y')$ such that $Y'_i = Y_i$ and $\|Y' - Y\|_1 \le \gamma_{t_\ell}$. This is because although these observations are biased, the biases are bounded due to the Lipchitz property of the profit function. We formally state this in the following proposition.

**Proposition 4** *Given initial inventory levels $Y$ and $Y'$, if $\|Y' - Y\|_1 \le \gamma$, then we have $\left|\rho_i(Y) - \rho_i(Y')\right| \le \gamma \cdot \max\{p_i, h_i\}$ for any product $i \in [I]$.*

To prove Proposition 4, define $\eta_i^n(Y, \omega)$ as the total sales of product $i$ from customer $n$ to the last customer given the inventory level observed by consumer $n$ is $Y$ and the demand sample path is $\omega$. We have $\eta_i^n(X^n, \omega) = d_i(A^n, U^n) + \eta_i^{n+1}(X^{n+1}, \omega)$, $X^{n+1} = X^n - d(A^n, U^n)$, $\eta_i^{N+1} = 0$, $X^1 = Y$. Let $u_i(Y, \omega) = \eta_i^1$. Then, it is not difficult to show that if $Y' = Y + e_i$, then $u_i(Y, \omega) \le u_i(Y', \omega) \le u_i(Y, \omega) + 1$ for any sample path $\omega$. If $Y' = Y + e_j$, $j \ne i$, then $u_i(Y, \omega) - 1 \le u_i(Y', \omega) \le u_i(Y, \omega)$ for any sample paths $\omega$. Thus, $\left|u_i(Y, \omega) - u_i(Y', \omega)\right| \le \gamma \ \forall \omega$ if $\left|Y' - Y\right|_1 \le \gamma$. Since $f_i(Y, \omega) = p_i u_i(Y, \omega) - h_i(Y_i - u_i(Y, \omega))$, we have $|f_i(Y, \omega) - f_i(Y', \omega)| \le \max\{p_i, h_i\}\gamma$ for any sample path $\omega$. Therefore, $\left|\rho_i(Y) - \rho_i(Y')\right| \le \max\{p_i, h_i\}\gamma$.

In Step 5, we require that $Y'_i = Y_i$, which is not needed in Proposition 4. This additional condition is to ensure that all the observations are *independent*. Due to Step 11, for each product $i$, we may update multiple estimators in one period. Then, in Step 5, we need to ensure that no estimators included have been updated simultaneously in any historical periods.

**Theorem 5** *There exists a non-empty set of parameters $\gamma$, such that the regret of Algorithm in Sect. 11.5.1 is bounded above by*

$$\min\left\{C_1 \cdot I \cdot \sqrt{K^I \cdot T \cdot \ln T},\ C_2 \cdot I \cdot \sqrt{K \cdot T \cdot \ln T} + \Delta_s \cdot T\right\}, \tag{11.6}$$

*where $C_1$ and $C_2$ do not depend on $K$, $I$, or $T$, and $\Delta_s$ does not depend on $K$ or $T$.*

Theorem 5 demonstrates the theoretical performance guarantee of the algorithm. The average regret is upper bounded by the minimum of two terms. These two terms represent the trade-off from choosing a proper $\gamma$. When $\gamma$ is relatively small, we introduce less bias in Step 5 of the algorithm but at the same use fewer data points for estimation. In this case, the regret is bounded above by $C_1 \cdot I \cdot \sqrt{K^I \cdot T \cdot \ln T}$. The term $K^I$ is largely due to not using enough data points for estimation, which may lead to slow convergence. On the other hand, when $\gamma$ is very large, the algorithm almost ignores the substitution effect in the system and tries to learn and optimize for each product independently. In this case, the algorithm uses more data points for estimation but inevitably introduces more biases into the estimation. This leads to the second term in (11.6). Note that $\Delta_s$ captures the regret due to the biases introduced. It can be further showed that $\Delta_s$ does not depend on $K$ or $T$ and $\Delta_s$ is upper bound by $O(I^2)$ (Gao and Zhang, 2021).

### *11.5.2 Numerical Example*

In order to have an efficient convergence, the algorithm needs to strike a balance by a proper choice on $\gamma$. In the following, we will show the performance of our algorithm with different choices of $\gamma$ and problem parameters of the inventory substitution problem.

**Experiment Settings** Consider an inventory system with three substitutable products: $\{A, B, C\}$. Assume that each customer has a preference list. In this numerical experiment, there are 15 types of customers with the preference lists $\{A\}$, $\{B\}$, $\{C\}$, $\{A, B\}$, $\{A, C\}$, $\{B, A\}$, $\{B, C\}$, $\{C, B\}$,$\{A, B, C\}$, $\{A, C, B\}$, $\{B, A, C\}$, $\{B, C, A\}$, $\{C, A, B\}$,$\{C, B, A\}$. If a customer has the preference list $\{A\}$, she will only purchase product $A$ and will not substitute to $B$ or $C$. A customer with preference list $\{A, B, C\}$ will first try to purchase $A$, and when facing the stock-out of $A$, she will try to purchase $B$, and when facing the stock-out of both $A$ and $B$, she will try to purchase $C$. Each type of customer arrives according to a Poisson process, with randomly generated rates. The unit-holding cost is $\{1, 2, 3\}$ for the three products. There are four possible price schemes for the three products: $\{5, 10, 15\}$, $\{10, 20, 30\}$, $\{20, 40, 60\}$, and $\{40, 80, 120\}$. Denote them as Case 1 to 4, respectively. The maximum inventory level is assumed to be 10. We use 5000 sample paths to estimate the average regret of the algorithm. Motivated by Russo and Van Roy (2014), we also tune the algorithm with a parameter $\lambda$, which is multiplied

**Table 11.3** Expected Average Regret for Algorithm in Sect. 11.5.1 at $T = 50, 100, 100, 500, 1000$, and 5000

| Case | $\gamma$ | $T = 50$ | 100 | 200 | 500 | 1000 | 5000 | Optimal Policy |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 20.41% | 18.28% | 13.06% | 12.13% | 11.10% | 10.79% | (3,4,5) |
| | $2^6$ | 13.46% | 14.88% | 12.59% | 10.15% | 8.14% | 6.53% | |
| | $2^8$ | 13.29% | 11.95% | 10.74% | 9.75% | 9.32% | 5.56% | |
| | $2^{10}$ | 13.96% | 12.21% | 11.04% | 10.03% | 9.53% | 8.67% | |
| | $\infty$ | 13.28% | 11.68% | 10.51% | 9.32% | 8.81% | 7.94% | |
| 2 | 0 | 16.26% | 15.40% | 10.56% | 7.46% | 7.02% | 6.79% | (3,4,6) |
| | $2^6$ | 9.30% | 9.66% | 7.86% | 6.55% | 5.44% | 4.79% | |
| | $2^8$ | 10.06% | 9.53% | 9.22% | 8.53% | 8.02% | 4.08% | |
| | $2^{10}$ | 10.40% | 9.78% | 9.20% | 8.41% | 7.94% | 7.12% | |
| | $\infty$ | 10.36% | 9.99% | 9.53% | 8.91% | 8.49% | 7.57% | |
| 3 | 0 | 10.08% | 7.05% | 6.69% | 6.02% | 5.90% | 5.33% | (3,5,6) |
| | $2^6$ | 8.42% | 8.06% | 5.86% | 4.81% | 3.79% | 3.76% | |
| | $2^8$ | 8.50% | 8.51% | 8.42% | 7.90% | 7.43% | 3.23% | |
| | $2^{10}$ | 8.56% | 8.47% | 8.25% | 7.86% | 7.52% | 6.88% | |
| | $\infty$ | 8.81% | 8.72% | 8.40% | 8.02% | 7.66% | 6.96% | |
| 4 | 0 | 9.10% | 6.72% | 6.42% | 5.41% | 5.19% | 3.97% | (5,6,7) |
| | $2^6$ | 7.40% | 6.76% | 4.31% | 3.07% | 2.25% | 2.58% | |
| | $2^8$ | 6.59% | 6.73% | 6.63% | 6.28% | 5.75% | 2.04% | |
| | $2^{10}$ | 6.32% | 6.37% | 6.32% | 6.08% | 5.75% | 5.23% | |
| | $\infty$ | 7.03% | 6.93% | 6.73% | 6.46% | 6.08% | 5.51% | |

to the UCB term, which is set to be $2^{-6}$ for all the cases. In order to test how the performance depends on the choice of parameter $\gamma$, we set $\gamma \in \{0, 2^6, 2^8, 2^{10}, \infty\}$ for all four cases.

**Convergence Results** Table 11.3 summarizes the numerical results of all the testing instances. For each case, the best $\gamma$ is highlighted in boldface. Note that a $\gamma$ with value 0 or $\infty$ never achieves the best performance in any of the cases we tested. This shows the importance of choosing a proper $\gamma$. For all four cases, we can observe that the convergence is faster when the profit/holding cost ratio is larger. In practice, the per-unit profit is usually much higher than the per-unit one-period holding cost. In conclusion, we can see that our algorithm combining Improvement Ideas 1 and 2 speeds up the convergence of the algorithm. For a problem with 1000 feasible inventory policies, the expected average regret for $T = 50$ periods is around 10% and keeps decreasing as $T$ increases.

## References

Agrawal, S., & Jia, R. (2019). Learning in structured MDPs with convex cost functions: Improved regret bounds for inventory management. In *Proceedings of the 2019 ACM conference on economics and computation* (pp. 743–744).

Besbes, O., & Muharremoglu, A. (2013). On implications of demand censoring in the newsvendor problem. *Management Science, 59*(6), 1407–1424.

Chao, X., Gong, X., Shi, C., & Zhang, H. (2015). Approximation algorithms for perishable inventory systems. *Operations Research, 63*(3), 585–601.

Chao, X., Gong, X., Shi, C., Yang, C., Zhang, H., & Zhou, S. X. (2018). Approximation algorithms for capacitated perishable inventory systems with positive lead times. *Management Science, 64*(11), 5038–5061.

Chen, B., & Chao, X. (2019). Dynamic inventory control with stockout substitution and demand learning. *Management Science, 66*(11), 5108–5127.

Chen, T., & Gong, X. (2018). Optimal control policy for stochastic inventory models with two substitutable products, working paper. Available at SSRN: https://ssrn.com/abstract=3217017.

Chen, X., Pang, Z., & Pan, L. (2014). Coordinating inventory control and pricing strategies for perishable products. *Operations Research, 62*(2), 284–300.

Cheung, W. C., & Simchi-Levi, D. (2019). Sampling-based approximation schemes for capacitated stochastic inventory control models. *Mathematics of Operations Research, 44*(2), 668–692.

Cooper, W. L. (2001). Pathwise properties and performance bounds for a perishable inventory system. *Operations Research, 49*(3), 455–466.

Fries, B. (1975). Optimal ordering policy for a perishable commodity with fixed lifetime. *Operational Research, 23*(1), 46–61.

Gao, X., & Zhang, H. (2021). An efficient learning framework for multi-product inventory systems with customer choices. Available at SSRN 3775303.

Honhon, D., & Seshadri, S. (2013). Fixed vs. random proportions demand models for the assortment planning problem under stockout-based substitution. *Manufacturing and Service Operations Management, 15*(3), 378–386.

Honhon, D., Gaur, V., & Seshadri, S. (2010). Assortment planning and inventory decisions under stockout-based substitution. *Operations Research, 58*(5), 1364–1379. doi:10.1287/opre.1090.0805.

Honhon, D., Jonnalagedda, S., & Pan, X. A. (2012). Optimal algorithms for assortment selection under ranking-based consumer choice models. *Manufacturing and Service Operations Management, 14*(2), 279–289.

Huh, W. H., & Rusmevichientong, P. (2009). A non-parametric asymptotic analysis of inventory planning with censored demand. *Mathematics of Operations Research, 34*(1), 103–123.

Huh, W. T., Janakiraman, G., Muckstadt, J. A., & Rusmevichientong, P. (2009a). An adaptive algorithm for finding the optimal base-stock policy in lost sales inventory systems with censored demand. *Mathematics of Operations Research, 34*(2), 397–416.

Huh, W. T., Janakiraman, G., Muckstadt, J. A., & Rusmevichientong, P. (2009b). Asymptotic optimality of order-up-to policies in lost sales inventory systems. *Management Science, 55*(3), 404–420.

Huh, W. H., Rusmevichientong, P., Levi, R., & Orlin, J. (2011). Adaptive data-driven inventory control with censored demand based on Kaplan-Meier estimator. *Operations Research, 59*(4), 929–941.

Janakiraman, G., & Roundy, R. O. (2004). Lost-sales problems with stochastic lead times: Convexity results for base-stock policies. *Operations Research, 52*(5), 795–803.

Levi, R., Pál, M., Roundy, R. O., & Shmoys, D. B. (2007). Approximation algorithms for stochastic inventory control models. *Mathematics of Operations Research, 32*, 284–302.

Levi, R., Perakis, G., & Uichanco, J. (2015). The data-driven newsvendor problem: New bounds and insights. *Operations Research, 63*(6), 1294–1306.

Li, Q., & Yu, P. (2014). Multimodularity and its applications in three stochastic dynamic inventory problems. *Manufacturing and Service Operations Management, 16*(3), 455–463.

Mahajan, S., & van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research, 49*(3), 334–351.

Mahajan, S., & Van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research, 49*(3), 334–351.

Nahmias, S. (1975). Optimal ordering policies for perishable inventory-II. *Operational Research, 23*(4), 735–749.

Nahmias, S. (1976). Myopic approximations for the perishable inventory problem. *Management Science, 22*(9), 1002–1008.

Nahmias, S. (1977a). Comparison between two dynamic perishable inventory models. *Operations Research, 25*(1), 175–184.

Nahmias, S. (1977b). Higher order approximations for the perishable inventory problem. *Operations Research, 25*(4), 630–640.

Nandakumar, P., & Morton, T. E. (1993). Near myopic heuristics for the fixed-life perishability problem. *Management Science, 39*(12), 1490–1498.

Porteus, E. L. (2002). *Foundations of stochastic inventory theory*. California: Stanford University Press.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243.

Shi, C., Chen, W., & Duenyas, I. (2016). Nonparametric data-driven algorithms for multiproduct inventory systems with censored demand. *Operations Research, 64*(2), 362–370.

Yuan, H., Luo, Q., & Shi, C. (2021). Marrying stochastic gradient descent with bandits: Learning algorithms for inventory systems with fixed costs. *Management Science, 67*(10), 6089–6115.

Zhang, H., Shi, C., & Chao, X. (2016). Technical note—approximation algorithms for perishable inventory systems with setup costs. *Operations Research, 64*(2), 432–440.

Zhang, H., Chao, X., & Shi, C. (2018). Technical note—perishable inventory systems: Convexity results for base-stock policies and learning algorithms under censored demand. *Operations Research, 66*(5), 1276–1286.

Zhang, C., Ayer, T., & White, C. C. (2019). 2-approximation policies for perishable inventory systems when FIFO is an optimal issuing policy. Available at SSRN 3469852.

Zhang, H., Chao, X., & Shi, C. (2020). Closing the gap: A learning algorithm for lost-sales inventory systems with lead times. *Management Science, 66*(5), 1962–1980.

Zipkin, P. (2008). On the structure of lost-sales inventory models. *Operations Research, 56*(4), 937–944.

# Chapter 12
# Joint Pricing and Inventory Control with Demand Learning


**Boxiao Chen**

## 12.1 Problem Formulation in General

Since the seminal paper of Whitin (1955), joint pricing and inventory control problems have attracted tremendous attention and been studied by hundreds of research papers in the literature. For a comprehensive review, see survey papers Petruzzi and Dada (1999), Elmaghraby and Keskinocak (2003), Yano and Gilbert (2005), and Chen and Simchi-Levi (2012). Traditional literature assumes the demand distribution is known and takes this information as model input, which is hardly satisfied in practice. In this chapter, we relax this assumption and discuss online algorithms to learn the demand only from historical data. As time goes by, the learning algorithms will learn the demand better and better, so that the solutions prescribed by the algorithms converge to the true optimal solution had the demand distribution been known.

In this section, we discuss the general setup for the problem of joint inventory and pricing. Consider a periodic review system in which a firm (e.g., a retailer) sells a non-perishable product over a planning horizon of $T$ periods. At the beginning of each period $t$, the firm observes on-hand inventory $x_t$ and determines an inventory order-up-to level $y_t$ and a price $p_t$, where $y_t \geq x_t,\ y_t \in \mathcal{Y} = [y^l, y^h]$ and $p_t \in \mathcal{P} = [p^l, p^h]$ with $y^l < y^h$ and $p^l < p^h$. For simplicity we assume that the system is initially empty, i.e., $x_1 = 0$. Demand for period $t$, denoted by $D_t(p_t)$, is stochastic and price dependent. Demand is satisfied as much as possible by on-hand inventory, and profits are collected by the firm. There might be a mismatch between supply and demand. If $y_t > D_t(p_t)$, any leftover inventories will be carried over to the next period, for each of which the firm pays a holding cost $h$. If $y_t < D_t(p_t)$,

B. Chen (✉)
College of Business Administration, University of Illinois Chicago, Chicago, IL, USA
e-mail: bbchen@uic.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_12

some demands are not fulfilled, and the firm pays a penalty cost $b$ for any unit of stockout. Per-unit ordering cost is normalized to 0 without loss of generality. The firm's objective is to maximize the $T$-period total profit.

If the distribution of $D_t(p_t)$ is known a priori to the firm (complete information scenario), then the optimization problem the firm wishes to solve is

$$\max_{\substack{(p_t,\, y_t) \,\in\, \mathcal{P} \times \mathcal{Y} \\ y_t \,\geq\, x_t}} \sum_{t=1}^{T} v(p_t, y_t), \tag{12.1}$$

where $v(p_t, y_t)$ is the instantaneous reward during period $t$. Let $V^*$ represent the maximum $T$-period expected profit generated from the optimal policy under complete information.

In practice, the demand distribution is unknown; therefore, the firm needs to develop an admissible policy which prescribes pricing and ordering decisions for each period. An admissible policy is represented by a sequence of prices and order-up-to levels, $\{(p_t, y_t), t \geq 1\}$, where $(p_t, y_t)$ depends only on realized data and decisions made prior to period $t$, and $y_t \geq x_t$, i.e., $(p_t, y_t)$ is adapted to the filtration generated by $\{(p_s, y_s), o_s : s = 1, \ldots, t-1\}$. Here $o_s$ represents the observable data of demand. Ideally, $o_s = D_s(p_s)$, meaning that demand is fully observable, but in some cases demand data is censored, which yields $o_s < D_s(p_s)$. Given any admissible policy $\pi$, the sequence of events for each period $t$ is described as follows:

1. At the beginning of period $t$, the retailer observes the initial inventory level $x_t$.
2. The retailer decides the selling price $p_t$ and the inventory level-up-to level $y_t \geq x_t$. New orderings, if there is any, arrive instantaneously.
3. Demand realizes and is satisfied to the maximum extent using on-hand inventory. Unsatisfied demand is backlogged or lost, and any leftover inventory is carried to the next period. The retailer observes data $o_t$.
4. At the end of period $t$, the retailer collects profit of the current period.

The firm's objective is to find an admissible policy to maximize the $T$-period total profit while learning the unknown demand distribution on the fly. The regret of policy $\pi$, denoted by $R^\pi(T)$, is defined as the total profit loss over $T$ periods, which is

$$R^\pi(T) = V^* - \mathbb{E}\left[\sum_{t=1}^{T} v(p_t, y_t)\right].$$

The smaller the regret, the better the policy.

In this chapter, we will discuss a number of models under the framework of joint inventory and pricing. These models differ in the following three dimensions.

1. Backlog versus lost-sales
   - In a backlog system, if $y_t < D_t(p_t)$, any unsatisfied demands will be backlogged and served in future periods, and $x_{t+1} = y_t - D_t(p_t)$.
   - In a lost-sales system, unmet demands will leave the market without any purchases, and $x_{t+1} = (y_t - D_t(p_t))^+$.
2. Unlimited price changes versus limited price changes
   - Most models we will discuss allow unlimited number of price changes, i.e., the retailer is allowed to change price every period.
   - We will discuss one model where the firm is not allowed to make price changes more than a certain number of times.
3. With versus without setup cost
   - If a setup cost is present, a fixed amount of fee will be charged whenever a positive amount of inventory is ordered.

In Sects. 12.2 and 12.3, we discuss the classic joint inventory and pricing problem with backlogged demand and lost sales, respectively. In Sect. 12.4, we consider scenarios with a limited number of price changes. In Sect. 12.5, we discuss the joint pricing and inventory control problem with setup cost. In Sect. 12.6, we discuss other models of joint pricing and inventory control that have been studied in the literature.

## 12.2 Nonparametric Learning for Backlogged Demand

In this section, we discuss the joint pricing and inventory control problem with backlogged demand, one of the most classical models under the topic of joint pricing and inventory control. We will discuss the model, algorithm, regret convergence results and proof sketch based on Chen et al. (2019).

Per-period demand can be either $D_t(p_t) = \lambda(p_t) + \epsilon_t$ (additive) or $D_t(p_t) = \lambda(p_t)\,\epsilon_t$ (multiplicative), where $\lambda(\cdot)$ is a strictly decreasing deterministic function and $\epsilon_t, t = 1, 2, \ldots, T$, are independent and identically distributed random variables with probability density function $f(\cdot)$ and cumulative distribution function $F(\cdot)$. Here we focus on the multiplicative demand form. Unsatisfied demands are backlogged, and one has $x_{t+1} = y_t - D_t(p_t)$ for all $t = 1, \ldots, T$. The instantaneous reward for period $t$ is $G(p_t, y_t) = p_t\mathbb{E}[D_t(p_t)] - h\mathbb{E}[y_t - D_t(p_t)]^+ - b\mathbb{E}[D_t(p_t) - y_t]^+$.

By Sobel (1981), myopic policy is optimal for this problem. Therefore, to optimize the $T$-period problem in (12.1), it suffices to solve the single-period problem

$$\max_{(p,\, y)\, \in\, \mathcal{P} \times \mathcal{Y}} \quad G(p, y), \tag{12.2}$$

where

$$G(p, y) = p\mathbb{E}[D(p)] - h\mathbb{E}[y - D(p)]^+ - b\mathbb{E}[D(p) - y]^+$$
$$= pe^{\lambda(p)}\mathbb{E}\left[e^{\epsilon}\right] - \left\{h\mathbb{E}\left[y - e^{\lambda(p)}e^{\epsilon}\right]^+ + b\mathbb{E}\left[e^{\lambda(p)}e^{\epsilon} - y\right]^+\right\}.$$

Let $Q(p, e^{\lambda(p)}) := \max_{y\in\mathcal{Y}} G(p, y)$, then problem (12.2) can be re-written as

**Problem CI**:

$$\max_{p\in\mathcal{P}} Q(p, e^{\lambda(p)})$$
$$:= \max_{p\in\mathcal{P}} \left\{pe^{\lambda(p)}\mathbb{E}\left[e^{\epsilon}\right] - \min_{y\in\mathcal{Y}} \left\{h\mathbb{E}\left[y - e^{\lambda(p)}e^{\epsilon}\right]^+ + b\mathbb{E}\left[e^{\lambda(p)}e^{\epsilon} - y\right]^+\right\}\right\}. \tag{12.3}$$

The inner optimization problem (minimization) determines the optimal order-up-to level that minimizes the expected holding and backlog cost for a given price $p$, and we denote it by $\overline{y}\left(e^{\lambda(p)}\right)$. The outer optimization solves for the optimal price $p$. Because $(p^*, y^*)$ is the optimal solution for (12.3), they satisfy $y^* = \overline{y}(e^{\lambda(p^*)})$.

The firm knows neither the function $\lambda(\cdot)$ nor the distribution of random variable $\epsilon_t$. In the backlog system, true demand realizations can be observed. Therefore, $o_t = D_t(p_t)$, and an admissible policy $(p_t, y_t)$ is adapted to the filtration generated by $\{(p_s, y_s), D_s(p_s) : s = 1, \ldots, t-1\}$.

**Learning Algorithm** A learning algorithm named DDA (shorthand for Data-Driven Algorithm) is proposed in Chen et al. (2019). DDA approximates $\lambda(p)$ by an affine function, and it constructs empirical and dependent error samples from the collected data, called centered samples. DDA divides the planning horizon into stages whose lengths are exponentially increasing (in the stage index). At the start of each stage, the firm sets two pairs of prices and order-up-to levels based on its current linear estimation of the demand-price function and the constructed centered samples of random error, and the collected demand data from this stage are used to update the linear estimation of the demand-price function and the empirical distribution of random error. These are then utilized to find the pricing and inventory decision for the next stage. The detailed algorithm design is presented in Algorithm 1.

As shown in Algorithm 1, for $i = 1, 2, \ldots$ in the DDA algorithm, iteration $i$ focuses on stage $i$ that consists of $2I_i$ periods. The algorithm sets the ordering quantity and selling price for each period in stage $i$ derived from the previous iteration. The first $I_i$ periods (from $t_i + 1$ to $t_i + I_i$) try to implement order-up-to $\hat{y}_{i,1}$ policy while the second $I_i$ periods try to implement order-up-to $\hat{y}_{i,2}$ policy. Because starting inventory level may be higher than the order-up-to level, $\hat{y}_{i,1}$ and $\hat{y}_{i,2}$ may not be achieved, and one challenge is to identify the impact of the carryover inventory constraint on the performance of a learning algorithm.

**Algorithm 1** Data-Driven Algorithm (DDA)

Input $v > 1$, $\rho > 0$ and $I_0 > 0$, and $\hat{p}_1, \hat{y}_{1,1}, \hat{y}_{1,2}$. Compute $I_1 = \lfloor I_0 v \rfloor$, $\delta_1 = \rho(2I_0)^{-\frac{1}{4}}$, and $\hat{p}_1 + \delta_1$.
**for** $i = 1, \ldots, n$ **do**
  **for** $t = t + i + 1, \ldots, t_i + I_i$ **do**
    Set $p_t = \hat{p}_i$, $y_t = \max\{\hat{y}_{i,1}, x_t\}$.
    Let $D_t = \log \tilde{D}_t(p_t)$.
  **end for**
  **for** $t = t_i + I_i + 1, \ldots, t_i + 2I_i$ **do**
    Set $p_t = \hat{p}_i + \delta_i$, $y_t = \max\{\hat{y}_{i,2}, x_t\}$.
    Let $D_t = \log \tilde{D}_t(p_t)$.
  **end for**
  Compute

$$(\hat{\alpha}_{i+1}, \hat{\beta}_{i+1}) = \operatorname*{argmin}_{\alpha,\beta} \left\{ \sum_{t=t_i+1}^{t_i+2I_i} \Big( D_t - (\alpha - \beta p_t) \Big)^2 \right\},$$

$$\eta_t = D_t - \frac{1}{I_i} \sum_{t=t_i+1}^{t_i+I_i} D_t, \qquad \text{for } t = t_i + 1, \ldots, t_i + I_i,$$

$$\eta_t = D_t - \frac{1}{I_i} \sum_{t=t_i+I_i+1}^{t_i+2I_i} D_t, \qquad \text{for } t = t_i + I_i + 1, \ldots, t_i + 2I_i.$$

  The data-driven optimization problem (Problem DD) is

$$\max_{(p,y)\in\mathcal{P}\times\mathcal{Y}} G_{i+1}^{DD}(p, y) = \max_{p\in\mathcal{P}} Q_{i+1}^{DD}\big(p, e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p}\big), \tag{12.4}$$

  where

$$G_{i+1}^{DD}\big(p, y\big) = p e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p} \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} e^{\eta_t}$$

$$- \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} \left( h \left( y - e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p + \eta_t} \right)^+ + b \left( e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p + \eta_t} - y \right)^+ \right),$$

  and

$$Q_{i+1}^{DD}\big(p, e^{\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p}\big) = \min_{y\in\mathcal{Y}} G_{i+1}^{DD}\big(p, y\big).$$

  If $\hat{\beta}_{i+1} > 0$, then solve problem DD and set the first pair of price and inventory level to

$$(\hat{p}_{i+1}, \hat{y}_{i+1,1}) = \arg \max_{(p,y)\in\mathcal{P}\times\mathcal{Y}} G_{i+1}^{DD}(p, y);$$

  otherwise, set

$$(\hat{p}_{i+1}, \hat{y}_{i+1,1}) = \left( \frac{p^l + p^h}{2}, \frac{y^l + y^h}{2} \right).$$

  Set $\hat{p}_{i+1,2} = \hat{p}_{i+1} + \delta_{i+1}$ (in case $\hat{p}_{i+1} + \delta_{i+1} \notin \mathcal{P}$, set $\hat{p}_{i+1,2} = \hat{p}_{i+1} - \delta_{i+1}$), and

$$\hat{y}_{i+1,2} = \arg \max_{y\in\mathcal{Y}} G_{i+1}^{DD}(\hat{p}_{i+1,2}, y).$$

**end for**

The algorithm applies the realized demand data and least-square method to update the linear approximation, $\hat{\alpha}_{i+1} - \hat{\beta}_{i+1} p$, of $\lambda(p)$ and computes a centered sample $\eta_t$ of random error $\epsilon_t$, for $t = t_i + 1, \ldots, t_i + 2I_i$. Note that $\eta_t$ is not a sample of the random error $\epsilon_t$. This is because $\epsilon_t = D_t(p_t) - \lambda(p_t)$ but $1/I_i \sum_{k=t_1+1}^{t_i+I_i} D_k \neq \lambda(p_t)$. For this reason, the constructed objective function for holding and shortage costs is not a sample average of the newsvendor problem. In the traditional SAA, mathematical expectations are replaced by true sample averages, see, e.g., Kleywegt et al. (2002); Levi et al. (2007, 2015). When only biased samples are available, techniques from statistics such as jackknife resampling can be applied to reduce bias for SAA (Wu et al., 1986). In this work, samples of $\epsilon_t$ cannot be observed, however,

$$\eta_t = D_t(p_t) - \frac{1}{I_i} \sum_{k=t_1+1}^{t_i+I_i} D_k = \epsilon_t - \frac{1}{I_i} \sum_{k=t_1+1}^{t_i+I_i} \epsilon_k$$

can be obtained. Since $\mathbb{E}[\epsilon_k] = 0$, $1/I_i \sum_{k=t_1+1}^{t_i+I_i} \epsilon_k$ converges to 0 in probability as $I_i$ grows, and one would expect $\eta_t \to \epsilon_t$ in probability as $t$ grows. Thus, DDA use $\eta_t$ in place of $\epsilon_t$ in computing proxy objectives. Since these samples are obtained from the original i.i.d. samples after subtracting the sample average, we call $\eta_t$ *centered samples*, and $\{\eta_t, t = t_i + 1, \ldots, t_i + 2I_i\}$ are dependent.

A data-driven optimization problem is then constructed. When $\hat{\beta}_{i+1} > 0$, the algorithm solves an optimization problem of a jointly concave function. Technical analyses in the paper show that the probability for $\hat{\beta}_{i+1} > 0$ converges to 1 as $i$ grows.

The DDA algorithm integrates a process of earning (exploitation) and learning (exploration) in each stage. The earning phase consists of the first $I_i$ periods starting at $t_i + 1$, during which the algorithm implements the optimal strategy for the proxy optimization problem $G_i^{DD}(p, y)$. In the next $I_i$ periods of learning phase that starts from $t_i + I_i + 1$, the algorithm uses a different price $\hat{p}_i + \delta_i$ and its corresponding order-up-to level. The purpose of this phase is to extract demand sensitivity information around the selling price. Note that, even though the firm deviates from the optimal strategy of the proxy problem in the second phase, the policies, $(\hat{p}_i + \delta_i, \hat{y}_{i,2})$ and $(\hat{p}_i, \hat{y}_{i,1})$, will be very close to each other as $i$ increases. Chen et al. (2019) show that they both converge to the clairvoyant optimal solution and the loss of profit from this deviation converges to zero.

**Regret Convergence** An upper bound for regret of the DDA policy is provided as $R^{DDA}(T) = V^* - \mathbb{E}\left[\sum_{t=1}^{T} G(p_t, y_t)\right] \leq C_1 T^{1/2}$, for some constant $C_1 > 0$. The lower bound for regret is $\Omega(T^{1/2})$, which is implied by Keskin and Zeevi (2014). This shows that the regret convergence rate for DDA is tight.

The intuitions for regret convergence are the following. Note that during cycle $i$, two distinct prices got implemented, based on which demand data is generated. The two prices are different by $\delta_i$, which decreases to 0 as $i$ increases. Therefore, the two

prices are getting closer, and the linear function yielded by linear approximation approaches the tangent line of $\lambda(\cdot)$, providing gradient information for future decisions.

**Proof Sketch** To compare the DDA policy with the clairvoyant optimal policy, i.e., the optimal solutions of problem DD (12) and problem CI (12.3), note that these two objective functions have significant differences. In problem CI, both $\lambda(p)$ and the distribution of $\epsilon$ are known, but in problem DD, $\lambda(p)$ is approximated by a linear function and distribution of $\epsilon$ is estimated using centered samples instead of true samples. Therefore, to analyze DDA, the authors' approach is to introduce several "intermediate" bridging problems, and in each step we compare two "adjacent" problems that differ along only one dimension.

First, for parameters $\alpha$ and $\beta > 0$, we introduce bridging problem B1 defined by

**Bridging Problem B1** :

$$\max_{p\in\mathcal{P}} \overline{Q}(p, e^{\alpha-\beta p})$$

$$:= \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p}\mathbb{E}\left[e^{\epsilon}\right] - \min_{y\in\mathcal{Y}} \left\{ h\mathbb{E}\left[y - e^{\alpha-\beta p+\epsilon}\right]^{+} + b\mathbb{E}\left[e^{\alpha-\beta p+\epsilon} - y\right]^{+} \right\} \right\}. \tag{12.5}$$

It is easy to see that, the only difference between problem B1 and problem CI in (12.3) is that, in problem B1 we replace the demand-price function in CI by an affine function $\alpha - \beta p$. Let $\overline{p}(\alpha, \beta)$ denote the optimal price for problem B1, and for given $p \in \mathcal{P}$, we let $\overline{y}(e^{\alpha-\beta p})$ denote its optimal order-up-to level, which is the optimal solution for the inner minimization problem in (12.5).

The second bridging problem, B2, is defined for each iteration $i$ of the DDA algorithm, and for any $\alpha$ and $\beta > 0$, it is given by

**Bridging Problem B2** :

$$\max_{p\in\mathcal{P}} \tilde{Q}_{i+1}(p, e^{\alpha-\beta p}) := \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p} \left( \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} e^{\epsilon_t} \right) \right. \tag{12.6}$$

$$\left. - \min_{y\in\mathcal{Y}} \left\{ \frac{1}{2I_i} \sum_{t=t_i+1}^{t_i+2I_i} \left( h\left(y - e^{\alpha-\beta p+\epsilon_t}\right)^{+} + b\left(e^{\alpha-\beta p+\epsilon_t} - y\right)^{+} \right) \right\} \right\}.$$

Compared with problem B1, it is seen that B2 is obtained from B1 after replacing the expectations in B1 by sample averages, hence B2 is the sample average approximation (SAA) of problem B1. Here $\epsilon_t$, $t = t_i + 1, \ldots, t_i + 2I_i$, represent the realizations of random errors during stage $i$. Let $\tilde{p}_{i+1}(\alpha, \beta)$ denote the optimal

price and $\tilde{y}_{i+1}(e^{\alpha-\beta p})$ the optimal order-up-to level for problem B2, which is the optimal solution for the inner minimization problem in (12.6).

The third bridging problem B3 is a variation of problem B2, which replaces the true random error $\epsilon_t$ by a biased error sample $\zeta_t$, $t = t_i + 1, \ldots, t_i + 2I_i$. That is, for

$$\zeta_{t=t_i+1}^{t_1+I_i} = \left(\zeta_{t_i+1}, \ldots, \zeta_{t_i+I_i}\right), \quad \zeta_{t=t_i+I_i+1}^{t_1+2I_i} = \left(\zeta_{t_i+I_i+1}, \ldots, \zeta_{t_i+2I_i}\right),$$

and parameters $\alpha$ and $\beta > 0$, we define the third bridging problem B3 as

**Bridging Problem B3** :

$$\max_{p\in\mathcal{P}} \breve{Q}_{i+1}\left(p, e^{\alpha-\beta p}, \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\right) := \max_{p\in\mathcal{P}} \left\{ pe^{\alpha-\beta p}\left(\frac{1}{2I_i}\sum_{t=t_i+1}^{t_i+2I_i} e^{\zeta_t}\right) - \min_{y\in\mathcal{Y}} \left\{ \frac{1}{2I_i}\sum_{t=t_i+1}^{t_i+2I_i} \left(h\left(y - e^{\alpha-\beta p+\zeta_t}\right)^+ + b\left(e^{\alpha-\beta p+\zeta_t} - y\right)^+\right)\right\}\right\}.$$

Note that when $(\alpha, \beta) = (\hat{\alpha}_{i+1}, \hat{\beta}_{i+1})$, and $\zeta_t = \eta_t$ for $t = t_1 + 1, \ldots, t_i + 2I_i$, problem B3 reduces to problem DD (12) in the DDA algorithm. Thus, problem B3 serves as a bridge between problem B2 and problem DD. We denote the optimal price of problem B3 by $\breve{p}_{i+1}\left((\alpha, \beta), \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\right)$ and its optimal order-up-to level, for given price $p$, by $\breve{y}_{i+1}\left(e^{\alpha-\beta p}, \zeta_{t=t_i+1}^{t_1+I_i}, \zeta_{t=t_i+I_i+1}^{t_1+2I_i}\right)$.

Based on their definitions, problem CI, bridging problems B1–B3, and problem DD require less and less information about the demand process. Problem CI has complete information about both $\lambda(\cdot)$ and the distribution of $\epsilon$; problem B1 does not know $\lambda(\cdot)$ but knows the distribution of $\epsilon$; problem B2 does not know either $\lambda(\cdot)$ or the distribution of $\epsilon$ but has access to true samples of $\epsilon$; problems B3 and DD do not have true samples and have to use biased samples. Chen et al. (2019) prove convergence for each pair of adjacent problems, and eventually establish convergence of problem DD to problem CI.

## 12.3 Nonparametric Learning for Lost-Sales System

Different from Sect. 12.2 that considers backlogged demand, in this section we consider lost sales and censored demand. This scenario happens when, in case of a stockout, rejected customers leave the store without purchasing. These customers cannot be observed by the retailer, and demand data is thus truncated by inventory levels. We will discuss the model, algorithms, and regret convergence results based on Chen et al. (2021a, 2020b).

Consider the additive demand model $D_t(p_t) = \lambda(p_t) + \epsilon_t$ with $\lambda(\cdot)$ being a non-increasing deterministic function and $\epsilon_t$, $t = 1, 2, \ldots, T$, being i.i.d. random variables with $\mathbb{E}[\epsilon_t] = 0$. We denote the CDF of $\epsilon_t$ by $F(\cdot)$, which is assumed to be continuous and differentiable, the PDF by $f(\cdot)$ such that $f(\epsilon_t) < \infty$ for any $\epsilon_t$, and the standard deviation of $\epsilon_t$ by $\sigma$. For notational convenience, we use $\epsilon_t$ and $\epsilon$ interchangeably because of the i.i.d. assumption. Demands are satisfied as much as possible by on-hand inventory, and unsatisfied demands are lost and unobservable. For system dynamics one has $x_{t+1} = (y_t - D_t(p_t))^+$. The instantaneous reward for period $t$ is $p_t\mathbb{E}[\min\{y_t, D_t(p_t)\}] - b\mathbb{E}[D_t(p_t) - y_t]^+ - h\mathbb{E}[y_t - D_t(p_t)]^+ = p_t\mathbb{E}[D_t(p_t)] - (b + p_t)\mathbb{E}[D_t(p_t) - y_t]^+ - h\mathbb{E}[y_t - D_t(p_t)]^+$.

The firm knows neither the function $\lambda(p_t)$ nor the distribution of the random term $\epsilon_t$ a priori, which must be learned from censored demands collected over time while maximizing the cumulative profit. In this system, demand is censored, therefore, $o_t = \min\{D_t(p_t), y_t\}$. For an admissible policy, $(p_t, y_t)$ is adapted to the filtration generated by $\{(p_s, y_s), \min\{D_s(p_s), y_s\} : s = 1, \ldots, t-1\}$ under censored demand.

If the underlying demand-price function $\lambda(p)$ and the distribution of the error term $\epsilon_t$ were known a priori, the clairvoyant optimal policy for this problem is a myopic policy (refer to Sobel 1981). Define the single-period problem by

$$Q(p, y) = p\mathbb{E}[D_1(p)] - (b + p)\mathbb{E}[D_1(p) - y]^+ - h\mathbb{E}[y - D_1(p)]^+.$$

To find the optimal pricing and inventory decisions, it suffices to maximize the single-period revenue $Q(p, y)$, which can be expressed as

$$\begin{aligned}
&\max_{p,y} \left\{ p\mathbb{E}[D_1(p)] - (b + p)\mathbb{E}[D_1(p) - y]^+ - h\mathbb{E}[y - D_1(p)]^+ \right\} \\
&\quad = \max_{p} \left\{ p\lambda(p) - \min_{y} \left\{ (b + p)\mathbb{E}\left[\lambda(p) + \epsilon - y\right]^+ + h\mathbb{E}\left[y - \lambda(p) - \epsilon\right]^+ \right\} \right\}.
\end{aligned}$$

Hence, we rewrite the clairvoyant problem as

$$\begin{aligned}
&\max_{p,y} Q(p, y) = \max_{p} G(p), \\
&\text{where} \quad G(p) = p\lambda(p) - \min_{y} \left\{ (b + p)\mathbb{E}\left[\lambda(p) + \epsilon - y\right]^+ + h\mathbb{E}\left[y - \lambda(p) - \epsilon\right]^+ \right\}.
\end{aligned}$$

This problem was first studied in Chen et al. (2021a), whose learning method and result will be briefly reviewed. Then we shift our focus to Chen et al. (2020b), which studies the problem in a more general setting and improves the convergence rate in Chen et al. (2021a).

### 12.3.1 Algorithms and Results in Chen et al. (2021a)

Chen et al. (2021a) assume $G(\cdot)$ is concave and $\lambda(p)$ is differentiable to a high order. They provide a spline approximation based learning algorithm (SALA) under an exploration-exploitation framework.

**Algorithm for Concave** $G(\cdot)$ The learning algorithm follows an exploration-exploitation framework and is based on spline approximation.

We now formally describe how a spline approximation for the demand-price function $\lambda(\cdot)$ is constructed. Before doing that, we first present a high-level view of the approximation method.

Spline approximation needs two integer inputs, $m > 0$ and $l > 0$, and it requires the specification of *knots, basis functions*, and *coefficients*. Knots, denoted as $w_i$, $i = 1, \ldots, 2m + l$, are equally spaced price points on the whole price interval, and there are in total $2m + l$ of them. The more knots a model has, the more observations of $\lambda(\cdot)$ the model uses to do estimation, which in general leads to a more accurate spline approximation. Let $\mathcal{L}\lambda(p)$ denote the spline approximation operator of a deterministic function $\lambda(p)$, and it can be represented as

$$\mathcal{L}\lambda(p) = \sum_{i=1}^{m+l} \gamma_i^{\lambda} \cdot N_i^m(p), \tag{12.7}$$

where $N_i^m(p)$, $i = 1, \ldots, m + l$, are the basis functions with coefficients $\gamma_i^{\lambda}$. The base function $N_i^m(p)$ is polynomial in $p$ with the highest order $m - 1$ and is constructed based on knots $w_i, \ldots, w_{i+m}$. The larger the $m$, the smoother the $N_i^m(p)$ and $\mathcal{L}\lambda(p)$. The coefficient $\gamma_i^{\lambda}$ is computed based on some specific price points on $[w_i, w_{i+m}]$ and the corresponding values of $\lambda(p)$ at these price points. To be more specific, price points used here include $w_i, \ldots, w_{i+m}$ and $\tau_{i1}, \ldots, \tau_{im}$ that will be defined shortly in Algorithm 2. The detailed procedure of spline approximation is also presented in Algorithm 2.

It follows from Schumaker (2007) that for the basis function $N_i^m(p)$, its $(m - 2)$-th order derivative exists and is continuous. Together with Theorem 4.9 in Schumaker (2007), one can verify that the basis function $N_i^m(p) = 0$ for $p \notin (w_i, w_{i+m})$ and $N_i^m(p) > 0$ for $p \in (w_i, w_{i+m})$.

Given the detailed construction of the spline approximation, we are ready to present the main learning algorithm termed SALA in Algorithm 3.

The learning algorithm SALA separates the planning horizon into a disjoint exploration phase and exploitation phase.

The algorithm specifies the parameters $m$ and $l$ for determining the density for spline approximation, the parameter $\Delta$ for determining the grid size for (sparse) discrete optimization problem, and the parameter $L$ for determining the length of the exploration phase. Note that these parameters are determined "optimally" via (12.10) to minimize the theoretical regret rate.

**Algorithm 2** Constructing a Spline Approximation (SA)

Let integers $m \geq 2$ and $l \geq 1$ be the inputs of a spline approximation. The (optimal) values of $m$ and $l$ will be specified later.

Let the set of $2m + l$ points $\{w_1, \ldots, w_{2m+l}\}$ be a partition of the interval

$$\left[ p^l - \frac{p^h - p^l}{l+1}(m-1),\, p^h + \frac{p^h - p^l}{l+1}(m-1) \right],$$

where each point $w_i$ is defined by

$$w_i = p^l + \frac{p^h - p^l}{l+1}(i-m), \qquad \text{for } i = 1, \ldots, 2m+l.$$

Note that $w_m = p_l$ and $w_{m+l+1} = p_h$ and there are $l$ equally spaced points strictly between $p_l$ and $p_h$. Also, there are $m-1$ extension points to the left of $p_l$ and $m-1$ extension points to the right of $p_h$. Thus, there are in total $2m + l$ equally spaced points for the above specified interval.

**for** $i = 1, 2, \ldots, m + l$ **do**

  $\varphi_{im}(x) = \Pi_{r=1}^{m-1}(x - w_{i+r})$.

  **for** $j = 1, 2, \ldots, m$ **do**

    $\tau_{ij} = w_i + (w_{i+m} - w_i)\frac{j-1}{m-1}$.

    $\psi_{ij}(x) = \Pi_{r=1}^{j-1}(x - \tau_{ir})$, with $\psi_{i1}(x) \equiv 1$.

    Then define

$$\alpha_{ij} = \sum_{r=1}^{j} \frac{(-1)^{r-1}\varphi_{im}^{(m-r)}(0)\psi_{ij}^{(r-1)}(0)}{(m-1)!}. \tag{12.8}$$

  **end for**

**end for**

Given a single variate real function $\lambda(\cdot)$ and a sequence of numbers $x_1 < x_2 < \cdots < x_{r+1}$, let $\mathcal{D}_{[x_1,\ldots,x_{r+1}]}\lambda$ be the operator that gives the $r$-th order divided difference of $\lambda(\cdot)$, defined by

$$\mathcal{D}_{[x_1,\ldots,x_{r+1}]}\lambda = \sum_{j=1}^{r+1} \frac{\lambda(x_j)}{\Pi_{i=1,i\neq j}^{r+1}(x_j - x_i)},$$

and if $r = 0$, $\mathcal{D}_{[x_1]}\lambda \equiv \lambda(x_1)$.

**for** $i = 1, \ldots, m + l$ **do**

  The *spline approximation coefficients* are

$$\gamma_i^{\lambda} = \sum_{j=1}^{m} \alpha_{ij} \cdot \mathcal{D}_{[\tau_{i1},\ldots,\tau_{ij}]}\lambda.$$

Moreover, for $p \in [p^l, p^h]$, define the $m$-th order *spline approximation basis functions* associated with knots $w_i, \ldots, w_{i+m}$ by

$$N_i^m(p) = (-1)^m(w_{i+m} - w_i)\mathcal{D}_{[w_i,\ldots,w_{i+m}]}(\max\{0, p-w\})^{m-1}. \tag{12.9}$$

In $\mathcal{D}_{[w_i,\ldots,w_{i+m}]}(\max\{0, p-w\})^{m-1}$, the argument $(\max\{0, p-w\})^{m-1}$ is considered as a function of $w$ for given $p$, and the resulting basis function $N_i^m(p)$ is a function of $p$.

**end for**

The spline approximation of function $\lambda(p)$, denoted by $\mathcal{L}\lambda(p)$, is given by (12.7)

**Algorithm 3** Spline Approximation Based Learning Algorithm (SALA)

Set input parameters

$$m = \max\left\{3, \lceil (\log T)^{\frac{1}{2}} \rceil\right\},\ L = \left\lceil T^{\frac{1}{2}+\frac{1}{3\sqrt{\log T}}} \right\rceil,\ l = \left\lceil (\log T)^{\frac{3}{2}} T^{\frac{1}{4\sqrt{\log T}}} \right\rceil,\ \Delta = T^{-\frac{1}{4}}. \tag{12.10}$$

Define a sparse discretized set of prices by

$$\mathcal{S} = \{p^l, p^l + \Delta, p^l + 2\Delta, \ldots, p^h\}, \tag{12.11}$$

which is the discrete search space for pricing decisions. We refer to $\mathcal{S}$ as the (sparse) grid.

**for** $i = 1, \ldots, l + m$ **do**
  **for** $j = 1, \ldots, l$ **do**
    **for** $t = (i-1)mL + (j-1)L + 1, \ldots, (i-1)mL + jL$ **do**
      Implement the following pricing and order-up-to decisions: $p_t = \tau_{ij}, y_t = \lceil \log L \log\log L \rceil$, where $\tau_{ij}$ is defined in Algorithm 2 spline approximation.
    **end for**
  **end for**
**end for**
**for** $i = 1, \ldots, l + m$ **do**
  **for** $j = 1, \ldots, l$ **do**
    Let the average empirical sales be $s_{ij} = \frac{\sum_{t=(i-1)mL+(j-1)L+1}^{(i-1)mL+jL} d_t \wedge y_t}{L}$.
  **end for**
  Let the empirical spline approximation coefficients be

$$\beta_i = \alpha_{i1} s_{i1} + \sum_{j=2}^{m} \sum_{v=1}^{j} \frac{\alpha_{ij} s_{iv}}{\Pi_{r=1, r\neq v}^{j} (\tau_{iv} - \tau_{ir})},$$

  where $\alpha_{ij}$ is defined in (8).
**end for**
The spline approximation of function $\lambda(p)$ using sales (or censored demand) is then given by $\hat{\lambda}(p) = \sum_{i=1}^{m+l} \beta_i N_i^m(p)$, where the basis function $N_i^m(p)$ is defined in (13).
**for** $i = 1, \ldots, l + m$ **do**
  **for** $j = 1, \ldots, l$ **do**
    **for** $t = (i-1)mL + (j-1)L + 1, \ldots, (i-1)mL + jL$ **do**
      Let

$$\eta_t = d_t \wedge y_t - s_{ij} \tag{12.12}$$

      be the *residual error*, which is used to approximate the random error (with some biases).
    **end for**
  **end for**
**end for**
Solve the following surrogate optimization problem on a sparse grid $\mathcal{S}$ (based on sales and spline approximation):

$$\max_{p,y} \hat{Q}(p, y) \triangleq \max_{p\in\mathcal{S}} \hat{G}(p), \quad \text{where}$$

$$\hat{G}(p) \triangleq p\hat{\lambda}(p) - \min_{y} \left\{ (b+p) \frac{\sum_{t=1}^{L(m+l)m} [\hat{\lambda}(p) + \eta_t - y]^+}{L(m+l)m} + h \frac{\sum_{t=1}^{L(m+l)m} [y - \hat{\lambda}(p) - \eta_t]^+}{L(m+l)m} \right\}.$$

Let $(\hat{p}, \hat{y}) = \arg\max \hat{Q}(p, y)$.
**for** $t = L(m+l)m + 1, \ldots, T$ **do**
  Set the price and target inventory level to $p_t = \hat{p}$, $y_t = x_t \vee \hat{y}$.
**end for**

SALA then enters the exploration phase of total length of $L(m+l)m$ periods, which is roughly on the order of $\sqrt{T}$. The price space is discretized into equally spaced prices $\{\tau_{ij}\}$'s (which will also be used for constructing a spline approximation). For each $i$ and $j$, SALA offers the price $\tau_{ij}$, together with the pre-specified target inventory level $y_t$, for an equal number of periods. We note here that the high-level reason for the target inventory level $y_t$ to be on the order of $\log L \log\log L$ is to ensure that the bias caused by demand censoring is appropriately bounded.

SALA leverages the sales collected over prices $\{\tau_{ij}\}$'s to carry out an empirical spline approximation $\hat{\lambda}(p)$ of the true demand-price function $\lambda(p)$. Also, SALA computes the so-called residual error $\eta_t$, which is used to approximate the random error $\epsilon_t$. It is important to note that $\hat{\lambda}(p)$ is constructed based on sales (or censored demand) and, therefore, it suffers a bias in estimating $\lambda(p)$, which must be quantified in the regret analysis. Similarly, due to demand censoring, $\eta_t$ is also a biased representation of $\epsilon_t$, in which the bias must also be quantified.

SALA essentially treats the empirical spline approximation $\hat{\lambda}(p)$ as the true demand-price function $\lambda(p)$ and the residual error $\eta_t$ as the true random error $\epsilon_t$, and constructs the corresponding *sample average approximation* (SAA) based surrogate optimization problem. Note that the surrogate optimization problem is solved sequentially: the inner problem is to find the optimal inventory target level for a given price, while the outer problem is to find the optimal price on the grid. The inner problem is convex in the inventory target level, which can be efficiently solved using first-order methods, whereas the outer problem is a one-dimensional discretized problem but solved on a sparse grid.

Finally, SALA completes the exploration phase and enter the exploitation phase. For the remaining planning horizon, SALA implements the optimal price and target inventory level suggested by the (sampled) surrogate optimization problem. Note that the length of the exploitation phase is $T - L(m+l)m$, which is roughly on the order of $T - \sqrt{T}$.

**Regret Convergence** In Chen et al. (2021a), it shows that the convergence of the spline approximation can be bounded as $\mathbb{P}\left\{\|\lambda'(p) - \hat{\lambda}'(p)\|_\infty \le C_2 T^{-1/4}\right\} > 1 - T^{-2}$ and $\mathbb{P}\left\{\|\lambda'(p) - \hat{\lambda}'(p)\|_\infty \le C_2 T^{-1/4}\right\} > 1 - T^{-2}$ for some constant $C_2 > 0$ and any $p \in \mathcal{P}$. Moreover, the convergence of error estimation is shown as $\mathbb{P}\left\{\left|\mathbb{E}[\epsilon - z^*(p)]^+ - \frac{1}{L(m+l)m}\sum_{t=1}^{L(m+l)m}(\eta_t - \hat{z}(p))^+\right| \le C_3 T^{-1/4}\right\} > 1 - 10T^{-2}$, where $z^*(p) = F^{-1}\left(\frac{b+p}{b+p+h}\right)$ and $\hat{z}(p) = \min\left\{\eta_j : \sum_{t=1}^{L(m+l)m}\mathbb{1}(\eta_t \le \eta_j) \ge \frac{b+p}{b+p+h}\right\}$, for some constant $C_3 > 0$ and any $p \in \mathcal{P}$. Based on these results, the regret convergence rate of SALA is upper bounded as $R^{SALA}(T) \le C_4 T^{1/2+\varepsilon}(\log T)^3 \log\log T$, where $\varepsilon = 1/\sqrt[3]{\log T} + 0.25/\sqrt{\log T}$ and constant $C_4 > 0$. Here note that for any constant $c > 0$, one has $\log\log T/\log T < \varepsilon < c$ (or equivalently, $\log T < T^\varepsilon < T^c$), for large enough $T$. Since the regret lower

bound for this problem is $\Omega(T^{1/2})$, the SALA algorithm matches the lower bound up to $T^{\varepsilon}$.

### 12.3.2 Algorithms and Results in Chen et al. (2020b)

Chen et al. (2020b) consider both concave and non-concave $G(\cdot)$, provide learning algorithms for the two scenarios, and show that the convergence rates of both algorithms match the theoretical lower bounds, respectively.

#### 12.3.2.1 Concave $G(\cdot)$

In this section, we discuss the scenario with concave $G(\cdot)$.

**Algorithm for Concave** $G(\cdot)$ A different algorithm is proposed in Chen et al. (2020b) for concave $G(\cdot)$, which approaches the optimal $y$ using bisection and optimal $p$ using trisection. The detailed algorithm is presented in Algorithms 4, 5, and 6.

With the SEARCHORDERUPTO routine in Algorithm 4, for every price $p \in [\underline{p}, \overline{p}]$ one can estimate, using relatively few selling periods, the near-optimal order-up-to level $\hat{y}_n$ so that $Q(p, \hat{y}_n) \approx Q(p, y^*(p)) = G(p)$, where $y^*(p)$ is the optimal inventory level under price $p$. It is tempting to use a similar strategy on $G(\cdot)$ which is

**Algorithm 4** Bisection search for order-up-to level $y$

**function** SEARCHORDERUPTO($p, n, C_1$)
  Initialize: $L_\tau = 0, U_\tau = \bar{y}, m_\tau = \bar{y}/2, \tau = 0, g_\tau = 0$;
  Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
  **while** $n$ review periods have not been reached **do**
    Set order-up-to level at $y_t = m_\tau$ and price at $p_t = p$;
    Observe censored demand and update $n_\tau \leftarrow n_\tau + 1$; $g_\tau \leftarrow g_\tau + (b+p)$ if no inventory is left; $g_\tau \leftarrow g_\tau - h$ if positive inventory is left;
    Construct confidence intervals $[\underline{g}(m_\tau), \overline{g}(m_\tau)] = \hat{g}_\tau \pm C_1/\sqrt{n_\tau}$, where $\hat{g}_\tau = g_\tau/n_\tau$;
    **if** $\tau < \lceil \log_2(n\bar{y}) \rceil$ and $\underline{g}(m_\tau) > 0$ **then**
      Update $L_{\tau+1} = m_\tau, U_{\tau+1} = U_\tau, m_{\tau+1} = (L_{\tau+1} + U_{\tau+1})/2, n_{\tau+1} = 0, \tau \leftarrow \tau + 1$;
      Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
    **else if** $\tau < \lceil \log_2(n\bar{y}) \rceil$ and $\overline{g}(m_\tau) < 0$ **then**
      Update $L_{\tau+1} = L_\tau, U_{\tau+1} = m_\tau, m_{\tau+1} = (L_{\tau+1} + U_{\tau+1})/2, n_{\tau+1} = 0, \tau \leftarrow \tau + 1$;
      Offer the lowest price $\underline{p}$ until current inventory level is below $m_\tau$;*
    **end if**
  **end while**
  Return $\hat{y}_n = m_\tau$ which is explored for the most number of times (largest $n_\tau$).
**end function**

* Review periods in these steps do *not* count towards the total budget of $n$ periods.

**Algorithm 5** Estimation of reward ($G(\cdot)$) differences at $p<p'$

1: **function** ESTIMATEGDIFFERENCE($p$, $\hat{y}$, $p'$, $\hat{y}'$, $n$)
2: Set prices and order-up-to levels at $(p, \hat{y})$ for $n$ periods, and let $\{o_t = \min\{\lambda(p) + \varepsilon_t, \hat{y}\}\}_{t\in\mathcal{T}_1}$ be the censored demands, where $\mathcal{T}_1$ is the $n$ periods in this step;
3: Set prices and order-up-to levels at $(p', \hat{y}')$ for the next $n$ periods, and let $\{o'_t = \min\{\lambda(p') + \varepsilon_t, \hat{y}'\}\}_{t\in\mathcal{T}_2}$ be the censored demands, where $\mathcal{T}_2$ is the $n$ periods in this step;
4: Define $\delta_t := \hat{y} - o_t$, $\delta'_t := \hat{y}' - o'_t$ and let $\hat{\nu}, \hat{\nu}'$ be the empirical distributions of $\{\delta_t\}_{t\in\mathcal{T}_1}, \{\delta'_t\}_{t\in\mathcal{T}_2}$, respectively. Let $F_{\hat{\nu}}, F_{\hat{\nu}'}$ be the CDFs of $\hat{\nu}, \hat{\nu}'$. Find $\hat{u}$ such that

$$\hat{u} := \sup\left\{u : F_{\hat{\nu}'}(u) \le \tfrac{h}{b+p+h}\right\};$$

5: Return the estimate reward difference $\hat{\Delta}_G(p, p')$ as

$$\begin{aligned}&\hat{\Delta}_G(p, p')\\ =&\left[\frac{1}{n}\sum_{t\in\mathcal{T}_2} p'o'_t - h\delta'_t\right] - \left[\frac{1}{n}\sum_{t\in\mathcal{T}_1} po_t - h\delta_t\right] + b\left[\hat{u}\times\frac{h}{b+p+h} - \frac{1}{n}\sum_{t\in\mathcal{T}_2}\delta'_t\mathbf{1}\{0<\delta'_t\le\hat{u}\}\right].\end{aligned}$$

6: **end function**

**Algorithm 6** The main algorithm: trisection search on prices

**Input**: time horizon $T$, price range $[\underline{p}, \overline{p}]$, parameters $C_1, C_2 > 0$.
Initialization: $\zeta = 0$, $L_\zeta = \underline{p}$, $U_\zeta = \overline{p}$.
**while** $T$ review periods have not been reached **do**
  Set $\alpha_\zeta = \frac{2}{3}L_\zeta + \frac{1}{3}U_\zeta$, $\beta_\zeta = \frac{1}{3}L_\zeta + \frac{2}{3}U_\zeta$, $N_\zeta = \lceil g(C_2/(\beta_\zeta - \alpha_\zeta)^4)\rceil$; **
  $\hat{y}_\zeta \leftarrow$ SEARCHORDERUPTO($\alpha_\zeta, N_\zeta, C_1$), $\hat{y}'_\zeta \leftarrow$ SEARCHORDERUPTO($\beta_\zeta, N_\zeta, C_1$);
  $\hat{\Delta}_G(\alpha_\zeta, \beta_\zeta) \leftarrow$ ESTIMATEGDIFFERENCE($\alpha_\zeta, \hat{y}_\zeta, \beta_\zeta, \hat{y}'_\zeta, N_\zeta$);
  **if** $\hat{\Delta}_G(\alpha_\zeta, \beta_\zeta) > 0$ **then**
    Update $L_{\zeta+1} \leftarrow \alpha_\zeta$, $U_{\zeta+1} \leftarrow U_\zeta$, $\zeta \leftarrow \zeta + 1$;
  **else**
    Update $L_{\zeta+1} \leftarrow L_\zeta$, $U_{\zeta+1} \leftarrow \beta_\zeta$, $\zeta \leftarrow \zeta + 1$;
  **end if**
**end while**
** We use $g(x) := (x + \lceil\log_2(x\bar{y})\rceil)\lceil\log_2(x + \lceil\log_2(x\bar{y})\rceil)\rceil$.

strongly concave to the price to find the optimal price $p^*$, which has been applied to pure pricing without inventory replenishment problems in the literature (Wang et al., 2014; Lei et al., 2014). Such an approach, however, encounters a major technical hurdle that neither the reward $G(\cdot)$ nor its derivative can be directly observed or even accurately estimated, due to the censoring of the demands and the lost-sales component in the objective function.

In this section we present the key idea of this paper that overcomes this significant technical hurdle. The important observation is that, in a bisection or trisection search method, it is *not* necessary to estimate $G(p)$ accurately. Instead, one only needs to accurately estimate the *difference* of rewards $G(p') - G(p)$ at two prices $p$, $p'$ in order to decide on how to progress, which can be accurately estimated even in the

presence of censored demands and lost sales. We sketch and summarize this idea below.

**The Key Idea of Algorithm 5—"Difference Estimator"** Let $p < p'$ be two different prices and recall the definition that $G(p) = p\mathbb{E}[\min\{\lambda(p) + \varepsilon, y^*(p)\}] - b\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+] - h\mathbb{E}[(y^*(p) - \lambda(p) - \varepsilon)^+]$. When $y^*(p)$ is relatively accurately estimated (from the previous section and Algorithm 4), the only term in $G(p)$ that cannot be directly observed without bias is the lost-sales penalty $-b\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+]$. Hence, to estimate $G(p') - G(p)$ accurately (Chen et al., 2020b) only need to estimate the difference

$$\mathbb{E}[(\varepsilon + \lambda(p) - y^*(p))^+] - \mathbb{E}[(\varepsilon + \lambda(p') - y^*(p'))^+]. \tag{12.13}$$

By the property of newsvendor solution, $y^*(p) = \lambda(p) + z_p$ where $z_p$ is such that $F_\mu(z_p) = \int_{-\infty}^{z_p} f_\mu(u)\mathrm{d}u = \phi(p) = \frac{b+p}{b+p+h}$, and similarly $y^*(p') = \lambda(p') + z_{p'}$ such that $F_\mu(z_{p'}) = \phi(p') = \frac{b+p'}{b+p'+h}$. Since $p < p'$, we have $z_p < z_{p'}$. Equation (12.13) can be subsequently simplified to

$$\begin{aligned}\mathbb{E}[(\varepsilon - z_p)^+] - \mathbb{E}[(\varepsilon - z_{p'})^+] &= \mathbb{E}[(\varepsilon - z_p)^+ - (\varepsilon - z_{p'})^+] \\ &= \underbrace{(z_{p'} - z_p) \times \Pr[\varepsilon \geq z_p]}_{\text{Part A}} - \underbrace{\mathbb{E}[(z_{p'} - \varepsilon)\mathbf{1}\{z_p \leq \varepsilon \leq z_{p'}\}]}_{\text{Part B}}.\end{aligned} \tag{12.14}$$

For Part A of Eq. (12.14), the $\Pr[\varepsilon \geq z_p]$ term has the closed-form, known formula of $\Pr[\varepsilon \geq z_p] = 1 - F_\mu(z_p) = 1 - \phi(p) = \frac{h}{b+p+h}$. To estimate $z_{p'} - z_p$, which is nonnegative, (Chen et al., 2020b) use the following observation:

$$1 - \phi(p) = \frac{h}{b+p+h} = \Pr[\varepsilon \geq z_p] \overset{(*)}{=} \Pr[(z_{p'} - \varepsilon)^+ \leq z_{p'} - z_p]. \tag{12.15}$$

Here the crucial Eq. (*) holds because $z_{p'} > z_p$, and, therefore, the event $\varepsilon \geq z_p$ is equivalent to either $\varepsilon > z_{p'}$ (for which $(z_{p'} - \varepsilon)^+$ is zero), or $\varepsilon \leq z_{p'}$ and $z_{p'} - \varepsilon \leq z_{p'} - z_p$. Furthermore, the random variable $(z_{p'} - \varepsilon)^+ = (y^*(p') - \lambda(p') - \varepsilon)^+$ is (approximately) *observable* when $y^*(p')$ is estimated accurately, because this is the leftover inventory at ordering-up-to level $y^*(p')$ and posted price $p'$. Therefore, one can collect samples of $(z_{p'} - \varepsilon)^+$, construct an empirical cumulative distribution function (CDF) and infer the value of $z_{p'} - z_p$ by inverting the empirical CDF at $h/(b + p + h)$. A similar approach can be taken to estimate Part B of Eq. (12.14), by plugging in the empirical distribution of the random variable $(z_{p'} - \varepsilon)^+\mathbf{1}\{0 \leq (z_{p'} - \varepsilon)^+ \leq z_{p'} - z_p\}$.

A pseudo-code description of the reward difference estimation routine is given in Algorithm 5. The design of Algorithm 5 roughly follows the key ideas demonstrated in the previous paragraph. The $o_t$ and $\delta_t$ random variables correspond to the censored demand and the leftover inventory at time period $t$, and the distribution

of $\delta_t$ (or $\delta'_t$) would be close to the distribution of $(z_p - \varepsilon)^+$ (or $(z_{p'} - \varepsilon)^+$). Using the observation in Eq. (12.15), $\hat{u}$ in Algorithm 5 would be a good estimate of $z_{p'} - z_p$ by inverting the empirical CDFs.

As the last component and the main entry point of the algorithm framework, (Chen et al., 2020b) describe a trisection search method to localize the optimal price $p^*$ that maximizes $G(\cdot)$, based on the strong concavity of $G(\cdot)$ in $p$ that is assumed for this scenario. The trisection principle for concave functions itself is not a new idea and has been explored in the existing literature on pure pricing without inventory replenishment problems (Lei et al., 2014; Wang et al., 2014). A significant difference, nevertheless, is that in this application the expected reward function $G(\cdot)$ cannot be observed directly (even up to centered additive noise) due to the presence of censored demands, and one must rely on the procedure described in the previous section to estimate the reward difference function $\Delta_G(\cdot, \cdot)$ instead. Below we describe the key idea for this component.

**The Key Idea of Algorithm 6** Recall that $G(p) = \max_{y \in [0, \bar{y}]} Q(p, y)$ and $\Delta_G(p, p') = G(p') - G(p)$. A trisection search algorithm is used to locate $p^* \in [\underline{p}, \overline{p}]$ that maximizes $G(\cdot)$, under the assumption that $G(\cdot)$ is twice continuously differentiable and strongly concave in $p$. The algorithm starts with $I_0 = [\underline{p}, \overline{p}]$ and attempts to shorten the interval by $2/3$ after each epoch $\zeta$, without throwing away the optimal price $p^*$ with high probability. Suppose at epoch $\zeta$ the interval $I_\zeta = [L_\zeta, U_\zeta]$ includes $p^*$, and let $\alpha_\zeta, \beta_\zeta$ be the trisection points of $I_\zeta$. Depending on the location of $p^*$ relative to $\alpha_\zeta, \beta_\zeta$, the updated, shrunk interval $I_{\zeta+1} = [L_{\zeta+1}, U_{\zeta+1}]$ can be computed. The above discussion shows that trisection search updates can be carried out by simply determining the signs of $\Delta_G(\alpha_\zeta, \beta_\zeta)$. A complete pseudo-code description of the procedure is given in Algorithm 6.

**Regret Convergence for Concave** $G(\cdot)$ The regret rate of the algorithm for concave $G(\cdot)$ is upper bounded as $R(T) \leq O\left(T^{1/2}(\ln T)^2\right)$ with probability $1 - O(T^{-1})$. This upper bound almost matches the theoretical lower bound of $\Omega(T^{1/2})$.

#### 12.3.2.2 Non-Concave $G(\cdot)$

In this section, we discuss the scenario with non-concave $G(\cdot)$.

**Algorithm for Non-Concave** $G(\cdot)$ For non-concave $G(\cdot)$, (Chen et al., 2020b) still rely on bisection to search for the optimal $y$, but for $p$, the previous trisection framework cannot be applied anymore due to loss of concavity. They design an active tournament algorithm based on the difference estimator to search for the optimal $p$.

Key idea 1: discretization. The price interval $[\underline{p}, \overline{p}]$ is first being partitioned into $J$ evenly spaced points $\{p(j)\}_{j \in [J]}$, with $J = \lceil T^{1/5} \rceil$. Because $G(\cdot)$ is twice continuously differentiable (implied by the first condition in Chen et al. (2020b))

and $p^* \in (\underline{p}, \overline{p})$, there exists $p_{j^*}$ for some $j^* \in [J]$ such that $G(p^*) - G(p_{j^*}) \leq O(|p^* - p_{j^*}|^2) \leq O(J^{-2}) = O(T^{-2/5})$, because $G'(p^*) = 0$. The problem then reduces to a multiarmed bandit problem over the $J$ arms of $\{p_j\}_{j\in[J]}$, with the important difference of the actual reward of each arm *not* directly observable due to the censored demands.

Key idea 2: active elimination with tournaments. With the sub-routines developed in Algorithms 4 and 5 in the previous section, we can in principle estimate the reward difference $\Delta_G(p, p')$ at two prices $p < p'$ up to an error on the order of $\widetilde{O}(1/\sqrt{n})$, with $\approx 2n$ review periods for each price and without incurring large regret. In Algorithm 6, we successfully applied this "pairwise comparison" oracle in a trisection approach to utilize the concavity of $G(\cdot)$. Without concavity of $G(\cdot)$, we are going to use an active elimination with tournaments approach to find the price with the highest rewards in $\{p_j\}_{j\in[J]}$.

More specifically, consider epochs $\gamma = 1, 2, \cdots$ with geometrically increasing sample sizes $n_\gamma$ implied by geometrically decreasing accuracy levels $\Delta_\gamma = 2^{-\gamma}$. At the beginning of each epoch $\gamma$, the algorithm maintains an "active set" $\mathcal{S}_\gamma \subseteq [J]$ of prices such that for all $p \in \mathcal{S}_\gamma$, $G(p_{j^*}) - G(p) \leq \Delta_\gamma$ where $\Delta_\gamma = \widetilde{O}(1/\sqrt{n_\gamma})$. Chen et al. (2020b) use a "tournament" approach to eliminate prices in $\mathcal{S}_\gamma$ that have large sub-optimality gaps. In particular, all prices in $\mathcal{S}_\gamma$ are formed into pairs and each pair is allocated $n_\gamma$ samples to either eliminate the inferior price in the pair, or to combine both prices into one and advance to the next round of the tournament. The tournament ends once there is only one price left, $\hat{p}_\gamma$. Afterwards a separate elimination procedure is invoked to retain all other prices that are close to $\hat{p}_\gamma$ in terms of performance. A detailed algorithm for non-concave $G(\cdot)$ is presented in Algorithm 7.

**Regret Convergence for Non-Concave** $G(\cdot)$ The regret convergence rate for non-concave $G(\cdot)$ is upper bounded as $R(T) \leq O\left(T^{3/5}(\ln T)^2\right)$ with probability $1 - O(T^{-1})$. Chen et al. (2020b) then prove the lower bound for non-concave $G(\cdot)$ and show that the upper bound matches the lower bound. They prove that there exist a problem instance such that for any learning-while-doing policy $\pi$ and the sequential decisions $\{p_t, y_t\}_{t=1}^T$ the policy $\pi$ produces, it holds for sufficiently large $T$ that $\sup_\lambda \mathbb{E}\left[V^* - \sum_{t=1}^T Q(p_t, y_t)\right] \geq C_5 \times T^{3/5}/\ln T$ for some constant $C_5 > 0$. The lower bound is established by a novel information-theoretical argument based on generalized squared Hellinger distance, which is significantly different from conventional arguments that are based on Kullback–Leibler divergence.

## 12.4 Parametric Learning with Limited Price Changes

Models discussed in Sects. 12.2 and 12.3 assume that price can be adjusted at the beginning of every period. In practice, however, retailers may hesitate changing prices too frequently. Cheung et al. (2017) discussed several practical reasons for not

**Algorithm 7** A discretization + tournament approach with non-concave $G(\cdot)$

**Input**: time horizon $T$, discretization parameter $J$, parameters $C_1, C_3 > 0$;
Let $\{p_j\}_{j=1}^{J}$ be $J$ prices that evenly partition $[\underline{p}, \overline{p}]$; $\mathcal{S}_0 = [J]$;
**for** $\gamma = 0, 1, 2, \cdots$ until $T$ review periods are reached **do**
  $\Delta_\gamma \leftarrow 2^{-\gamma}$, $n_\gamma \leftarrow \lceil g(C_3/\Delta_\gamma^2)\rceil^{***}$, $\mathcal{V}_{\gamma,0} \leftarrow \mathcal{S}_\gamma$, $\ell \leftarrow 0$; ▷ the tournament phase
  **while** $|\mathcal{V}_{\gamma,\ell}| > 1$ **do**
    Group prices in $\mathcal{V}_{\gamma,\ell}$ into pairs;
    If $|\mathcal{V}_{\gamma,\ell}|$ is odd then transfer one arbitrary price to form $\mathcal{V}_{\gamma,\ell+1}$; else set $\mathcal{V}_{\gamma,\ell+1} = \emptyset$;
    **for** each pair of prices $p, p'$ in $\mathcal{V}_{\gamma,\ell}$ **do**
      $\hat{y} \leftarrow$ SEARCHORDERUPTO$(p, n_\gamma, C_1)$, $\hat{y}' \leftarrow$ SEARCHORDERUPTO$(p', n_\gamma, C_1)$;
      $\hat{\Delta}_G(p, p') \leftarrow$ ESTIMATEGDIFFERENCE$(p, \hat{y}, p', \hat{y}', n_\gamma)$;
      Update $\mathcal{V}_{\gamma,\ell+1} \leftarrow \mathcal{V}_{\gamma,\ell+1} \cup \{p'\}$ if $\hat{\Delta}_G(p, p') > 0$ and $\mathcal{V}_{\gamma,\ell+1} \leftarrow \mathcal{V}_{\gamma,\ell+1} \cup \{p\}$ otherwise;
    **end for**
    $\ell \leftarrow \ell + 1$;
  **end while**
  Obtain $\hat{p}_\gamma$ as the only price in $\mathcal{V}_{\gamma,\ell}$ and initialize $\mathcal{S}_{\gamma+1} \leftarrow \emptyset$;
  ▷ the elimination phase
  **for** each $p \in \mathcal{S}_\gamma$ **do**
    $\hat{y}_1 \leftarrow$ SEARCHORDERUPTO$(\hat{p}_\gamma, n_\gamma, C_1)$, $\hat{y}_2 \leftarrow$ SEARCHORDERUPTO$(p, n_\gamma, C_1)$;
    $\hat{\Delta}_G(\hat{p}_\gamma, p) \leftarrow$ EstimateGDifference$(\hat{p}_\gamma, p)$;
    If $\hat{\Delta}_G(\hat{p}_\gamma, p) \geq -\Delta_\gamma$ then update $\mathcal{S}_{\gamma+1} \leftarrow \mathcal{S}_{\gamma+1} \cup \{p\}$;
  **end for**
**end for**

*** Recall that we use $g(x) := (x + \lceil \log_2(x\bar{y})\rceil)\lceil \log_2(x + \lceil \log_2(x\bar{y})\rceil)\rceil$.

allowing frequent price changes, including customers' negative responses (e.g., that may cause confusion and affect the seller's brand reputation) and the cost associated with such changes (e.g., due to changing price labels in brick-and-mortar stores, etc.). In this section, we introduce a constraint that only allows the retailer to change prices no more than a certain number of times. Clearly, such a constraint limits the firm's ability to learn demand.

Demand in period $t, t \in \{1, 2, \ldots, T\}$, is random and depends on the selling price $p_t$, and its distribution function belongs to some family parameterized by $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^k$, $k \geq 1$, where $\mathcal{Z}$ is a compact and convex set. Let $D_t(p_t, \mathbf{z})$ be the demand in period $t$ with probability mass function $f(\cdot; p_t, \mathbf{z})$, cumulative distribution function $F(\cdot; p_t, \mathbf{z})$, and support $\{d^l, d^l + 1, \ldots, d^h\}$ with $d^l$ being a nonnegative integer and $d^h \leq +\infty$, and let $d_t$ denote the realization of $D_t(p_t, \mathbf{z})$. The firm knows $f(\cdot; p_t, \mathbf{z})$ up to the parameter vector $\mathbf{z}$, which has to be learned from sales data.

Chen and Chao (2019) consider the backlog system and (Chen et al., 2020a) consider the lost-sales system with censored demand. This section will be mainly devoted to discussing algorithms and results in Chen et al. (2020a), where the firm can only observe sales data but not the actual demand when stockout occurs. Therefore, $o_t = \min\{D_t(p_t, \mathbf{z}), y_t\}$, and $(p_t, y_t)$ is adapted to the filtration generated by $\{(p_s, y_s), o_s : s = 1, \ldots, t-1\}$ under censored demand. Let $p_t \in \mathcal{P} = [p^l, p^h]$ and $y_t \in \mathcal{Y} = \{y^l, y^l + 1, \ldots, y^h\}$, where the bounds of support $0 \leq p^l \leq p^h <$

$+\infty$ and $0 \leq y^l \leq y^h < +\infty$ are known. Assume for any $p_t \in \mathcal{P}$ it holds that $\mathbb{E}[D_t(p_t, \mathbf{z})] > 0$. The state transition is $x_{t+1} = (y_t - D_t(p_t, \mathbf{z}))^+$.

The expected total profit over the planning horizon, given an admissible policy $\phi = ((p_1, y_1), (p_2, y_2), \ldots, (p_T, y_T))$, is

$$V^{\phi}(T, \mathbf{z}) = \sum_{t=1}^{T} \Big\{ p_t \mathbb{E}[\min\{D_t(p_t, \mathbf{z}), y_t\}] - \left\{ h\mathbb{E}\left[y_t - D_t(p_t, \mathbf{z})\right]^+ + b\mathbb{E}\left[D_t(p_t, \mathbf{z}) - y_t\right]^+ \right\} \Big\} \tag{12.16}$$

and the prices need to satisfy the *limited price change constraint* for some given integer $m \geq 1$:

$$\sum_{t=1}^{T-1} \mathbf{1}(p_t \neq p_{t+1}) \leq m, \tag{12.17}$$

where $\mathbf{1}(A)$ is the indicator function taking value 1 if statement A is true and 0 otherwise.

The single-period objective function is

$$G(p, y, \mathbf{z}) = p\mathbb{E}[D(p, \mathbf{z})] - h\mathbb{E}\left[y - D(p, \mathbf{z})\right]^+ - (b + p)\mathbb{E}\left[D(p, \mathbf{z}) - y\right]^+, \tag{12.18}$$

where $D(p, \mathbf{z})$ is a generic random demand when the true parameter is $\mathbf{z}$ and the price is $p \in \mathcal{P}$. For the underlying system parameter vector $\mathbf{z}$, let $(p^*, y^*)$ be a maximizer of $G(p, y, \mathbf{z})$. If $\mathbf{z}$ is known, then the firm could set $(p^*, y^*)$ every period without changing the price, and this is the clairvoyant solution, for which the $T$-period total profit is denoted as $V^*$.

Demand models are categorized into two groups, (1) the well-separated case and (2) the general case. Two probability mass functions are said to be identifiable if they are not identically the same.

### 12.4.1 Well-Separated Demand

The family of distributions $\{f(\cdot; p, z) : z \in \mathcal{Z}\}$ is called well-separated if for any $p \in \mathcal{P}$, the class of probability mass functions $\{f(\cdot; p, z) : z \in \mathcal{Z}\}$ is identifiable, i.e., $f(\cdot; p, z_1) \neq f(\cdot; p, z_2)$ for $z_1 \neq z_2 \in \mathcal{Z}$.

If a family of distributions is well-separated, then no matter what selling price $p$ is charged, the sales data will allow the firm to learn about the parameter $z$. This shows that, in the well-separated case, pricing exploration can be a side benefit from exploitation, thus no active pricing exploration is necessary.

**Algorithm 8** $m$ price changes for the well-separated case

1: Input $\hat{p}_1$, $\hat{y}_1$.
2: Let $I_i = \lceil T^{i/(m+1)} \rceil$, for $i = 1, \ldots, m$, and $I_{m+1} = T - \sum_{i=1}^{m} I_i$. Let $t_1 = 0$, and $t_i = \sum_{j=1}^{i-1} I_j$ for $i = 2, \ldots, m+2$.
3: **for** stage $i \le m+1$ **do**
4: Set

$$\tilde{y}_i = \begin{cases} \hat{y}_i, & \text{if } \hat{y}_i > d^l, \\ \min\{\max\{\hat{y}_i + \Delta, y^l\}, y^h\}, & \text{if } \hat{y}_i = d^l. \end{cases}$$

5: **for** $t = t_i + 1, \ldots, t_{i+1}$ **do**
6: $p_t = \hat{p}_i$, $y_t = \max\{x_t, \tilde{y}_i\}$, $x_{t+1} = \max\{y_t - d_t, 0\}$.
7: **end for**
8: Compute the MLE estimator for $z$ by

$$\hat{z}_i = \arg \max_{z \in \mathcal{Z}} \left\{ \sum_{\{t \in \{t_i+1, \ldots, t_{i+1}\} : d_t < y_t\}} \log f(d_t; \hat{p}_i, z) + \sum_{\{t \in \{t_i+1, \ldots, t_{i+1}\} : d_t \ge y_t\}} \log\left(1 - F(y_t - 1; \hat{p}_i, z)\right) \right\}. \tag{12.19}$$

9: Solve the data-driven optimization problem

$$(\hat{p}_{i+1}, \hat{y}_{i+1}) = \arg \max_{(p, y) \in \mathcal{P} \times \mathcal{Y}} G(p, y, \hat{z}_i). \tag{12.20}$$

10: **end for**

Chen et al. (2020a) consider two scenarios of limited-price constraint for well-separated demand. The first scenario is that the number of price changes is restricted to be no more than a given integer $m \ge 1$ that is independent of the length of planning horizon $T$, while for the second scenario, the number of allowed price changes is at most $\beta \log T$ for the $T$-period problem for some constant $\beta > 0$.

**Algorithm for $m$ Price Changes Under Well-Separated Demand** The main idea of the algorithm is to estimate the known parameter $z$ by maximum likelihood estimation based on censored demand. The detailed algorithm is presented in Algorithm 8.

As shown in Algorithm 8, exploration in the inventory space is needed. If $\hat{y}_i$ equals $d^l$, then implementing $\hat{y}_i$ will not yield any information about the demand. Hence the algorithm imposes $\tilde{y}_i = \hat{y}_i + \Delta$, which ensures to reveal some demand information with a positive probability. Then the algorithm constructs an MLE estimator using censored data, $\min\{d_t, y_t\}$, which are neither independent nor identically distributed. This is because, inventory level $y_t$ depends on carryover inventory $x_t$ that is a function of earlier inventory level and demand, and earlier demand depends on the pricing decisions. Assumption 1(i) in the paper guarantees that, with a high probability (its complement has a probability decaying exponentially fast in

$I_i$), the objective function in (12.19) is strictly concave, thus there exists a unique global maximizer.

**Regret Convergence for $m$ Price Changes Under Well-Separated Demand** Chen et al. (2020a) provide both regret upper and lower bounds for well-separated demand with $m$ price changes. The regret upper bound is $R(T) \leq C_6\ T^{\frac{1}{m+1}}$ for some constant $C_6 > 0$. The lower bound is provided as following. There exist problem instances such that the regret for any admissible learning algorithm that changes price at most $m$ times is lower bounded by $R(T) \geq C_7\ T^{\frac{1}{m+1}}$ for some constant $C_7 > 0$ and large enough $T$.

One fundamental challenge to prove this lower bound is that the times of price changes are dynamically determined, i.e., they are increasing random stopping times. An adversarial parameter class is constructed, among which a policy needs to identify the true parameter. The parameter class is constructed in a hierarchical manner such that when going further down the hierarchy the parameters are harder to distinguish. A delicate information-theoretical argument is employed to prove the lower bound. Here we only illustrate the high-level idea using a special case $m = 2$.

Chen et al. (2020a) construct a problem instance in which the inventory order-up-to level for each period is fixed and high enough so that any realization of the demand can be satisfied under any price. Therefore, the effect of lost sales and censored data is eliminated and the original joint pricing and inventory control problem is reduced to a dynamic pricing problem with fixed inventory control strategies. Suppose the demand follows a Bernoulli distribution with a single unknown parameter $z \in [0, 1]$.

Let $(p_0, p_1, p_2)$ be the $m + 1 = 3$ different prices of a policy $\pi$, $(T_0, T_1, T_2)$ be the number of time periods each price is committed to, with $T_2 = T - T_0 - T_1$. The paper constructs an adversarial parameter class consisting of $2^{m+1} = 8$ parameters, among which policy $\pi$ needs to identify the true parameter. These parameters are constructed in a hierarchical way. The 8 parameters are first partitioned into two 4-parameter groups, with the parameters in each group being close to each other, and the two groups are about $1/4$ apart. Each 4-parameter group can then be divided into two 2-parameter groups, with a distance of $T^{-1/6}$ between them. Within each 2-parameter group, the two parameters are $T^{-1/3}$ apart. A policy needs to work down the hierarchy levels to locate the true parameter, and the further it works down, the harder to differentiate between groups/parameters.

The proof first shows the tradeoff in deciding $(p_0, T_0)$ at the first hierarchy level. Assume without loss of generality that $z$ resides in the first branch of the tree. Because policy $\pi$ does not have any observations when deciding $p_0$, there is a constant probability that $p_0$ is selected to favor the other branch. This high risk yields that $T_0$ cannot be longer than $O(T^{1/3})$, because otherwise the regret accumulated during $T_0$ would immediately imply an $\Omega(T^{1/3})$ regret.

If $T_0$ is upper bounded by $O(T^{1/3})$, the tradeoff in deciding $(p_1, T_1)$ is as follows. With so few demand observations during $T_0$, policy $\pi$ will *not* be able to distinguish groups on the second level. Therefore, assuming the true $z$ resides in the first group, it can (and will) be shown that $p_1$ is selected to favor the wrong (second) group with

a constant probability. Given this risk and that the parameters between the first and second groups are distanced at $T^{-1/6}$, $T_1$ cannot be longer than $O(T^{2/3})$ to yield an $\Omega(T^{1/3})$ regret. The same argument then carries over to the third level when deciding $p_2$. After summing up the regrets from all the three levels, it is shown that the total regret of policy $\pi$ cannot be better than $\Omega(T^{1/3})$.

In making real decisions it may happen that $T$ is not clearly specified at the beginning. The firm requires that the price change be not too often, but it usually allows more price changes for longer planning horizon. Chen et al. (2020a) propose a learning algorithm where the number of price changes is restricted to $\beta \log T$ for some constant $\beta > 0$.

**Algorithm for $\beta \log T$ Price Changes Under Well-Separated Demand** The algorithm runs very similarly to the one for $m$ price changes, except that now the number of periods in $i$ is given by $I_i = \lceil I_0 v^i \rceil$, $i = 1, 2 \ldots, N$, and there is a total of $N = O(\log T)$ iterations.

**Regret Convergence for $\beta \log T$ Price Changes Under Well-Separated Demand** The regret convergence rate for the algorithm with less than $\beta \log T$ price changes is upper bounded as $R(T) \leq C_8 \log T$, for a constant $C_8 > 0$ and large enough $T$. The lower bound is also provided. There exist problem instances such that the regret for any learning algorithm satisfies $R(T) \geq C_9 \log T$ for some constant $C_9 > 0$ and $T \geq 1$.

### *12.4.2 General Demand*

Now we consider the more general case that the parameters in probability mass function $f(\cdot; p, \mathbf{z})$ is a $k$-dimensional vector, i.e., $\mathbf{z} = (z_1, \ldots, z_k) \in \mathcal{Z} \subset \mathbb{R}^k$ for some integer $k \geq 1$. For a set of given prices $\mathbf{p} = (p_1, \ldots, p_k) \in \mathcal{P}^k$, and correspondingly realized demands $\mathbf{d} = (d_1, \ldots, d_k) \in \{d^l, d^l + 1, \ldots, d^h\}^k$, define

$$Q^{\mathbf{p},\mathbf{z}}(\mathbf{d}) = \prod_{j=1}^{k} f(d_j; p_j, \mathbf{z}).$$

The family of distributions $\{Q^{\mathbf{p},\mathbf{z}}(\cdot) : \mathbf{z} \in \mathcal{Z}\}$ is said to belong to the general case if there exist $k$ price points $\bar{\mathbf{p}} = (\bar{p}_1, \ldots, \bar{p}_k) \in \mathcal{P}^k$ such that the family of distributions $\{Q^{\bar{\mathbf{p}},\mathbf{z}}(\cdot) : \mathbf{z} \in \mathcal{Z}\}$ is identifiable, i.e., $Q^{\bar{\mathbf{p}},\mathbf{z}_1}(\cdot) \neq Q^{\bar{\mathbf{p}},\mathbf{z}_2}(\cdot)$ for any $\mathbf{z}_1 \neq \mathbf{z}_2$ in $\mathcal{Z}$.

Suppose we are allowed to make up to $m$ price changes during the planning horizon. We consider the case of $m \geq k$ in this section, as in the case of $m < k$ no algorithm will be able to identify the $k$ unknown parameters and, therefore, the regret would be linear in $T$.

**Algorithm for General Demand** The algorithm follows an exploration-exploitation framework, and the unknown parameter vector $\mathbf{z}$ is estimated by MLE. Detailed algorithm is presented in Algorithm 9.

**Algorithm 9** $m \geq k$ price changes for the general case

Input $\bar{y} \in \mathcal{Y}$ for the initial inventory order-up-to level, and constant $s > 0$.
Let $I = \lceil T^{1/2}/k \rceil$.
**for** $i = 1, \cdots, k$ **do**
  **for** $t = (i-1)I + 1, \ldots, iI$ **do**
    Set $p_t = \bar{p}_i$.
  **end for**
  For $t = (i-1)I + 1$, set $y_t = \max\{x_t, \bar{y}\}$, thus $x_{t+1} = \max\{y_t - d_t, 0\}$;
  **for** $t = (i-1)I + 2, \ldots, iI$ **do**
    Set

$$y_t = \begin{cases} y_{t-1}, & \text{if } d_{t-1} < y_{t-1}; \\ \min\left\{(1+s)y_{t-1}, \lceil \log T \rceil\right\}, & \text{otherwise.} \end{cases}$$

$$x_{t+1} = \max\{y_t - d_t, 0\}.$$

  **end for**
**end for**
Estimate $\mathbf{z}$ by the MLE estimator

$$\hat{\mathbf{z}} = \arg \max_{\mathbf{z} \in \mathcal{Z}} \left\{ \sum_{\{t \in \{1,\ldots,kI\}: y_t > d_t\}} \log f(d_t; p_t, \mathbf{z}) + \sum_{\{t \in \{1,\ldots,kI\}: y_t \leq d_t\}} \log\left(1 - F(y_t - 1; p_t, \mathbf{z})\right) \right\}. \tag{12.21}$$

Solve the data-driven optimization problem $(\hat{p}, \hat{y}) = \max_{(p,y) \in \mathcal{P} \times \mathcal{Y}} G(p, y, \hat{\mathbf{z}})$.
**for** $t = kI + 1, \ldots, T$ **do**
  $p_t = \hat{p}$, $y_t = \max\{x_t, \hat{y}\}$, and $x_{t+1} = \max\{0, y_t - d_t\}$.
**end for**

As shown in Algorithm 9, during the exploration phase, Algorithm-II experiments with $k$ prices (thus $k - 1$ price changes). Because of censored data, the true demand realizations exceeding inventory level cannot be observed. To make sure to receive sufficient demand data, every time a stockout occurs, the algorithm increases the order-up-to level by a certainty percentage. Because $d^h$ may be infinity, this does not mean that the data censoring issue will be totally resolved, but with high probability. In the MLE step, the sales data $\min\{d_t, y_t\}$ are correlated and non-identically distributed, because inventory levels $y_t$ are dependent through the "raising inventory" decisions as well as the carryover inventories. Propositions in Chen et al. (2020a) state that, despite the dependent data, the MLE possesses the desired property. The empirical optimal solution is implemented for the rest of the planning horizon, resulting in $k$ price changes.

**Regret Convergence for General Demand** Chen et al. (2020a) provide the regret upper bounded for the general demand case as follows: if the demand is unbounded $d^h = +\infty$, then the regret for general demands is upper bounded by $R(T) \leq C_{10} T^{1/2} \log T$; if the demand is bounded $d^h < +\infty$, then the regret for general

demands is upper bounded by $R(T) \leq C_{10}T^{1/2}$, for some constant $C_{10} > 0$. The theoretical lower bound for this problem is $\Omega(T^{1/2})$, which is established in Broder and Rusmevichientong (2012) for a dynamic pricing problem with infinite initial inventory.

## 12.5 Backlog System with Fixed Ordering Cost

In this section, we consider the presence of fixed ordering cost, which is a fixed cost that is incurred by the firm whenever a positive amount of inventory is ordered.

Demand is modeled as $D = D_0(p) + \beta$, where $D_0 : [0, 1] \to [\underline{d}_0, \overline{d}_0]$ is the (expected) demand function and $\beta$ is the random noise with 0 mean. Unsatisfied demands are backlogged. Chen et al. (2021b) consider both linear models and generalized linear models for $D_0(p)$ with *unknown* parameters $\theta_0$. The distribution for $\beta$ is unknown in the nonparametric sense. Let $k > 0$ be the fixed ordering cost, $c > 0$ be the variable ordering cost of ordering one unit of inventory, and $h : \mathbb{R} \to \mathbb{R}^+$ be the holding cost (when the remaining inventory level is positive) or the backlogging cost (when the remaining inventory level is negative). The instantaneous reward for period $t$ is

$$\mathfrak{r}_t = -k \times \mathbf{1}\{y_t > x_t\} - c(y_t - x_t) + p_t(D_0(p_t) + \beta_t) - h(y_t - D_0(p_t) - \beta_t),$$

and the firm would like to maximize the $T$-period total reward.

With known demand curve $D_0$ and noise distribution $\mu_0$, the work of Chen and Simchi-Levi (2004a) proves that, under mild conditions, for both the average and discounted profit criterion there exists an $(s, S, \mathbf{p})$ policy that is optimal in the long run. Under an $(s, S, \mathbf{p})$-policy, the retailer will only order new inventories when $x_t < s$, and after the ordering of new inventories maintain $y_t = S$. The function $\mathbf{p}$ prescribes the pricing decision that depends on the initial inventory level of the same period.

The performance of a particular $(s, S, \mathbf{p})$ policy can be evaluated as follows. Define $H_0(x, p; \mu)$ as the *expected* immediate reward of pricing decision $p$ at inventory level $x$, without ordering new inventories. It is easy to verify that

$$H_0(x, p; \mu) = -\mathbb{E}_\mu[h(x - D_0(p) - \beta)] + pD_0(p) - cD_0(p). \tag{12.22}$$

For a certain $(s, S, \mathbf{p})$ policy, define quantities $I(s, x, \mathbf{p}; \mu)$ and $M(s, x, \mathbf{p}; \mu)$ as follows:

$$I(s, x, \mathbf{p}; \mu) = \begin{cases} H_0(x, \mathbf{p}(x); \mu) + \mathbb{E}_\mu[I(s, x - D_0(\mathbf{p}(x)) - \beta, \mathbf{p}; \mu)], & x \geq s, \\ 0, & x < s; \end{cases} \tag{12.23}$$

$$M(s,x,\mathbf{p};\mu)=\begin{cases}1+\mathbb{E}_{\mu}[M(s,x-D_0(\mathbf{p}(x))-\beta,\mathbf{p};\mu)], & x\geq s,\\ 0, & x<s;\end{cases} \tag{12.24}$$

Define $r(s,S,\mathbf{p};\mu)$ as

$$r(s,S,\mathbf{p};\mu)=\frac{-k+I(s,S,\mathbf{p};\mu)}{M(s,S,\mathbf{p};\mu)}. \tag{12.25}$$

When $I(s,S,\mathbf{p};\mu_0)$ and $M(s,S,\mathbf{p};\mu_0)$ are bounded, Lemma 2 from Chen and Simchi-Levi (2004a) shows that $\lim_{T\to\infty} R_T(\pi)=r(s,S,\mathbf{p};\mu_0)$.

**Learning Algorithm** The learning algorithm proposed in Chen et al. (2021b) is based on an $(s,S,\mathbf{p})$-policy with evolving inventory levels $(s,S)$ and pricing strategies $\mathbf{p}$. Because unsatisfied demands are backlogged, the decision maker can observe true demand realizations. A regularized least-squares estimation is used to estimate $\theta_0$, and a sample average approximation approach is used to construct an empirical distribution for $\beta$.

Next we present the detailed learning algorithm. For linear models, $\mathfrak{D}(\eta(p)|\theta_0)=\eta(p)^{\top}\theta_0$, and the unknown parameter $\theta_0$ is estimated by the (regularized) least-squares estimation, i.e., let

$$\hat{\theta}_{\text{Linear}}:=\arg\min_{\theta\in\mathbb{R}^{\mathfrak{d}}}\left\{\frac{1}{2}\sum_{t\in\mathcal{H}}\left|d_t-\langle\eta(p_t),\theta\rangle\right|^2+\frac{1}{2}\|\theta\|_2^2\right\}. \tag{12.26}$$

For generalized linear models, $\mathfrak{D}(\eta(p)|\theta_0)=\upsilon(\eta(p)^{\top}\theta_0)$ for $\upsilon(\cdot)$ as a given *link function*. Let the unknown parameter $\theta_0$ be estimated by

$$\hat{\theta}_{\text{GLM}}:=\arg\min_{\theta\in\Theta}\left\|\sum_{t\in\mathcal{H}}(\upsilon(\eta(p_t)^{\top}\theta)-d_t)\eta(p_t)\right\|_{\Lambda^{-1}}. \tag{12.27}$$

Let $b\in\{1,2,\cdots\}$ be a particular epoch and $\mathcal{H}_{b-1}=\mathcal{B}_1\cup\cdots\cup\mathcal{B}_{b-1}$ be the union of all epochs prior to $b$. For time period $t\in\mathcal{H}_{b-1}$, let $p_t$ be the advertised price and $d_t=D_0(p_t)+\beta_t$ be the realized demand. Let the *estimate* $\hat{\theta}_b$ of the unknown regression parameter $\theta_0$ be computed by (12.26) if demand is linear or (12.27) if demand is generalized linear given samples from $\mathcal{H}_{b-1}$. Define $\Lambda_b:=I_{\mathfrak{d}\times\mathfrak{d}}+\sum_{t\in\mathcal{H}_{b-1}}\eta(p_t)\eta(p_t)^{\top}$. For every $p\in[0,1]$, define $\Delta_b(p)$ as

$$\Delta_b(p):=\gamma\sqrt{\eta(p)^{\top}\Lambda_b^{-1}\eta(p)},$$

where $\gamma>0$ is the oracle-specific parameter. We then define an upper estimate of $D_0$, $\bar{D}_b$, as

$$\bar{D}_b(p):=\min\left\{\overline{d}_0,\underline{d}_0+L^2(1-p),\mathfrak{D}(\eta(p)|\hat{\theta}_b)+\Delta_b(p)\right\}, \tag{12.28}$$

where $\overline{d}_0, \underline{d}_0$ are maximum and minimum demands and $L$ is the Lipschitz constant. Note that the Lipschitz continuity of $\eta(p)$ and $\Lambda_b \succeq I$ imply the continuity of $\Delta_b(\cdot)$ in $p$, which further implies the continuity of $\bar{D}_b(\cdot)$ in $p$.

One key challenge in the learning-while-doing setting is the fact that all of the important quantities $H_0$, $I$, $M$ and $r$ involve expectational evaluated under the noise distribution $\mu_0$, an object which we do not know a priori. In this section, we give details on how empirical distributions are used to approximate $\mu_0$.

At the beginning of epoch $b$, let $\mathcal{E}_{<b} \subseteq \mathcal{B}_1 \cup \cdots \mathcal{B}_{b-1}$ be a *non-empty subset* of historical selling periods used to approximate the noise distribution $\mu_0$. We define the empirical noise distribution $\hat{\mu}_b$ as

$$\hat{\mu}_b := \frac{1}{|\mathcal{E}_{<b}|} \sum_{t \in \mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})], \tag{12.29}$$

where $\mathbb{I}[\beta']$ is the point mass at $\beta'$ and $b(t)$ denotes the epoch to which selling period $t$ belongs. Note that samples in $\{d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})\}_t$ are *dependent* because both $p_t$ and $\hat{\theta}_{b(t)}$ are dependent across periods. Due to technical reasons, $\mathcal{E}_{<b}$ is *not* chosen to include all selling periods prior to epoch $b$. Instead, we construct $\mathcal{E}_{<b}$ such that all $t \in \mathcal{E}_{<b}$ have small estimation errors of $D_0$ on the advertised prices.

To further upper bound the deviation of $H_0(x, p; \hat{\mu}_b)$ from $H_0(x, p; \mu_0)$, we need to demonstrate that the empirical distribution $\hat{\mu}_b$ is close to the true noise distribution $\mu_0$. Because such deviations must include the estimation errors of $D_0$ by $\bar{D}_{b(t)}$ themselves, it is crucial to select time periods $t \in \mathcal{B}_1 \cup \cdots \mathcal{B}_{b-1}$ during which the error $\Delta_{b(t)}(p_t)$ is small. To this end, we define $\mathcal{E}_{<b}$ as

$$\mathcal{E}_{<b} := \left\{ t \in \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{b-1} : \Delta_{b(t)}(p_t) \leq \kappa/\sqrt{b} \right\}, \tag{12.30}$$

where $\kappa > 0$ is a scaling algorithm parameter, set as $\kappa = 2\underline{d}^{-3/2}\overline{d}S^{3/2}\gamma\sqrt{\mathfrak{d}\ln(TL^2)}$. Note that $\kappa$ will only depend logarithmically on $T$. As is shown in the proof of the paper, the selection of $\kappa$ leads to $|\mathcal{E}_{<b}| \geq b/2$, meaning that the set is non-empty, and, therefore, the definition in Eq. (12.30) is proper. The idea of the construction of $\mathcal{E}_{<b}$ in Eq. (12.30) is as follows. Note that $d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)}) = \beta_t + (\mathfrak{D}(\eta(p_t)|\theta_0) - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)}))$. While $\beta_t$ is the desired sample from the noise distribution, $\mathfrak{D}(\eta(p_t)|\theta_0) - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})$ is incurred due to the estimation error of $\hat{\theta}_{b(t)}$, which may be very large. Also note that the absolute value of this estimation error is upper bounded by $\Delta_{b(t)}(p_t)$. Constructing $\mathcal{E}_{<b}$ as in Eq. (12.30) allows us to only exploit selling periods during which the estimation errors are sufficiently small. This ensures that the obtained (approximate) noise samples $\{d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})\}_{t \in \mathcal{E}_{<b}}$ are of high quality.

With the upper-confidence bounds $\bar{D}_b$ and the approximate noise distribution $\hat{\mu}_b$ constructed at the beginning of epoch $b$, (Chen et al., 2021b) use the dynamic programming approach detailed in the work of Chen and Simchi-Levi (2004a) to obtain an approximately optimal strategy $(s_b, S_b, \mathbf{p}_b)$ to be carried out during epoch $b$.

First define an upper bound estimate $\bar{H}_b(x, p; \hat{\mu}_b)$ on $H_0(x, p; \hat{\mu}_b)$ as

$$\bar{H}_b(x, p; \hat{\mu}_b) := -\mathbb{E}_{\hat{\mu}_b}[h(x - \bar{D}_b(p) - \beta)] + p\bar{D}_b(p) - c\bar{D}_b(p) + (c + L')\Delta_b(p), \tag{12.31}$$

where $L'$ is a constant defined in Assumption (A3) of the paper.

For any $s \in [\underline{s}, \bar{s}]$, $S \in [\underline{S}, \bar{S}]$, $r \in \mathbb{R}$, demand function $D : [0, 1] \to [\underline{d}, \infty)$, noise distribution $\mu$ and their associated $H : \mathbb{R} \times [0, 1] \to \mathbb{R}$, define

$$\phi^{(s,S)}(x; D, r, \mu) := \begin{cases} \sup_{p\in[0,1]} H(x, p; \mu) - r + \mathbb{E}_\mu[\phi^{(s,S)}(x - D(p) - \beta; D, r, \mu)], & x \geq s; \\ 0, & x < s. \end{cases} \tag{12.32}$$

With $D = \bar{D}_b$ and $H = \bar{H}_b(\cdot, \cdot; \hat{\mu}_b)$, the functions $\phi^{(s,S)}(x; \bar{D}_b, r, \hat{\mu}_b)$ can be computed for every $s \in [\underline{s}, \bar{s}]$, $S \in [\underline{S}, \bar{S}]$ and $r \in \mathbb{R}$, since both $H(\cdot, \cdot; \hat{\mu}_b)$ and the expectation with respect to $\hat{\mu}_b$ can be evaluated. For every $(s, S)$, define

$$\bar{r}_b(s, S) := \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\} \tag{12.33}$$

and let the pricing strategy $\mathbf{p}$ (associated with inventory levels $s, S$) be the optimal solution to the $\phi^{(s,S)}(\cdot; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b)$ dynamic programming; that is, $\mathbf{p}(x)$ is defined such that $\phi^{(s,S)}(x; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b) = \bar{H}_b(x, \mathbf{p}(x); \hat{\mu}_b) - \bar{r}_b(s, S) + \mathbb{E}_{\hat{\mu}_b}[\phi^{(s,S)}(x - \bar{D}_b(\mathbf{p}(x)) - \beta; \bar{D}_b, \bar{r}_b(s, S), \hat{\mu}_b)]$ for all $x$.

Comparing equations in (12.32)–(12.33) with those in (12.22)–(12.25), it is easy to observe connections between them. $r(s, S, \mathbf{p}; \mu)$ in (12.25) represents the expected per-period profit, which includes both the immediate reward $H$ and the fixed ordering cost $k$. On the other hand, $\phi^{(s,S)}(S; D, r, \mu)$ in (12.32) accumulates the immediate reward $H$ over time and subtracts a constant $r$ every period. If the constant $r$ in (12.32) equals the expected per-period profit involving both $H$ and $k$, intuitively one would expect $\phi^{(s,S)}(S; D, r, \mu)$ to be equal to $k$. Lemma 3 of Chen and Simchi-Levi (2004b) confirms this connection, which shows that $\phi^{(s,S)}(S; D, r^*(s, S), \mu) = k$, where $r^*(s, S) = \sup_{\mathbf{p}} r(s, S, \mathbf{p}; \mu)$. Therefore, $\bar{r}_b(s, S)$ can be considered as an empirical approximation of $r^*(s, S)$.

We finally remark that in practice, one may discretize the choices of $s$, $S$, $x$, and $p$ in the dynamic programming scheme described above with granularity $T^{-1}$. This leads to a computationally efficient algorithm. On the other hand, by the Lipschitz property of $\overline{H}_b(\cdot, \cdot; \hat{\mu}_b)$, it can be shown that the error caused by discretization is at most $O(T^{-1})$, which does not affect the order of the overall regret.

The proposed algorithm is based on an $(s, S, \mathbf{p})$-policy with evolving inventory levels $(s, S)$ and pricing strategies $\mathbf{p}$. As mentioned earlier, in the learning algorithm the $T$ time periods are partitioned into *epochs*, labeled as $\mathcal{B}_1, \mathcal{B}_2, \cdots$. Re-stocking only occurs at the first time period of each epoch $\mathcal{B}_b$, $b \in \{1, 2, \cdots\}$. Each epoch $\mathcal{B}_b$ is also associated with inventory levels $(s_b, S_b)$ and pricing strategy $\mathbf{p}_b$, such that for the first time period $t_b \in \mathcal{B}_b$, the re-stocked inventory level is $y_{t_b} = S_b$; the epoch

**Algorithm 10** The main algorithm: dynamic inventory control and pricing with unknown demand

1: **Input**: problem parameters $k, c, h$, time horizon $T$, the regression-oracle-specific parameter $\gamma$.
2: **Output**: inventory and pricing decisions $y_t$, $p_t$ for each $t \in [T]$.
3: **for** epoch $b = 1, 2, 3, \cdots$ **do**
4: Compute the model estimate $\hat{\theta}_b$ using the regression oracle $O$ and samples from $\mathcal{H}_{b-1}$;
5: Construct upper-confidence bounds $\bar{D}_b$ as in Eqs. (12.28, 12.31);
6: Construct $\hat{\mu}_b = \frac{1}{|\mathcal{E}_{<b}|} \sum_{t \in \mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})]$, where $\mathcal{E}_{<b}$ is constructed in Eq. (12.30);
7: For every $s \in [\underline{s}, \bar{s}]$, $S \in [\underline{S}, \bar{S}]$ compute $\phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b)$ as in Eq. (12.32) and find $\bar{r}_b(s, S) = \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\}$;
8: Select $(s_b, S_b) = \arg\max_{s,S} \bar{r}_b(s, S)$ and let $\mathbf{p}_b$ be the optimal pricing decisions associated with dynamic programming $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b(s_b, S_b), \hat{\mu}_b)$;
9: For the first time period $t_b$ in epoch $\mathcal{B}_b$ set $y_{t_b} = S_b$ and $p_{t_b} = \mathbf{p}_b(S_b)$; for the rest of epoch $\mathcal{B}_b$ set $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$; epoch $\mathcal{B}_b$ terminates once $x_t < s_b$;
10: **end for**

$\mathcal{B}_b$ terminates whenever $x_t < s_b$, and for all $t \in \mathcal{B}_b \backslash \{t_b\}$, $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$. Algorithm 10 gives a pseudo-code description of the proposed algorithm.

Updates of the $(s, S, \mathbf{p})$ policies being implemented occur at the beginning of each epoch, as detailed from Step 4 to Step 8 in Algorithm 10. More specifically, at the beginning of epoch $b$ when policy update is due, the algorithm first collects all realized demand information from previous epochs to construct model estimate $\hat{\theta}_b$ (of the demand-rate curve) and noise distribution $\hat{\mu}_b$. With estimates $\hat{\theta}_b$ and $\hat{\mu}_b$, dynamic programming (reflected in $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b, \hat{\mu}_b)$) is computed to obtain an approximately optimal pricing function $\mathbf{p}_b$, as well as the inventory levels $s_b, S_b$.

**Regret Convergence** Regret of the algorithm described above is upper bounded by $\widetilde{O}(T^{1/2})$ with probability $1 - O(T^{-1})$, where $\pi^*$ is the optimal policy that maximizes $r(s, S, \mathbf{p}; \mu_0)$. In the $\widetilde{O}(\cdot)$ notation we omit polynomial dependency on $\log T$ and other problem parameters. With $k = c = 0$ and $h(\cdot) \equiv 0$, the problem becomes a pure pricing problem with unknown linear demand functions. As long as $\tau > 1$, the work of Broder and Rusmevichientong (2012) proves an $\Omega(T^{1/2})$ lower bound for any admissible pricing policies. Therefore, the $\widetilde{O}(T^{1/2})$ regret established here is optimal.

In Algorithm 10, a dynamic programming needs to be carried out after each epoch $b$ to obtain a new policy $(s_b, S_b, \mathbf{p}_b)$. Because each epoch lasts at most $\overline{S}/\underline{d} = O(1)$ selling periods, the algorithm requires $\Omega(T)$ DP calculations which can be computationally expensive. Chen et al. (2021b) then propose an improved algorithm that only needs $O(\tau \log T)$ DP calculations to achieve virtually the same regret, which is much more computationally efficient.

**Algorithm with Infrequent DP Updates** The detailed description is presented in Algorithm 11.

Note that in Algorithm 11, a new $(s, S, \mathbf{p})$ policy is computed only if $2^\iota$, $\iota \in \{1, 2, \cdots, \}$ epochs are met, or the determinant of the sample covariance

**Algorithm 11** Dynamic inventory control and pricing with infrequent DP solutions

1: **Input**: problem parameters $\kappa, c, h$, time horizon $T$, the regression oracle specific parameter $\gamma$.
2: **Output**: inventory and pricing decisions $y_t$, $p_t$ for each $t \in [T]$.
3: Initialize: $\hat{\theta}_0 = 0^{\mathfrak{d}}$, $\Lambda_1 = I_{\mathfrak{d}\times\mathfrak{d}}$ and $\zeta_1 = 1$;
4: **for** epoch $b = 1, 2, 3, \cdots$ **do**
5: **if** $\det(\Lambda_b) \geq 2\zeta_b$ or $b = 2^\iota$ for some $\iota \in \mathbb{N}$ **then**
6: Update $\zeta_{b+1} = \det(\Lambda_b)$ and compute the model estimate $\hat{\theta}_b$ using the regression oracle $\mathcal{O}$ and samples from $\mathcal{H}_{b-1}$;
7: Construct upper-confidence bounds $\bar{D}_b$ as in Eqs. (12.28,12.31);
8: Construct $\hat{\mu}_b = \frac{1}{|\mathcal{E}_{<b}|}\sum_{t\in\mathcal{E}_{<b}} \mathbb{I}[d_t - \mathfrak{D}(\eta(p_t)|\hat{\theta}_{b(t)})]$, where $\mathcal{E}_{<b}$ is constructed in Eq. (12.30);
9: For every $s, S \in [\underline{s}, \overline{S}]$ compute $\phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b)$ as in Eq. (12.32) and find $\bar{r}_b(s, S) = \inf\{r \in \mathbb{R} : \phi^{(s,S)}(S; \bar{D}_b, r, \hat{\mu}_b) = k\}$;
10: Select $(s_b, S_b) = \arg\max_{s,S} \bar{r}_b(s, S)$ and let $\mathbf{p}_b$ be the optimal pricing decisions associated with dynamic programming $\phi^{(s_b,S_b)}(\cdot; \bar{D}_b, \bar{r}_b(s_b, S_b), \hat{\mu}_b)$;
11: **else**
12: Set $\hat{\theta}_b = \hat{\theta}_{b-1}$, $\zeta_{b+1} = \zeta_b$, $\overline{D}_b = \overline{D}_{b-1}$, $\hat{\mu}_b = \hat{\mu}_{b-1}$,$s_b = s_{b-1}$, $S_b = S_{b-1}$ and $\mathbf{p}_b = \mathbf{p}_{b-1}$;
13: **end if**
14: If the current inventory level exceeds $S_b$, set $p_t = 0$ until inventory level falls below $S_b$; *
15: For the first time period $t_b$ in epoch $\mathcal{B}_b$ set $y_{t_b} = S_b$ and $p_{t_b} = \mathbf{p}_b(S_b)$; for the rest of epoch $\mathcal{B}_b$ set $y_t = x_t$ and $p_t = \mathbf{p}_b(x_t)$; epoch $\mathcal{B}_b$ terminates once $x_t < s_b$;
16: Update $\Lambda_{b+1} = \Lambda_b + \sum_{t\in\mathcal{B}_b} \eta(p_t)\eta(p_t)^\top$;
17: **end for**

* Note that this step may only happen when the policy changes. It does not belong to any epoch; and since it happens very infrequently, its incurred regret can be bounded separately.

$\Lambda_b$ doubles. This greatly reduces the number of DP calculations from $O(T)$ to $O(\tau \log T)$.

**Regret Convergence for Infrequent DP Updates** For the algorithm with infrequent DP updates, the regret is upper bounded by $\widetilde{O}(T^{1/2})$ with probability $1 - O(T^{-1})$.

## 12.6 Other Models

Burnetas and Smith (2000) is one of the earliest papers, if not the first one, that studies joint pricing and inventory control with unknown demand distribution. They assume the lost-sales cost is zero and inventory perishes at the end of each period. The pricing mechanism is modeled as a multiarmed bandit problem, while the order quantity decision is made based on a stochastic approximation procedure. Burnetas and Smith (2000) proves policy convergence of their proposed algorithm. Katehakis et al. (2020) consider the joint optimization problem with discrete backlogged demand in different settings with or without a leading price. Keskin et al. (2021) study the joint pricing and inventory control problem with learning in a changing environment under a parametric demand-rate function and assume lost sales are

observable. They provide learning algorithms whose convergence rates match the theoretical lower bound.

## References


Broder, J., & Rusmevichientong, P. (2012), Dynamic pricing under a general parametric choice model. *Operations Research, 60*(4), 965–980.

Burnetas, A. N., & Smith, C. E. (2000). Adaptive ordering and pricing for perishable products. *Operations Research, 48*(3), 436–443.

Chen, B., & Chao, X. (2019). Parametric demand learning with limited price explorations in a backlog stochastic inventory system. *IISE Transactions, 51*(6), 605–613.

Chen, X., & Simchi-Levi, D. (2004a). Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case. *Operations Research, 52*(6), 887–896.

Chen, X., & Simchi-Levi, D. (2004b). Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The infinite horizon case. *Mathematics of Operations Research, 29*(3), 698–723.

Chen, B., Chao, X., & Ahn, H. S, (2019). Coordinating pricing and inventory replenishment with nonparametric demand learning. *Operations Research, 67*(4), 1035–1052.

Chen, B., Chao, X., & Wang, Y. (2020a). Data-based dynamic pricing and inventory control with censored demand and limited price changes. *Operations Research, 68*(5), 1445–1456.

Chen, B., Wang, Y., & Zhou, Y. (2020b). Optimal policies for dynamic pricing and inventory control with nonparametric censored demands. Available at SSRN 3750413.

Chen, B., Chao, X., & Shi, C. (2021a). Nonparametric learning algorithms for joint pricing and inventory control with lost-sales and censored demand. *Mathematrics of Operations Research, 46*(2), 726–756.

Chen, B., Simchi-Levi, D., Wang, Y., & Zhou, Y. (2021b). Dynamic pricing and inventory control with fixed ordering cost and incomplete demand information. *Management Science, forthcoming*.

Chen, X., & Simchi-Levi, D. (2012). Pricing and inventory management. *The Oxford Handbook of Pricing Management, 1*, 784–824.

Cheung, W. C., Simchi-Levi, D., & Wang, H. (2017). Dynamic pricing and demand learning with limited price experimentation. *Operations Research, 65*(6), 1722–1731.

Elmaghraby, W., & Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science, 49*(10), 1287–1309.

Katehakis, M. N., Yang, J., & Zhou, T. (2020). Dynamic inventory and price controls involving unknown demand on discrete nonperishable items. *Operations Research, 68*(5), 1335–1355.

Keskin, N. B., & Zeevi, A. (2014). Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research, 62*(5), 1142–1167.

Keskin, N. B., Li, Y., & Song, J. S. J. (2021). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science, 68*(3), 1938–1958.

Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization, 12*(2), 479–502.

Lei, Y. M., Jasin, S., & Sinha, A. (2014). Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. Ross School of Business Paper (1252)

Levi, R., Roundy, R. O., & Shmoys, D. B. (2007). Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research, 32*(4), 821–839.

Levi, R., Perakis, G., & Uichanco, J. (2015). The data-driven newsvendor problem: New bounds and insights. *Operations Research, 63*(6), 1294–1306.

Petruzzi, N. C., & Dada, M. (1999). Pricing and the newsvendor problem: A review with extensions. *Operations Research, 47*(2), 183–194.

Schumaker, L. (2007). *Spline functions: Basic theory*. Cambridge, UK: Cambridge University Press.

Sobel, M. J. (1981). Myopic solutions of Markov decision processes and stochastic games. *Operations Research, 29*(5), 995–1009.

Wang, Z., Deng, S., & Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research, 62*(2), 318–331.

Whitin, T. M. (1955). Inventory control and price theory. *Management Science, 2*(1), 61–68.

Wu, C. F. J., et al. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics, 14*(4), 1261–1295.

Yano, C. A., & Gilbert, S. M. (2005). Coordinated pricing and production/procurement decisions: A review. *Managing Business Interfaces* (pp. 65–103).

# Chapter 13
# Optimization in the Small-Data, Large-Scale Regime


**Vishal Gupta**

## 13.1 Why Small Data?

Despite the promises of Big Data, data in modern operations research applications can be scarce. Worse, this data scarcity is typically unavoidable. For example, in some systems, such as financial markets, data are rapidly time-varying. Consequently, only the most recent data are indicative of current conditions and obtaining additional relevant data is impossible. In other settings, data collection can be expensive, either financially or operationally. For example, when optimizing early childhood interventions to prevent adult obesity, it might take years to observe a single data point. Finally, in some settings such as medicine and education, data are highly regulated by privacy laws. These laws prohibit decision-makers from directly accessing protected data, leaving them instead to work with either (1) coarser, aggregated "summary" data (see, e.g., Gupta et al., 2020 for discussion) or (2) anonymized data that are deliberately contaminated to protect privacy (see, e.g., Dwork, 2008). In all these settings, we either cannot access as much data as we would ideally like or cannot access the kind of data we would ideally like. In this sense, data are fundamentally scarce, and any estimates of uncertain parameters in the system necessarily have low precision.

In the context of operations management, specifically, data scarcity sometimes arises as a result of personalization or customization. Indeed, real-world applications often require making thousands of separate decisions simultaneously, each customized to a particular, person, product, and instant of time. Such personalization exacerbates data scarcity, since there may only be a few similar people, products, and times historically from which to draw.

V. Gupta (✉)
USC Marshall School of Business, Los Angeles, CA, USA
e-mail: guptavis@usc.edu



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_13

This "personalization induced data scarcity" is not simply a pathological possibility, but rather a commonplace occurrence. For example, Gupta and Rusmevichientong (2021) studies data from a large online retailer that sells hundreds of thousands of products per quarter. The authors show that even among the most popular product categories, half of all product types sold have fewer than 10 total sales in the last quarter. Similarly, the MovieLens25M dataset (Harper and Konstan, 2015) consists of 25 million ratings of 62,000 movies by 162,000 users. Despite this size, 60% of movies have 10 or fewer ratings. Finally, Liu and Li (2017) observe that even when using real-time GPS traffic data from millions of drivers, many arcs in urban road network are traveled relatively infrequently, leading to "stale" data that are too old to be meaningful. Similar examples, with large datasets describing a huge number of uncertain parameters but where most parameters have a fairly limited amount of relevant data, abound throughout operation research.

In the absence of strong modeling assumptions, data scarcity limits our ability to estimate uncertain quantities effectively. Hence, most uncertain parameters in these settings necessarily admit, at best, low-precision estimates. We term decision-making settings with these features—i.e., *many* uncertain parameters, each with a *low-precision estimate*—the small-data, large-scale regime.

Despite the prevalence of applications in the small-data, large-scale regime, however, most data-driven optimization methods are inspired by and analyzed in the large-sample regime, i.e., the setting where the available data are increasing, and all uncertain parameters admit increasingly precise estimates. Many data-driven algorithms behave *very* differently in these two regimes, suggesting provably good theoretical performance in the large-sample regime might tell us nothing about an algorithm's practical performance in the small-data, large-scale regime.

Consequently, this chapter focuses on the small-data, large-scale regime, with particular emphasis on unique phenomena not typically seen in the large-sample regime. Our goal is twofold: (1) understand how these phenomena impact the performance of certain "traditional" data-driven optimization algorithms and (2) exploit these new phenomena to design better algorithms tailored to applications in this regime.

Philosophically, the distinction between large-sample and small-data, large-scale regimes mirrors the distinction between the macroscopic and molecular scales in physics. We now know that certain phenomena, like statistical and quantum mechanical effects, are essentially negligible when modeling everyday objects at the macroscopic scale such as cars, people, and buildings. However, at the molecular scale, these forces dominate other forces such as gravity and friction, and hence objects at this scale behave in "unintuitive" ways. Indeed, the guiding principle of nanotechnology is that one can engineer systems at the molecular scale to directly exploit these unintuitive phenomena to achieve performance not possible at the macroscopic sale.

Our goal in studying the small-data, large-scale regime is similar. We seek to describe and understand the new "unintuitive" phenomena that emerge in this regime in order to exploit them in the aforementioned applications, much in the same way nanotechnology does for the molecular scale.

### 13.1.1 Structure

The remainder of this chapter is organized as follows: we first introduce a somewhat stylized data-driven optimization model that allows us to easily contrast the small-data, large-scale and large-sample regimes. We then highlight unique phenomena arising in this regime and show that algorithms designed with large-sample intuition can have very poor behavior in the small-data, large-scale regime. In the second part of the chapter, we develop an alternative approach based on debiasing to illustrate that there do exist—at least in our stylized model—simple algorithms that have excellent behavior in both regimes.

## 13.2 Contrasting the Large-Sample and Small-Data, Large-Scale Regimes

### 13.2.1 Model

We begin with the optimization model

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{\mu}^\top \boldsymbol{x}, \tag{13.1}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a known, feasible region, and $\boldsymbol{\mu} \in \mathbb{R}^n$ is an unknown, deterministic vector representing uncertain parameters. Throughout, we assume that we observe a random variable $\boldsymbol{Z} \in \mathbb{R}^n$ representing an estimate of $\boldsymbol{\mu}$ such that

$$\mathbb{E}[\boldsymbol{Z}] = \boldsymbol{\mu}, \quad \text{and} \quad \mathbb{E}\left[(Z_j - \mu_j)^2\right] = 1/\nu_j \quad j = 1, \ldots, n. \tag{13.2}$$

In words, $Z_j$ is an unbiased estimator of $\mu_j$ with precision $\nu_j$. (Recall precision is the reciprocal of variance.) We make no assumptions on the convexity or shape of $\mathcal{X}$; it may be a discrete set or involve nonlinear, non-convex constraints. For convenience, we define $\nu_{\min} \equiv \min_j \nu_j$ and $\nu_{\max} \equiv \max_j \nu_j$.

Albeit stylized, Problem (13.1) subsumes network optimization applications with uncertain edge costs such as minimum spanning tree, shortest path, the traveling salesman, and matching on graphs (Bertsimas and Tsitsiklis, 1997). As noted in Elmachtoub and Grigas (2021), with a clever reformulation, Problem (13.1) also subsumes some inventory optimization applications with uncertain demands like the economic lot-sizing problem. Finally, through nonlinear transformations (which may introduce non-convexities in $\mathcal{X}$), Problem (13.1) can also model certain multiproduct pricing problems and portfolio optimization problems (Gupta et al., 2021). In this sense, Problem (13.1) represents a general setting under which to study the small-data, large-scale regime.

Our use of the probabilistic model, Eq. (13.2), however, deviates somewhat from the traditional operations literature. Equation 13.2 abstracts away from the data generation mechanism and instead focuses on the properties of the estimators $Z_j$ built from that data. Importantly, this framework allows us to describe and analyze both the large-sample and small-data, large-scale regimes in a variety of data settings with a minimal amount of mathematical overhead.

Namely, instances of Problem (13.1) under Eq. (13.2) fall under the small-data, large-scale regime when $n$ is very large relative to $\nu_{\mathsf{max}}$ (a large number of uncertainties, but all estimates are imprecise). By contrast, such instances fall under the large-sample regime when $n$ is small relative to $\nu_{\mathsf{min}}$ (a fixed number of uncertainties, and all estimates are very precise). One can formalize these definitions by introducing an asymptotic sequence of instances of Problem (13.1) (see Gupta and Rusmevichientong, 2021 for details), but the extra formalism offers little insight in what follows, and, hence, we prefer these loose descriptions.

We next provide some examples illustrating how these definitions of both regimes in terms of $n$, $\nu_{\mathsf{min}}$ and $\nu_{\mathsf{max}}$, provide a unified framework for analyzing several different data settings.

### Independent, Identically Distributed (I.I.D) Data

Following Gupta and Rusmevichientong (2021), suppose that for each $j = 1, \ldots, n$, we observe $\{\xi_{j1}, \ldots, \xi_{j,N_j}\}$ i.i.d. draws of a random variable $\xi_j$ with mean $\mu_j$. A natural estimator for $\mu_j$ is the sample average $Z_j \equiv N_j^{-1} \sum_{k=1}^{N_j} \xi_{k,N_j}$, which is unbiased. Notice the precision of $Z_j$ is proportional to $N_j$. Thus, our intuitive notion of large-sample asymptotics, i.e., $N_j \to \infty$ for all $j$, corresponds to $\nu_{\mathsf{min}} \to \infty$. By contrast, our intuitive notion of small-data, i.e., $N_j$ small and fixed for all $j$, corresponds to $\nu_{\mathsf{max}}$ small and fixed. Large-scale naturally corresponds to large $n$. In this way, both large-sample and small-data, large-scale regimes can be described entirely by the precisions and dimension $n$ in Eq. (13.2) without explicitly modeling the i.i.d. sampling.

### Weakly Stationary Time Series

Building on our previous example, suppose now that the sequence $(\xi_{j1}, \ldots, \xi_{j,N_j})$ is not i.i.d. for each $j$ but follows a weakly stationary time series. One can confirm that sample mean is still an unbiased estimate for $\mu_j$, but its precision depends not only on $N_j$ but also on the auto-covariance structure of the time series. In particular, for a highly autocorrelated time series, information accumulates slowly, and $N_j$ must be fairly large before one can learn $\mu_j$ precisely.

Fortunately, we can still discuss both regimes without explicitly specifying this covariance structure by again appealing to the precisions $\nu_{\mathsf{min}}$ and $\nu_{\mathsf{max}}$. In the large sample setting, $\nu_{\mathsf{min}}$ will be large relative to $n$, while in the small-data, large-scale setting, $\nu_{\mathsf{max}}$ will be small relative to $n$, and $n$ will be large.

**Regression Settings with Contextual Information**

Finally, suppose that we observe independent observations $(\xi_j, \boldsymbol{W}_j)$ for $j = 1, \ldots, n$, where $\mathbb{E}\left[\xi_j\right] = \mu_j$ and $\boldsymbol{W}_j \in \mathbb{R}^p$ is a fixed covariate that is informative for the $j$th uncertain parameter. For example, in logistics and routing applications, $\mu_j$ might represent the travel time on road $j$, and $\boldsymbol{W}_j$ might encode relevant information like the speed limit and length of road $j$. In such a setting, it is common to estimate $\mu_j$ by $Z_j \equiv \boldsymbol{W}_j^\top \boldsymbol{\beta}^{\mathsf{OLS}}, \quad j = 1, \ldots, n$, where

$$\boldsymbol{\beta}^{\mathsf{OLS}} \in \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{n} (\xi_j - \boldsymbol{W}_j^\top \boldsymbol{\beta})^2$$

is the ordinary least-squares fit, perhaps after transforming the covariates $\boldsymbol{W}_j$.

The behavior of these estimates depend subtly on the interplay between $n$, $p$, *and* the eigenspectrum of the matrix $\boldsymbol{W} = (\boldsymbol{W}_1^\top, \ldots, \boldsymbol{W}_n^\top)^\top \in \mathbb{R}^{n \times p}$. However, under the usual homoscedastic assumptions, the precision of $Z_j$ is known to be proportional to

$$\nu_j \propto \left( \boldsymbol{W}_j^\top \left( \boldsymbol{W}^\top \boldsymbol{W} \right)^{-1} \boldsymbol{W}_j \right)^{-1}.$$

Hence, we can still describe the large-sample and small-data, large-scale regimes without explicitly having to specify details about the structure of $\boldsymbol{W}$. Namely, this model is in the large-sample regime if $\nu_{\mathsf{min}}$ is large relative to $n$ and is in the small-data, large-scale regime if $n$ is large relative to $\nu_{\mathsf{max}}$.

Finally, we note that we have *not* assumed that $\boldsymbol{Z}$ is multivariate Gaussian, but in many of the estimation settings described above, one would expect intuitively that $\boldsymbol{Z}$ is approximately distributed as a multivariate Gaussian. Hence, we will often consider this special case to develop intuition.

We next use our above model to highlight a first important difference in these regimes.

### *13.2.2 Failure of Sample Average Approximation (SAA)*

Sample average approximation (SAA), also called empirical risk minimization (ERM) in the machine learning literature, is arguably the most fundamental data-driven optimization procedure. Many other popular procedures including

regularized ERM and distributionally robust optimization are, at least intuitively, motivated as refinements of SAA.

In our setting, the SAA procedure plugs in the estimator $\boldsymbol{Z}$ for the unknown $\boldsymbol{\mu}$ in Problem (13.1) and returns the resulting solution:

$$\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z}) \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{Z}^\top \boldsymbol{x}. \tag{13.3}$$

Under fairly mild assumptions, SAA has excellent performance in the large-sample regime. In our setting specifically, one can prove the following:

**Theorem 1 (SAA in Large-Sample Regime)** *Consider an instance of Problem* (13.1) *under Eq.* (13.2) *where* $\mathcal{X} \subseteq [0,1]^n$. *The expected sub-optimality of SAA relative to the full-information optimum satisfies*

$$0 \le \mathbb{E}\left[\boldsymbol{\mu}^\top \boldsymbol{x}^{SAA}(\boldsymbol{Z})\right] - \boldsymbol{\mu}^\top \boldsymbol{x}^* \le \frac{2n}{\sqrt{\nu_{min}}}.$$

In particular, in large-sample settings when $\nu_{\mathsf{min}}$ is large relative to $n$, SAA performs comparably to the full-information solution. For clarity, recall in the i.i.d. setting of our previous example, $\nu_{\mathsf{min}} \propto \min_j N_j$, and hence Theorem 1 shows expected performance of SAA converges to the full-information optimum at the "usual" rate of $O(N_{\min}^{-1/2})$. The proof of Theorem 1 is quite standard and, hence, omitted.

Since $\nu_{\mathsf{max}} \ge \nu_{\mathsf{min}}$, the above bound is vacuous in the small-data, large-scale regime, i.e., when $n$ is large relative to $\nu_{\mathsf{max}}$. This is not merely a weakness in analysis; SAA can have very poor performance in this regime, as seen in the following example:

**Poor Performance of SAA in Small-Data, Large-Scale Regime**

Consider an instance of Problem (13.1) under Eq. (13.2) where $Z_j \sim N(\mu_j, 1/\nu_j)$ is normally distributed,

$$(\mu_j, \nu_j) = \begin{cases} (0, 0.01) & \text{if } j \text{ is odd,} \\ (-1, 1) & \text{if } j \text{ is even,} \end{cases}$$

and

$$\mathcal{X} = \left\{ \boldsymbol{x} \in [0,1]^n : \sum_{j=1}^{n} x_j = 0.01n \right\}.$$

For convenience, assume $0.01n$ is an integer. In words, the problem seeks to identify the worst 1% of the $\mu_j$ given the noisy estimates $Z_j$. The full-information optimal value is $-0.01n$ obtained by choosing any $.01n$ even components.

The SAA solution $x_j^{\mathsf{SAA}} = \mathbb{I}\{Z_j \le q_n\}$, where $q_n$ is any solution to the equation

$$\frac{1}{n}\sum_{j=1}^{n} \mathbb{I}\{Z_j \le q\} = .01.$$

Write

$$\frac{1}{n}\sum_{j=1}^{n} \mathbb{I}\{Z_j \le q\} \;=\; \frac{1}{2}\cdot\frac{2}{n}\sum_{j:j \text{ odd}} \mathbb{I}\{Z_j \le q\} + \frac{1}{2}\cdot\frac{2}{n}\sum_{j:j \text{ even}} \mathbb{I}\{Z_j \le q\},$$

and note each sum consists of $n/2$ terms. Since the $Z_j$ are i.i.d. for odd $j$, we have by the uniform law of large numbers that

$$\frac{2}{n}\sum_{j:j \text{ odd}} \mathbb{I}\{Z_j \le q\} \to_p \mathbb{P}(Z_1 \le q) = \Phi\left(q\sqrt{v_1}\right) = \Phi(0.1q),$$

uniformly in $q$ as $n \to \infty$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Similarly, $\frac{2}{n}\sum_{j:j \text{ even}} \mathbb{I}\{Z_j \le q\} \to_p \mathbb{P}(Z_2 \le q) = \Phi(q+1)$ as $n \to \infty$. Hence, $q_n \to_p q^*$, where

$$\frac{1}{2}\Phi\left(0.1q^*\right) + \frac{1}{2}\Phi\left(q^*+1\right) = 0.01.$$

The value $q^*$ can be determined numerically as $q^* \approx -20.54$. Then, an entirely analogous argument shows the scaled performance of SAA satisfies

$$\frac{1}{n}\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z}) \to_p \frac{1}{2}\cdot 0 \cdot \mathbb{P}\left(Z_1 \le q^*\right) + \frac{1}{2}\cdot 1 \cdot \mathbb{P}\left(Z_2 \le q^*\right).$$

Hence, the relative performance of SAA to the full-information optimum satisfies

$$\frac{\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})}{\boldsymbol{\mu}^\top \boldsymbol{x}^*} \to_p \frac{\mathbb{P}(Z_2 < q^*)}{-0.02} < -10^{-83},$$

a negligibly small fraction.

Worse, had we simply chosen a feasible solution at random, our expected performance would be $-0.005n$, yielding 50% relative performance to the full-information optimum. Thus, SAA performs substantively worse than random guessing in this example.

A clever reader might argue that the crux of the issue in the preceding example is that SAA does not leverage the precision information $v_j$ and hence is "tricked" into selecting many of the odd components. A more clever algorithm that leveraged this information could avoid such a mistake.

Although this intuition is partially true, it is not the whole story. Indeed, Gupta and Rusmevichientong (2021) establishes the following theorem which shows that *no* data-driven algorithm exists which can achieve more than a fraction of the full-information performance in the small-data, large-scale regime. This behavior sharply contrasts Theorem 1.

**Theorem 2 (Full-Information Optimum Is Unattainable)** *Given any data-driven algorithm* $\boldsymbol{x}(\cdot)$ *such that* $\boldsymbol{x}(\boldsymbol{Z}) \in [0,1]^n$ *almost surely, there exists an instance of Problem* (13.1) *with* $\mathcal{X} = [0,1]^n$, *and* $v_j = 1$, $\mu_j \in \{-1,+1\}$ *and* $Z_j \sim \mathcal{N}(\mu_j, 1/v_j)$ *for all* $j$, *such that*

$$\frac{\mathbb{E}\left[\boldsymbol{\mu}^\top \boldsymbol{x}(\boldsymbol{Z})\right]}{\boldsymbol{\mu}^\top \boldsymbol{x}^*} < 0.842.$$

The bound is not tight but already highlights a distinct phenomenon in the small-data, large-scale regime, not present in the large-sample regime. No algorithm, even one with knowledge of the precisions, can expect to achieve a large fraction of full-information performance for all instances.

### *13.2.3 Best-in-Class Performance*

Since full-information performance is not generally achievable, we instead establish a different benchmark to assess data-driven procedures. To this end, we next define a notion of "best-in-class" performance for a given policy class. For simplicity of exposition, we focus our discussion on plug-in policies:

**Definition 1 (Plug-In Policy)** Given functions $f_j : \mathbb{R} \mapsto \mathbb{R}$, we define the *plug-in policy* $\boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z})$ corresponding to $\boldsymbol{f}(\cdot) = (f_1(\cdot), \ldots, f_n(\cdot))^\top$ to be

$$\boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{f}(\boldsymbol{Z})^\top \boldsymbol{x}, \tag{13.4}$$

where $\boldsymbol{f}(\boldsymbol{Z}) \in \mathbb{R}^n$ is the vector with $j$th component $f_j(Z_j)$. Given a set of functions $\mathcal{F}$, we further define the corresponding class of plug-in polices to be $\left\{\boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) : \boldsymbol{f} \in \mathcal{F}\right\}$.

We stress that the component functions $f_j(Z_j)$ in the definition may differ by $j$ or depend on auxiliary information.

Plug-in policies are computationally attractive because computing the policy for a fixed $\boldsymbol{f}(\cdot)$ requires solving an optimization problem of the same form as

Problem (13.1). Thus, if there exists a specialized algorithm for solving Problem (13.1)—as is the case with many transportation, inventory management, and pricing problems—the same algorithm can be used to evaluate $\boldsymbol{x}^f(\boldsymbol{Z})$.

We next consider some examples:

**Sample Average Approximation (SAA) as a Plug-In Policy**

SAA is an example of a plug-in policy where $f_j(Z_j) = Z_j$.

**Plug-Ins for Linear Classes**

Consider our previous regression setup where $\boldsymbol{W}_j$ encodes (known) covariate information for $\mu_j$. A classical predict-then-optimize approach might first find the ordinary least-squares estimate $\boldsymbol{\beta}^{\mathsf{OLS}}$ and then solve Problem (13.1) after replacing $\mu_j$ by $\boldsymbol{W}_j^\top \boldsymbol{\beta}^{\mathsf{OLS}}$. This policy is not a plug-in policy in the sense of Definition 1 because the vector $\boldsymbol{\beta}^{\mathsf{OLS}} = \boldsymbol{\beta}^{\mathsf{OLS}}(\boldsymbol{Z})$ itself depends on the entire data vector $\boldsymbol{Z}$. However, with probability 1, this policy is a member of the corresponding plug-in policy class for linear functions:

$$\mathcal{F}^{\mathsf{Linear}} = \{\boldsymbol{Z} \mapsto (\boldsymbol{W}_1^\top \boldsymbol{\beta}, \ldots, \boldsymbol{W}_n^\top \boldsymbol{\beta})^\top : \boldsymbol{\beta} \in \mathbb{R}^p\}.$$

In Elmachtoub and Grigas (2021), the authors argue that there exist plug-in policies in this larger class that significantly outperform the plug-in policy corresponding to $\boldsymbol{\beta}^{\mathsf{OLS}}$.

Observe that members of $\mathcal{F}^{\mathsf{Linear}}$ are constant valued (they do not depend on $\boldsymbol{Z}$), and, hence, the corresponding plug-in policies also do not depend on $\boldsymbol{Z}$. We call such classes of plug-in policies *non-data-driven.* Non-data-driven policy classes are common in machine learning but do not cover all examples of interest in data-driven optimization. For example, the SAA policy does depend on $\boldsymbol{Z}$ and hence does not belong to the non-data-driven plug-in policy class corresponding to $\mathcal{F}^{\mathsf{Linear}}$.

We next describe a data-driven plug-in policy class that does contain SAA as a member:

**Plug-Ins Based on Mixed-Effects Regression**

Define

$$\mathcal{F}^{\mathsf{ME}} = \left\{ \boldsymbol{Z} \mapsto \left( \frac{\nu_1}{\nu_1 + \tau} Z_1 + \frac{\tau}{\nu_1 + \tau} \boldsymbol{W}_1^\top \boldsymbol{\beta}, \ \ldots, \ \frac{\nu_n}{\nu_n + \tau} Z_n + \frac{\tau}{\nu_n + \tau} \boldsymbol{W}_n^\top \boldsymbol{\beta} \right)^\top \right.$$
$$\left. : \ \tau \in \mathbb{R}_+, \boldsymbol{\beta} \in \mathbb{R}^p \right\}.$$

In words, the members of $\mathcal{F}^{\mathsf{ME}}$ proxy each $\mu_j$ as an interpolation between $Z_j$ and a linear fit based on $\boldsymbol{\beta}$, where $\tau$ controls the degree of interpolation and the precision $\nu_j$ attenuates the effect. This form of interpolation arises naturally in a mixed-effects regression model of the unknown $\boldsymbol{\mu}$ where we assume $\boldsymbol{W}_j$ corresponds to some shared (fixed) effects and there is some unknown, random effect for each $j$. Moreover, the plug-in policy corresponding to $\tau = 0$ is $\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})$ (cf. Problem (13.3)), and the plug-in policies corresponding to $\tau = \infty$ exactly recover the plug-in policies corresponding to $\mathcal{F}^{\mathsf{Linear}}$. Thus, $\mathcal{F}^{\mathsf{ME}}$ strictly generalizes $\mathcal{F}^{\mathsf{Linear}}$.

Given any plug-in policy class, we define its "best" member, depending on the data $\boldsymbol{Z}$.

**Definition 2 (Oracle Policy)** Given a class $\mathcal{F}$ of functions, the *oracle plug-in policy* $\boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})$ is defined by

$$\boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z}) = \boldsymbol{x}^{f_{\mathsf{OR}}}(\boldsymbol{Z}) \text{ where } f_{\mathsf{OR}} \in \operatorname*{arg\,min}_{f \in \mathcal{F}} \boldsymbol{\mu}^\top \boldsymbol{x}^f(\boldsymbol{Z}). \tag{13.5}$$

The oracle policy minimizes the true performance, similar to the full-information solution $\boldsymbol{x}^*$ (cf. Problem (13.1)). However, unlike $\boldsymbol{x}^*$, the oracle policy is restricted to use a member of the given class. We stress that the oracle policy is defined with respect to a particular realization of the data $\boldsymbol{Z}$ and is, thus, random.

By construction, no plug-in policy from $\mathcal{F}$ outperforms its oracle member. In particular, this statement holds even for policies which are not themselves plug-in policies but are (with probability 1) contained in a plug-in policy class, such as the predict-then-optimize policy with $\boldsymbol{\beta}^{\mathsf{OLS}}$. In this sense, the oracle policy is a strong benchmark. On the other hand, computing $\boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})$ *seemingly* requires knowledge of $\boldsymbol{\mu}$, so it is not clear that we can identify a member of the given class with performance comparable to $\boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})$ using only the data at hand. (We show later that this is indeed possible in certain cases.)

Importantly, oracle policies for well-chosen policy classes often enjoy favorable properties. For example, the element of $\mathcal{F}^{\mathsf{ME}}$ corresponding to parameters $(\tau, \boldsymbol{\beta})$ can be interpreted as the posterior mean estimate of $\boldsymbol{\mu}$ assuming the data are drawn from the following Bayesian model:

$$\begin{aligned} \mu_j &\sim \mathcal{N}(\boldsymbol{W}_j^\top \boldsymbol{\beta}, 1/\tau) \quad \text{independently across } j = 1, \ldots, n, \\ Z_j \mid \boldsymbol{\mu} &\sim \mathcal{N}(\boldsymbol{\mu}_j, 1/\nu_j) \quad \text{independently across } j = 1, \ldots, n. \end{aligned}$$

Consequently, the corresponding plug-in policy is the Bayes optimal policy for this model. A standard result in Bayesian statistics is that under very mild assumptions, Bayes polices are admissible, i.e., no other data-driven policy Pareto-dominates their performance across all values of $\boldsymbol{\mu}$, whether or not the prior is correctly specified.

Hence, since the oracle policy must perform at least as well as each element of the class, it too inherits this favorable property and is non-dominated.

In this sense, comparing the performance of a given data-driven algorithm to performance of an oracle policy from a suitable policy class is arguably a more natural approach than comparing to the (unachievable) full-information optimal performance. Indeed, much of the existing literature in small-data, large-scale optimization focuses on identifying policies with performance comparable to an oracle policy, i.e., near-best-in-class performance, and we will do the same throughout the remainder.

### *13.2.4 Shortcomings of Cross-Validation*

To summarize, we have reduced our study to the problem of identifying a policy with near-best-in-class performance. A standard approach to such problems is cross-validation. In this section, we show that the performance of cross-validation in the small-data, large-scale regime is complex; in general, it might perform quite poorly, however, in some special cases it has provably good performance. These two distinct behaviors sharply contrast with the strong performance of cross-validation in the large-sample regime, highlighting yet another new phenomenon that emerges in the small-data, large-scale regime.

While there are many variants of cross-validation, we focus below on hold-out validation for simplicity. At a high-level, hold-out validation uses half the available data, i.e., *training data*, to train a policy and then estimates the performance of that policy on the remaining half of the data, i.e., *hold-out data*. One typically then compares the performance of different policies on the hold-out data to select a member of a policy class. The hope is that this procedure identifies a policy with near-best-in-class performance.

Since our general model Eq. (13.2) abstracts away from the data generation procedure to model hold-out validation, we will need some additional assumptions and notation. Our setup will mirror our previous example of "Independent, Identically Distributed (I.I.D.)" from Sect. 13.2.1.

Specifically, we assume that we observe

$$\{\xi_{j,1}, \ldots, \xi_{j,N_j}\} \text{ drawn i.i.d. such that } \mathbb{E}\left[\xi_{j,1}\right] = \mu_j, \quad j = 1, \ldots, n. \tag{13.6}$$

(For convenience, assume $N_j$ is even for each $j$.) We then estimate $\mu_j$ by $Z_j \equiv \frac{1}{N_j} \sum_{k=1}^{N_j} \xi_{j,k}$. Our estimate of $\mu_j$ based on the training data is $Z_j^{\text{train}} \equiv \frac{2}{N_j} \sum_{k \le N_j/2} \xi_{j,k}$. Similarly, our estimate of $\mu_j$ based on the hold-out set is $Z_j^{\text{hold}} \equiv \frac{2}{N_j} \sum_{k > N_j/2} \xi_{j,k}$.

With this notation, given a class $\mathcal{F}$, policy selected by hold-out cross-validation is

$$\boldsymbol{x}^{\mathsf{HO}}(\boldsymbol{Z}) = \boldsymbol{x}^{f_{\mathsf{HO}}}(\boldsymbol{Z}) \text{ where } f_{\mathsf{HO}} \in \arg\min_{f \in \mathcal{F}} \boldsymbol{Z}^{\mathsf{HO}\top} \boldsymbol{x}^{f}(\boldsymbol{Z}^{\mathsf{train}}). \tag{13.7}$$

Intuitively, the objective function of Problem (13.7) is meant to estimate $\boldsymbol{\mu}^\top \boldsymbol{x}^f(\boldsymbol{Z})$, i.e., the objective defining the oracle policy in Problem (13.5).

The next example adapted from Gupta et al. (2021) shows that in the small-data, large-scale regime, this procedure might provide a poor estimate of oracle performance for a fixed policy and, hence, might fail to identify the best-in-class policy.

**Cross-Validation Can Perform Poorly**

Consider an instance of Problem (13.1) under Eq. (13.2) in which $\mathcal{X} = [0,1]^n$. Suppose $N_j = 2$ for all $j$ and

$$\xi_j \sim \begin{cases} \mathcal{N}(-1,1) & \text{if } j \le 0.14n \\ \mathcal{N}(1,1) & \text{otherwise.} \end{cases}$$

Thus, the precision of each $Z_j$ is 2, and $Z_j^{\mathsf{train}} = \xi_{j1}$ while $Z_j^{\mathsf{hold}} = \xi_{j2}$. For convenience, assume $0.14n$ is an integer.

Finally, take $\mathcal{F} = \{\boldsymbol{Z} \mapsto \boldsymbol{1}, \boldsymbol{Z} \mapsto \boldsymbol{Z}\}$ to have only two members. The corresponding plug-in policies are (1) the zero policy that has all components equal to zero and (2) the SAA solution $\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})$.

By inspection, the oracle performance of the zero policy is 0. On the other hand, following an argument entirely analogous to our example in Sect. 13.2.2, one can see that as $n \to \infty$, the scaled, oracle performance of $\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})$ converges to

$$\frac{1}{n}\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z}) \to_p -0.14\Phi(\sqrt{2}) + 0.86\Phi(-\sqrt{2}) \approx -0.0614.$$

Hence, an oracle would prefer SAA.

Next, consider hold-out cross-validation. Cross-validation correctly estimates the performance of the zero policy to be 0. On the other hand, the scaled cross-validation performance of SAA is

$$\frac{1}{n}\sum_{j=1}^{n} \xi_{j2} \mathbb{I}\{\xi_{j1} \le 0\} \to_p -0.14\Phi(1) + 0.86\phi(-1) \approx 0.0186.$$

This is a very poor estimate of the oracle SAA performance. Moreover, hold-out cross-validation incorrectly suggests choosing the zero policy as best-in-class almost surely as $n \to \infty$.

In summary, hold-out cross-validation fails in two ways in the previous example: First, it provides a poor estimate of the SAA policy that remains poor even as $n \to \infty$. This shortcoming alone would not be enough to dismiss cross-validation as an inviable approach. Indeed, if cross-validation misestimated the performance of all policies by the same constant amount, it could still be used to identify a best-in-class policy. However, as seen above, cross-validation also fails in a second way; it misestimates differently for different policies and hence picks a poor policy from the policy class.

As discussed in Gupta et al. (2021), the key issue behind the shortcoming of cross-validation in this setting is that the hold-out objective Problem (13.7) does not actually estimate the oracle objective $\boldsymbol{\mu}^\top \boldsymbol{x}^f(\boldsymbol{Z})$ but rather estimates the objective $\boldsymbol{\mu}^\top \boldsymbol{x}^f(\boldsymbol{Z}^{\text{train}})$. In the large-sample regime where precisions are high, $\boldsymbol{x}^f(\boldsymbol{Z})$ and $\boldsymbol{x}^f(\boldsymbol{Z}^{\text{train}})$ are reasonably close, so cross-validation is an effective strategy. However, in the small-data, large-scale regime where precisions are low, sacrificing half that precision when training the policy causes $\boldsymbol{x}^f(\boldsymbol{Z})$ and $\boldsymbol{x}^f(\boldsymbol{Z}^{\text{train}})$ to be quite different. Hence, cross-validation does not identify a near-best-in-class policy.

That said, as mentioned, there are special cases where cross-validation does identify a best-in-class policy in the small-data, large-scale regime. Indeed, the above intuition suggests that for non-data-driven plug-in policy classes, e.g., the class induced by $\mathcal{F}^{\text{Linear}}$, cross-validation might correctly identify a best-in-class policy since $\boldsymbol{x}^f(\boldsymbol{Z}) = \boldsymbol{x}^f(\boldsymbol{Z}^{\text{train}})$ for all data realizations. This intuition is made formal in the following theorem:

**Theorem 3 (Cross-Validation for Non-data Driven Plug-in Classes)** *Consider a non-data-driven plug-in policy class induced by the set of functions $\mathcal{F}$. Assume*

*(i)* $2 \le |\mathcal{F}| < \infty$.
*(ii) The data sets* $\{\xi_{j,k} : k = 1, \ldots, N_j\}$ *are independent across* $j$.
*(iii)* $\xi_{j,k} - \mu_j$ *is a sub-Gaussian random variables with variance proxy at most* $\sigma^2$ *for all* $j$ *and* $k$.

*Then, there exists an absolute constant $C$ such that for any $0 < \epsilon < \frac{1}{2}$ and any instance of Problem (13.1) where $\mathcal{X} \subseteq [0,1]^n$, with probability at least $1 - \epsilon$, we have that*

$$0 \le \boldsymbol{\mu}^\top \boldsymbol{x}^{HO}(\boldsymbol{Z}) - \boldsymbol{\mu}^\top \boldsymbol{x}^{OR}(\boldsymbol{Z}) \le C\sigma\sqrt{n \log |\mathcal{F}| \log(1/\epsilon)}.$$

***Proof*** Since the plug-in policies do not depend on $\boldsymbol{Z}$, we write $\boldsymbol{x}^f$ instead of $\boldsymbol{x}^f(\boldsymbol{Z})$. Similarly, we write $\boldsymbol{x}^{\text{OR}}$ and $\boldsymbol{x}^{\text{HO}}$.

The first inequality is immediate from the definition of $\boldsymbol{x}^{\mathsf{OR}}$. For the second, observe that

$$\begin{aligned}\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{HO}} - \boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{OR}} &= \left(\boldsymbol{\mu} - \boldsymbol{Z}^{\mathsf{HO}}\right)^\top \boldsymbol{x}^{\mathsf{HO}} + \boldsymbol{Z}^{\mathsf{HO}\top}\left(\boldsymbol{x}^{\mathsf{HO}} - \boldsymbol{x}^{\mathsf{OR}}\right) \\ &\quad + \left(\boldsymbol{Z}^{\mathsf{HO}} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\mathsf{OR}} \\ &\leq \left(\boldsymbol{\mu} - \boldsymbol{Z}^{\mathsf{HO}}\right)^\top \boldsymbol{x}^{\mathsf{HO}} + \left(\boldsymbol{Z}^{\mathsf{HO}} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\mathsf{OR}} \\ &\leq 2 \sup_{\boldsymbol{f}\in\mathcal{F}} \left| \left(\boldsymbol{Z}^{HO} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\boldsymbol{f}} \right|,\end{aligned}$$

where the first inequality follows from the definition of $\boldsymbol{x}^{\mathsf{HO}}$. For a fixed $\boldsymbol{f}$, the random variable $\left(\boldsymbol{Z}^{HO} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\boldsymbol{f}}$ is mean zero and sub-Gaussian. From our independence assumption, its variance proxy is at most

$$\sigma^2 \sum_{j=1}^{n} (x_j^{\boldsymbol{f}})^2 \leq \sigma^2 n,$$

since $\mathcal{X} \subseteq [0,1]^n$. Thus, our upper bound is the supremum of at most $2|\mathcal{F}|$ mean-zero, sub-Gaussian random variables. By Massart's lemma (Wainwright, 2019, eq. 2.67), we can bound

$$\mathbb{E}\left[ 2 \sup_{\boldsymbol{f}\in\mathcal{F}} \left| \left(\boldsymbol{Z}^{HO} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\boldsymbol{f}} \right| \right] \leq 4\sigma\sqrt{n \log |\mathcal{F}|}.$$

To prove the stronger high-probability result claimed in the theorem, we need to show that the supremum concentrates at this value. To that end, (Pollard, 1990, Lemma 3.2) shows[1] that there exists an absolute constant $C_1$ such that

$$\mathbb{E}\left[ \exp\left( \frac{2 \sup_{\boldsymbol{f}\in\mathcal{F}} \left| \left(\boldsymbol{Z}^{HO} - \boldsymbol{\mu}\right)^\top \boldsymbol{x}^{\boldsymbol{f}} \right|}{C_1 \sigma \sqrt{n \log |\mathcal{F}|}} \right)^2 \right] \leq 5.$$

Applying Markov's inequality and collecting constants then completes the proof. □

[1] Pollard (1990) states this result in terms of the $\Psi$-Orlicz norm. Recall for any random variable $Y$, we define $\|Y\|_\Psi = \inf\{C > 0 : \mathbb{E}\left[\exp(Y^2/C^2)\right] \leq 5\}$. The $\Psi$-Orlicz norm is closely related to the sub-Gaussian parameter of a random variable. See, e.g., (Gupta and Rusmevichientong, 2021, Appendix A) or Rivasplata (2012).

Theorem 3 asserts the sub-optimality of cross-validation scales like $O_p(\sqrt{n})$. In most settings of interest, the oracle performance $\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})$ scales like $O_p(n)$. Thus, Theorem 3 proves that in these settings the relative sub-optimality of the policy chosen by cross-validation relative to the oracle policy is vanishing at a rate of $O_p(1/\sqrt{n})$. In this sense, cross-validation identifies a near-best-in-class policy asymptotically in the small-data, large-scale regime for non-data-driven plug-in policy classes.

Most of the regularity conditions in Theorem 3 can be weakened. For example, by leveraging classical results for suprema of sub-Gaussian processes, we can relax the finiteness of $\mathcal{F}$ to requiring that $\mathcal{F}$ has finite metric entropy. In this way, one can show that hold-out cross-validation does asymptotically select a best-in-class policy from $\mathcal{F}^{\mathsf{linear}}$ in the small-data, large-scale regime, provided the dimension of $W_j$ is not too large.

Theorem 3 contrasts with the behavior in our previous example; cross-validation does not fail in either of the two aforementioned ways. The above proof bounds the error over *all* policies in the policy class simultaneously. Hence, with high probability, cross-validation asymptotically correctly estimates the performance of *every* policy in the policy class and can identify a best-in-class policy asymptotically. This contrasting behavior when treating data-driven and non-data-driven plug-in policies highlights a subtlety of cross-validation in the small-data, large-scale regime that is not present in the large-sample regime.

Finally, while somewhat beyond the scope of this chapter, we remark that Gupta and Kallus (2021) have shown additional new phenomenon for cross-validation in a slightly different setting. Loosely, they show that if we randomize the amount of data $N_j$ for each component in a particular fashion, then cross-validation *does* allow us to identify a best-in-class policy for many data-driven policy classes in the small-data, large-scale regime with high probability. However, even with this randomization, cross-validation does *not* correctly estimate the oracle performance of any given policy in those classes; rather it uniformly misestimates their performance by an unknown multiplicative constant. In this sense, randomizing the amount of data appears to be a "middle-ground" between our earlier counterexample and Theorem 3, addressing one of the failures of cross-validation but not the other.

Developing a complete theory of cross-validation in the small-data, large-scale regime remains an open question. In the next section, we pursue an entirely different avenue for policy selection.

## 13.3 Debiasing In-Sample Performance

Since the shortcomings of cross-validation stem from sacrificing part of the data when training and part of the data when evaluating the performance of a policy, one might consider instead selecting a policy by optimizing

$$\min_{f \in \mathcal{F}} \mathbf{Z}^\top \boldsymbol{x}^f(\mathbf{Z}), \tag{13.8}$$

so that all the data are used in both steps. Unfortunately, for most interesting policy classes, this strategy fails, due to the well-known in-sample bias or "over-fitting" problem. The next theorem illustrates the issue:

**Theorem 4 (SAA Optimizes a Biased Objective)** *Suppose there exists an* $\boldsymbol{f}_{SAA} \in \mathcal{F}$ *such that* $\boldsymbol{x}^{f_{SAA}}(\mathbf{Z}) = \boldsymbol{x}^{SAA}(\mathbf{Z})$. *Then,*

$$\boldsymbol{f}_{SAA} \in \arg\min_{f \in \mathcal{F}} \mathbf{Z}^\top \boldsymbol{x}^f(\mathbf{Z}).$$

***Proof*** Write

$$\mathbf{Z}^\top \boldsymbol{x}^{\mathsf{SAA}}(\mathbf{Z}) \geq \min_{f \in \mathcal{F}} \mathbf{Z}^\top \boldsymbol{x}^f(\mathbf{Z}) \;\geq\; \min_{\boldsymbol{x} \in \mathcal{X}} \mathbf{Z}^\top \boldsymbol{x} = \mathbf{Z}^\top \boldsymbol{x}^{\mathsf{SAA}}(\mathbf{Z}),$$

where the first inequality follows because $\boldsymbol{x}^{\mathsf{SAA}}(\mathbf{Z}) = \boldsymbol{x}^{f_{\mathsf{SAA}}}(\mathbf{Z})$, the second inequality follows because $\boldsymbol{x}^f(\mathbf{Z}) \in \mathcal{X}$ for all $\boldsymbol{f} \in \mathcal{F}$ by construction, and the last equality follows by definition of $\boldsymbol{x}^{\mathsf{SAA}}(\mathbf{Z})$. Thus, we have equality throughout, proving the theorem. □

Consequently, for any sufficiently rich plug-in policy class, optimizing Problem (13.8) returns the SAA solution, which we have already seen can perform quite poorly in the small-data, large-scale regime.

Some reflection shows that at least part of the issue here is that $\mathbf{Z}^\top \boldsymbol{x}^f(\mathbf{Z})$ is a biased estimate of the oracle objective $\boldsymbol{\mu}^\top \boldsymbol{x}^f(\mathbf{Z})$ whenever $\boldsymbol{x}^f(\mathbf{Z})$ depends on $\mathbf{Z}$ (i.e., for truly data-driven plug-in classes).

Hence, our approach to identifying a best-in-class policy will be to first debias this estimator.

### *13.3.1 Stein Correction*

We leverage a classical result for Gaussian distributions attributed to Charles Stein and frequently called Stein's lemma:

**Lemma 1 (Stein's Lemma)** *Suppose* $Y \sim \mathcal{N}(\mu, \sigma^2)$. *Then, for any function* $g : \mathbb{R} \mapsto \mathbb{R}$ *that is almost everywhere differentiable and for which both expectations are defined, we have*

$$\mathbb{E}\left[(Y - \mu) g(Y)\right] = \sigma^2 \mathbb{E}\left[g'(Y)\right].$$

***Proof*** We first treat the case where $\mu = 0$ and $\sigma = 1$. Then, using integration by parts,

$$\mathbb{E}[Yg(Y)] = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} yg(y)e^{-y^2/2}dy = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} g'(y)e^{-y^2/2}dy = \mathbb{E}\left[f'(Y)\right],$$

proving the special case. For general $(\mu, \sigma)$, define the function $\overline{g}(t) = g(\mu + \sigma t)$, so that

$$\mathbb{E}[Yg(Y)] = \mathbb{E}[(\mu + \sigma\xi)\overline{g}(\xi)] = \mathbb{E}[\mu\overline{g}(\xi)] + \sigma\mathbb{E}[\xi\overline{g}(\xi)],$$

where $\xi \sim \mathcal{N}(0, 1)$. Applying the lemma to the last expectation yields

$$\mathbb{E}[Yg(Y)] = \mathbb{E}[\mu\overline{g}(\xi)] + \sigma\mathbb{E}\left[\overline{g}'(\xi)\right] = \mathbb{E}[\mu g(Y)] + \sigma^2\mathbb{E}\left[g'(Y)\right].$$

Rearranging completes the proof. □

Stein's lemma provides a tool to estimate the bias of $\boldsymbol{Z}^\top \boldsymbol{x}^f(\boldsymbol{Z})$ when $\boldsymbol{Z}$ is a multivariate Gaussian; namely, write,

$$\mathbb{E}\left[(\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z})\right] = \sum_{j=1}^{n} \mathbb{E}\left[(Z_j - \mu_j)\mathbb{E}\left[x_j^f(\boldsymbol{Z}) \mid Z_j\right]\right].$$

Define the function $g_j(t) \equiv \mathbb{E}\left[x_j^f(\boldsymbol{Z}) \mid Z_j = t\right]$. Then, applying Stein's lemma to each element of the sum shows

$$\mathbb{E}\left[(\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z})\right] = \sum_{j=1}^{n} \frac{1}{v_j}\mathbb{E}\left[g_j'(Z_j)\right].$$

Of course, the challenge is that we do not have a simple expression for $g'(Z_j)$. Instead, we approximate this derivative by a central finite step difference, i.e., we heuristically argue that for small $h$,

$$g_j'(Z_j) = \frac{g_j(Z_j + h) - g_j(Z_j - h)}{2h} + O(h^2).$$

Hence, we might expect that

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z})\right] &= \sum_{j=1}^{n} \frac{\mathbb{E}\left[g_j(Z_j + h) - g_j(Z_j - h)\right]}{2hv_j} + O(nh^2) \\ &= \sum_{j=1}^{n} \frac{\mathbb{E}\left[x_j^f(\boldsymbol{Z} + h\boldsymbol{e}_j) - x_j^f(\boldsymbol{Z} - h\boldsymbol{e}_j)\right]}{2hv_j} + O(nh^2),\end{aligned}$$

where $\boldsymbol{e}_j$ is the $j$th coordinate vector.

Gupta and Rusmevichientong (2021) makes the above heuristic argument rigorous by dealing with potential points of non-differentiability and precisely quantifying the remainder. Indeed, they prove a slightly stronger theorem which applies when $\boldsymbol{Z}$ is possibly not multivariate Gaussian but is well-approximated by a multivariate Gaussian. For simplicity of exposition, we summarize their result in the Gaussian case only:

**Theorem 5 (Bias of the Stein Correction for Gaussian Estimates)** *Suppose that for each $j = 1, \ldots, n$, we have that $Z_j \sim \mathcal{N}(\mu_j, 1/v_j)$, independently across $j$. Finally, let*

$$B^{\boldsymbol{f}}(\boldsymbol{Z}, h) \equiv \sum_{j=1}^{n} \frac{x_j^{\boldsymbol{f}}(\boldsymbol{Z} + h\boldsymbol{e}_j) - x_j^{\boldsymbol{f}}(\boldsymbol{Z} - h\boldsymbol{e}_j)}{2hv_j}. \tag{13.9}$$

*Then, for any $0 < h < \frac{1}{2}$ and any plug-in policy $\boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z})$, we have that*

$$\left| \mathbb{E}\left[ \boldsymbol{\mu}^\top \boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) \right] - \mathbb{E}\left[ \boldsymbol{Z}^\top \boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) \right] + B^{\boldsymbol{f}}(\boldsymbol{Z}, h) \right| \leq 4h^2 n.$$

Theorem 5 asserts that by choosing $h$ small enough, we can estimate the performance $\boldsymbol{\mu}^\top \boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z})$ of a plug-in policy in an almost unbiased fashion by the bias-corrected quantity $\boldsymbol{Z}^\top \boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) - B^{\boldsymbol{f}}(\boldsymbol{Z})$. At first glance, this analysis suggests choosing $h$ arbitrarily small. As we will see, $h$ controls a bias-variance trade-off for our estimator; small $h$ does induce small bias but comes at the cost of large variance.

Given the central role of Stein's lemma in its derivation, we term $B^{\boldsymbol{f}}(\boldsymbol{Z})$ the *Stein Correction*. Evaluating $B^{\boldsymbol{f}}(\boldsymbol{Z})$ from the data is straightforward but computationally cumbersome, since in principle we must compute $2n$ different plug-in policies corresponding to the $\pm h$ perturbations of the $n$ components. Gupta and Rusmevichientong (2021) and Gupta et al. (2021) each discuss possible refinements that exploit either duality or the sensitivity analysis of the underlying Problem (13.1) to speed up the computation.

Finally, we remark that in the non-Gaussian case, Gupta and Rusmevichientong (2021) generalize the above result so that the error term contains an additional term that does not vanish as $h \to 0$ and depends on the degree to which $\boldsymbol{Z}$ is non-Gaussian.

### 13.3.2 From Unbiasedness to Policy Selection

Theorem 5 suggests the following procedure for identifying a near-best-in-class policy: choose some small $h > 0$, and then select

$$\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) = \boldsymbol{x}^{\boldsymbol{f}_{\mathsf{Stein}}}(\boldsymbol{Z}) \text{ where } \boldsymbol{f}_{\mathsf{Stein}} \in \arg\min_{\boldsymbol{f} \in \mathcal{F}} \boldsymbol{Z}^\top \boldsymbol{x}^{\boldsymbol{f}}(\boldsymbol{Z}) - B^{\boldsymbol{f}}(\boldsymbol{Z}, h). \tag{13.10}$$

Unfortunately, Theorem 5 alone is not enough to ensure that this procedure identifies a near-best-in-class policy, even asymptotically in the small-data, large-scale regime. Namely, since Theorem 5 only treats the bias of our estimator, we need also to establish that certain random quantities concentrate at their expectations.

More specifically, let $f_{\mathsf{Stein}}, f_{\mathsf{OR}} \in \mathcal{F}$ be the functions such that $\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) = \boldsymbol{x}^{f_{\mathsf{Stein}}}(\boldsymbol{Z})$ and $\boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z}) = \boldsymbol{x}^{f_{\mathsf{OR}}}(\boldsymbol{Z})$. Then, write

$$
\begin{aligned}
&\boldsymbol{\mu}^\top \left(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})\right) \\
&\quad = (\boldsymbol{\mu} - \boldsymbol{Z})^\top \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) + B^{f_{\mathsf{Stein}}}(\boldsymbol{Z}, h) \\
&\qquad + \boldsymbol{Z}^\top \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - B^{f_{\mathsf{Stein}}}(\boldsymbol{Z}, h) - \boldsymbol{Z}^\top \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z}) + B^{f_{\mathsf{OR}}}(\boldsymbol{Z}, h) \qquad (13.11) \\
&\qquad + (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z}) - B^{f_{\mathsf{OR}}}(\boldsymbol{Z}, h) \\
&\quad \le (\boldsymbol{\mu} - \boldsymbol{Z})^\top \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) + B^{f_{\mathsf{Stein}}}(\boldsymbol{Z}, h) + (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z}) \\
&\qquad - B^{f_{\mathsf{OR}}}(\boldsymbol{Z}, h),
\end{aligned}
$$

where the inequality follows from the definition of $\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z})$ (cf. Problem (13.10)). Rearranging and upper bounding by the worst case in the policy class shows

$$
\begin{aligned}
\boldsymbol{\mu}^\top \left(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})\right) &\le 2 \sup_{f \in \mathcal{F}} \left| (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) + B^f(\boldsymbol{Z}, h) \right|. \\
&\le 2 \sup_{f \in \mathcal{F}} \left| (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) \right| + 2 \sup_{f \in \mathcal{F}} \left| B^f(\boldsymbol{Z}, h) \right| \\
&\le 2 \sup_{f \in \mathcal{F}} \left| (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) - \mathbb{E}\left[ (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) \right] \right| \\
&\quad + 2 \sup_{f \in \mathcal{F}} \left| B^f(\boldsymbol{Z}, h) - \mathbb{E}\left[ B^f(\boldsymbol{Z}, h) \right] \right| \\
&\quad + 2 \sup_{f \in \mathcal{F}} \left| \mathbb{E}\left[ (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) - B^f(\boldsymbol{Z}, h) \right] \right|.
\end{aligned}
$$

Theorem 5 bounds the last term. Thus,

$$
\begin{aligned}
&\underbrace{\boldsymbol{\mu}^\top \left(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})\right)}_{\text{Sub-Optimality of Our Procedure}} \\
&\quad \le 2 \sup_{f \in \mathcal{F}} \left| (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) - \mathbb{E}\left[ (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^f(\boldsymbol{Z}) \right] \right| \qquad (13.12a) \\
&\qquad + 2 \sup_{f \in \mathcal{F}} \left| B^f(\boldsymbol{Z}, h) - \mathbb{E}\left[ B^f(\boldsymbol{Z}, h) \right] \right| \\
&\qquad + 4h^2 n. \qquad (13.12b)
\end{aligned}
$$

To prove that $\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z})$ has near-best-in-class performance, we must argue that the above two suprema are vanishingly small in the small-data, large-scale regime relative to the oracle performance.

When can we expect these suprema to be vanishingly small? To develop some intuition, we first study a special case in which Problem (13.1) decouples into $n$ separate optimization problems.

**Theorem 6 (Near-Best-In-Class Performance for Decoupled Feasible Regions)** *Consider an instance of Problem* (13.1) *under Eq.* (13.2) *where the feasible region admits a factorization of the form* $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ *for some sets* $X_j \subseteq [0, 1]$ *for* $j = 1, \ldots n$. *Suppose further that* $\boldsymbol{Z}$ *is a multivariate Gaussian with independent components. Finally, consider a plug-in policy class induced by the function class* $\mathcal{F}$ *where* $2 < |\mathcal{F}| < \infty$. *Then, there exists a constant* $C$ *not depending on* $h$, $n$, *or* $\mathcal{F}$ *such that for any* $0 < \epsilon < \frac{1}{2}$,

$$0 \leq \underbrace{\boldsymbol{\mu}^\top \left(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})\right)}_{\text{Sub-Optimality of Our Procedure}} \leq C \log(1/\epsilon)\sqrt{\log|\mathcal{F}|} \cdot \frac{\sqrt{n}}{h} + Cnh^2.$$

*In particular, if we let* $h = O(n^{-1/6})$, *then the sub-optimality of our procedure is* $O_p(n^{2/3})$.

Recall that in most applications, we expect that $\boldsymbol{\mu}^\top \boldsymbol{x}^{\mathsf{OR}}(\boldsymbol{Z})$ itself will scale like $O_p(n)$. Hence, in these applications, the lemma proves that the relative sub-optimality of our procedure is vanishing in the small-data, large-scale limit.

***Proof*** Our approach will be to bound the two suprema in Eq. (13.12). We first write them explicitly

$$Eq.\ (13.12a) = \sup_{f \in \mathcal{F}} \left| \sum_{j=1}^{n} (Z_j - \mu_j) x_j^f(\boldsymbol{Z}) - \mathbb{E}\left[(Z_j - \mu_j) x_j^f(\boldsymbol{Z})\right] \right|,$$

$$Eq.\ (13.12b) = \sup_{f \in \mathcal{F}} \left| \sum_{j=1}^{n} \frac{x_j(\boldsymbol{Z} + h\boldsymbol{e}_j) - x_j(\boldsymbol{Z} - h\boldsymbol{e}_j) - \mathbb{E}\left[x_j(\boldsymbol{Z} + h\boldsymbol{e}_j) - x_j(\boldsymbol{Z} - h\boldsymbol{e}_j)\right]}{2h\nu_j} \right|.$$

The argument of each suprema is the sum of mean-zero random variables. Under our assumption on $\mathcal{X}$, the $j$th component of the solution $x_j(\boldsymbol{Z})$ only depends on $Z_j$ but does not depend on $Z_k$ for $k \neq j$. Thus, the terms of these sums are independent. This observation is crucial. Said differently, both suprema can be interpreted as suprema of an empirical process and hence analyzed with standard techniques (see, e.g., Pollard (1990) for a canonical reference).

To that end, we first bound the supremum in Eq. (13.12b). For a fixed $\boldsymbol{f}$, each term in the sum has magnitude at most $\frac{1}{h\nu_{\mathsf{min}}}$. Hence, each term is sub-Gaussian with variance proxy at most $\frac{1}{h\nu_{\mathsf{min}}}$. Since the terms are independent, the entire sum (for a fixed $\boldsymbol{f}$) is sub-Gaussian with variance proxy at most $\frac{n}{h\nu_{\mathsf{min}}}$. Finally, since the

suprema is over a finite set, we expect the supremum cannot grow too large. Indeed, by Massart's lemma (Wainwright, 2019, Eq. (2.67)), we know that

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|B^f(\boldsymbol{Z},h)-\mathbb{E}\left[B^f(\boldsymbol{Z},h)\right]\right|\right] \leq 2\sqrt{\frac{n\log|\mathcal{F}|}{h\nu_{\min}}}.$$

To prove a stronger, high-probability bound, we invoke the discussion leading up to (Pollard, 1990, Eq. (7.4)). This discussion shows that there exists a constant $C_1$ such that with probability at least $1-\epsilon/2$, this supremum is at most $C_1\log(1/\epsilon)\sqrt{\frac{n}{h}}\cdot\sqrt{\log|\mathcal{F}|}$. (See also Theorem A.1 of Gupta et al., 2021 for clarification.)

We now treat the supremum in Eq. (13.12a). Intuitively, the analysis is similar, but it is more tedious to establish that each term of the sum is sub-Gaussian. Instead, we invoke a generic result from empirical process theory that encapsulates the relevant argument. Specifically, note that $\left|(Z_j-\mu_j)x_j^f(\boldsymbol{Z})\right| \leq \left|Z_j-\mu_j\right|$. Hence, the vector $|\boldsymbol{Z}-\boldsymbol{\mu}|$ with $j$th component $\left|Z_j-\mu_j\right|$ is an envelope for the process. Moreover, by Lemma A.1, Part (iv) of Gupta and Rusmevichientong (2021), the Orlicz norm[2] $\|\,|\boldsymbol{Z}-\boldsymbol{\mu}|\,\|_2\|_\Psi$ is at most $\sqrt{\frac{2n}{\nu_{\min}}}$. Hence, by Theorem A.1 of Gupta et al. (2021), there exists a constant $C_2$ such that with probability at least $1-\epsilon/2$, Eq. (13.12a) is at most $C_2\log(1/\epsilon)\sqrt{n\log|\mathcal{F}|}$.

Combining both bounds and collecting constants proves the theorem. □

Theorem 6 already highlights the aforementioned trade-off with $h$. As we let $h\to 0$, the error due to misestimating the bias vanishes, but the stochastic error stemming from Eq. (13.12b) blows up.

Using fairly standard machinery from empirical process theory, it is straightforward to generalize Theorem 6 to the setting where $|\mathcal{F}|$ is infinite, but $\mathcal{F}$ has finite metric entropy. We refer the interested reader to Pollard (1990). Similarly, our analysis of the two suprema above only required that the components $Z_j$ were sub-Gaussian and independent. Hence, by leveraging the more general form of Theorem 5 in Gupta and Rusmevichientong (2021), one can also easily generalize Theorem 6 to the case where $\boldsymbol{Z}$ is only approximately Gaussian.

Unfortunately, for more interesting optimization problems where $\mathcal{X}$ does not factorize, the proof of Theorem 6 breaks down. The issue is that even for a fixed $\boldsymbol{f}$, the terms of the sums composing the suprema are *not* independent because $x_j^f(\boldsymbol{Z})$ potentially depends on the entire vector $\boldsymbol{Z}$. The nature of this dependence hinges on the structure of $\mathcal{X}$ in Problem (13.1) in a potentially complex way.

[2] See footnote 1 for details on the Orlicz-norm.

Nonetheless, Theorem 6 provides a blueprint for how one might analyze these cases; namely,

1. Use the structure of $\mathcal{X}$ to argue that the terms $x_j^f(\boldsymbol{Z}_j)$ are only "weakly dependent" across $j$. More precisely, we must argue that the sums inside the suprema of Eq. (13.12) each concentrate at a rate $o_p(n)$ for a fixed $\boldsymbol{f} \in \mathcal{F}$.
2. Use empirical process theory to bound each of the suprema with these weakly dependent sums in terms of the "size" of $\mathcal{F}$, i.e., either its cardinality $|\mathcal{F}|$ or its metric entropy.

Although not trivial, this blueprint underlies the more advanced results in Gupta and Rusmevichientong (2021). Indeed, therein the authors consider the special case where $\mathcal{X}$ is polyhedral of the special form $\{\boldsymbol{x} \in [0,1]^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}n\}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. When $m \ll n$, the authors use a duality argument to show that the relevant terms of the sum are not too dependent, and hence the above program goes through as described. For a different debiasing procedure, Gupta et al. (2021) also follows a similar blueprint for problems that suitably decouple after fixing a small number of decision variables or removing a small number of constraints. Summarizing these results is beyond the scope of this chapter.

### *13.3.3 Stein Correction in the Large-Sample Regime*

Interestingly, although we motivated $\boldsymbol{x}^{\text{Stein}}(\boldsymbol{Z})$ by the need for debiasing in the small-data, large-scale regime, this policy has excellent performance in the large-sample regime, as well:

**Theorem 7 (Stein Correction Achieves Full-Information in Large Sample Regime)** *Consider an instance of Problem (13.1) under Eq. (13.2) such that $\mathcal{X} \subseteq [0,1]^n$. Suppose there exists $\boldsymbol{f}_{SAA} \in \mathcal{F}$ such that $\boldsymbol{x}^{f_{SAA}}(\boldsymbol{Z}) = \boldsymbol{x}^{SAA}(\boldsymbol{Z})$. Then,*

$$0 \leq \underbrace{\mathbb{E}\left[\boldsymbol{\mu}^\top(\boldsymbol{x}^{Stein}(\boldsymbol{Z}) - \boldsymbol{x}^*)\right]}_{\text{Expected Sub-Optimality to Full-Info.}} \leq \frac{1}{h\nu_{min}} + \frac{2n}{\sqrt{\nu_{min}}}.$$

The result should be compared to Theorem 1. Indeed, the Stein Correction adds at most $\frac{1}{h\nu_{\min}}$ to the expected error compared to SAA. Moreover, in the large-sample limit, $\nu_{\min} \to 0$, so this term is negligibly small compared to the SAA error. In other words, the Stein Correction enjoys performance comparable to the SAA performance in the large-sample regime.

***Proof*** The first inequality follows from the definition of $\boldsymbol{x}^*$ in Problem (13.1). Let $\boldsymbol{f}_{\text{Stein}} \in \mathcal{F}$ be the optimizer of Problem (13.10).

Then, write

$$\begin{aligned}\boldsymbol{\mu}^\top(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^*) \;=\; &(\boldsymbol{\mu} - \boldsymbol{Z})^\top \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) + \boldsymbol{Z}^\top(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})) \\ &+ \boldsymbol{Z}^\top(\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z}) - \boldsymbol{x}^*) + (\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{x}^*.\end{aligned}$$

By optimality of $\boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})$ in Problem (13.3), the third term above is non-positive. We can use the Cauchy–Schwarz inequality to upper bound the first and last terms by $\|\boldsymbol{Z} - \boldsymbol{\mu}\|_1$ since $\boldsymbol{x}^*, \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) \in \mathcal{X} \subseteq [0,1]^n$. Thus,

$$\begin{aligned}\boldsymbol{\mu}^\top(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^*) \;&\leq\; 2\|\boldsymbol{Z} - \boldsymbol{\mu}\|_1 + \boldsymbol{Z}^\top(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})). \\ &=\; 2\|\boldsymbol{Z} - \boldsymbol{\mu}\|_1 + B^{f_{\mathsf{Stein}}}(\boldsymbol{Z}, h) - B^{f_{\mathsf{SAA}}}(\boldsymbol{Z}, h) \\ &\quad+\; \boldsymbol{Z}^\top \boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - B^{f_{\mathsf{Stein}}}(\boldsymbol{Z}, h) - \boldsymbol{Z}^\top \boldsymbol{x}^{\mathsf{SAA}}(\boldsymbol{Z})) \\ &\quad+ B^{f_{\mathsf{SAA}}}(\boldsymbol{Z}, h).\end{aligned}$$

By the optimality of $f_{\mathsf{Stein}}$ in Problem (13.10), the last line of the last inequality is non-positive. Moreover, $\sup_{f \in \mathcal{F}} \left|B^f(\boldsymbol{Z}, h)\right| \leq \frac{1}{2h\nu_{\min}}$ by construction. Combining shows

$$\boldsymbol{\mu}^\top(\boldsymbol{x}^{\mathsf{Stein}}(\boldsymbol{Z}) - \boldsymbol{x}^*) \;\leq\; 2\|\boldsymbol{Z} - \boldsymbol{\mu}\|_1 + \frac{1}{h\nu_{\min}}.$$

To complete the proof, take expectations of both sides and observe that by Jensen's inequality,

$$\mathbb{E}\left[\|\boldsymbol{Z} - \boldsymbol{\mu}\|_1\right] = \sum_{j=1}^n \mathbb{E}\left[\left|Z_j - \mu_j\right|\right] \leq \sum_{j=1}^n \sqrt{\mathbb{E}\left[(Z_j - \mu_j)^2\right]} = \sum_{j=1}^n \frac{1}{\sqrt{\nu_j}} \leq \frac{n}{\sqrt{\nu_{\min}}}.$$

Substituting above completes the proof. □

Theorem 7 is a heartening result. It shows that it is possible to design algorithms with provably good performance in both large-sample and small-data, large-scale regimes.

### *13.3.4 Open Questions*

The debiasing approach to optimization in the small-data, large-scale regime is still nascent. At time of writing, there are a number of exciting open questions. For what kinds of optimization problems might we expect that the components of the solution $x_j^f(\boldsymbol{Z})$ are only weakly dependent? Is this weak-dependence strictly necessary in order to construct provably good procedures, or is it an artifact of our analysis?

From a computational perspective, how should we efficiently solve Problem (13.10)? In general, this problem is discontinuous and non-convex. If the space of functions $\mathcal{F}$ is fairly complex, simple enumeration may not be feasible. How then should we identify good policies?

More generally, are there better debiasing schemes than the Stein Correction? Gupta et al. (2021) considers the special case of affine plug-in policies and provides an alternate debiasing scheme that explicitly leverages optimization structure via Danskin's theorem. What are the benefits and drawbacks of these various schemes? Might we design even better schemes for particular, specialized optimization problems in inventory or revenue management? What other approaches beyond debiasing exist to attack problems in this new setting?

## 13.4 Conclusion

As the degree of personalization and customization increases in operations management and operations research applications, the ubiquity of the small-data, large-scale regime will only increase. Our goal in this chapter was to highlight some new phenomena that emerge in this regime and to argue that these new phenomena can dramatically affect our intuition about and the performance of data-driven optimization algorithms for these applications. While developing a comprehensive theory for this regime remains outstanding, we hope that our initial steps will further motivate researchers to develop customized algorithms for these new, exciting applications that explicitly leverage these phenomena.

## References

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (vol. 6). Belmont: Athena Scientific.

Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (pp. 1–19). Berlin: Springer.

Elmachtoub, A. N., & Grigas, P. (2021). Smart "predict, then optimize". Management Science. https://doi.org/10.1287/mnsc.2020.3922

Gupta, V., Han, B. R., Kim, S. H., & Paek, H. (2020). Maximizing intervention effectiveness. *Management Science, 66*(12), 5576–5598.

Gupta, V., Huang, M., & Rusmevichientong, P. (2021). Debiasing in-sample policy performance for small-data, large-scale optimization. https://arxiv.org/abs/2107.12438

Gupta, V., & Kallus, N. (2021). Data pooling in stochastic optimization. Management Science. https://doi.org/10.1287/mnsc.2020.3933

Gupta, V., & Rusmevichientong, P. (2021). Small-data, large-scale linear optimization with uncertain objectives. *Management Science, 67*(1), 220–241.

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems, 5*(4), 1–19.

Liu, Y., & Li, Z. (2017). A novel algorithm of low sampling rate GPS trajectories on map-matching. *EURASIP Journal on Wireless Communications and Networking, 2017*(1), 1–5.

Pollard, D. (1990). Empirical processes: Theory and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics, JSTOR* (pp. i–86)

Rivasplata, O. (2012). Subgaussian random variables: An expository note. https://doi.org/10.13140/RG.2.2.36288.23040. http://www.stat.cmu.edu/~arinaldo/36788/subgaussians.pdf

Wainwright, M. J. (2019). High-dimensional statistics: A non-asymptotic viewpoint (vol. 48). Cambridge: Cambridge University Press.

# Part V
# Healthcare Operations

# Chapter 14
# Bandit Procedures for Designing Patient-Centric Clinical Trials


**Sofia S. Villar and Peter Jacko**

## 14.1 Introduction

Multi-armed bandit problems (MABPs) define a special class of an optimal control problem. The MABP is a well-studied and a well-suited framework to model resource allocation under uncertainty in a wide variety of contexts. As Whittle (1980) put it: *The multi-armed bandit problem (as it has become known) is important as one of the simplest non-trivial problems in which one must face the conflict between taking actions which yield immediate reward and taking actions (such as acquiring information, or preparing the ground) whose benefit will come only later. It has proved difficult enough to become a classic, and has now a large literature.*

The MABP has developed over its history as a key example of a problem that has attracted considerable attention from both the Operations Research (OR) and Machine Learning (ML) literature, thus having an exceptional potential to act as a bridge between these two communities. As well, the MABP had its origins in the medical statistics literature, when Thompson (1933) published his work back in the 1930s, and one can easily argue today that its potential to improve health applications is high (Villar et al., 2015; Press, 2009).

S. S. Villar
MRC Biostatistics Unit, University of Cambridge, Cambridge, UK
e-mail: sofia.villar@mrc-bsu.cam.ac.uk

P. Jacko (✉)
Department of Management Science, Lancaster University, Lancaster, UK

Berry Consultants, Abingdon, UK
e-mail: p.jacko@lancaster.ac.uk



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_14

However, despite the great theoretical attention from both OR and ML literature and the considerable potential of its application in practice, both the links between OR and ML as well as the uptake in practice remain relatively under-explored. The potential of the MABP to act as bridge between the OR and ML communities remains low because the perspective in tackling the problem has had a markedly different focus in the two fields. While in OR, formulations based on optimizing expected discounted (or average) rewards are the most common, in ML the dominant goal is that suggested by Robbins where the average regret is minimized. In both cases, the most common horizon considered is infinite, and the focus usually is on asymptotic forms of optimality. Second, the uptake of so-called bandit methods in healthcare practice, and specially in clinical trials, is still virtually non-existing. This may very well be surprising to the reader as across all of this theoretical literature, the use of bandit models to optimally design clinical trials is posited as the typical motivating application. Yet, as it was explored in Villar et al. (2015) and we will further discuss in this chapter, little of the resulting theory has ever been used in the actual design and analysis of clinical trials. The focus on infinite horizon problems for OR and ML is one of the reasons for lack of practical impact but (as we will discuss later) not the only one.

At this point, the reader may also wonder why could a MABP be a perfect fit to optimize the design of clinical trials. The development of a drug or medical therapy follows a regulated and lengthy process which may take between 10 and 15 years (from discovery to being available for patients). Drugs are tested in humans only after laboratory testing, and it is divided into a series of successive clinical trials traditionally known as phase I, II, III, and IV trials. These phases are usually separate clinical studies, and each has a unique objective. Typically, Phase I trials establish safety and tolerability in healthy volunteers, Phase II trials study the drugs' efficacy and adverse effects at different dosages in patients, Phase III trials establish the effectiveness and safety of the drug compared with placebo or standard of care, and Phase IV trials determine general risks and benefits after approval.

A clinical trial is an experiment designed to produce data in order to answer a specific question about a medical intervention (e.g., a drug's superiority versus a standard of care). A typical Phase III clinical trial would compare a single new intervention to a standard of care (which could be simply placebo) with the aim of establishing superiority (or non-inferiority) in terms of a certain efficacy metric. Many Phase II trials compare multiple variants of the same intervention (e.g., drug dosages, treatment durations, or treatment combinations), while some recent Phase II trials include and compare multiple (independent) interventions in one trial. Currently, there is a growing number of trials which might not be easily categorized into these four phases, and even the regulators seem to tend to move away from such strict definitions, and instead, talk about *exploratory* and *confirmatory* trials. Some trials might even answer several questions and/or run across various phases, such as the so-called *seamless Phase II/III trials* or *platform trials*.

Bandit problems formalize the tension between two goals when collecting data to aid decision-making under uncertainty. Those goals are, the desire to *learn* (or *explore*) about the different alternatives (i.e., to learn about the new interventions)

and that to *earn* (or, *exploit*) from that learning to achieve a certain overall objective (i.e., to treat more patients effectively). Therefore, one could argue that in a confirmatory clinical trial, there could be an aim of balancing two separate goals: (1) to correctly identify the best intervention (learning) and (2) to treat patients as effectively as possible during the trial (earning). These two goals may appear to some as naturally complementary, but for those familiar with the MAPB it should be clear that this is not the case. If one is considering the case of a finite population of patients, then correctly identifying the best intervention requires some patients to be allocated to all interventions, and therefore the former acts to limit the possibility of treating more patients with a superior intervention.

As we will describe in this chapter, designing a clinical trial using a MABP solution will entail defining a so-called *response-adaptive* allocation procedure, which (together with specification of other aspects, e.g., statistical analysis methods to be used at the end of the trial) would be part of an adaptive design of a clinical trial. Traditional clinical trials, which have been the dominant design paradigm until the very last decade, follow a linear schematic: design, conduct and analysis of data according to a pre-specified plan. This approach allows for no form of change to the experiment based on the accumulated data. In contrast to this, adaptive designs permit pre-planned changes (or adaptations) to occur after interim looks of the trials data. The key element is that while one can be flexible and adapt based on the observed data, this should be done without undermining its integrity or validity. This latter part and the difficulties it poses for new designs will play a key role explaining the lower uptake of bandit results in practice. The interested reader may read Pallmann et al. (2018); Burnett et al. (2020) for a non-technical introduction to adaptive designs.

While adaptive designs broadly defined have generated a lot of interest in the clinical trials community recently, particularly after the COVID-19 crisis (Stallard et al., 2020), bandit models, methods and algorithms as a class of procedures potentially very useful to deliver adaptive designs for patient-centric trials remain largely unused in practice. Recent work has discussed the reasons for this lower uptake in detail (Villar et al., 2015), discussing what the potential benefits of their use can be as well as the challenges to its application in clinical trial practice. In this chapter, we revisit the ideas presented in the work above and build on them to explain what has changed since and what still calls for further research.

The structure of this chapter is as follows. In the following section, we introduce terminology, assumptions and notation. In this chapter, we shall follow the convention (for simplicity of presentation only) that two-arm clinical trials represent typical Phase III (confirmatory) trials, while multi-armed trials reflect Phase II (exploratory) trials. This is an oversimplification as one could imagine two-armed trials that are exploratory or multi-armed ones that are confirmatory, but it would aid presentation of statistical and design concepts that are relevant in one case more than in another.

## 14.2 The Bayesian Beta-Bernoulli MABP

In this section, we present a Bayesian formulation of a finite-horizon multi-armed problem with binary outcomes as a collection of Markov decision processes (MDPs), which provides a framework for finding the Bayes-optimal allocation procedure by dynamic programming. Our problem of interest has the following defining elements: time, arms (interventions), and each arm is modelled as an MDP with states (information), actions (allocations), transition probabilities and expected one-period rewards (patient outcomes).

**Time**
Patients arrive (i.e., are recruited) sequentially (i.e., one by one) at random moments in continuous time. Since we do not discount the future, we can without loss of generality focus only on the moments of patients' arrivals, which we call discrete time epochs and see as regularly spaced. That is, equivalently, we can consider that patients arrive at time epochs $t \in \mathcal{T} := \{0, 1, 2, \dots, T-1\}$, where $T < +\infty$ is the number of patients in the trial, i.e., the trial size. To clarify, the $(t+1)$-st patient arrives at time epoch $t$. Note that $t = T$ is the time epoch denoting the end of the trial, when the outcome of the last patient is observed and no patient arrives.

**Arms**
We consider arms (or, interventions) labelled by $k \in \mathcal{K} := \{0, 1, \dots, K\}$, where arm $k = 0$ refers to a *control intervention* (typically, a standard of care for the studied disease), and arms $k = 1, \dots, K$ refer to novel (experimental) interventions. A patient must be allocated to exactly one intervention (although this intervention may well be defined as a combination therapy), and such allocation results in a binary type of outcome from that intervention: 0 (failure) or 1 (success). The outcome set is denoted by $O = \{0, 1\}$. In a clinical trial context, the success outcome represents, e.g., response to intervention, remission of tumor, etc. Patient outcomes are uncertain, i.e., modelled as Bernoulli-distributed with parameter $p_k$ (the success probability), independent across arms. Taking the Bayesian approach, the initial prior for the success probability of arm $k$ is Beta distribution with parameters $(\widetilde{s}_k(0), \widetilde{f}_k(0))$, which can be interpreted as the number of pseudo-successes and pseudo-failures observed before making the first allocation in the experiment. The rewards are immediate, meaning that the outcome of an allocated patient is observed before the next decision needs to be made.

**States**
The *state space* for arm $k$, $\mathcal{X}_k := \{\boldsymbol{x}_k := (s_k, f_k) \in \subsetneq \mathcal{T} \cup \{T\})^2 : s_k + f_k \leq T\}$, represents all the possible two-dimensional vectors of available information on the unknown parameter $p_k$ at any time during the trial. Note that we exclude the prior information (i.e., pseudo-observations) from the state definition because it does not change over time and because in this way the model is as small as possible, which is beneficial from the computational point of view. However, to simplify some expressions, we also define the pseudo-state $\widetilde{\boldsymbol{x}}_k := (\widetilde{s}_k, \widetilde{f}_k)$ with $\widetilde{s}_k := \widetilde{s}_k(0) + s_k$, $\widetilde{f}_k := \widetilde{f}_k(0) + f_k$.

**Actions**
The *action set* $\mathcal{A}_k$ for arm $k$ is a binary set representing the action of drawing a sample observation from arm $k$ ($a_k = 1$) or not ($a_k = 0$). In a clinical context, the action variable stands for the choice of allocating next patient to arm $k$ or not.

**Transition Probabilities**
The Markovian *transition law* $\mathcal{P}_k(\boldsymbol{x}'_k|\boldsymbol{x}_k, a_k)$ describing the evolution of the information state variable on arm $k$ in state $\boldsymbol{x}_k$ under action $a_k$ from one time epoch to the next is given by

$$\boldsymbol{x}'_k = \begin{cases} (s_k + 1, f_k), & \text{if } a_k = 1 \text{ w.p. } \frac{\widetilde{s}_k}{\widetilde{s}_k + \widetilde{f}_k}, \\ (s_k, f_k + 1), & \text{if } a_k = 1 \text{ w.p. } \frac{\widetilde{f}_k}{\widetilde{s}_k + \widetilde{f}_k}, \\ \boldsymbol{x}_k, & \text{if } a_k = 0 \text{ w.p. } 1, \end{cases} \tag{14.1}$$

where w.p. stands for "with probability". Note that under action 1, the transition probabilities are defined by the mean of the current posterior distribution, which, due to conjugacy, is a Beta distribution with parameters $\left(\widetilde{s}_k, \widetilde{f}_k\right)$.

**Expected One-Period Reward**
The expected reward on arm $k$ in state $\boldsymbol{x}_k$ under action $a_k$ is

$$\mathcal{R}^{a_k}_{k,\boldsymbol{x}_k} = \frac{\widetilde{s}_k}{\widetilde{s}_k + \widetilde{f}_k} a_k, \tag{14.2}$$

where in accordance with the above specified dynamics, expected reward is the Bayes-expected number of successes from the current patient, computed using the current posterior Beta distribution.

Note that both the transition law and the rewards depend on the prior distributions, although we do not indicate it in the notation. The system dynamics is captured by the joint state process $(\boldsymbol{x}_k(t))_{k\in\mathcal{K}}$ for all $t \in \mathcal{T} \cup \{T\}$ and by the joint action process $(a_k(t))_{k\in\mathcal{K}}$ for all $t \in \mathcal{T}$. The actions are restricted by the fact that every patient in the trial must be allocated to one and only one arm, i.e., $\sum_{k\in\mathcal{K}} a_k(t) = 1$ for all $t \in \mathcal{T}$. This restriction implies a restriction on the joint state process so that $\sum_{k\in\mathcal{K}}(s_k(t) + f_k(t)) = t$ for all $t \in \mathcal{T} \cup \{T\}$.

A rule is required to operate the resulting (sometimes called *weakly coupled*) MDP, which indicates which action to take for each arm $k \in \mathcal{K}$ for every possible combination of states of the arms at every time $t \in \mathcal{T}$. Such a rule forms a sequence of actions resulting in a joint action process $(a_k(t))_{k\in\mathcal{K}}$ and it is known as a *policy*, denoted by $\pi \in \Pi$, where $\Pi$ is the set of all the feasible policies satisfying the above action constraint.

To complete the specification of the multi-armed bandit model as an *optimal control model*, the problem's *objective function* must be selected. The typical performance objective in the Bayesian Beta-Bernoulli MABP in a trial with $T$ patients is to maximize the *Bayes-expected number of successes*. For a feasible

policy $\pi \in \Pi$, the Bayes-expected number of successes is, i.e., the total value function conditional on the initial joint prior parameters $\widetilde{\boldsymbol{x}}(0)$,

$$\mathrm{ENS}^{\pi}_{\widetilde{\boldsymbol{x}}(0)} = E^{\pi}_{\widetilde{\boldsymbol{x}}(0)}\left[\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} \mathcal{R}^{a_k(t)}_{k,\boldsymbol{x}_k(t)}\right] = E^{\pi}_{\widetilde{\boldsymbol{x}}(0)}\left[\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} \frac{\widetilde{s}_k(t)}{\widetilde{s}_k(t)+\widetilde{f}_k(t)} a_k(t)\right], \tag{14.3}$$

where $E^{\pi}_{\widetilde{\boldsymbol{x}}(0)}[\cdot]$ denotes Bayesian expectation with joint Beta prior parameters $\widetilde{\boldsymbol{x}}(0) := (\widetilde{\boldsymbol{x}}_k(0))_{k\in\mathcal{K}}$ under policy $\pi$. The multi-armed bandit optimal control problem is mathematically summarized as the problem of finding an optimal policy $\pi^*$, i.e., a feasible policy ($\pi^* \in \Pi$) that optimizes the performance objective. Formally, the optimal policy is

$$\pi^* = \operatorname*{argmax}_{\pi\in\Pi} \mathrm{ENS}^{\pi}_{\widetilde{\boldsymbol{x}}(0)}, \tag{14.4}$$

and the optimal Bayes-expected number of successes is

$$\mathrm{ENS}^{*}_{\widetilde{\boldsymbol{x}}(0)} = \max_{\pi\in\Pi} \mathrm{ENS}^{\pi}_{\widetilde{\boldsymbol{x}}(0)}. \tag{14.5}$$

Note that the right-hand side of (14.4) suggests that $\pi^*$ should depend on the prior $\widetilde{\boldsymbol{x}}(0)$, but the MDP theory implies that there is an optimal policy which is stationary (i.e., it prescribes joint action $(a_k(t))_{k\in\mathcal{K}}$ only as a function of the posterior joint state $\widetilde{\boldsymbol{x}}(t) := \left(\widetilde{s}_k(t), \widetilde{f}_k(t)\right)_{k\in\mathcal{K}}$ without a direct dependence on $t$), and thus we assume $\pi^*$ is such and drop its dependence on the prior parameters.

The optimal policy $\pi^*$ is, nevertheless, in general different for different trial sizes $T$, because larger $T$ tends, for a given state, to lead to an allocation that provides a larger amount of learning about the arms' unknown success probability parameters in order to increase the expected number of successes from the remaining patients.

### *14.2.1 Discussion of the Model*

The above model is probably the simplest model for the multi-armed bandit problem cast as an optimization problem. Analogous modelling approach can in theory be employed for other distribution of outcomes (discrete, continuous, etc.), although the state would need to be redefined as an appropriate sufficient statistic, and the transition law and reward would need to be adjusted correspondingly (see, e.g., Williamson and Villar, 2020). However, in practice, these often quickly become computationally unfeasible to be solved by dynamic programming, and approximate approaches need to be employed.

The action set can be generalized by making the actions randomized and/or by specifying an action to take when the original two actions are equivalent. In some states, one could modify the action set to either have a single action (for instance, allowing only allocation to a pre-specified arm or allowing only equal fixed

randomization in the initial stage of the trial) or have more actions (for instance, allowing for stopping of the trial if the treatment difference seems to be large).

The model can be extended to include discounting of the future patients' outcomes and/or to be optimized over an infinite horizon using standard approaches from the theory of Markov decision processes, but we believe that the undiscounted finite-horizon formulation is the most relevant for healthcare applications.

The rewards can be generalized, for instance, by including penalties in some undesirable states in order to improve a particular statistical operating characteristic, such as in those that would lead to an extremely unbalanced allocation in order to improve power and estimation (as in Williamson et al., 2017, 2021) or by using any other utility function of interest.

The above-defined model thus requires only the horizon $T$ and the prior parameters to be set by the trial designer. The standard choice in the bandit literature is to set the horizon equal to the size of the trial, but in clinical trials it may sometimes be more reasonable to optimize over the size of the patient population, assuming that one of the arms is chosen at the end of the trial and is applied to the after-trial patients. The standard choice for the prior parameters is the so-called Bayes' prior $\left(\widetilde{s}_k(0), \widetilde{f}_k(0)\right) = (1, 1)$, which is considered non-informative, although other priors with mean 0.5 are also considered uninformative, e.g., Jeffrey's prior $(0.5, 0.5)$ or Haldane's prior $(0, 0)$. Note that Haldane's prior essentially reduces the optimization problem to a frequentist objective, where the posterior mean equals the sample mean, which is the maximum likelihood estimator of the mean, as shown in Bowden and Trippa (2017).

## 14.3 Metrics for Two-Armed Problem (Confirmatory Trials)

The two-armed bandit problem with binary outcomes is probably the most studied version of all the bandit problems, intriguing researchers from several disciplines for almost a century (for a review, see, e.g., Jacko, 2019b). At the same time, clinical trials with two arms are probably the most common setup of clinical trials in practice, especially used for *confirmatory trials* which are typically defined with an objective of generating convincing evidence of efficacy (and safety) in order to seek regulatory approval. These are traditionally referred to as the *randomized controlled trials*, where "controlled" indicates that a novel intervention is being concurrently compared to another one (typically, the current standard of care), i.e., there are at least two arms, in order to control for seasonality effects, time trends, population changes and other shocks, and "randomized" indicates that patients are allocated to interventions using a procedure which ensures that patients and their doctors are not able to predict with certainty which intervention will be allocated next, in order to help avoiding the selection bias and other types of biases (see, e.g. Rosenberger and Lachin, 2015, for a discussion of importance of randomization). Throughout this section, we assume $K = 1$, having a control arm $k = 0$ and an experimental arm $k = 1$.

Traditionally, the *randomization ratio* is taken as 1:1, called the *equal fixed randomization* (EFR). This is done without any rigorous justification, often relying on a widespread myth that the 1:1 ratio maximizes statistical power, which is however true only under the assumption of equal variances of the efficacy of the two arms. That might be a somewhat reasonable assumption in some cases of continuous outcomes modelled using the normal distribution, but it is not appropriate for binary outcomes as the variance of the Bernoulli distribution is dependent on its mean (Robertson et al., 2021) and also for other types of outcomes such as time-to-event (Sverdlov et al., 2011). Clinical trials might also be too small to invoke the recommended conditions for approximation of binomial samples by the normal distribution. Understanding of that and consideration of patient outcome (e.g., for deadly diseases that have no current treatment) lead some clinical trial designers to implement other fixed ratios in an ad hoc manner, e.g., 2:1, typically allocating higher probability to the novel intervention. Note that the 1:1 randomization ratio is often interpreted in the academic literature as that every patient's allocation is randomized with probability of 0.5 to either arm, but the ratio is in practice implemented essentially as a permutation of allocations within blocks of patients, e.g., in every block of 60 patients, there are 30 patients allocated to each arm, i.e., in practice it is a *per-block allocation ratio* rather than a *per-patient randomization probability*.

Several stakeholders are involved in confirmatory trials, and thus several metrics are of interest: the regulatory agencies would constrain the *Type I error* (typically at the one-sided level of 0.025), intervention sponsors would require high *statistical power* (typically at the level of 0.8 or 0.9) and small *trial size* (or, more generally, a good balance of expected trial costs and expected post-approval revenues), patient organizations would require high *patient benefit* (i.e., health benefit for in-trial patients) and health economics agencies and clinicians would require *accurate and precise estimation* of the interventions efficacy (or of their difference).

### *14.3.1 Accurate and Precise Estimation*

Unequal fixed (i.e., not adaptive to observed successes and failures on each arm) randomization is well understood in the biostatistics literature, but the researchers in other disciplines and practitioners seem to be largely unaware of those results. For the two-armed setting, there are closed formulae that give ratios that are optimal for different objectives. Any fixed procedure that allocates at least one patient to each intervention provides basis for an unbiased estimation of the efficacy of each arm using the *maximum likelihood estimator* (MLE), which equals the mean of observed successes, and for statistical testing (Rosenberger et al., 2019).

While perfect accuracy (i.e., unbiased estimation) can be achieved by fixed randomization, using the MLE after an adaptive procedure always leads to a bias

(Bowden and Trippa, 2017), which is typically negative but can also be positive (Nie et al., 2018). In that case, improving accuracy by using other estimators that are unbiased can be done at a cost of decreased precision (e.g., the mean-squared error). To the best of our knowledge, maximization of the precision using adaptive procedures is not well understood, but there are some promising recent research lines (Hadad et al., 2021), although existing estimation methods typically do not apply to deterministic allocation procedures. Note also that it is linked to the maximization of statistical power. In practice, the block randomization is sometimes implemented in a stratified way and/or using the so-called *minimization algorithms* that balance the covariates in order to increase the precision of estimators.

### *14.3.2 Statistical Errors*

Statistical hypothesis testing is usually required by the regulators at the end of a confirmatory trial in order to apply for a marketing approval. This is usually done in a frequentist approach (but Bayesian approaches are also sometimes allowed after a discussion with the regulator). For one-sided test comparing two proportions, we specify the null hypothesis and the alternative hypothesis as follows:

$$H_0 : p_1 \leq p_0 \tag{14.6}$$

$$H_1 : p_1 > p_0 \tag{14.7}$$

One-sided testing is more appropriate than two-sided testing whenever the regulator is interested in limiting the probability of approving the novel intervention (arm 1) despite being worse than or equal to the control intervention (arm 0), which is called the *Type I error*, formally defined as the probability of rejecting the null hypothesis if it is true. On the other hand, the sponsor of the novel intervention is interested in achieving a high probability of getting the novel intervention approved if it is indeed better than the control intervention, which is called the *statistical power*, formally defined as the probability of rejecting the null hypothesis if it is false.

A variety of tests have been proposed for a comparison of proportions of two binomial distributions, including z-tests (unpooled or pooled; with or without continuity correction), Fisher's exact test (and its modifications such as Boschloo's test) or simulation-based randomization tests. However, there is no consensus on which test is the most appropriate because they all have certain disadvantages. The z-tests are based on approximation of binomial distribution by normal distribution and therefore are suitable for large samples; typically, it is suggested that there should be a minimum number of both successes and failures on each arm (5 or 10). Fisher's exact test is considered too conservative, yielding the Type I error sometimes notably below the given significance level. Other tests, including randomization tests, become computationally intractable for large samples.

For a given Type I error, the ratio that maximizes the statistical power if using the (unpooled) z-test is Neyman's allocation ratio $\sqrt{\theta_C(1-\theta_C)} : \sqrt{\theta_D(1-\theta_D)}$ (Melfi and Page, 1998), which is the ratio of standard deviations of Bernoulli distributions with means $\theta_C$ and $\theta_D$ (we remark a connection with optimal designs of ranking and selection problems presented in Ryzhov, 2021, equation (4)). We can see that Neyman's allocation coincides with 1:1 when the efficacies of the two interventions are either equal (i.e., $\theta_C = \theta_D$) or equally distant from 0.5 (i.e., $\theta_C = 1-\theta_D$). The monotonicity properties of the standard deviation formula imply that the intervention whose efficacy is closer to 0.5 is allocated more patients. So, the inferior intervention, which might be considered undesirable from the patient-benefit perspective, is allocated more patients if and only if $\theta_C > 1-\theta_D$. For instance, if $\theta_C = 0.5$ and $\theta_D = 0.2$ (or 0.8), the ratio that maximizes the statistical power is 5:4, while $\theta_D = 0.1$ (or 0.9) gives the ratio 5:3; ratio 2:1 is optimal for instance if $\theta_C = 0.5$ and $\theta_D \approx 0.067$ (or 0.933) or if $\theta_C = 0.2$ (or 0.8) and $\theta_D \approx 0.042$ (or 0.958). However, as Neyman's allocation ratio depends on the efficacies of the two arms, which are unknown, it needs to be implemented adaptively in a "learning by doing" fashion, typically by adaptively estimating the efficacies using the accumulating observations (Rosenberger et al., 2001).

### *14.3.3 Patient Benefit*

In order to measure the benefit for patients in the trial, we define the *expected number of successes* under procedure $\boldsymbol{\pi}$ if the probabilities of success are $\boldsymbol{p}$,

$$\mathrm{ENS}_{\boldsymbol{p}}^{\boldsymbol{\pi}} = E_{\boldsymbol{p}}^{\boldsymbol{\pi}}\left[\sum_{t=0}^{T-1}\sum_{k=0}^{K} p_k a_k(t)\right], \tag{14.8}$$

where $E_{\boldsymbol{p}}^{\boldsymbol{\pi}}[\cdot]$ denotes expectation under procedure $\boldsymbol{\pi} \in \Pi$ prescribing the vector $\boldsymbol{a}(t)$ of allocation processes $a_k(t) \in \{0, 1\}$ if the probabilities of success are $\boldsymbol{p}$. (Note the slight abuse of notation, with (14.3) being a Bayesian expectation depending on the prior parameters, while (14.8) being a frequentist expectation depending on the true success probabilities.) An alternative measure of patient benefit is the *expected proportion of allocations on the superior arm* under procedure $\boldsymbol{\pi}$ if the probabilities of success are $\boldsymbol{p}$,

$$\mathrm{EPASA}_{\boldsymbol{p}}^{\boldsymbol{\pi}} = \frac{1}{T} E_{\boldsymbol{p}}^{\boldsymbol{\pi}}\left[\sum_{t=0}^{T-1} a_{k^*}(t)\right], \tag{14.9}$$

where $k^* := \min \operatorname{argmax}_{k\in\{0,\ldots,K\}} p_k$ is the lexicographically first of all the superior arms in the trial. The means of EPASA and ENS are linear transformations (so, produce an equivalent performance ordering of procedures) in the case of two arms,

but their variability is not so easily linked (and they are not equivalent in the case of more than two arms because EPASA does not capture how the allocations are split among the non-superior arms).

Kelly (1981) derived an allocation procedure which is optimal to be used at the beginning of a trial (assuming an infinite trial size) with the objective of providing the maximum Bayes-expected patient benefit. It is known as the *least failures first* (LFF) rule, and it sequentially allocates patients to the intervention with fewer observed failures, breaking the ties in favour of the intervention with more observed successes (breaking the double ties arbitrarily). It is easy to see that this procedure continues allocating patients to the same intervention as long as observing successes and it switches to the other intervention after the first or after the second observed failure since the last switch. See also Jacko (2019b) for a discussion of its similarity to the "stay-with-a-winner&switch-on-a-loser" rule known in the biostatistics literature as the "Play-the-winner" rule (Zelen, 1969). This procedure in the long term converges to the ratio $(1-\theta_D) : (1-\theta_C)$, which is the same asymptotic ratio as of the "Play-the-winner" rule (Zelen, 1969) and of a specific configuration of the "Randomized play-the-winner" with its parameter $\alpha = 0$ (Wei and Durham, 1978; Rosenberger, 1999). For instance, if $\theta_C = 0.5$ and $\theta_D = 0.2$ (or 0.8), the ratio is 8:5 (or 2:5), while $\theta_D = 0.1$ (or 0.9) gives the ratio 9:5 (or 1:5); ratio 2:1 is optimal for instance if $\theta_C = 0.5$ and $\theta_D = 0.0$ or if $\theta_C = 0.8$ and $\theta_D = 0.6$.

For a finite trial size, the maximum Bayes-expected patient benefit can be obtained only computationally, using dynamic programming (DP) methods such as the exact (optimal) method of backward recursion or approximate (near-optimal) methods such as the Whittle index rule and the Gittins index rule. All these methods result in allocation procedures which are not only adaptive (to observed successes and failures on each arm) but also non-myopic meaning that they depend on the trial size $T$. The backward recursion and the Whittle index rule have this dependence direct by defining the (remaining) time horizon of the optimization problem at every moment by the (remaining) trial size. The Gittins index rule has this dependence only indirectly by choosing the discount factor which should reflect the trial size. Jacko (2019b); Pilarski et al. (2021) illustrated that efficient coding in performance-oriented programming languages (such as Julia and C++) allows for using these computational methods for offline calculation of the allocation procedures (stored in lookup tables) for trials sizes of up to several thousand on standard computers. The backward recursion method is only practical when the number of arms is small, but the sub-optimality of some index rules is practically negligible (see Sect. 14.4).

Other allocation ratios that are patient-benefit optimal given a constraint on the variance of a function comparing the two interventions were developed in Rosenberger et al. (2001).

### 14.3.4 Trial Size

While all the above approaches try to optimize a metric for a given trial size, a very common approach in practice is actually to minimize the trial size given (some of) the above metrics as constraints. This is because shorter trials are cheaper (recent studies report a cost of over $100,000 per in-trial patient for some diseases) and, if approved, lead to a longer patent-protected marketing period.

### 14.3.5 Multiple Metrics

Besides the single-metric optimization, typically subject to a single constraint, researchers have developed procedures that come close to optimizing several metrics. These are usually tunable procedures, in which some parameters can be set to (directly or indirectly) give higher or lower weight to a particular metric. We will discuss two such families of procedures: the tunable Upper Confidence Bound ($\alpha$UCB) procedures and the Constrained Randomized Dynamic Programming (CRDP) procedures.

Following Bubeck and Cesa-Bianchi (2012, Section 2), we consider the popular $\alpha$UCB procedure which allocates each arm once in the initial two periods and then deterministically allocates every patient to the arm with currently the largest index (breaking ties randomly) of the form

$$\frac{s_k(t)}{s_k(t)+f_k(t)}+\sqrt{\frac{\alpha\cdot\ln(t+1)}{s_k(t)+f_k(t)}} \tag{14.10}$$

where $\alpha \geq 0$. The original procedure introduced in Auer et al. (2002) used $\alpha = 2$. Theoretical upper bounds currently exist for $\alpha > 1$, but researchers have noticed empirically that lower values of $\alpha$ typically lead to better performance and some used $\alpha = 1$, see, e.g., Cserna et al. (2017). Numerical experiments of finite trials have revealed that approximately the best patient benefit is robustly achieved with $\alpha = 0.18$ (Jacko, 2019b) or $\alpha = 0.19$ (Pilarski et al., 2021). Note that setting $\alpha = 0$ recovers the (frequentist) myopic procedure which at every period selects the arm with highest sample mean.

Williamson et al. (2017) proposed an extension of the DP procedure called CRDP in which (i) the original identity between selected actions and arm allocations is disrupted by a random perturbation (i.e., adding randomization) and (ii) it is allowed to introduce penalties in undesirable end-of-trial states (i.e., adding constraining). They proposed that a good trade-off between patient benefit and statistical properties may be achieved by setting the randomization parameter to 0.9 and by penalizing the states with less than $0.15T$ observations on either arm. Note that DP is recovered by setting the randomization parameter to 1.0, while EFR is recovered by setting it to 0.5 (and not penalizing any states).

## 14.4 Illustrative Results for Two-Armed Problem

We re-examine the experimental setting presented in Villar et al. (2015, Section 5.1), and we reprint results from the original Villar et al. (2015, Table 5) in Table 14.1 (adapting the notation and terminology to this chapter) for easy reference. The table shows the results for a variety of two-arm procedures under both the null and alternative hypotheses. The size of the trial was set to be $T = 148$ to ensure that a traditional balanced design with EFR attains at least 80% power when rejecting $H_0$ with a (maximum) one-sided 5% Type-I error rate using the z-test.

In Table 14.2, we re-evaluate the measures of the trials designed using some of these procedures. The table presents results in the same scenarios as originally presented in Table 14.1, but it makes several extensions, improvements and corrections. First, we complement the results presented originally by including additional procedures and provide a more robust picture due to the employment of several statistical tests and confidence levels. While the original table was obtained by simulations, the results presented here are for all the procedures obtained by *exact calculations* using the backward recursion (and are thus accurate up to a computer's numerical precision), as proposed in Jacko (2019b). A few measures are also calculated slightly differently. EPASA originally included the prior (i.e., 2 pseudo-allocations on each arm), so it slightly underestimated the value of EPASA reported here, which is based on observed (or realized) allocations only. Hypothesis testing is performed using both a z-test and Fisher's exact test for comparing two binomial distributions. The z-test was originally based on uncorrected variances (in order to allow for calculation of the variance even for arms with a single observation), while here we use it with corrected variances (using Bessel's correction to obtain an unbiased variance estimator); we moreover require that each arm has at least one observed success and at least one observed failure at the end of the trial in

**Table 14.1** Comparison of procedures in a two-arm trial of size $T = 148$ by simulation. 1.645: the critical value used in z-test (two-sided; confidence level approximately 0.9); $F_a$: Fisher's adjusted test (two-sided). TS: Thompson sampling; RBI: randomized belief index; RGI: randomized Gittins index; CB: current belief; WI: Whittle index; GI: Gittins index (with discount factor 0.99. Reprinted (adapted) from Villar et al. (2015, Table 5)

| | Crit. | $H_0 : p_0 = p_1 = 0.3$ | | | $H_1 : p_0 = 0.3\,,\ p_1 = 0.5$ | | |
|---|---|---|---|---|---|---|---|
| | value | Type I E | EPASA (SD) | ENS (SD) | Power | EPASA (SD) | ENS (SD) |
| EFR | 1.645 | 0.052 | 0.500 (0.04) | 44.34 (5.62) | 0.809 | 0.501 (0.04) | 59.17 (6.03) |
| TS | 1.645 | 0.066 | 0.499 (0.10) | 44.39 (5.58) | 0.795 | 0.685 (0.09) | 64.85 (6.62) |
| 2UCB | 1.645 | 0.062 | 0.499 (0.10) | 44.30 (5.60) | 0.799 | 0.721 (0.07) | 66.03 (6.57) |
| RBI | 1.645 | 0.067 | 0.502 (0.14) | 44.40 (5.57) | 0.763 | 0.737 (0.07) | 66.43 (6.54) |
| RGI | 1.645 | 0.063 | 0.500 (0.11) | 44.40 (5.61) | 0.785 | 0.705 (0.07) | 65.46 (6.40) |
| CB | $F_a$ | 0.046 | 0.528 (0.44) | 44.34 (5.55) | 0.228 | 0.782 (0.35) | 67.75 (12.0) |
| WI | $F_a$ | 0.048 | 0.499 (0.35) | 44.37 (5.59) | 0.282 | 0.878 (0.18) | 70.73 (8.16) |
| GI | $F_a$ | 0.053 | 0.501 (0.26) | 44.41 (5.58) | 0.364 | 0.862 (0.11) | 70.21 (7.11) |

**Table 14.2** Comparison of different two-arm procedures for a trial of size $T = 148$ by exact calculation; all values are rounded to three digits. The first two columns report the Type I error under the null hypothesis and power under the alternative hypothesis, respectively, of one-sided tests. F-test: Fisher's exact test; {0.91, 0.95, 0.98}: one-sided confidence levels; SD: uncorrected standard deviation. Note that ENS (SD) under the null hypothesis is 44.400 (5.575) for all procedures

| | $H_0 : p_0 = p_1 = 0.3$ | | | | | $H_1 : p_0 = 0.3\,,\ p_1 = 0.5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | z-test | | F-test | | | z-test | | F-test | | | |
| | 0.95 | 0.98 | 0.91 | 0.95 | EPASA (SD) | 0.95 | 0.98 | 0.91 | 0.95 | EPASA (SD) | ENS (SD) |
| EFR | 0.051 | 0.021 | 0.058 | 0.024 | 0.500 (0.041) | 0.805 | 0.676 | 0.755 | 0.589 | 0.500 (0.041) | 59.200 (5.960) |
| LFF | 0.054 | 0.023 | 0.057 | 0.024 | 0.500 (0.029) | 0.804 | 0.672 | 0.746 | 0.567 | 0.586 (0.033) | 61.735 (6.199) |
| 2UCB | 0.063 | 0.031 | 0.068 | 0.033 | 0.500 (0.101) | 0.786 | 0.637 | 0.707 | 0.497 | 0.727 (0.077) | 65.915 (6.543) |
| 1UCB | 0.073 | 0.038 | 0.079 | 0.040 | 0.500 (0.142) | 0.751 | 0.581 | 0.652 | 0.432 | 0.785 (0.090) | 67.638 (6.724) |
| 0.5UCB | 0.089 | 0.049 | 0.095 | 0.050 | 0.500 (0.199) | 0.650 | 0.442 | 0.547 | 0.308 | 0.838 (0.103) | 69.219 (6.894) |
| 0.25UCB | 0.097 | 0.051 | 0.105 | 0.051 | 0.500 (0.271) | 0.462 | 0.243 | 0.379 | 0.173 | 0.872 (0.134) | 70.221 (7.299) |
| 0.18UCB | 0.091 | 0.047 | 0.101 | 0.047 | 0.500 (0.308) | 0.356 | 0.158 | 0.308 | 0.104 | 0.877 (0.163) | 70.356 (7.740) |
| 0UCB | 0.001 | 0.000 | 0.001 | 0.000 | 0.500 (0.483) | 0.012 | 0.007 | 0.011 | 0.004 | 0.692 (0.445) | 64.883 (14.51) |
| 37C+0.8RDP | 0.063 | 0.030 | 0.068 | 0.031 | 0.500 (0.181) | 0.746 | 0.600 | 0.663 | 0.478 | 0.714 (0.060) | 65.527 (6.240) |
| 22C+0.9RDP | 0.077 | 0.040 | 0.085 | 0.040 | 0.500 (0.259) | 0.650 | 0.492 | 0.565 | 0.371 | 0.801 (0.097) | 68.116 (6.696) |
| 15C+0.95RDP | 0.091 | 0.048 | 0.101 | 0.049 | 0.500 (0.298) | 0.580 | 0.412 | 0.504 | 0.314 | 0.840 (0.118) | 69.270 (7.021) |
| 0.95RDP | 0.090 | 0.047 | 0.104 | 0.048 | 0.500 (0.313) | 0.511 | 0.346 | 0.454 | 0.264 | 0.856 (0.144) | 69.726 (7.455) |
| 0.99RDP | 0.077 | 0.031 | 0.097 | 0.034 | 0.500 (0.344) | 0.323 | 0.170 | 0.308 | 0.123 | 0.882 (0.166) | 70.504 (7.849) |
| 37C+DP | 0.063 | 0.030 | 0.068 | 0.031 | 0.500 (0.209) | 0.715 | 0.575 | 0.634 | 0.461 | 0.734 (0.050) | 66.128 (6.159) |
| 30C+DP | 0.068 | 0.032 | 0.073 | 0.036 | 0.500 (0.244) | 0.675 | 0.523 | 0.586 | 0.407 | 0.776 (0.066) | 67.371 (6.320) |
| 22C+DP | 0.076 | 0.040 | 0.086 | 0.039 | 0.500 (0.282) | 0.604 | 0.453 | 0.522 | 0.344 | 0.820 (0.089) | 68.682 (6.600) |
| 15C+DP | 0.092 | 0.047 | 0.105 | 0.047 | 0.500 (0.313) | 0.536 | 0.376 | 0.467 | 0.288 | 0.854 (0.114) | 69.666 (6.962) |
| 7C+DP | 0.089 | 0.029 | 0.116 | 0.032 | 0.500 (0.343) | 0.411 | 0.250 | 0.369 | 0.219 | 0.880 (0.151) | 70.441 (7.590) |
| DP | 0.073 | 0.026 | 0.094 | 0.028 | 0.500 (0.352) | 0.263 | 0.116 | 0.262 | 0.078 | 0.888 (0.172) | 70.696 (7.964) |
| WI | 0.065 | 0.022 | 0.090 | 0.024 | 0.500 (0.363) | 0.233 | 0.102 | 0.240 | 0.069 | 0.887 (0.184) | 70.667 (8.185) |
| ORACLE | 0.000 | 0.000 | 0.000 | 0.000 | 0.500 (0.500) | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 (0.000) | 74.000 (6.083) |

order for the z-test to be employed, otherwise the null hypothesis is not rejected; and we use the exact critical value instead of the rounded 1.645. The Fisher test was originally two-sided, while here we report a one-sided variant, which might not be fully equivalent due to the asymmetry of this test; moreover, the one-sided significance level was originally adjusted (increased) to achieve the one-sided Type I error of around 0.05, while here we present results for significance levels of 0.05 and 0.09 (i.e., confidence levels of 0.95 and 0.91). Both the original and our table report standard deviation, even though the original table in Villar et al. (2015) referred to it as "s.e.".

As discussed in Villar et al. (2015), if one compares a traditional EFR procedure to response-adaptive procedures (including bandit procedures) in the two-armed setting, the first realization is that power is always higher in EFR, but its patient benefit metrics are always lower. Adaptive procedures have their power reduced because they induce correlation among intervention allocations; for the deterministic policies like the DP and UCB, this effect is the most severe because they almost permanently skew intervention allocation towards an intervention as soon as one exhibits a certain advantage over the other arms. This table shows the tension between learning (high power) and earning (high EPASA and ENS) and how different procedures settle for a different balance between these two objectives.

Both tables show that even EFR leads to an inflated Type I error using the z-test because of not having at least a certain number of both successes and failures on each arm in order for the normal distribution to be an acceptable approximation of the binomial distribution. Academic literature typically recommends that number to be 5 or 10. In this scenario, we would need to require to have at least 11 successes and 11 failures on each arm in order to obtain a Type I error below the significance level of 0.05 (giving Type I error 0.0497 and power 0.8033). Looking at Table 14.2, LFF also leads to a slightly inflated Type I error under the z-test, but the power is almost the same as that of EFR, while bringing a notable patient benefit of 2.535 additional expected successes. Under the F-test, the Type I errors of these two procedures are practically identical and notably below the significance level, while the power of LFF is slightly lower.

Table 14.2 also includes ORACLE, which is the procedure that assumes that the success probabilities are known, so it allocates all the patients to the superior arm; in case of a tie (i.e., under the null hypothesis), it randomly picks one of the arms at the beginning of the trial and sticks to it. Under the alternative hypothesis, this procedure provides an upper bound for EPASA and ENS and a benchmark for SD of ENS (which is almost the same as that of EFR). Under the null hypothesis, it leads to the highest SD of EPASA of 0.500. Note that 0UCB comes close to it (0.483) because this procedure is essentially a (frequentist) myopic procedure allocating the patients to the arm with the currently highest sample mean. A Bayesian version of the myopic procedure is CB in Table 14.1, which allocates using the current belief (the mean of the posterior distribution). All the three procedures are extremely aggressive and they almost never end the trial with at least 1 success and 1 failure on each arm, and so their Type I error and power are extremely low (unless the significance level is adjusted). We also see that under the alternative hypothesis,

both 0UCB and CB are outperformed by many other procedures, and their SDs of ENS and of EPASA are notably larger than those of all the other procedures. It is thus clear that these two procedures are not good choices.

In terms of patient benefit, we look at both tables and focus on ENS under the alternative hypothesis (because EPASA was calculated slightly differently, as described above). The highest ENS is achieved by DP (70.696), closely followed by WI (70.667) in Table 14.2. We believe that WI (70.73) in Table 14.1 is better than DP only due to simulation error, but we do highlight that WI is an excellent approximation to the DP. There are several runners-up with less than 1% ENS suboptimality: 0.99RDP (70.504), 7C+DP (70.441), 0.18UCB (70.356) and GI (70.21). This patient benefit suboptimality comes with higher Type I error and higher power, but there are notable differences between these procedures, depending on the test and confidence level used, with no overall winner. For instance, in three out of the four tests, 7C+DP has lower Type I error than 0.18UCB, but notably higher power and higher ENS; in three out of the four tests, 0.99RDP has higher or equal power and higher ENS than 0.18UCB, but notably lower Type I error; and in the two tests at higher confidence level, 7C+DP has lower Type I error and higher power than 0.99RDP, but lower ENS.

Table 14.2 illustrates the flexibility of each of the three families of procedures: UCB, CRDP and CDP. For the CDP family, we increase the constraining parameter by approximately $0.05T$, penalizing if there are fewer than 7, 15, 22, 30, 37 observations on each arm. For the CRDP family, we include 0.99RDP and 0.95RDP to illustrate the performance of unconstrained procedures, and then we set the constraining parameter by approximately 0.05 above the complement of the randomization parameter (e.g., for 37C+0.8RDP, the complement of the randomization parameter 0.8 is 0.2, so we set the constraining parameter to $0.25T$). Note that varying the parameters of CRDP and CDP leads to a monotone change in ENS, but varying the $\alpha$ in the UCB leads to a concave change, as there is a maximum around $\alpha = 0.18$, and lower values quickly deteriorate the performance. For all three families, we can see that *the Type I error is concave, while power is monotone.* These non-monotonicities give scope for parameter optimization if the designer knows the relative importance of the three metrics.

In order to compare among these three families, note that 2UCB, 37C+0.8RDP and 37C+DP are quite similar in the Type I error, under all four tests, and also quite similar in ENS, but there seems to be a mild difference in the power, with 2UCB dominating the other two. Another triple for comparison would be 0.5UCB, 15C+0.95RDP and 15C+DP, for which the conclusion would be similar, except for the F-test at 0.95 confidence level, at which 15C+0.95RDP becomes the best in power. Finally, comparing 0.18UCB, 0.99RDP and 7C+DP, 7C+DP is the best in power for all tests. Note however that 37C+0.8RDP 15C+0.9RDP and 0.99RDP are randomized procedures, while the other two families are deterministic.

We note that TS in Table 14.1 performs relatively poorly in ENS, outperforming only EFR and LFF, while losing only a bit of power and inflating the Type I error comparing to these two procedures. This may be surprising for the reader, but we

note that the table reports a finite sample performance of this asymptotically optimal procedure.

In terms of statistical testing (excluding 0UCB and ORACLE from this discussion due to their extremely low Type I errors), there are important differences between the z-test and F-test at confidence level 0.95. The Type I error (expected to be 0.05) of the z-test is inflated by all the procedures, from 0.051 (EFR) up to 0.097 (0.25UCB), while that of the F-test is controlled well (the only inflation is to 0.051 of 0.25UCB), showing its most extreme conservatism for EFR (0.024), LFF (0.024) and DP (0.028). In general, there is a strong correlation of Type I errors between these two tests, z-test achieving approximately twice the Type I error of the F-test. There are also notable differences in power, as the F-test achieves power of between 0.185 and 0.342 lower than the z-test. For the z-test at 0.98, the Type I error is also inflated by all the procedures, from 0.021 (EFR) up to 0.051 (0.25UCB). An attentive reader however might notice that the Type I errors reported for z-test at 0.98 and for F-test at 0.95 are very similar across all the procedures. In fact, except for 22C+DP, for which the relation is opposite by 0.001, the former always leads to a lower or equal Type I error. At the same time, it always leads to a higher power. Similarly, z-test at 0.95 is better than F-test at 0.91 as it always results in a lower Type I error and in a notably higher power. The F-test is often cited as conservative, however, Table 14.2 shows that at 0.91 confidence level that is not always true, especially for some of the more aggressive procedures, which can even inflate the Type I error. To the best of our knowledge, this is the first time that inflation of the Type I error by Fisher's exact test has been reported in the literature. These observations suggest that in the null and alternative hypotheses scenarios we have presented, it might be preferable to use z-test over F-test. However, we emphasize that we have discussed only a single pair of scenarios of the null and alternative hypotheses, the performance of statistical tests for binomial samples is very sensitive to the specific scenario parameters and the appropriateness of using these tests is highly dependent on the specifics of each procedure, so we would refrain from any generalizations. In practice, the trial designer could replicate our analysis and study a variety of plausible scenarios. In theory, inference with data obtained by adaptive procedures remains an important open question and requires further research. Some recent examples of work in this area include (Hadad et al., 2021; Zhang et al., 2020; Deliu et al., 2021).

The tables do not include any measures related to estimation, because that on its own has trade-offs between precision and accuracy, which has been left out of this chapter.

## 14.5 Discussion

In this section, we close the chapter by discussing how (and when) bandit models can be specified to design a clinical trial beyond the traditional assumptions considered in here. These include the presence or possibility of delayed responses,

other practicalities such as dropouts (or patients lost to follow up) and/or missing responses, safety concerns, early evidence of efficacy or futility and unavailability of prior distributions. We also discuss how bandit models as those reviewed here, which are typically defined for binary outcomes, can be used in practice to accommodate for a primary endpoint that is non-binary through the use of an appropriate surrogate endpoint. Finally, we discuss how the computational limitations of optimal bandit approaches (i.e., those like CRDP for finite size trials) can be mitigated by using an efficient programming language and a more effective coding syntax to allow for designing and evaluating trials with several thousands of patients.

For many of the practicalities discussed below, we discuss how the MDP model of CRDP could be amended, as some of these have been recently explored in the literature. We are not aware of how other procedures perform in the presence of them or how could they be adjusted to incorporate each practicality.

### *14.5.1 Safety Concerns*

Many trials in practice are forced to stop recruitment due to safety concerns by observing secondary endpoints or adverse events, which have nothing to do with the observed (primary endpoint) outcomes on which a response-adaptive procedure is typically based. A designer using a response-adaptive procedure may need to incorporate the possibility of stopping for safety concerns to introduce more control over the number of observations from each arm. This can be done by incorporating the probability of such stopping in the MDP model of the DP and CRDP procedures (which we jointly refer to as (CR)DP) and by specifying constraints or by keeping the degree of randomization relatively balanced in early stages. We are not aware of how that could be incorporated to procedures, which are agnostic to the trial size, apart from UCB in which we could perhaps adaptively change the parameter $\alpha$ as the trial evolves.

### *14.5.2 Prior Distributions*

All the results presented in this chapter assume for each arm Bayes' prior $Beta(1, 1)$, which is the uniform distribution and is commonly considered non-informative. This is the standard choice for binary outcomes in methodological papers using Bayesian framework. Trial designers can however consider an informative one based on data from previous trials. The (CR)DP easily allows also for implementing a decreasingly informative prior (Donahue and Sabo, 2021) by modifying the rewards and transition probabilities between states.

In some situations, there is no previous reliable data or willingness to specify the prior distributions for each arm. In that case, the trial could have an initial phase

in which a non-adaptive randomization procedure is used, and bandit approach is employed only after that phase accumulates sufficient amount of information, which will be taken as the prior distribution for the (CR)DP procedures. In Williamson and Villar (2020), some sensitivity analysis for different informative priors in a continuous endpoint case paired with a randomized index procedure is illustrated.

### *14.5.3 Delayed Responses*

Williamson et al. (2021) evaluated how the (CR)DP procedure performs in two-armed trials with both fixed and random delays in responses (i.e., in observations of outcomes). This is an important question in practice which is natural to ask about any response-adaptive procedure. To summarize, they illustrated that one gains slightly in terms of power and bias through the delay, so in that sense delay could be viewed as a positive attribute from the statistical point of view (which seems somewhat counter-intuitive), but one loses in terms of patient benefit which is the main advantage of using such response-adaptive procedures over alternatives. However, this loss is not overly concerning and for a relatively large, fixed delay length, for example, one-third of the sample size 75, the percentage of patients on the superior arm when $p_0 = 0.5$ and $p_1 = 0.1$ is approximately 23% higher for CRDP and 30% higher for DP than the traditional approach of EFR. Furthermore, when compared to the performance of the most commonly studied procedure for delayed responses scenarios (Hardwick et al., 2006), namely the Delayed Randomized Play-the-Winner Rule (DRPWR), there are still considerable improvements with respect to the patient benefit for (CR)DP. Therefore, this evaluation has shown that the (CR)DP procedures perform well in trials with delayed responses since they continue to dominate in terms of the patient benefit over other procedures for a range of (expected) delay lengths.

The investigation in Williamson et al. (2021) leads to a conclusion that it may not be necessary to adjust the CRDP optimization horizon (i.e., to decrease $T$ by the delay length $d$) if the delay is large enough to satisfy the desired constraints already by the equal fixed-randomization of the first $d + 1$ patients, and essentially such constraints may not need to be included in the optimization model at all. For smaller delays, if the designer decides to adjust the horizon, it might be beneficial for fine-tuning of the procedure to also appropriately adjust the constraining parameters taking into account the observations of the patients which will be revealed after the recruitment of the last patient. Another option the designer has is to reach the desired trial design objectives for statistical operating characteristics (high power, small bias) by modifying the randomization probabilities, either for the early patients that are fixed-randomized before the first observation or for the remaining patients that are allocated using the CRDP procedure or both.

Special attention needs to be paid if there is a possibility of overly delayed responses so that these are not observed by the time of the final analysis. In that case,

(CR)DP with non-adjusted horizon may not even reach the final stage in which the constraints are specified, so adjusting the horizon seems to be a preferred approach.

### *14.5.4 Dropouts and Missing Responses*

When designing a randomized controlled trial, the designer needs to account for the possibility of dropouts and missing responses, i.e., patients who are recruited and get allocated to one of the arms, but we fail to observe their response, either because they leave the trial or because their outcome is erroneous. A simple approach the designers can take is to estimate the probability of missing responses and inflate the trial size so that the expected number of observations excluding the missing responses is the desired one. With (CR)DP, we can take this possibility into account by adjusting the procedure optimization horizon by a constant, e.g., for a trial size $T$, taking the procedure horizon $T - m$, where $m$ is an estimate of the number of missing responses, and correspondingly specify the constraints for the final stage $T - m$. It is also possible to consider a random number of missing responses, which would keep the procedure horizon $T$ but would include constraints not only in the final stage but also in previous stages which we would like to avoid. In that case, the state-transition probabilities of the MDP model of the (CR)DP procedure could be modified to account for the probability of observed dropouts or erroneous outcomes.

### *14.5.5 Early Evidence of Efficacy or Futility*

Although the trial size is usually planned based on existing data and/or expert opinion about the expected intervention effect (i.e., difference between the two intervention success probabilities), such estimates likely come with a large variance and bias. Both frequentist and Bayesian concepts have been developed to identify situations during the trial which would identify sufficient evidence of efficacy or futility of an intervention. In case of evidence of futility of a novel intervention, recruitment to this arm should be stopped to keep patient benefit for the remaining in-trial patients at least at the level of the current standard of care. In case of evidence of efficacy of a novel intervention, there are two common design approaches: (1) a decision as a result of an interim analysis is made to stop the recruitment to the novel arm, and the intervention to "graduate" to another separate trial to confirm efficacy, or (2) the trial seamlessly transforms to such a confirmatory trial without an explicit interim analysis.

Both cases can be incorporated in the MDP model of the CRDP procedure. For instance, consider a state of the trial with 5 observations on each arm, with the most extreme data: 5 successes and 0 failures on one arm and 0 successes and 5 failures on the other arm. Fisher's exact test would give a one-tailed p-value of 0.004 based on this data, showing evidence of difference between the two arms. In case of an

interim analysis which would stop recruitment for futility of the novel arm, the MDP model of the CRDP procedure can be modified by assuming that all the remaining in-trial patients will be allocated to the control arm, i.e., by modifying the reward of that state and by modifying the state-transition probabilities to "jump" to the end of the trial. In case of an interim analysis which would stop recruitment for efficacy of the novel arm, the MDP model can be modified by assuming that all the remaining in-trial patients will be randomized in the new separate trial, i.e., by modifying the reward of that state and by modifying the state-transition probabilities to "jump" to the end of the trial. In case of a seamless transformation of the trial, the degree of randomization of the subsequent states can be defined differently from the degree of randomization of the subsequent states that do not show such a strong evidence, so, effectively, further generalizing the CRDP procedure to allow for randomization $p$ to depend not only on arm $j$ and time stage $t$ as in Williamson et al. (2021) but also on the state (i.e., numbers of successes and failures) itself.

### *14.5.6 Non-binary Outcomes*

Development of an analogous randomization procedure to (CR)DP when the primary endpoint is non-binary is theoretically possible but computationally will become infeasible for much smaller trial sizes than the current variant for binary outcomes. The designer could still however employ the binary outcomes (CR)DP by using a dichotomization of the primary endpoint or by using an auxiliary endpoint correlated with the primary endpoint. Although dichotomization may not lead to as high patient benefit as theoretically achievable using the original endpoint, if meaningfully defined it could lose only a negligible amount and thus still bring important patient benefit over alternative response-adaptive procedures. The degree of randomization could be adjusted in order to reflect the designer's confidence in the correlation between the primary and auxiliary endpoints. See, for instance, Williamson and Villar (2020) for such an investigation for normally distributed outcomes.

### *14.5.7 Exploratory Trials*

In a two-armed setting, we discussed and illustrated the conflict between patient benefit (patient outcomes) and relevant statistical features (error levels and estimation metrics). In the two-arm setting, there is little scope for a bandit procedure to be superior to EFR in terms of the latter metrics. In a multi-armed setting (as for example large platform trials are), this is not necessarily the case, and depending on the main objective of the trial (e.g., the specific statistical power definition used) and the type of bandit procedure, one can find alternatives that may be superior to EFR in both the statistical features and patient benefit. Exploratory trials, which are

**Table 14.3** Comparison of procedures in a four-arm trial of size $T = 423$ by simulation. $F_a$: Fisher's adjusted test; Type I E: family-wise Type I error; CGI: controlled Gittins index. Reprinted (adapted) from Villar et al. (2015, Table 6)

| | Crit. | $H_0 : p_0 = p_1 = p_2 = p_3 = 0.3$ | | | $H_1 : p_0 = p_1 = p_2 = 0.3, p_3 = 0.5$ | | |
|---|---|---|---|---|---|---|---|
| | value | Type I E | EPASA (SD) | ENS (SD) | Power | EPASA (SD) | ENS (SD) |
| EFR | 2.128 | 0.047 | 0.250 (0.02) | 126.86 (9.41) | 0.814 | 0.250 (0.02) | 148.03 (9.77) |
| TS | 2.128 | 0.056 | 0.251 (0.07) | 126.93 (9.47) | 0.884 | 0.529 (0.09) | 172.15 (13.0) |
| 2UCB | 2.128 | 0.055 | 0.251 (0.06) | 126.97 (9.41) | 0.877 | 0.526 (0.07) | 171.70 (11.9) |
| RBI | 2.128 | 0.049 | 0.250 (0.03) | 126.77 (9.40) | 0.846 | 0.368 (0.04) | 158.34 (10.4) |
| RGI | 2.128 | 0.046 | 0.250 (0.03) | 126.80 (9.36) | 0.847 | 0.358 (0.03) | 157.26 (10.3) |
| CB | $F_a$ | 0.047 | 0.269 (0.39) | 126.89 (9.61) | 0.213 | 0.677 (0.41) | 184.87 (36.8) |
| GI | $F_a$ | 0.048 | 0.248 (0.18) | 126.68 (9.40) | 0.428 | 0.831 (0.10) | 198.25 (13.7) |
| CGI | 2.128 | 0.034 | 0.250 (0.02) | 127.16 (9.46) | 0.925 | 0.640 (0.08) | 182.10 (12.3) |
| ORACLE | | 0.000 | 0.250 (0.43) | 126.90 (9.42) | 0.000 | 1.000 (0.00) | 211.50 (10.3) |

often multi-armed, are moreover not meant to directly lead to a regulatory approval and thus may not need to perform in statistical operating characteristics as strictly as confirmatory trials would need to.

This was illustrated in Villar et al. (2015, Table 6) reproduced here as Table 14.3 for easy reference. The results in there show how some randomized and semi-randomized bandit procedures (i.e., TS, 2UCB, RBI, RGI) exhibit an advantage over EFR both in the achieved power and in ENS. These procedures continue to allocate patients to all arms during the trial while skewing allocation to the best performing arm, hence, ensuring that by the end of the trial the control arm will have a similar number of observations as with EFR, while the best arm will (in expectation) have a larger number. Among these procedures, TS and 2UCB exhibit the best performance in power and ENS as they are both greater than those achieved by EFR, although they cause a slight inflation of the Type I error. While RBI and RGI were performing somewhat similarly to TS and 2UCB in the two-armed setting shown in Table 14.1, their performance in ENS terms is notably inferior in the multi-armed setting shown in Table 14.3.

The deterministic index-based procedures CB and GI increase the advantage in ENS over EFR even more, while the Type I error is controlled using an adjusted Fisher test. However, this conservative test causes a severe reduction in power of these procedures. A simple way to overcome the severe loss of statistical power of the deterministic procedures in the multi-armed setting introduced in Villar et al. (2015) suggests to use a composite procedure in which the (random) allocation to the control arm is protected and the allocation to experimental arms is guided by a deterministic procedure. For example, in Table 14.3, results are shown for a procedure in which one in every $K$ patients (note that $K$ is the number of experimental arms) is allocated to the control group, while the allocation of the remaining patients among the experimental treatments is done using the Gittins index procedure. This procedure was referred in there as the controlled Gittins index (CGI) procedure. Simulation results show that a simple procedure like CGI manages

to solve the trade-off quite successfully, in the sense that it achieves the highest power, lowest Type I error and an ENS very close to that achieved by the myopic CB procedure but with a third of the variability that CB exhibits.

### *14.5.8 Large Trials*

Williamson et al. (2017) developed the (CR)DP procedure in the context of rare diseases and thus focused on relatively small trial sizes. They provided an "efficient algorithm" for (CR)DP implemented in the statistical software R and reported that the maximum time horizon that "can be computed on a standard laptop using R is $T = 215$" and that computations are "feasible on a standard performance workstation (1 TB of RAM) for $215 < T < 600$". Jacko (2019b,a) however showed that much larger horizons are possible to compute on standard computer (with 32 GB RAM) if using a more efficient programming language (Julia) and a more effective coding syntax, with up to $T = 4500$ for online calculation and $T = 1500$ for offline calculation (storing the whole (CR)DP procedure allocations in an array for saving on a hard disk).

The (CR)DP procedure could be in theory generalized to more than 2 arms, but in practice that might lead to computationally infeasible model. Alternatives that closely approximate the DP procedure are the Whittle index and the Gittins index (Villar et al., 2015; Villar, 2018; Jacko, 2019b). However, their modifications to include constraints like in the CRDP procedure have not been developed yet and may not always be possible, especially for constraints that depend on more than one arm, because the Whittle and Gittins indices crucially function by decomposing the trial-level optimization problem into single-arm optimization subproblems. Nevertheless, single-arm constraints such as about the number of observations from each arm should be implementable. If constraints are not required, then the degree of randomization can be easily implemented using the Whittle or Gittins index instead of the DP procedure in the alternative interpretation described in Williamson et al. (2021).

## References

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning, 47*(2–3), 235–256.

Bowden, J., & Trippa, L. (2017). Unbiased estimation for response adaptive clinical trials. *Statistical Methods in Medical Research, 26*(5), 2376–2388.

Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning, 5*(1), 1–122.

Burnett, T., Mozgunov, P., Pallmann, P., Villar, S. S., Wheeler, G. M., & Jaki, T. (2020). Adding flexibility to clinical trial designs: an example-based guide to the practical use of adaptive designs. *BMC Medicine, 18*(1), 1–21.

Cserna, B., Petrik, M., Russel, R. H., & Ruml, W. (2017). Value directed exploration in multi-armed bandits with structured priors. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*.

Deliu, N., Williams, J. J., & Villar, S. S. (2021). Efficient inference without trading-off regret in bandits: An allocation probability test for Thompson sampling. *Preprint arXiv:2111.00137*.

Donahue, E., & Sabo, R. T. (2021). A natural lead-in approach to response-adaptive allocation for continuous outcomes. *Pharmaceutical Statistics, 20*, 1–10.

Hadad, V., Hirshberg, D. A., Zhan, R., Wager, S., & Athey, S. (2021). Confidence intervals for policy evaluation in adaptive experiments. *Proceedings of the National Academy of Sciences, 118*(15), e2014602118.

Hardwick, J., Oehmke, R., & Stout, Q. F. (2006). New adaptive designs for delayed response models. *Journal of Statistical Planning and Inference, 136*, 1940–1955.

Jacko, P. (2019a). *BinaryBandit: An efficient Julia package for optimization and evaluation of the finite-horizon bandit problem with binary responses*. Management Science Working Paper 2019:4, Lancaster University Management School.

Jacko, P. (2019b). *The finite-horizon two-armed bandit problem with binary responses: A multidisciplinary survey of the history, state of the art, and myths*. Management Science Working Paper 2019:3, Lancaster University Management School. arXiv:1906.10173.

Kelly, F. (1981). Multi-armed bandits with discount factor near one: the Bernoulli case. *Annals of Statistics, 9*(5), 987–1001

Melfi, V., & Page, C. (1998). Variability in adaptive designs for estimation of success probabilities. In *New developments and applications in experimental design*, Lecture Notes-Monograph Series (Vol. 34, pp. 106–114).

Nie, X., Tian, X., Taylor, J., & Zou, J. (2018). Why adaptively collected data have negative bias and how to correct for it. In *International Conference on Artificial Intelligence and Statistics* (pp. 1261–1269). PMLR.

Pallmann, P., Bedding, A. W., Choodari-Oskooei, B., Dimairo, M., Flight, L., Hampson, L. V., Holmes, J., Mander, A. P., Sydes, M. R., Villar, S. S., Wason, J. M. S., Weir, C. J., Wheeler, G. M., Yap, C. & Jaki, T. (2018). Adaptive designs in clinical trials: why use them, and how to run and report them. *BMC Medicine, 16*(1), 1–15.

Pilarski, S., Pilarski, S., & Varró, D. (2021). Optimal policy for Bernoulli bandits: Computation and algorithm gauge. *IEEE Transactions on Artificial Intelligence, 2*(1), 2–17.

Press, W. H. (2009). Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *Proceedings of the National Academy of Sciences, 106*(52), 22387–22392.

Robertson, D. S., Lee, K. M., Lopez-Kolkovska, B. C., & Villar, S. S. (2021). Response-adaptive randomization in clinical trials: From myths to practical considerations. *Preprint arXiv:2005.00564*.

Rosenberger, W. F. (1999). Randomized play-the-winner clinical trials: Review and recommendations. *Controlled Clinical Trials, 20*(4), 328–342.

Rosenberger, W. F., & Lachin, J. M. (2015). *Randomization in clinical trials: Theory and practice*. Wiley.

Rosenberger, W. F., Stallard, N., Ivanova, A., Harper, C. N., & Ricks, M. L. (2001). Optimal adaptive designs for binary response trials. *Biometrics, 57*(3), 909–913.

Rosenberger, W. F., Uschner, D., & Wang, Y. (2019). Randomization: The forgotten component of the randomized clinical trial. *Statistics in Medicine, 38*(1), 1–12.

Ryzhov, I. O. (2021). Optimal learning and optimal design. In *The elements of joint learning and optimization in operations management*. Berlin: Springer.

Stallard, N., Hampson, L., Benda, N., Brannath, W., Burnett, T., Friede, T., Kimani, P. K., Koenig, F., Krisam, J., Mozgunov, P., Posch, M., Wason, J., Wassmer, G., Whitehead, J., Williamson, S. F., Zohar, S., Jaki, T. (2020). Efficient adaptive designs for clinical trials of interventions for COVID-19. *Statistics in Biopharmaceutical Research, 12*(4), 483–497.

Sverdlov, O., Tymofyeyev, Y., & Wong, W. K. (2011). Optimal response-adaptive randomized designs for multi-armed survival trials. *Statistics in Medicine, 30*(24), 2890–2910.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3/4), 285–294.

Villar, S. S. (2018). Bandit strategies evaluated in the context of clinical trials in rare life-threatening diseases. *Probability in the Engineering and Informational Sciences, 32*, 229–245.

Villar, S. S., Bowden, J., & Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical Science, 30*(2), 199–215.

Wei, L. J., & Durham, S. (1978). The randomized play-the-winner rule in medical trials. *Journal of the American Statistical Association, 73*(364), 840–843.

Whittle, P. (1980). Multi-armed bandits and the Gittins index. *Journal of the Royal Statistical Society, Series B, 42*(2), 143–149.

Williamson, S. F., Jacko, P., & Jaki, T. (2022). Generalisations of a Bayesian decision-theoretic randomisation procedure and the impact of delayed responses. *Computational Statistics and Data Analysis, 174*, 107407.

Williamson, S. F., Jacko, P., Villar, S. S., & Jaki, T. (2017). A Bayesian adaptive design for clinical trials in rare diseases. *Computational Statistics and Data Analysis, 113C*, 136–153.

Williamson, S. F., & Villar, S. S. (2020). A response-adaptive randomization procedure for multi-armed clinical trials with normally distributed outcomes. *Biometrics, 76*(1), 197–209.

Zelen, M. (1969). Play the winner rule and the controlled clinical trial. *Journal of the American Statistical Association, 64*(325), 131–146.

Zhang, K., Janson, L., & Murphy, S. (2020). Inference for batched bandits. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems*, (Vol. 33, pp. 9818–9829). Curran Associates.

# Chapter 15
# Dynamic Treatment Regimes for Optimizing Healthcare


**Nina Deliu and Bibhas Chakraborty**

## 15.1 Introduction

Treatment of diseases or disorders, including both chronic conditions and acute illnesses, often involves a series of decisions over time to address evolving characteristics of patients' conditions. For example, treatment of cancer involves a succession of decisions at key milestones in the disease progression: initially, patients are generally treated with a powerful chemotherapy, known as *induction therapy*, to induce remission of the disease; then, if the patient *responds* (i.e., shows sign of remission), the clinician tries to maintain remission for as long as possible by prescribing a *maintenance therapy*, otherwise, the clinician prescribes a *second-line* or *salvage induction* therapy to try to induce remission. Of course there exist many possible induction and maintenance therapies. The specific sequence of possible therapies is generally chosen by a clinician in order to elicit the best outcome possible, e.g., long survival with little toxicity, or to maximize a single outcome of interest. Similarly, management of mental health and behavioral disorders, or

N. Deliu (✉)
MRC - Biostatistics Unit, University of Cambridge, Cambridge, UK

Department of Methods and Models for Economics, Territory and Finance, Sapienza University of Rome, Rome, Italy
e-mail: nina.deliu@uniroma1.it

B. Chakraborty
Centre for Quantitative Medicine and Program in Health Services and Systems Research, Duke-NUS Medical School, National University of Singapore (NUS), Singapore, Singapore

Department of Statistics and Data Science, NUS, Singapore, Singapore

Department of Biostatistics and Bioinformatics, Duke University, Durham, NC, USA
e-mail: bibhas.chakraborty@duke-nus.edu.sg



X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_15

other medical conditions such as obesity, requires a series of decisions in which the physician may start, stop, maintain, modify, or adjust interventions on the basis of a patient's response and other characteristics. To the patient, this sequence of treatments seems like standard treatment; to the clinician, it represents a series of decisions, based on information from previous patients with similar treatment history, characteristics, and behaviors; and to the statistician, this constitutes a *dynamic treatment regime* or *regimen* (Murphy et al., 2001; Murphy, 2003; Lavori and Dawson, 2004; Chakraborty and Moodie, 2013; Chakraborty and Murphy, 2014). Dynamic treatment regimes (DTRs) offer a vehicle to operationalize the sequential decision making process involved in the personalized clinical practice, and thereby a potential way to improve it. Thus, conceptually, a DTR can also be viewed as a *decision support system* of a clinician (or more generally, any decision maker), described as a key element of the *chronic care model* (Wagner et al., 2001).

DTRs are known by a variety of different names, with adaptive interventions (Collins et al., 2004; Nahum-Shani et al., 2012a), treatment strategies (Lavori et al., 2000; Thall et al., 2007b), adaptive treatment strategies (Murphy, 2005a; Murphy et al., 2007; Lavori and Dawson, 2008), and treatment policies (Lunceford et al., 2002; Wahed and Tsiatis, 2004; Dawson and Lavori, 2012) being the most common ones. In the context of multi-stage decision making, a DTR is defined as a sequence of decision rules, one per stage of intervention, that dictate how to personalize treatments to patients based on their baseline and evolving history (*time-varying*, *dynamic state*), repeatedly adjusting over time in response to ongoing performance (Almirall et al., 2014; Nahum-Shani et al., 2018). Thus, treatment regime is "dynamic" within a person over time, varying because the person or disease is changing, with the goal of obtaining the best results for that individual. Note that some authors have used dynamic treatment regime to refer to the fact that a regime involves multiple decision points, without regard to the nature of its rules, thus tacitly implying that a single decision regime is "nondynamic." Here, we follow the original definition of Murphy et al. (2001), in which "dynamic" refers to a regime with decision rules incorporating baseline and evolving patient information. This definition is thus consistent with the view of any single-stage regime as "nondynamic," but at the same time with "nondynamic" multi-stage decisions that do not incorporate time-varying information (but uses only baseline).

Existing frameworks for DTRs (Almirall et al., 2014; Collins et al., 2004) highlight four components that play an important role in designing these interventions: (i) the critical decision points, specifying the time points at which patients' outcomes (e.g., response), are assessed and decisions are made to continue, alter, add, or subtract treatment; at each of this points, (ii) the treatment options; (iii) the tailoring variables to personalize treatment; and (iv) a decision rule which links the tailoring variables to specific interventions. Treatment options are not limited to different medications or drugs but can also include different dosages (duration, frequency, or amount (Voils et al., 2012)), modes of delivery (e.g., oral or injection), timing schedules, behavioral interventions, or no further treatment. Tailoring variables refer to patient and treatment information available up to the time of the critical decision and may include previous treatment and disease history,

genetic information, diagnostic test results, etc. Once the four elements are defined, each decision rule takes a subject's individual characteristics and treatment history observed up to that stage as inputs and outputs a recommended treatment strategy at that stage. However, one should take into account that any combination of the four elements mentioned above does not provide a sufficient set for developing a practical DTR. Indeed, a DTR must be constructed with thought and be a treatment regimen either used by physicians in the past or that physicians would consider using. Furthermore, it is exceedingly important that the DTR be *viable* (Wang et al., 2012), *realistic* (van der Laan and Petersen, 2007a), and *feasible* (Robins, 1986, 2004) to capture those that may experience common contingencies in the course of treatment. For instance, aggressive initial treatment may set the stage for better overall success or lead to toxicities or other side effects which may hinder success of subsequent treatment; thus, the trade-off between short term and long-term outcomes should occupy a central place. In addition, because of the tremendous heterogeneity among people and within diseases, the same treatment may not be the best treatment for everyone or may not even be the best treatment for an extended period of time for a single person.

The study of evidence-based (i.e., data-driven) DTRs comprises an emerging and important line of methodological research within the domain of *personalized medicine*, a medical paradigm that transitions from the *one-size-fits-all* ideology, emphasizing systematic use of individual characteristics to optimize that individual's health care. In contrast with traditional single-stage treatments in which all individuals are assigned the same level and type of treatment, DTRs explicitly incorporate the heterogeneity in treatment across individuals and the heterogeneity in treatment across time within an individual (Murphy, 2003), providing an attractive framework of personalized treatments in longitudinal settings. In addition, by treating only subjects who show a need for treatment, DTRs hold the promise of reducing non-compliance by subjects due to overtreatment or undertreatment (Lavori et al., 2000; Collins et al., 2004), and at the same time are attractive to public policy makers, allowing a better allocation of public and private funds for more intensive treatment of the needy (Murphy, 2003).

The main research goals in this personalized treatments arena concern: (i) to construct *optimal DTRs*, i.e., to identify the sequence of treatments that result in the most favorable outcome of interest possible (i.e., with the highest utility) and (ii) to compare two or more preconceived DTRs in terms of their utility. In the current literature, a DTR is usually said to be optimal if it optimizes a mean *long-term outcome* (e.g., an outcome observed at the end of the final stage of intervention). However, at least in principle, other utility functions (e.g. median or other quantiles, or some other feature of the outcome distribution) can be employed as optimization criteria. Thus, any attempt to achieve the above goals in a data-driven way essentially requires knowing or estimating the *utility* functions (or some variations). For example, Murphy (2003) defines multiple stage-specific *regret* (i.e., *loss*) functions, and Robins (2004) defines stage-specific *blip* functions (alternatively known as *welfare contrasts* in econometrics). In their proposed framework of *structural nested mean models*, they provided methodologies to estimate the parameters of regret

or blip functions and thereby to identify the optimal DTR. On the other hand, *Q-learning*, a *reinforcement learning* (RL) method originally developed in computer science but later adapted to statistics, targets estimating and maximizing the utility function (conditional expectation of the primary outcome), rather than minimizing the regret or any other blip. All these approaches will be discussed in great details in Sect. 15.4, along with their relative merits and demerits.

With the above broad picture of DTRs in mind, in this chapter we aim to provide a comprehensive overview of this cutting-edge area of research. We start with a mathematical formalization of the DTRs problems, followed by existing methodologies and solutions, which includes the RL framework as well, and end with statistical and practical considerations when dealing with DTRs in real life.

## 15.2 Mathematical Framework

We now present a more formal definition of dynamic treatment regimes, introducing the basic mathematical framework of DTRs as a general decision making problem and presenting conventions and notations we adopt throughout this chapter.

Traditionally, personalized medicine concerns single-stage decision making, where the clinician has to decide on the optimal treatment for an individual patient, given their *baseline* or study-entry (*nondynamic*) *covariates*. Suppose the clinician observes a certain random characteristic (e.g., a demographic variable, a biomarker, the result of a diagnostic test, or other clinical values) of the patient, which we denote with $x_1$, and based on that has to decide whether to prescribe treatment $a$ or treatment $a'$. A decision rule, say $d$, could be, for example: "give treatment $a'$ to the patient if their individual characteristic $X_1$ is higher than a pre-specified threshold, and treatment $a'$ otherwise." Throughout this chapter, we will use case letters to denote a realized/observed variable, and capital letters for the (unobserved) random variable. In this simple setting, $d$ is a mapping from currently available information $X_1$, sometimes also referred to as *state*, into the space of possible decisions, or *actions*, say $\mathcal{A} \doteq \{a, a'\}$. In the treatment regimes literature, this single-stage decision rule is known as *individualized treatment regime* (Zhao et al., 2012a) or *individualized treatment rule* (Qian and Murphy, 2011). Any decision, medical or otherwise, is then statistically evaluated in terms of its *utility*, say $\mathcal{U}(A)$, which refers to the utility of taking a random action $A \in \mathcal{A}$. The utility function can be specified in various ways, depending on the specific problem; it can be a summary of one outcome of interest, or a composite outcome: for example, in Wang et al. (2012) the utility is a compound score numerically combining information on treatment efficacy, toxicity, and the risk of disease progression. However, one of the most common ways would be to set $\mathcal{U}(A, X_1) \doteq \mathbb{E}(Y|A, X_1)$, i.e., the conditional expectation of outcome $Y$ given the state $X_1$ and action $A$. The outcome $Y$ is generally a function of the baseline covariates $X_1$, the selected treatment $A$, and the new set of patient's covariates, say $X_1$, evaluated after giving treatment $A$. Alternatively, one can define $\mathcal{U}(A, X_1) \doteq \mathbb{E}(Y^A|X_1)$, where $Y^A$ is the *potential*

*outcome* of decision $A$; we will introduce this notion in Sect. 15.2.1. Clearly, the primary goal is to find the *optimal decision rule*, i.e., the one that outputs the action that maximizes the utility at the given state $X$; this is personalized decision making since the optimal decision depends on the state.

This decision-theoretic framework is adopted also in other literatures different from treatment regimes. For instance, in econometrics literature (Manski, 2000, 2002, 2004; Dehejia, 2005; Hirano and Porter, 2009), a similar framework is used for the evaluation of social welfare programs, where the role of a clinician is replaced by a social planner, different welfare programs serve as different treatment choices, and the state again consists of individual characteristics.

Constructing DTRs involves solving, or estimating quantities relevant in, a multi-stage decision problem. Thus, the focus of this chapter is multi-stage decision problems rather than the considerably simpler single-stage problems, for which interested readers may consult Hirano and Porter (2009) and Qian and Murphy (2011). However, we will use the single-stage decision framework at times to develop certain ideas to be ultimately used in the more complicated setting of multiple decisions.

Formally, a DTR is a sequence of decision rules, say $\mathbf{d} \doteq \{d\}_{t\geq 1}$, indexed by a finite or indefinite number $T$ of decision points at which a treatment must be selected from among a set of available, feasible options; and decisions have to be made based on the time-varying (dynamic) covariates of an individual. Given the stage or time of intervention $t$ (here we assume a discrete time space, i.e., $t \in \mathbb{N}$), we denote with $X_t$, $A_t$, and $Y_t$ the patient's covariates, the treatment option, and the outcome at time $t$, respectively. The full data trajectory of a single patients is represented as a sequence of covariates, treatments received and outcomes observed after treatment assignment, i.e., $\mathcal{T} \doteq \{X_t, A_t, Y_{t+1}\}_{t\geq 1}$. We assume that $\{Y_t\}_{t>1}$ are continuous variables that are coded so that higher values are preferred. In some problems, there may only be an end-of-study outcome of interest $Y \doteq Y_{T+1}$ instead of multiple intermediate outcomes; for example, in the *attention deficit hyperactivity disorder* (ADHD) study (Pelham et al., 2002) for evaluating the effects of a treatment on children with ADHD, the target outcome was school performance score at the end of study.

Define now $\mathbf{X_t} \doteq (X_1, \ldots, X_t)$, $\mathbf{A_t} \doteq (A_1, \ldots, A_t)$ and $\mathbf{Y_t} \doteq (Y_2, \ldots, Y_t)$, and similarly $\mathbf{x_t}$, $\mathbf{a_t}$, and $\mathbf{y_t}$, where again the upper and lower case letters denote random variables and their particular realization, respectively. Also define the *history* $\mathbf{H_t}$ as all the information available at time $t$ prior to decision $A_t$, i.e., $\mathbf{H_t} \doteq (\mathbf{X_t}, \mathbf{A_{t-1}}, \mathbf{Y_t})$; similarly $\mathbf{h_t}$. Generally, the outcome $Y_{t+1}$ at next time step $t+1$ is conceptualized as a known function of the history $\mathbf{H_t}$ at time $t$, the current decision $A_t$ and the subsequent state $X_{t+1}$ once decision is made, i.e., $Y_{t+1} = Y_{t+1}(\mathbf{H_t}, A_t, X_{t+1})$. Note that, by definition, $\mathbf{H_1} = X_1$. Let $\mathcal{H}_t \doteq \mathcal{X}_1 \times \prod_{\tau=2}^{t} \mathcal{X}_\tau \times \mathcal{A}_{\tau-1} \times \mathcal{Y}_\tau$ denote the support of $\mathbf{H_t}$, for $t \in \mathbb{N}$, with $\mathcal{X}_t$, $\mathcal{A}_t$ and $\mathcal{Y}_t$ the support of the state, the set of available options and the support of the outcome variable at time $t$.

Armed with this notation, we can now more formally define a dynamic treatment regime as a set of decision rules, also known as *policy*

$$\mathbf{d} \doteq \{d_t\}_{t\geq 1} = \{d_1, d_2 \dots\} = \{d_1(X_1), d_2(\mathbf{H_2}), \dots\}, \tag{15.1}$$

where each stage-$t$ rule $d_t$ is a function that maps an individual's history $\mathbf{H_t} \in \mathcal{H}_t$ to a treatment option in $\mathcal{A}_t$, that is, $d_t : \mathcal{H}_t \to \mathcal{A}_t$, for all $t \geq 1$.

An important distinction in the formulation of a treatment regime relates to the deterministic (or nonrandom) vs random set of decision rules. A rule $d_t(\mathbf{H_t})$ is said to be nonrandom if, at each time $t \geq 1$, given history $\mathbf{H_t}$, it assigns one and only one treatment option from among those in $\mathcal{A}_t$; while a random policy will assign treatments according to some pre-specified probabilities depending on $\mathbf{H_t}$. In the vast majority of applications, as well as in this chapter, interest is restricted to nonrandom regimes. We refer to Murphy et al. (2001) for discussion of random regimes.

The optimal DTR is the set of decision rules, denoted by $\mathbf{d}^* = \{d_t^*\}_{t\geq 1}$, that maximizes the expected utility. If we denote, as mentioned before, the utility as a conditional expectation of an outcome given action $A$ and the initial state (let us assume for now we only have a final outcome of interest $Y \doteq Y_{T+1}$), the optimal regime $\mathbf{d}^* = \{d_t^*\}_{t=1,\dots,T}$, is given by

$$\boldsymbol{d^*} \doteq \arg\max_{\boldsymbol{d}} \mathbb{E}_{\boldsymbol{d}}[\mathcal{U}(A, X_1)] \doteq \arg\max_{\boldsymbol{d}} \mathbb{E}_{\boldsymbol{d}}[Y|X_1], \tag{15.2}$$

where the expectation is taken with respect to the probability distribution of full data trajectory $\mathcal{T}$ induced by assigning treatment according to policy $\boldsymbol{d}$, which we denoted by $\mathbb{E}_{\boldsymbol{d}}$.

This expected utility is also known as the (*state*) *value* of a specific treatment regime $\boldsymbol{d}$ for an individual with baseline information or state $X_1$, so that to incorporate the notion of personalization. Other authors may refer to this value as the marginal expectation of $Y$, i.e., $\mathbb{E}_{\boldsymbol{d}}[Y]$, with marginalization occurring over the space of all possible baseline information $\mathcal{X}_1$. We will clearly make the distinction in this work, by using the terms *value*, *state value*, and *action value* (which will be introduced later), referring to the marginal expectation, the expectation conditioned on the state as in (15.2), and the expectation conditioned on both state and action, respectively.

### *15.2.1 Potential Outcomes Framework*

In order to understand the methods for constructing and estimating DTRs we will be discussing later in Sect. 15.4, and to allow the quantification of treatment effects from observational or experimental data, we present the foundations for the underlying *potential outcomes* or *counterfactual framework*. The potential outcomes framework was first introduced to analyze causal effects of time-independent treatments in randomized trials (Neyman, 1923) and then extended to observational studies (Rubin, 1974) and time-dependent treatments in observational and random-

ized studies (Robins, 1986). By far, it represents the most popular approach to mathematically defining a causal effect and constitutes a basis for the modern causal inference.

Potential outcomes are the set of all possible values of a state or outcome variable for an individual, each of which is associated with a unique treatment regime (sequence). Thus, it also includes those regimes different from the one they were actually observed to follow (hence, counter to fact). In a simple one-stage study in which subjects can receive either treatments $a$ and $a'$, we denote the set of (unobserved) potential outcomes for an individual with baseline information $X_1$ by $\left(X_2^a, Y_2^a, X_2^{a'}, Y_2^{a'}\right)$, where $X_2^a$ and $Y_2^a$ refer to the potential state and outcome that were to be observed if assigned to treatment $a$. Clearly, in line with what we saw at the beginning of this section, $Y_2^a \doteq Y_2(X_1, a, X_2^a)$. This framework applies to the study of DTRs as well as to other different areas where estimating causal effects are of central interest. For instance, in educational sciences, one may want to understand how an educational intervention (e.g., an online course format) changes student achievement relative to other levels of the intervention (e.g., in-person course format). In this case, a student will have one potential outcome (achievement) when assigned to an online course and another one when assigned to an in-person course; even if for an individual only one outcome will be observed (the one associated with the assigned intervention), they are all possible if they could have been assigned to a different option. The difference between the student's potential outcome with their assigned intervention (say online format) and the potential outcome for a different intervention (in-person attendance) is the causal effect of the online intervention relative to the in-person alternative.

In order to define what we mean by a causal effect, for each individual (or subject, or unit) we thus assume the existence of the potential outcomes, $Y_2^a, Y_2^{a'}$, corresponding to what value the outcome would take if we did assign $a$ or $a'$, respectively. Then, to calculate the causal effect on a given individual we would need to somehow compute the so-called *individual-level causal parameter* given by $Y_2^a - Y_2^{a'}$. However, since we cannot observe all the potential outcomes on the same individual, typically *population-level causal parameter* (e.g., $\mathbb{E}[Y_2^a] - \mathbb{E}[Y_2^{a'}]$) is considered instead. In order to connect the potential outcomes with observed data, ensuring $\hat{\mathbb{E}}[Y_2|A = a]$ is an unbiased estimate of $\mathbb{E}[Y_2^a]$, the following three assumptions about the assignment mechanism must hold.

1. *Stable unit treatment value assumption* (SUTVA), which assumes that each participant's potential outcome is not influenced by the treatment applied to other participants (Rubin, 1978, 1980). This assumption connects the potential outcomes to the observed data such that, for each $t$, $X_t^{\mathbf{a_t}} = X_t(\mathbf{a_t}) \doteq X_t$ and $Y_t^{\mathbf{a_t}} = Y(\mathbf{a_t}) \doteq Y_t$, when regime $\mathbf{a_t} \doteq (a_1, \ldots, a_t)$ is actually followed. This agreement between potential outcomes under the observed treatment and the observed data is sometimes referred to as *axiom of consistency*.

2. *No unmeasured confounders* (NUC), which states that conditional on the patient's history $\mathbf{H_t}$ up to time $t$, the treatment assignment $A_t$ at time $t$ is independent of future potential outcomes of the individual (Robins, 1997). That is, for any regime $\mathbf{a_t}$,

$$A_t \perp \left(X_{t+1}^{\mathbf{a_t}}, Y_{t+1}^{\mathbf{a_t}}, X_{t+2}^{\mathbf{a_{t+1}}}, Y_{t+1}^{\mathbf{a_{t+1}}}, \dots\right) \Big| \mathbf{H_t}, \quad \forall t \geq 1. \tag{15.3}$$

   This assumption always holds under either complete or sequential randomization, but must be evaluated on subject matter grounds in observational studies.
3. *Positivity*, which defines the set of *feasible* regimes so that for every covariate-treatment history up to time $t$ that has a positive probability of being observed, there must be a positive probability that the corresponding treatment dictated by the treatment regime will be observed (Robins, 1994). Formally, if we denote with $\boldsymbol{\pi}$ the probability distribution of actions given the history, a feasible regime $\mathbf{d}(\mathbf{h}) = \mathbf{a}$ satisfies

$$\pi_t(d_t(\mathbf{H_t})|\mathbf{H_t} = \mathbf{h_t}) > 0, \quad \forall \mathbf{h_t} \in \mathcal{H}_t, \forall t \geq 1. \tag{15.4}$$

   That is, feasibility requires some subjects to follow regime $\mathbf{d}$ to guarantee non-parametric estimation of its performance.

   As we will see later in Sect. 15.4.2, the notation "$\pi$" is not arbitrary. It translates the notion of "exploration policy" meant for the action process generation, and in a case of a randomized trial it consists of the randomization probabilities.

Under the consistency, sequential randomization, and positivity assumptions, the conditional distributions of the observed data are the same as the conditional distributions of the potential outcomes. It follows that an optimal treatment regime may be obtained using the observed data.

## 15.3 Data Sources for Constructing DTRs

For the study of DTRs, either for developing new regimes or for evaluating/comparing existing regimes, three sources of data have been generally considered the in literature: (1) longitudinal observational studies, (2) sequentially randomized studies, and (3) dynamical systems models. While research based on the first type of data, i.e., observational studies, have been the focus of the majority of real-life studies conducted so far, experimental data source is experiencing a period of rapid growth and currently represents the gold standard for developing DTRs. The third data source has received much less attention in the study of DTRs; however, it represents a common choice in the statistics and machine learning literature aiming at developing and improving existing DTRs methodologies. Despite their artificial

nature, dynamical systems strongly rely on biological or behavioral models to simulate patient trajectories under different DTRs.

In this section we review these different types of data sources, their advantages and drawbacks, and practical considerations to account for in the development or evaluation of a DTR, in relation also with the causal inference assumptions required to perform valid analysis. In doing so, we present and discuss some key examples.

### *15.3.1 Longitudinal Observational Data*

Observational studies are the most common source of data to shed light on potential DTRs, representing a particularly preferred option in scenarios in which a trial would be either cost-prohibitive or of concern from an ethical or logistical perspective (e.g., in several chronic diseases such as diabetes or HIV). In addition, they offer the potential to inform construction of promising regimes to be tested in a confirmatory trial (Kidwell, 2015).

Observational data sources include electronic medical records and other administrative (e.g., hospital) databases (Rosthøj et al., 2006; Cain et al., 2010; Cotton and Heagerty, 2011), randomized encouragement trials (Moodie et al., 2009), and cohort studies (van der Laan and Petersen, 2007b). In all these cases the treatments are not randomized within the study; in particular, the reasons why different individuals receive differing treatments or the reasons why one individual receives different treatments at different times are not known with certainty. The main limitation of observational data is their inability to draw reliable causal inference due to the potential presence of time-varying confounders and intermediate effects. We briefly remind that, given a covariate $X$, and a response variable $Y$, a *mediating* or *intermediate* variable is a third hypothetical variable, say $Z$, which is influenced by $X$ and in turn causes changes in $Y$, influencing thus the relationship between $X$ and $Y$ (MacKinnon et al., 2012). For example, psychotherapy $(X)$ may result in an increased outcome $(Y)$ because it improves compliance with anti-depressant medication $(Z)$, which in turn acts on the outcome $(Y)$. In contrast, a variable $Z$ is said to *confound* a relationship between a treatment $X$ and an outcome $Y$ if it is a common cause of both the treatment and the outcome, thus causing a spurious association (Pearl, 2000). If the effect of $Z$ on both $X$ and $Y$ is not accounted for, it may appear that there is a relationship between $X$ and $Y$ when actually their pattern of association may be due entirely to changes in $Z$. For example, the hypothesis that drinking coffee $(X)$ causes heart disease $(Y)$ may be explained by another factor. Coffee drinkers may smoke more cigarettes $(Z)$ than non-coffee drinkers, so smoking is a confounding variable in the study of the association between coffee drinking and heart disease. The increase in heart disease may be due to the smoking and not the coffee. Actually, recent studies have shown coffee drinking to have substantial benefit in heart health (Stevens et al., 2021).

Typical methods employed in observational settings include *G-estimation* (Robins, 2004, 1994, 1989) and *inverse probability of treatment weighting* (IPTW) of

marginal structural models (Rosthøj et al., 2006; Cotton and Heagerty, 2011; Moodie et al., 2009; van der Laan and Petersen, 2007b; Robins et al., 2008), in which the study of confounders and their potential risk of affecting future treatments plays a central role. Indeed, most DTR research in statistics has concentrated on how best to use observational data to make causal inferences, as it can be particularly tricky and relies critically on all the (unverifiable) assumptions discussed in Sect. 15.2.1.

More recently, a relevant number of studies proposed the use of *deep learning* models (Liu et al., 2017a; Raghu et al., 2017; Atan et al., 2018; Liu et al., 2019), which have the potential to automatically perform *feature extraction* and deal with the high dimensionality in the observational setting. Nevertheless, despite the causal inference limitation, observational data may better reflect the heterogeneity of both patient populations and treatment implementation. Therefore, this data source may represent actual treatment practice better than trial data (Mahar et al., 2021). Notably, some authors suggest that optimal DTR-based treatment decisions should be estimated using observational data, where possible, before proceeding to a randomized design stage (Chakraborty and Moodie, 2013; Wallace and Moodie, 2014), which may be neither feasible nor ethical. However, given this high-dimensional setting (in both number of treatments and patient information), standard statistical methods are difficult to implement or require a relevant simplification (using domain knowledge) in the number of stages and actions.

All the mentioned statistical strategies will be formally covered in depth later in Sect. 15.4. Now, we present two study examples for evaluating optimal DTR based on an observational data source: considering the same registry data, the first study applies common DTR modelling techniques, which uses more traditional statistics (regression), while the second example considers deep learning methods.

#### 15.3.1.1 The CIBMTR Registry: Two Study Examples for Constructing DTRs with Observational Data

The Center for International Blood and Marrow Transplant Research (CIBMTR) maintains one of the world's largest observational databases of clinical information on hematopoietic cell transplantation (HCT), including nearly all allogeneic transplants and approximately 80% of the autologous transplants performed in the USA. Information on the CIBMTR's history, structure, and all the variables collected can be found in Horowitz (2008). These data are available for clinical decision making, and research purposes, and have been used by many researchers for DTR construction.

*Example 1* For instance, in the context of acute graft-versus-host disease (GVHD), a frequent complication of allogeneic hematopoietic cell transplantation (AHCT), Krakow et al. (2017) proposed a DTR for immunosuppressive management for maximizing disease-free survival 2 years post-AHCT. In AHCT, immunosuppressive therapeutics are often administered sequentially to prevent and (if needed) treat GVHD. Because of limited monetary resources, logistical

challenges, and the heterogeneity (yet relative scarcity) of patients who develop the most severe post-transplant complications, there is a dearth of large RCTs for guiding practice. Thus, registry data have the potential to allow the exploration for developing precision medicine approaches. Krakow et al. (2017) used Q-learning, a widely used reinforcement learning (RL) algorithm (traditionally based on a regression model), plus other statistical tools, for evaluating the construction of optimal DTR. In doing this, they simplified the problem to avoid computational complexity and model instability as follows. First, their analysis carefully defined and focused on 2 stages of treatment only: (1) first-line GVHD prophylaxis and (2) second-line salvage or treatment for persons who developed GVHD and experienced unsuccessful first-line treatment. Second, they reduced the action space to two treatment options at each stage: $A_1 \in \mathcal{A}_1 = \{a_{1,1} =$ "nonspecific, highly T-cell lymphodepleting" (NHTL) prophylaxis; $a_{1,2} =$ "standard" prophylaxis$\}$; then, if GVHD that requires salvage, $A_2 \in \mathcal{A}_2 = \{a_{2,1} =$ NHTL salvage; $a_{2,2} =$ "standard" salvage$\}$. Finally, all covariates were tested for interaction with the treatments and included in the model when necessary, and important domain variables were a priori added in order to account for confounding. Thus, several considerations for transparently developing a DTR using non-randomized data with "standard" statistical approaches need to be made:

- **Pre-analysis considerations:** (1) establish clear inclusion/exclusion criteria that should match when a training and validation cohort is used; (2) determine how lost to follow-up patients are handled; (3) determine how missing data are handled; (4) determine the choice and form of the model(s) and of the variables to be entered into the model; (5) clearly define the stages for the multi-stage decision making.
- **Analysis considerations:** define the step-by-step analysis process, e.g., the iterative process of a Q-learning algorithm.
- **Post-analysis considerations:** (1) assess model fit within the current data set; (2) assess model stability (for example, describe the stability of recommendations across sets of patients who share similar salient covariates); (3) evaluate whether the study is adequately powered; (4) assess robustness and external validity (with a validation data set).
- **Clinical translation:** (1) assess whether the projected magnitude of benefit (and harms or costs) from implementing the DTR on a population level would justify moving forward with its implementation; (2) prospectively test the developed DTR in a randomized trial (if the DTR shows statistical validity and clinical worth).

*Example 2* Taking into account the same registry, context, and problem, Liu et al. (2017a, 2019) applied a different statistical framework, i.e., deep reinforcement learning, for obviating some of the pre-analysis considerations required in the previous example (Krakow et al., 2017). The framework is indeed particularly suitable for (i) automatically extracting and organizing the discriminative information

from the data, and (ii) exploring the high-dimensional action and state spaces and make personalized treatment recommendations. Thus, it has a distinctive feature compared to traditional statistical and reinforcement learning techniques to be highly scalable for large state spaces (without requiring a model selection) and action space (that requires, however, to be enumerable). In the actual CIBMTR registry action space, the GVHD prophylaxis contains 127 drug combinations (of 14 drugs), and the 100-day acute GVHD treatment contains 283 drug combinations (of 18 drugs); in addition multiple stages are present. The authors considered 5 stages, include in the deep model all the contextual information, and used only actions with highest probabilities, since actions with small probability have too small number of samples in the observational medical data sets. While their approach requires less data pre-processing and are able to provide great exploratory evidence to generate new hypotheses for subsequent research, it cannot directly dictate treatment to new patients, due to lack of interpretability. As Zhang et al. (2018) points out, an estimated treatment regime that is interpretable in a domain context may be of greater value than an unintelligible treatment regime built using "black-box" estimation methods (as deep learning models).

### *15.3.2 Sequentially Randomized Studies*

As discussed so far, observational data offer a cost-acceptable option and reflect the population's heterogeneity, but they also present several challenges that make estimation non-trivial and often subject to various hidden biases. Hence, randomized data, when available, are preferable for more accurate estimation and stronger statistical inference (Rubin, 1974; Holland, 1986; Rosenbaum, 1991). This is especially important when dealing with DTRs since the hidden biases can compound over stages. Randomized trials are the "gold standard" in study design, as randomization coupled with compliance allows causal interpretations to be drawn from statistical association. However, the scope of usual randomized controlled trials is to evaluate or confirm the efficacy of newly developed treatments, not for developing treatment regimens per se. In addition, they are not effective when there are two or more decision times since a sequence of randomizations is needed to best infer the optimal treatment sequence (Chakraborty and Moodie, 2013).

A special class of randomized designs, tailor-made for the purpose of developing optimal DTRs, is represented by *sequential multiple assignment randomized trial* (SMART) designs (Lavori and Dawson, 2004; Murphy, 2005a). By far, SMARTs are the most effective designs in these multi-stage medical settings, providing the highest-quality evidence of regimen efficacy by reducing confounding bias through randomization. However, SMARTs are more complex to design and implement than standard trial designs and, therefore, are resource intensive.

A SMART design is characterized by multiple stages of treatment, each stage corresponding to one of the critical decision time point in which a randomization

may occur. At each subsequent stage, re-randomizations may depend on information collected after previous treatments, but prior to assigning the new treatment, e.g., how well the patient responded to the previous treatment. Based on the extent of multiple randomizations, different types of SMARTs can be defined. These include SMARTs in which only non-responders are re-randomized and SMARTs in which both responders and non-responders are re-randomized. In addition, randomization could be made only to one of the initial treatment or to all the initial treatments. A more thorough discussion is given in Lei et al. (2012). Independently on the randomization process of a SMART, each randomization must be ethically acceptable, meaning that, given the history, the treatments or actions among which the patient is being randomized must be equally desirable. This criterion, applied at each stage, is the same as the usual requirement of equipoise in conventional randomized trials (Thall, 2015). If the goal of a study is to evaluate multi-stage DTRs rather than individual treatments (by using a SMART design), it is essential to define in advance the actual regimes that will be studied, ensuring these are viable (Wang et al., 2012). Practical considerations for designing such trials were discussed in Lavori and Dawson (2004), while several examples are proposed in Lei et al. (2012). In this work we discuss a more recent example in the context of weight loss management.

#### 15.3.2.1 The SMART Weight Loss Management Study

In order to make the discussion more concrete, we now report the context and the schematic of a recent SMART for weight loss management developed in Pfammatter et al. (2019). The study addresses two main primary aims: (1) to determine the optimal first-line weight loss treatment for a population of adults with obesity, and (2) to determine the optimal treatment augmentation tactic for early non-responders. For achieving this, the study is designed as follows. At program entry, all individuals are uniformly randomized to one of two first-stage interventions: either mobile app alone (App) or mobile app combined with coaching (App + Coaching). Participants achieving in 12 weeks $< 0.5$ lbs weight loss on average per week are classified as non-responders and re-randomized to one of two second-stage augmentation tactics: either modest augmentation, which consists of adding a supportive text messages (TXT), or vigorous augmentation, consisting in adding a TXT combined with Coaching or meal replacement (MR). Re-randomization following non-response occurs only once per participant. Responders continue the initial treatment option, and weight is assessed for all individuals at baseline, 3, 6, and 12 months, with weight change from baseline to 6 months being the primary outcome. A schematic of this SMART design is presented in Fig. 15.1.

Because different subsequent intervention options are considered for responders (continue) and non-responders (modest vs. vigorous augmentation), response status is embedded as a tailoring variable in this SMART by design. Such multi-stage restricted randomizations give rise to several DTRs that are embedded in the SMART. These allow the investigator to estimate the values or utility of the regimes

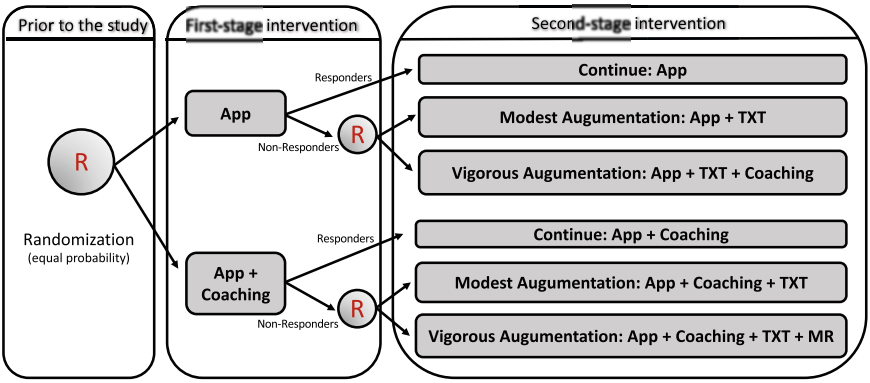


**Fig. 15.1** Schematic of the design of the sequential multiple assignment randomized trial (SMART) Weight Loss Management Study. App denotes a mobile app, TXT a supportive text message and MR meal replacement. Response is defined as a weight loss of at least 0.5 lb on average per week

"embedded" in the study (these are also known as *exploration policies* in the RL literature). For instance, one of the secondary aims of the SMART Weight Loss study is to determine the optimal sequence of treatment tactics by comparing effects on 6 month weight loss and cost-effectiveness. This can only be achieved by comparing the four treatment sequences embedded in the SMART design. For more details on embedded regimes through SMARTs, we point to Chakraborty and Moodie (2013).

The SMART example discussed above involves two stages of treatment and/or experimentation. In general it may involve as many as wanted. In this regard, it bears similarity with some other common designs, including the *adaptive designs*, widely discussed in Berry (2001, 2004), and more recently in Bhatt and Mehta (2016) and Burnett et al. (2020).

Adaptive trial designs have been proposed as a means to improve the quality, speed, cost, and efficiency of randomized clinical trials by modifying one or more aspects of a trial based on interim data (Bhatt and Mehta, 2016). The fundamental characteristic of an adaptive clinical trial is to dynamically adjust the design of the ongoing trial while more patients are enrolled. For example, one may prematurely stop the trial due to safety, futility, and/or efficacy (i.e., *group sequential designs*, Jennison and Turnbull, 2000, 2013) or may change the randomization probability in order to allow the allocation of a higher number of participants to the potential best treatment (i.e., *response-adaptive randomization (RAR) designs*, Rosenberger and Lachin, 2015). While both adaptive designs and SMARTs involve multiple stages at which a change in the design (e.g., randomization) occur, an important distinction exists. In a SMART design, each subject moves through multiple stages of treatment, while in adaptive designs each stage involves different subjects (e.g., in a SMART the same patient is randomized multiple times, while in RAR designs,

it is randomized only once with a probability that depends on data collected on previous patients). Thus, SMART designs involve *within-subject* adaptation of treatment, while adaptive designs involve *between-subject* adaptation. A SMART design can thus be employed for developing DTRs that could benefit future patients (not participants of the current trial), while RAR designs try to provide the most efficacious treatment to each patient in the trial. In addition, in a SMART, unlike in an adaptive design, the design elements such as the final sample size, randomization probabilities, and treatment options are pre-specified.

Other design similarities could be found with classical *crossover trial designs* and *factorial designs*. See Chakraborty and Moodie (2013) and Kidwell (2015) for a discussion on the main differences between SMARTs and these types of design, as well as for a thorough descriptions and characterization of SMARTs in general. More recently, there have also been proposed some works for incorporating adaptive design elements into the SMART design framework (see e.g., Cheung et al., 2015).

### *15.3.3 Dynamical Systems Models*

An indirect approach to constructing optimal DTRs, by providing a basis for improving the design and implementation of a regime, is to use a tool from control engineering, known as dynamical systems models (Ogunnaike and Ray, 1994; Seborg et al., 2016). By dynamical systems models we mean a multivariate time-varying process, in which changes to input variables (some of which can be manipulated) lead to changes in output variables that affect outcomes of interest. The idea of this approach is, first, to develop a dynamical systems model, which can then be used to build an artificial data set and finally to employ algorithms from control theory, such as *dynamic programming* (DP) or constrained optimization algorithms to construct an optimal DTR (Rivera et al., 2007). To develop such a model in an attractive way, aligning its process with biological, behavioral, or social theories, one may use observational or sequentially randomized data sets, or, alternatively, experts' opinion. See, for instance, Bennett and Hauser (2013), in which a framework for simulating clinical decision making from electronic medical records data is proposed.

While dynamical system models constitute a common approach in engineering, economics, and business, their use has now spread to areas of the behavioral and medical sciences as well, with time-varying treatment regimes representing a key example. In Thall et al. (2007a) a Bayesian framework is adopted in simple, low dimensional problems, while Rosenberg et al. (2007), Banks et al. (2011), and Kwon et al. (2014) discuss the use of ordinary differential equations for building dynamical systems models in the context of AIDS treatment. Within behavioral sciences, Rivera et al. (2007) and Navarro-Barrientos et al. (2011) show how dynamical systems models might be used to describe behavioral dynamics and thus form the basis for DTRs in obesity and addiction treatment, and, more recently, the value

they may provide in evaluating effective smoking cessation treatments (Bekiroglu et al., 2017).

To briefly illustrate the benefits of dynamical systems, let us begin with a regime aimed at substance use treatment (a more complex example is discussed later in this section). In such a case, the outcome (substance use) varies over time and is influenced by numerous time-varying variables, e.g., stress, or, most notably, the assigned treatment. Differently from variable "stress" (intrinsic characteristic of the individual), the treatment can be decided by the researcher. Thus, this variable can be manipulated in a dynamical model for understanding how changes in this variable may influence changes in the outcome of interest. The main question in DTR literature is in fact how to choose a DTR so as to optimize the outcome. A complex dynamical systems model, in conjunction with adequate algorithms and computer simulations, may give support in answering this question and improve the DTR construction process. Indeed, a control design with subsequent computer simulations provides a means for exploring various scenarios, by varying not only the decision rules, but also other design variables or characteristics of the experiment participants, in order to investigate the likely effects on key outcomes. Results provided by these extensive simulations offer valuable information that can be used to choose decision rules and other aspects of the design so as to optimize the intervention (Rivera et al., 2007). As in the case of observational studies, the resulting optimized intervention can then be evaluated in a randomized clinical trial or SMART.

#### 15.3.3.1 A Dynamical Systems Model for Behavioral Weight Change

The example we report here is developed in Navarro-Barrientos et al. (2011) in the context of obesity and weight loss. The primary goal of their work is to improve the understanding of behavioral weight change interventions by expressing these as dynamical systems. More specifically, they develop a dynamical system for daily weight change incorporating both a physiological and a psychological dynamical aspect. The former, which we name "energy balance model," describes the net effect of energy intake from food minus energy consumption (physical activity). This three-compartment model was validated using data from the *Minnesota Semi-Starvation Experiment* (Keys et al., 1950). The latter focuses on behavior, explaining how intentions, subjective norms, attitudes, and other system variables that may be impacted by an intervention, can result in healthy eating habits and increased physical activity over time. The authors use the widely accepted *Theory of Planned Behavior* (TPB) for this model (Ajzen and Madden, 1986), which we call "behavioral model." The general conceptual diagram for the integrated dynamical system model is reported in Fig. 15.2.

As suggested earlier, a key benefit of dynamic modelling is to evaluate how the outcome of interest (weight change) responds to changes in intervention or other input variables (e.g., intervention dosages, exogeneous influences) over time. In addition, these kind of mathematical models can be used to answer questions

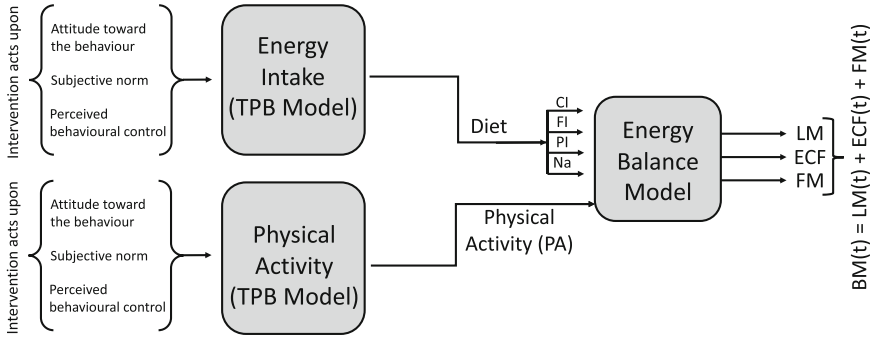


**Fig. 15.2** General input-output block diagram representation of the dynamical system model for body mass. It is conceived as a combination of two theoretical models, i.e., the "energy balance model" and the "behavioral model," based on the *Theory of Planned Behavior* (TPB) model, that can be decomposed in an "energy intake" and a "physical activity" part. Primary inputs to the overall model consist of interventions that act upon components of the TPB model. The latter influence the "diet" (comprised of carbohydrate intake ($CI$), fat intake ($FI$), protein intake ($PI$) and sodium intake ($Na$)) and "physical activity," which in turn, determines the outcome components. The output compartments consist of "lean mass" ($LM$), "fat mass" ($FM$), and "extracellular fluid" ($ECF$), whose sum gives as the "body mass" ($BM$)

regarding what variables to measure, how often, and the speed and functional form of the outcome responses as a result of decisions regarding the timing, spacing, and dosage levels of intervention components (Navarro-Barrientos et al., 2011). Indeed, by defining the functional form of the outcome in terms of its inputs and intermediate components, it is possible to calculate how small variations of each of the input components would diffuse over the functional form and would affect the outcome. The authors define the weight at time $t$ as the body mass at time $t$ ($BM(t)$) and express $BM(t) = FM(t) + ECF(t) + LM(t)$. Each of the three components depends on the previous components (as illustrated in Fig. 15.2). More particularly, each of them is defined by its own differential equation; for instance,

$$\frac{dFM(t)}{dt} = \frac{(1 - p(t))\, f(CI(t), FI(t), PI(t), Na(t), PA(t))}{\rho(FM)},$$

where $FM$ denotes the "fat mass" outcome component, $CI$, $FI$, $PI$, $Na$ and $PA$ the intermediate input components, standing for "carbohydrate intake," "fat intake," "protein intake," "sodium intake," and "physical activity," which are modelled through an appropriate biological function $f$. The term $\rho(FM)$ is a constant equal to 9400 kcal/kg, while $p(t)$ is the $p$-ratio parameter that assigns a percentage of the imbalance to the compartments "fat mass" ($FM$) and "fat-free mass" ($FFM$), respectively (Dugdale and Payne, 1977). As the reader may expect, both "physical activity" and "diet" are determined in turn by the behavioral model through other adequate functional forms based on an underlying behavioral theory. Now, once all the relationships are determined based on a well-established theory, the model

represents a static (i.e., steady-state) system that does not capture any changing behavior over time. In order to include dynamic effects, and generate the dynamical systems description, the authors propose the use of a fluid analogy which parallels the problem of inventory management in supply chains (Schwartz et al., 2006), and the principle of conservation of mass. By analyzing the first (and higher order) derivatives, one can understand through simulations the effects over time of different interventions on an outcome in different participants and estimate a suitable alternative for the participant under study. We point readers interested in the topic to the original paper of Navarro-Barrientos et al. (2011) who developed this dynamical systems model.

## 15.4 Methods for Constructing DTRs

In this section we present a review of the many existing approaches for constructing DTRs, while also affording space to discuss more general issues that relate to the estimation problem. We start with a brief digression on DTRs' origins, within the causal inference literature, and then move to more novel and currently widely employed reinforcement learning based techniques.

### *15.4.1 Origins and Development of DTRs*

The study of dynamic treatment regimes has his origins in the causal inference literature. It was pioneered by Robins (1986, 1997, 1994), with the introduction of *structural nested mean models* (SNMMs) and a number of estimating equation-based methods for finding optimal time-varying treatment regimes. SNMMs, which model the difference in the mean outcomes under different treatment regimes, rather than the full outcome model, were designed for estimating the joint effect of a sequence of treatments in the presence of confounding variables (Robins, 1986). In this setting, standard regression methods, which attempt to estimate causal effects simultaneously are inappropriate, whether or not one adjusts for or conditions on the confounders. Over an extended period of time, three basic approaches for dealing with such confounding were introduced by Robins: the parametric *G-formula* or *G-computation* (Robins, 1986), the *structural nested models* (SNMs), which include SNMMs as a subclass, with the associated method of *G-estimation* (Robins, 1989, 1994) and the *marginal structural models* (MSMs) with the associated method of *inverse probability of treatment weighting* (Robins, 2000). In spite of advantages and strong connections with popular estimation methods, SNMs and G-estimation are not as popular as MSMs and the associated IPW methods; possible reasons are extensively discussed in Vansteelandt et al. (2014).

A number of methods have subsequently been proposed within statistics, including frequentist and Bayesian likelihood-based approaches (Thall et al., 2000, 2002,

2007b). However, all these methods first estimate the data-generation process via a series of parametric conditional models, then estimate the optimal DTRs based on the inferred data distributions. These approaches easily suffer from model misspecification due to the inherent difficulty of modeling accumulative time-dependent and high-dimensional information in the models (Zhao et al., 2015).

In 2003 and 2004, the first semi-parametric methods for estimating the optimal DTR (from longitudinal data) were proposed by Murphy (2003), immediately followed by Robins (2004). Their methods use *approximate dynamic programming* (ADP) techniques and can thus be somehow considered as the first prototypes of reinforcement learning approaches for estimating optimal DTRs. Subsequently, reinforcement learning, previously confined to computer science and control theory, was fully introduced into the DTR literature, with the work of Murphy (2005b), who proposed the well-known Q-learning with function approximation approach (Watkins and Dayan, 1992; Sutton and Barto, 2018), which we will be discussing shortly.

### *15.4.2 Reinforcement Learning: A Potential Solution*

Reinforcement learning (RL), perfectly resembling the sequential decision making problem, represents one of the main current approach for developing DTRs. Generally speaking, RL is an area of machine learning (ML) concerned with determining optimal action selection policies in sequential decision making processes (Sutton and Barto, 2018; Bertsekas, 2019). As introduced in Chap. 2 of this Book, the general framework is based on continuous interactions between a *decision maker* or *learning agent* and the *environment* it wants to learn about. At each interaction stage or time step $t$ the agent receives some representation of the environment's *state* or *context*, $X_t \in \mathcal{X}_t$, which is used for making a decision, or selecting an *action* $A_t$ from a set of admissible actions $\mathcal{A}_t$. As a result, one time step later, the environment responds to the agent's action by making a transition into a new state $X_{t+1} \in \mathcal{X}_{t+1}$ and (typically) providing a *reward* $Y_{t+1} \in \mathcal{Y}_{t+1} \subset \mathbb{R}$. By repeating this process for each $t \in \mathbb{N} = \{1, 2, \dots, \}$, the result is a *trajectory* of states visited, actions pursued, and rewards received.

Using the same notation as in Sect. 15.2, we denote the context, the actions, the rewards, and the histories as $\mathbf{X_t} \doteq (X_1, \dots, X_t)$, $\mathbf{A_t} \doteq (A_1, \dots, A_t)$, $\mathbf{Y_t} \doteq (Y_2, \dots, Y_t)$, and $\mathbf{H_t} \doteq (\mathbf{X_t}, \mathbf{A_{t-1}}, \mathbf{Y_t})$. We assume that these longitudinal histories are sampled independently according to a fixed distribution $P_{\boldsymbol{\pi}}$ given by:

$$p_1(x_1) \prod_{t \geq 1} \pi_t(a_t|\mathbf{h_t})\, p_{t+1}(x_{t+1}, y_{t+1}|\mathbf{h_t}, a_t), \tag{15.5}$$

where:

- $p_1$ is the initial probability distribution specifying the initial state $X_1$.
- $\boldsymbol{\pi} \doteq \{\pi_t\}_{t\geq 1}$ represents the *exploration policy* and it determines the sequence of actions generated throughout the decision making process. More specifically, $\pi_t$ maps histories of length $t$, $\mathbf{h_t}$, to a probability distribution over the action space $\mathcal{A}_t$, i.e., $\pi_t(\cdot|\mathbf{h_t})$. The "|" in the middle of $\pi_t(\cdot|\mathbf{h_t})$ merely reminds that it defines a probability distribution over $\mathcal{A}_t$ for each $\mathbf{h_t} \in \mathcal{H}_t$. Sometimes, the action $A_t$ to take at each time step $t$ is uniquely determined by the history, therefore, $\pi_t$ is a simple function of $\mathbf{h_t}$, i.e., $\pi(\mathbf{h_t}) = a_t$. In other words, policy $\pi_t$ as a step in a sequence of decision rules $\{\pi_t\}_{t\geq 0}$ is an action. We call it *deterministic policy*, in contrast with the *stochastic policy* where an action to take is probabilistically determined.
- $\{p_t\}_{t\geq 1}$ are the unknown *transition probability distributions* and they completely characterize the dynamics of the environment. At each time $t \in \mathbb{N}$, the transition probability $p_t$ assigns to each state-action-reward sequence $(\mathbf{x_{t-1}}, \mathbf{a_{t-1}}, \mathbf{y_{t-1}}) = (\mathbf{h_{t-1}}, a_{t-1})$ of the trajectory up to time $t-1$ a probability measure over $\mathcal{X}_t \times \mathcal{Y}_t$, i.e., $p_t(\cdot, \cdot|\mathbf{h_{t-1}}, a_{t-1})$.

Generally, the reward $Y_{t+1}$ at next time step $t+1$ is conceptualized as a known function of the history $\mathbf{H_t}$ at time $t$, the current action $A_t$ and the next state $X_{t+1}$, i.e., $Y_{t+1} = Y_{t+1}(\mathbf{H_t}, A_t, X_{t+1})$.

The goal of the RL problem is learning an optimal way of choosing the set of actions or learning an *optimal policy*, so as to maximize the expected future return, say $\mathbf{R_t}$, where with the latter we refer to the cumulative sum of immediate rewards, or, more generally, a *discounted* version of it, i.e., $\mathbf{R_t} \doteq Y_{t+1} + \gamma Y_{t+2} + \gamma^2 Y_{t+3} + \cdots = \sum_{\tau\geq t} \gamma^{\tau-t} Y_{\tau+1},\ t \in \mathbb{N}$. If $\gamma = 1$, the return is well defined (finite) as long as the time-horizon is finite, i.e., $t \in [0, T]$, with $T < \infty$; if $T$ is fixed and known in advance, e.g., in clinical trials, the agent faces a *finite-horizon* problem; if $T$ is not pre-specified and can be arbitrarily large (the typical case of EHRs), but finite, we call it *indefinite-horizon* problem. The term *infinite-horizon* problem is used for $T = \infty$.

More formally, denoted with $\mathbf{d_t^*} \doteq \{d_t^*\}_{\tau\geq t}$ the optimal policy at time $t$, the goal is to find $\mathbf{d_t^*}$ such that

$$\boldsymbol{d_t^*} = \arg\max_{\boldsymbol{d_t}} \mathbb{E}_{\boldsymbol{d}}[\mathbf{R_t}] = \arg\max_{\boldsymbol{d_t}} \mathbb{E}_{\boldsymbol{d}}\left[\sum_{\tau\geq t} \gamma^{\tau-t} Y_{\tau+1}\right], \quad \forall t \in \mathbb{N}, \tag{15.6}$$

where the expectation is meant with respect to a trajectory distribution analogous to (15.5), say $P_{\mathbf{d}}$, where the fixed exploration policy $\boldsymbol{\pi}$ that generated the data is replaced by an arbitrary policy $\mathbf{d}$ we use to estimate the data. Indeed, in many decision problems, the *target policy* or *estimation policy* we want to learn about, say $\mathbf{d}$, might be different from the *exploration policy* $\boldsymbol{\pi}$ that generated the data. This happens, for instance, when we use trajectories generated from another trial. The set of decision rules $\mathbf{d} = \{d_t\}_{t\geq 1}$, or policy, is typically referred to as DTR, and each trajectory from the decision process corresponds to the complete history $\mathbf{H_t} \in \mathcal{H}_t$ of

**Table 15.1** Notation and terminology of reference in reinforcement learning (RL) and dynamic treatment regimes (DTRs)

| Notation | Terminology | |
|---|---|---|
| | RL | DTRs |
| $i$ | Trajectory | Patient |
| $t$ | Time | Stage/Interval |
| $X$ | State/Context | Covariates |
| $A$ | Action/Arm | Treatment/Intervention |
| $Y$ | Reward | Outcome |
| $\mathbf{H}$ | History | Time-Varying History |
| $\boldsymbol{\pi}/\mathbf{d}$ | Policy | Dynamic Treatment Regime |

baseline and time-varying covariates, assigned treatments, and observed outcomes of a single patient. Table 15.1 serves as a reference for the correspondence between RL and DTR terminologies.

While several methods exist for policy learning (Sutton and Barto, 2018), by optimal policy we generally mean the one with the greatest *value*, i.e., the greatest expected return by following it when starting from a given state (*state-value* or simply *value*) or a given state-action pair (*action-value* or *Q-value*). Thus, efficiently estimating the value function is one of the most important component of almost all RL algorithms, and it occupies a central place in the medical decision making paradigm.

The stage $t$ *state-value function* or *value function* of a fixed policy $\boldsymbol{d_t}$ maps a starting history $\mathbf{h_t}$ (with terminal state $X_t = x_t$) to the expected return. Formally, $\forall t \in \mathbb{N}$ and $\forall \mathbf{h_t} \in \mathcal{H}_t$, we denote it by $V_t \doteq V_{\boldsymbol{d}_t} : \mathcal{H}_t \to \mathbb{R}$ and define it as

$$V_t(\mathbf{h_t}) \doteq V_{\boldsymbol{d}_t}(\mathbf{h_t}) \doteq \mathbb{E}_{\boldsymbol{d}}\left[\mathbf{R_t} | \mathbf{h_t} = \mathbf{h_t}\right] = \mathbb{E}_{\boldsymbol{d}}\left[\sum_{\tau \geq t} \gamma^{\tau - t} Y_{\tau+1} \middle| \mathbf{h_t} = \mathbf{h_t}\right]. \tag{15.7}$$

To ensure that the conditional expectation in $V_t(\mathbf{h_t})$ is well defined, each history $\mathbf{h_t} \in \mathcal{H}_t$ should have positive probability ($\mathbb{P}(\mathbf{h_t} = \mathbf{h_t}) > 0$). Note that, by definition, at stage $t = 1$, $V_1(\mathbf{h_1}) = V_{\boldsymbol{d_1}}(x_1) \doteq V(x_1)$; while for the terminal stage, if any, the state-value function is 0.

Similarly, we define the stage $t$ *action-value function* for policy $\boldsymbol{d_t}$, also known as *Q-value* or *Q-function*, as the expected return at time $t$, when starting from a history $\mathbf{h_t}$, taking an action $a_t$ and following the policy $\boldsymbol{d_t}$ thereafter. Denoting it by $Q_t \doteq Q_{\boldsymbol{d}_t} : \mathcal{H}_t \times \mathcal{A}_t \to \mathbb{R}$, we have that, $\forall t \in \mathbb{N}$, $\forall \mathbf{h_t} \in \mathcal{H}_t$, and $\forall a_t \in \mathcal{A}_t$,

$$Q_t(\mathbf{h_t}, a_t) \doteq \mathbb{E}_{\boldsymbol{d}}\left[\mathbf{R_t} | \mathbf{h_t} = \mathbf{h_t}, A_t = a_t\right] = \mathbb{E}_{\boldsymbol{d}}\left[\sum_{\tau \geq t} \gamma^{\tau - t} Y_{\tau+1} \middle| \mathbf{h_t} = \mathbf{h_t}, A_t = a_t\right], \tag{15.8}$$

where, analogous to (15.7), $\mathbf{h_t}$ and $A_t$ are randomly selected such that $\mathbb{P}(\mathbf{h_t} = \mathbf{h_t}) > 0$ and $\mathbb{P}(A_t = a_t) > 0$. At stage $t$, the *optimal Q-function* $Q_t^* \doteq Q_{\boldsymbol{d}_t^*}$ and the *optimal value function* $V_t^* \doteq V_{\boldsymbol{d}_t^*}$ for policy $\boldsymbol{d_t}$ are defined as follows:

$$Q_t^*(\mathbf{h_t}, a_t) \doteq \max_{\boldsymbol{d_t}} Q_t(\mathbf{h_t}, a_t), \quad \forall \mathbf{h_t} \in \mathcal{H}_t, \forall a_t \in \mathcal{A}_t, \tag{15.9}$$

$$V_t^*(\mathbf{h_t}) \doteq \max_{\boldsymbol{d_t}} V_t(\mathbf{h_t}) \doteq \max_{a_t \in \mathcal{A}_t} Q_t^*(\mathbf{h_t}, a_t), \quad \forall \mathbf{h_t} \in \mathcal{H}_t. \tag{15.10}$$

Because an optimal state-value function is optimal for any fixed $\mathbf{h_t} \in \mathcal{H}_t$, it follows that the optimal policy at time $t$ must satisfy $d_t^*(\mathbf{h_t}) \in \arg\max_{\boldsymbol{d_t}} V_t(\mathbf{h_t}) = \arg\max_{a_t \in \mathcal{A}_t} Q_t^*(\mathbf{h_t}, a_t)$. A fundamental property of value functions used throughout RL is that they satisfy particular recursive relationships, known as Bellman equations. For any policy $\mathbf{d}$, the following consistency condition, expressing the relationship between the value of a state and the values of successor states, holds:

$$V_t(\mathbf{h_t}) = \mathbb{E}_{\boldsymbol{d}}\left[Y_{t+1} + \gamma V_{t+1}(\mathbf{h_{t+1}}) | \mathbf{h_t} = \mathbf{h_t}\right], \quad \forall \mathbf{h_t} \in \mathcal{H}_t, \forall t \in \mathbb{N}. \tag{15.11}$$

Based on this property and the definitions given in (15.9)–(15.10), at each time $t$, and $\forall \mathbf{h_t} \in \mathcal{H}_t$ and $\forall a_t \in \mathcal{A}_t$, with discrete state and action spaces, the following rules, known as Bellman optimality equations (Bellman, 1965), are satisfied:

$$V_t^*(\mathbf{h_t}) = \mathbb{E}\left[Y_{t+1} + \gamma V_{t+1}^*(\mathbf{H_{t+1}}) \mid \mathbf{h_t} = \mathbf{h_t}\right], \tag{15.12}$$

$$Q_t^*(\mathbf{h_t}, a_t) = \mathbb{E}\left[Y_{t+1} + \gamma \max_{a_t \in \mathcal{A}_t} Q_t^*(\mathbf{h_t}, a_t) \mid \mathbf{h_t} = \mathbf{h_t}, A_t = a_t\right]. \tag{15.13}$$

Here, the expectation is taken with respect to the transition distribution $p_{t+1}$ only, which does not depend on the policy, thus the subscript $\boldsymbol{d}$ can be omitted. This property allows estimation of value functions recursively, from $T$ backwards in time. In finite-horizon *dynamic programming* (DP), this technique is known as *backward induction* and represents one of the main methods in for solving the Bellman equation.

### *15.4.3 Taxonomy of Existing Methods*

Methodology for constructing *optimal DTRs*, i.e., the ones that, if followed, yield the most favorable (typically long-term) mean outcome, is of considerable interest within the domain of precision medicine and comprises a growing body of research in both computer science and statistics (Chakraborty and Moodie, 2013). If from one side, DTRs problems, perfectly resembling the RL design, attracted the attention of ML researchers, from the other side, the necessity of quantifying causal relationships, rather than mere associations, called for the intervention of causal inference community. Indeed, the main challenge in DTRs is that, since the underlying system dynamics are often unknown, inferring the consequences of executing a policy $\mathbf{d} = \{d_t\}_{t \geq 1}$ and understanding the causal effects on an outcome is not immediate.

Most of the current work in DTRs relies on the finite-horizon setting ($T < \infty$, and known in advanced), and the strongly connected *offline learning* procedures. Typically, in finite-horizon problems, estimation of the optimal DTR is obtained from offline data assuming we have access to the collection of observed trajectories for all patients (offline learning). Only recently, the indefinite-horizon setting, particularly suitable for chronic diseases where the number of stages cannot be a-priori specified and can be arbitrarily large, has been addressed by the DTR literature. Note that we use the term "indefinite," and not "infinite," in line with the finite life expectancy of an individual. Generally speaking, there are two fundamental learning mechanisms for deriving optimal policies: *direct* and *indirect methods*. Direct methods seek optimal policies by directly looking for the policy that maximizes an objective (typically the expected return or value function) within a class of policies. On the contrary, indirect methods attempt, first, to estimate a value function, and then to determine an optimal policy based on the learned value function. In the computer science literature, direct and indirect methods are sometimes referred to as *model-free* and *model-based* algorithms (Sutton and Barto, 2018), even if more subtle classifications (Sugiyama, 2015) tend to make a clearer division between the two categories.

Given the rich literature on methods for developing DTRs, before diving into some of those, we provide the reader with a roadmap in Fig. 15.3 that may serve as a guide for moving within this major section.

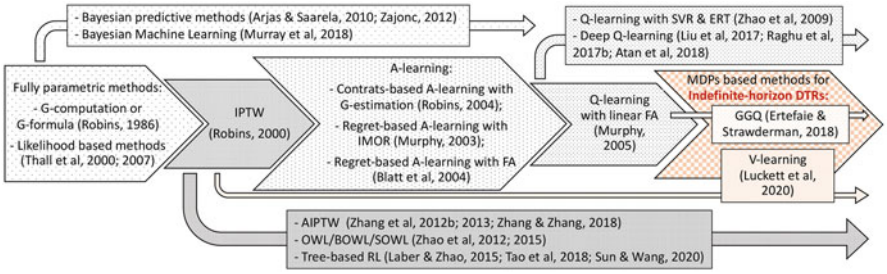


**Fig. 15.3** Schematic of existing methods (in a temporal line) for developing Dynamic Treatment Regimes (DTRs) in both finite and indefinite horizon. Grey colored blocks denote direct methods, i.e., the ones based on Inverse Probability of Treatment Weighting (IPTW); while white dotted blocks denote indirect approaches. AIPTW = Augmented IPTW; OWL = Outcome Weighted Learning; BOWL = Backward OWL; SOWL = Simultaneous OWl; FA = Function Approximation; RL = Reinforcement Learning; SVR = Support Vector Regression; ERT = Extremely randomized Tree; GGQ = Greedy-Gradient Q-learning; MDPs = Markov Decision Processes

### *15.4.4 Finite-Horizon DTRs*

Most of the existing methods in the DTRs literature fall in the finite-horizon setting. These are designed to optimize a utility function over a fixed period of time, say $T$. More specifically, given a finite-horizon trajectory $\mathcal{T} \doteq \{(X_1, A_1, Y_2, \ldots, X_T, A_T, Y_{T+1})\}$, with $X_1$ some pre-treatment information, $X_2, \ldots, X_T$ the evolving information, $A_1, \ldots, A_T$ the assigned treatments, and $Y_2, \ldots, Y_T$ the intermediate and the final ($Y_{T+1}$) outcomes, a sample (or *batch*) of $N$ finite-horizon available patients' trajectories, each of the above form, is used for estimating an optimal DTR $\mathbf{d}^* = \{d_t^*\}_{t \geq 1}$. Throughout this section, we consider deterministic policies, which map histories $\mathbf{h}$ directly into actions or decisions, i.e., $\mathbf{d}(\mathbf{h}) = \mathbf{a}$.

#### 15.4.4.1 Indirect Methods

With indirect methods we refer to a class of methods that focus on estimating an optimal objective function (typically, an expectation of the outcome variable such as the Q-function), and then get the associated policy, rather than directly looking for an optimal policy (e.g., within a class). These methods are mainly based on iterative techniques such as dynamic programming (DP) and approximate dynamic programming (ADP) and include Q-learning (Murphy, 2005b), where the conditional mean outcome is modelled, and other approaches that model contrasts of conditional mean outcomes, for which we use the term *Advantage-learning* (A-learning). The latter has as an example the SNMMs with the G-estimation proposal of Robins (2004). Traditional statistical likelihood-based methods (Thall et al., 2000, 2002), including the parametric G-computation (Robins, 1986) and Bayesian methods (Thall et al., 2007b), also fall into this category. We point to Vansteelandt et al. (2014) and Tsiatis et al. (2019) for readers interested in these traditional approaches.

**Q-Learning with Function Approximation** Q-learning (Watkins and Dayan, 1992) represents one of the most popular (off-policy) *temporal-difference* (TD) approaches (Sutton and Barto, 2018) and probably the most common strategy employed in DTRs research. In particular, a more recent version of Q-learning, i.e., *Q-learning with function approximation* (FA), offers a powerful and scalable tool to overcome both the modelling requirements and the computational burden for solving an RL problem through backward induction or dynamic programming.

The main idea of Q-learning with FA is first, to estimate the Q-functions using an approximator, e.g., regression models, neural networks or decision trees, and then to derive the estimated policy based on the estimated Q-functions. More specifically, we start by assuming an approximation space for each of the $T$ Q-functions in (15.8), e.g., $\mathcal{Q}_t \doteq \{Q_t(\mathbf{h_t}, a_t; \theta_t) : \theta_t \in \Theta_t\}$, with parameter space $\Theta_t$ being a subset of the Euclidean space. According to the results shown in Sect. 15.4.2, estimating an optimal stage-$t$ policy is equivalent to estimating an optimal Q-

function, or in this case, an optimal parameter $\hat{\theta}_t$, i.e.,

$$\begin{aligned}\hat{d}_t^*(\mathbf{h_t}) &= \arg\max_{a_t \in \mathcal{A}_t} \hat{Q}_t^*(\mathbf{h_t}, a_t) \\ &\doteq \arg\max_{a_t \in \mathcal{A}_t} Q_t^*(\mathbf{h_t}, a_t; \hat{\theta}_t) \doteq d_t^*(\mathbf{h_t}; \hat{\theta}_t)), \quad t = 1, \ldots, T.\end{aligned}$$

Then, according to Bellman optimality, we estimate an optimal regime $\hat{\mathbf{d}}^* = (d_1^*(x_1; \hat{\theta}_1), d_2^*(\mathbf{h_2}; \hat{\theta}_2), \ldots, d_T^*(\mathbf{h_t}; \hat{\theta}_T))$ by recursively estimating $Q_t^*$ backwards through time $t = T, T-1, \ldots, 1$. Formally, defining $Q_{T+1}^* \doteq 0$, we proceed as follows:

$$\begin{aligned}Q_T^*(\mathbf{h_t}, a_T; \hat{\theta}_T) &\doteq \hat{\mathbb{E}}[Y_{T+1} | \mathbf{h_t} = \mathbf{h_t}, A_T = a_T] \qquad (15.14)\\ d_T^*(\mathbf{h_t}; \hat{\theta}_T)) &= \arg\max_{a_T \in \mathcal{A}_T} Q_T^*(\mathbf{h_t}, a_T; \hat{\theta}_T) \\ Q_{T-1}^*(\mathbf{h_{T-1}}, a_{T-1}; \hat{\theta}_{T-1}) &\doteq \hat{\mathbb{E}}[Y_T \\ &\quad + \max_{a_T \in \mathcal{A}_T} Q_T^*(\mathbf{h_t}, a_T; \hat{\theta}_T) | \mathbf{H_{T-1}} = \mathbf{h_{T-1}}, A_{T-1} = a_{T-1}] \\ d_{t-1}^*(\mathbf{h_{t-1}}; \hat{\theta}_{t-1})) &= \arg\max_{a_{T-1} \in \mathcal{A}_{T-1}} Q_{T-1}^*(\mathbf{h_{T-1}}, a_{T-1}; \hat{\theta}_{T-1}) \\ &\ldots \\ Q_1^*(x_1, a_1; \hat{\theta}_1) &\doteq \hat{\mathbb{E}}[Y_2 + \max_{a_2 \in \mathcal{A}_2} Q_2^*(\mathbf{h_2}, a_2; \hat{\theta}_2) | X_1 = x_1, A_1 = a_1] \\ d_1^*(x_1; \hat{\theta}_1)) &= \arg\max_{a_1 \in \mathcal{A}_1} Q_1^*(x_1, a_1; \hat{\theta}_1).\end{aligned}$$

We sometimes refer to this procedure as *batch Q-learning*, as learning occurs only after the collection of a set of $N$ trajectories.

Several Q-learning function approximators have been proposed in the literature, with the regression modeling being a natural approach given that Q-functions are conditional expectations. Letting $\theta_t \doteq (\beta_t, \psi_t)$, we can parameterize the $t$-th stage optimal Q-function as

$$Q_t^*(\mathbf{h_t}, A_t; \beta_t, \psi_t) = \beta_t^T \mathbf{H_{t0}} + (\psi_t^T \mathbf{H_{t1}}) A_t, \quad t = 1, \ldots, T, \qquad (15.15)$$

where $\mathbf{H_{t0}}$ and $\mathbf{H_{t1}}$ are two (possibly different) vector summaries of the history $\mathbf{H_t}$, with $\mathbf{H_{t0}}$ denoting the "main effect of history" and $\mathbf{H_{t1}}$ denoting the "treatment effect of history." The collections of variables $\mathbf{H_{t0}}$ are often termed *predictive*, while $\mathbf{H_{t1}}$ are said *prescriptive* or *tailoring variables*. Parameters' estimates $\hat{\theta}_t \doteq (\hat{\beta}_t, \hat{\psi}_t)$ are obtained by solving suitable estimating equations such as *ordinary least squares* (OLS) or *weighted least squares* (WLS). Given a sample $\left\{X_{1i}, A_{1i}, Y_{2i}, \ldots, X_{Ti}, A_{Ti}, Y_{(T+1)i}, X_{(T+1)i}\right\}_{i=1}^N$ of i.i.d. trajectories,

WLS (whose choice might be dictated by heteroscedastic errors) will estimate $\hat{\theta}_t$ by solving

$$0 = \sum_{i=1}^{N} \frac{\partial Q_t^*(\mathbf{H_{ti}}, A_{ti}; \theta_t)}{\partial \theta_t} \Sigma_t^{-1}(\mathbf{H_{ti}}, A_{ti}) \times [Y_{(t+1)i} + \max_{a_{(t+1)i} \in \mathcal{A}_{(t+1)i}} Q_{t+1}^*(\mathbf{H_{(t+1)i}}, a_{(t+1)i}; \hat{\theta}_{t+1}) - Q_t^*(\mathbf{H_{ti}}, A_{ti}; \theta_t)],$$

where $\Sigma_t$ is a working variance-covariance matrix. Taking $\Sigma_t$ to be a constant yields the OLS estimator.

In order for $\hat{\mathbf{d}}^*$ to be a consistent estimator for the true optimal regime $\mathbf{d}^*$, it is important to recognize that all the models for the Q-functions should be correctly specified (Schulte et al., 2014). For addressing this problem, several FA alternatives to the simple linear one in (15.15), such as *support vector regression* and *extremely randomized trees* (Zhao et al., 2009), or *deep neural networks* (Liu et al., 2017a; Raghu et al., 2017; Atan et al., 2018) have been proposed. We now illustrate the latter approximation technique, which has gained a relevant attention in the recent years.

**Deep Q-Learning** The tremendous success achieved in recent years by Q-learning, and more generally RL, in many complex domains has been largely enabled by the use of advanced FA techniques such as *deep neural networks* (Mnih et al., 2015; Jonsson, 2019). We call this approach deep Q-learning (DQL). In DQL, a neural network (Goodfellow et al., 2016) is used to approximate the Q-function. More specifically, at each time $t$, a DNN is used to fit a model for the Q-function in a supervised way: states and actions $\{(\mathbf{H}_{t,i}, A_{t,i})\}_{i=1,\dots,N}$ are given as inputs, and the Q-values of all possible actions are generated as outputs $\{Q_t(\mathbf{H}_{t,i}, A_{t,i}; \hat{\mathbf{W}}, \hat{\mathbf{b}})\}_{i=1,\dots,N}$, leading to a labelled set of data $\mathcal{D} = \{(\mathbf{H}_{t,i}, A_{t,i}), Q_t(\mathbf{H}_{t,i}, A_{t,i}; \hat{\mathbf{W}}, \hat{\mathbf{b}})\}_{i=1,\dots,N}$. $\mathbf{W}$ and $\mathbf{b}$ represent the unknown *weight* and *bias* parameters of the DNN, respectively. Figure 15.4 shows a schematic of a *feed-forward neural network* (FFNN) used within RL. It is characterized by a set of neurons, structured in layers, where each neuron processes the information forward from one layer to the next one. Collected data $\mathcal{D}$ is stored and continuously updated by the user in memory for updating Q-function parameters' estimates. Next action is determined by an exploration scheme (typically $\epsilon$-greedy) which probabilistically chooses between the action with the highest Q-value and a random action. For updating the Q-network, we minimize a loss function, generally the MSE between our target Q-value and our current Q-output, and this is efficiently done by a technique known as back-propagation or stochastic gradient descent (Goodfellow et al., 2016).

Within the DTR literature, DQL implementations for estimating optimal regimes have been proposed in Liu et al. (2017a) and Raghu et al. (2017), for the graft-versus-host disease after transplantation and sepsis treatment, respectively. Both works use observational medical data and are built on the DQN developed in Mnih

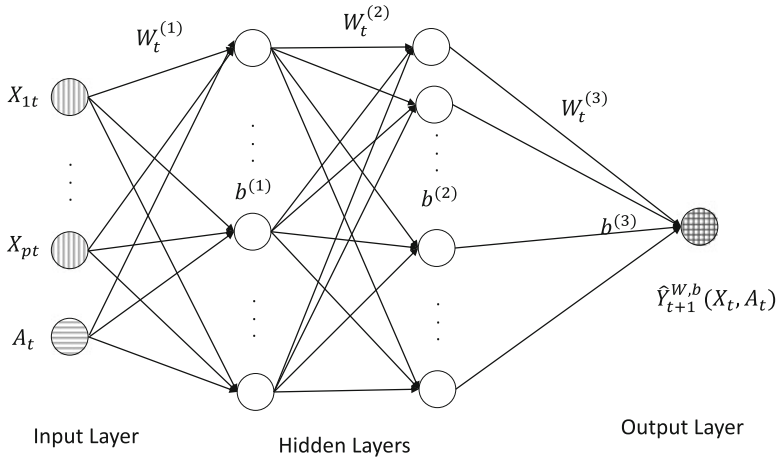


**Fig. 15.4** Representation of a feed-forward neural network with four layers used within Q-learning. In the first (input) layer, we introduce our input data, covariates $X_{1t}, \ldots, X_{pt}$ and treatment $A_t$ at time $t$, which are non-linearly transformed according to their weights $W^{(1)}$ and a bias parameter $b^{(1)}$ through the neurons of the first hidden layer. The final (output) layer, generates the predicted outcome value (reward) $\hat{Y}_{t+1}^{\mathbf{W},\mathbf{b}}(X_t, A_t)$, with $\mathbf{W}$ and $\mathbf{b}$ representing the parameters of the deep neural network

et al. (2015). More recently, Atan et al. (2018) proposed a more sophisticated approach for constructing effective treatment policies when the observed data is biased and lacks counterfactual information. Here, the problem is separated into two stages: first the bias is reduced by learning a representation map using an auto-encoder architecture (Goodfellow et al., 2016) for the neural network, then a FFNN is used on the transformed data to estimate an optimal DTR. An alternative DRL approach was also proposed in Wang et al. (2020), who, rather than using a supervised learning, illustrated the use of a more recent DL architecture, namely the adversarial networks (Goodfellow et al., 2016).

As already mentioned in Sect. 15.3, these deep structures allow for model flexibility and process features without the need of domain knowledge, being particularly suitable for real-life complexity, high dimensionality, and adaptivity. Compared to their shallow counterpart, they are more capable of automatic feature representation and capturing complicated relationships. However, one general limitation of indirect methods such as Q- and A-learning (which we will discuss shortly), independently on the FA, is that the optimal DTRs are estimated in a two-step procedure: one estimates either the Q-functions or the contrast/regret functions using the data; then these functions are either maximized or minimized to infer the optimal DTR. In the presence of high-dimensional information, even with flexible non-parametric techniques such as SVR and DL, it is possible that these conditional functions are poorly fitted, and thus the derived DTR may be far from optimal. Moreover, this approach may not necessarily result in maximal long-term clinical benefit, as

demonstrated by Zhao et al. (2012a) who shifted to parameterize and estimate the treatment rule directly.

**A-Learning with Function Approximation** A-learning (Murphy, 2003; Robins, 2004; Blatt et al., 2004), where "A" stands for the "advantage" in response incurred if the optimal treatment were given instead of the one actually given, is a general term used to describe a class of alternative methods to Q-learning, predicated on the fact that the entire Q-functions need not to be specified to estimate the optimal regime. Models can be posited only for parts of the expectation involving contrasts among treatments, as opposed to modeling the conditional expectation itself as in Q-learning. Recalling that $\mathbf{d}^* \doteq \{d_t^*\}_{t=1,\dots,T}$ denotes the optimal DTR, and denoting with $\underline{\mathbf{d}}_t^* \doteq \{d_\tau^*\}_{\tau=t,\dots,T}$ the optimal regime from $t$ onwards, $\mathbf{d}^{\text{ref}} \doteq \{d_t^{\text{ref}}\}_{t=1,\dots,T}$ a regime of reference we want to make comparisons with, and with 0 the "zero-treatment" (standard or placebo), popular contrast examples include:

$$g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, a_t, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right) - g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, d_t^{\text{ref}}, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right), \tag{15.16}$$

$$g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, a_t, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right) - g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, 0, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right), \tag{15.17}$$

$$g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, a_t, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right) - g\left(\mathbb{E}[Y_{t+1}^{\mathbf{a_{t-1}}}, d_t^*, \underline{\mathbf{d}}_{t+1}^* | \mathbf{h_t} = \mathbf{h_t}]\right), \tag{15.18}$$

where $g(\cdot)$ is a known *link function*, typically taken to be the identity link. *Optimal blip-to-reference* in (15.16) and *optimal blip-to-zero* in (15.17) evaluate removal of an amount ("blip") of treatment at stage $t$ on the subsequent average outcome, when optimal treatment regime $\mathbf{d}_{t+1}^*$ is followed from $t+1$ onwards: "blips" are represented by the treatment of reference $d_t^{\text{ref}}$ and the "zero-treatment," respectively. These are used in Robins' work (Robins, 2004), in which G-estimation was introduced. On the other side, the *regret* function in (15.18), proposed by Murphy (2003), evaluates the increase in the benefit-to-go that we forego by making decision $a_t$ rather than the optimal decision $d_t^*$ at time $t$.

While Robins (2004) advocates optimal blip functions and Murphy (2003) regrets, one can notice that they are mathematically equivalent (Moodie et al., 2007). In addition, they both propose SNMM-type parameterization of the conditional intermediate causal effects, or contrasts, which, without loss of generality, for $t = 1, \dots, T$, has the form $\gamma_t(\mathbf{h_t}, a_t; \psi_t)$, with $\gamma_t$ a known $(T - t + 1)$-dimensional function smooth in $\psi_t$. For all $\mathbf{h_t}, a_t$, the parameterization requires $\gamma_t(\mathbf{h_t}, 0; \psi_t) = 0$, and typically, is chosen to be such that $\gamma_t(\mathbf{h_t}, a_t; 0) = 0$, so that $\psi_t = 0$ encodes the null hypothesis of no treatment effect. As in Q-learning, to estimate the optimal treatment regime, an approximation space for the $t$-th advantage functions is assumed. However, it is important to note that, while the model formulation is equivalent, the estimation technique differs. In Robins (2004), an optimal DTR, under some assumptions (Chakraborty and Moodie, 2013), is estimated through backward recursive G-estimation; in Murphy (2003) a technique known as *iterative minimization of regrets* (IMOR) is proposed. We point to original authors' works (Robins, 2004; Murphy, 2003) for readers interested in these approaches, and to

the more general work of Schulte et al. (2014) for a comparison between their approaches and Q-learning. In an extensive two-stage simulation study, Schulte et al. (2014) found that Q-learning is more efficient than A-learning when: (i) all models are correctly specified (nearly twice more efficient estimating the second-stage parameters, and modestly so for first-stage parameters; and (ii) when the propensity model required in A-learning is misspecified. However, if the Q-functions were misspecified, there were values of the parameters for which gains in efficiency exhibited by Q-learning were clearly outweighed by the bias incurred, making A-learning preferable in terms of mean squared error.

**Bayesian Approaches** Several Bayesian methods have been studied and used in practice to identify optimal DTRs (Thall et al., 2007b; Murray et al., 2018; Arjas and Saarela, 2010; Zajonc, 2012; Xu et al., 2016). The majority of these are likelihood-based methods, requiring thus a joint estimation of data trajectory, and then either apply DP or a full numerical search of the action space to identify the optimal DTR. An alternative approach, which bridges the gap between Bayesian inference and existing RL-based DTR approaches, such as Q-learning, was proposed by Murray et al. (2018) with the so-called *Bayesian Machine Learning* (BML). This approach allows both for patient's preferences and physician's expert knowledge to be incorporated in the model, and the flexibility of novel ADP techniques. The BML proposal fits a series of Bayesian regression models (the authors recommend using Bayesian non-parametric regression models), one for each stage, in reverse sequential order. One distinguishing feature of BML is that it treats the counterfactual response variables as missing values, and multiply imputes them from their posterior predictive distribution, which is derived from the previously fitted regression models. A detailed presentation of Bayesian methodologies and the many modeling choices required for a Bayesian estimation of a DTR is beyond the scope of this chapter; however, a great number of resources are available to the interested reader (see, e.g., Chen et al., 2010).

#### 15.4.4.2 Direct RL Methods

Direct methods seek to maximize the return (i.e., the discounted sum of future rewards, see Sect. 15.4.2) by learning the optimal policy or value directly, without involving intermediate quantities such as Q-functions. These methods typically do not assume models for conditional means or other aspects of the conditional distributions of the outcomes; in this sense they are called "non-parametric." However, they may consider a parametrization of the class of policies.

In direct methods, indeed, first a class of policies or regimes $\mathcal{D}$, often indexed by a parameter, say $\psi \in \Psi$, is pre-specified. Then, for each candidate regime $d \in \mathcal{D}$, an estimate $\hat{V}_d = \hat{V}_d(X_1)$ of the corresponding value is obtained. Recall from Sect. 15.4.2 that the value is the mean of the return marginalized over all observations that might be impacted by the treatment; see (15.7). The regime that maximizes this value function represents the optimal treatment regime $d^*$

$$\hat{d}^* \doteq \arg\max_{d \in \mathcal{D}} \hat{V}_d = \arg\max_{\psi \in \Psi} \hat{V}_{d_\psi}. \tag{15.19}$$

For a simple example of a parametric class of policies, consider DTRs that use a suitable summary of the available history (*tailoring variable*) to indicate when to change treatment: if the *tailoring variable* falls below/above a threshold $\psi$, treatment is changed. Another common example is given by the *soft-max* class of functions $\mathcal{D} \doteq \{\pi(a_k|\mathbf{x}, \psi) = e^{-\mathbf{x}^T\psi_k} / \sum_{j=1}^K e^{-\mathbf{x}^T\psi_j} : \psi \in \Psi, k = 1, \ldots, K\}$, where $a_1, \ldots, a_K$ denote the $K$ possible treatments and $\psi \doteq (\psi_1^T, \ldots, \psi_K^T)$ the vector of parameters for the $K$ treatments indexing the class of policies.

Most of the statistical work in this area is based on the IPTW technique (Robins, 1994). It is used, for instance, in estimating MSMs (Robins, 2000) or value functions (Zhang et al., 2012a, 2013); in classification-based frameworks, such us *outcome weighted learning* (Zhao et al., 2012a, 2015; Liu et al., 2018), and in combination with ML approaches, such as decision trees (Laber and Zhao, 2015; Tao et al., 2018; Sun and Wang, 2021).

**Inverse Probability of Treatment Weighting** IPTW is a general technique that can be used in DTRs for inferring causal effects from observational data, under the standard assumptions for causal inference reported in Sect. 15.2.1.

In case of primary analysis of a randomized trial, particularly a SMART design (see Sect. 15.3), often the *target policy* $\mathbf{d}$ we want to learn about corresponds to the fixed *exploration policy* $\boldsymbol{\pi}$ that generated the trajectories: it consists in the randomization probabilities and is known by design. Thus, estimating an optimal regime based on (15.19) is relatively straightforward. In contrast, when this information is not available, as in the case of the most common observational studies, the value function has to be estimated for an arbitrary treatment policy $\mathbf{d}$ using an empirical sample of $N$ trajectories (*off-policy learning*). Making use of the importance sampling technique, which assumes $P_d$ absolutely continuous with respect to $P_\pi$, we change the distribution under which we compute the value function. In doing that, we basically weight our returns according to the relative probability of their trajectories occurring under the target and exploration policies:

$$V_d = \mathbb{E}_d[Y] = \int Y \, dP_d = \int Y \left(\frac{dP_d}{dP_\pi}\right) dP_\pi = \tag{15.20}$$

$$= \int \left(\prod_{t=1}^{T} \frac{\mathbb{I}[A_t = d_t(H_t)]}{\pi_t(A_t|H_t)}\right) Y \, dP_\pi \doteq \int w_{d,\pi} Y \, dP_\pi.$$

Estimating an optimal regime means estimating an optimal value. This is achieved with the mean estimator, i.e., $\hat{V}_d \doteq \mathbb{P}_N \left[w_{d,\pi} Y\right]$, where $\mathbb{P}_N$ denotes the empirical average over $N$ patients' trajectories. This estimator is known to be an unbiased estimator, but its variance is unbounded. To this purpose, to obtain a more stable estimator, the weights $w_{d,\pi}$ are normalized by their sample mean, leading to the IPTW estimator (Robins, 2000)

$$\hat{V}_d^{IPTW} \doteq \frac{\mathbb{P}_N \left[ w_{d,\pi} Y \right]}{\mathbb{P}_N \left[ w_{d,\pi} \right]}. \tag{15.21}$$

When $\pi$ is known (e.g., SMART design), the IPW estimator is consistent. However, it is highly variable due to the presence of the non-smooth indicator functions inside the weights.

An alternative version, which integrates the properties of the IPTW estimator with those of the regression based estimator, assuming models for both the propensity score and the (conditional) mean outcome, is the *augmented inverse probability of treatment weighting* (AIPTW) estimator (Zhang et al., 2013, 2012b; Tao and Wang, 2017; Zhang and Zhang, 2018a), where, with models posited for either Q-functions or contrast functions, a Q-learning or A-learning strategy was combined with the IPTW estimation. By requiring only one of the two models to be correctly specified, it ensures a double robustness property which enjoys protection against model misspecification and performance at least comparable to that of the competing methods.

IPTW represents a basis for other existing direct methods. For instance, it constitutes one of the most common approach for estimating MSMs, introduced in the causal inference literature for controlling for confounding through assigning each participant a weight (Robins, 2000), and it allowed the development of the general framework proposed by Zhang et al. (2012b) and Zhao et al. (2012b), who recast the estimation of the optimal decision rule as a classification problem. We illustrate now this framework and the specific OWL approach (Zhao et al., 2012b), with some of the subsequent developments.

**Outcome Weighted Learning** As an alternative direct approach, Zhao et al. (2012b) reformulated the problem of optimal DTR estimation as a weighted classification problem, with weights retrospectively determined from clinical outcomes (from here "Outcome Weighted Learning"); and proposed to solve it with tools from the ML literature.

In the case of two treatments, expressed as $A \in \{-1, 1\}$, Qian and Murphy (2011) first showed that the problem can be formulated as a weighted 0–1 loss in a weighted binary classification problem, where $d^*$ can be estimated as

$$\begin{aligned} \hat{d}^* &\doteq \arg\max_{d \in \mathcal{D}} \hat{V}_d = \arg\max_{d \in \mathcal{D}} \mathbb{P}_N \left[ \frac{\mathbb{I}[A = d(H)]}{\pi(A|H)} Y \right] \\ &= \arg\min_{d \in \mathcal{D}} \mathbb{P}_N \left[ \frac{\mathbb{I}[A \neq d(H)]}{\pi(A|H)} Y \right]. \end{aligned}$$

However, as solving the problem is hard due to the discontinuous indicator function, Zhao et al. (2012b) proposed to address it with a convex surrogate loss function for the 0–1 loss, which corresponds to the *hinge loss* used for *support vector machine* (SVM) optimization (Hastie et al., 2009). Considering that $d(H)$

can always be represented as $sign(f(H))$ for some suitable function $f$, the corresponding minimization problem proposed by the authors can be given as

$$\hat{f}^* \doteq \arg\min_{f \in \mathcal{F}} \mathbb{P}_N \left[ \frac{Y}{\pi(A|H)} \phi(Af(H)) + \lambda_N ||f(H)||^2 \right], \tag{15.22}$$

where $\lambda_N$ is a tuning penalty parameter that can be chosen via cross-validation, and $\phi(x) \doteq \max(1 - x, 0)$ is the hinge loss.

Although the seminal work of Zhao et al. (2012b) allows the use of different loss functions, the specific settings (non-negative rewards, single stage, binary treatments) opened many problems for its practical employment, some of which have been addressed by subsequent DTR literature (see, for instance, the recent work of Zhang et al., 2020). Among these, Zhao et al. (2015) and Liu et al. (2018) proposed an extension to multiple stages, integrating the OWL estimator with a RL framework.

**Tree-Based Methods** Again, by integrating tools from the ML literature, first Laber and Zhao (2015), in the context of individualized (single stage) treatment regimes, and then Tao et al. (2018) and Sun and Wang (2021) for dynamic regimes, proposed the tree-based approach (Breiman, 2001) for directly estimating optimal DTRs. The underlying idea of Tao et al. (2018) is, first, to define and estimate a *purity*, i.e., a target measure or output which needs to be optimized, and then, to improve the purity with a decision tree. Improvement is performed by splitting a *parent node* into *child nodes* repeatedly, and by choosing a split among all possible splits at each node so that the resulting child nodes are the purest (e.g., having the lowest misclassification rate). The mean outcome (or value function) is used as purity measure, and its estimation is carried out with the IPTW estimator (Robins, 2000), or alternatively a kernel smoother in the case of continuous treatments (Laber and Zhao, 2015), and the AIPTW estimator (Zhang et al., 2012b), respectively. Differently, Sun and Wang (2021) proposed a stochastic tree-based reinforcement learning which uses Bayesian additive regression trees, and then stochastically constructs an optimal regime using a Markov chain Monte Carlo (MCMC) tree search algorithm. In the multi-stage setting, estimation is implemented recursively using backward induction, starting from $t = T + 1$ and using the outcome $Y_{T+1}$ directly.

By combining the properties of a tree-based learning (straightforward to understand and interpret, and capable of handling various types of data without distributional assumptions) with those of the AIPTW (semi-parametric robust estimator), the tree-based approaches are robust, efficient and more interpretable and flexible (compared to the OWL, or the DQL, for instance) in the identification of optimal DTRs.

### 15.4.5 Indefinite-Horizon DTRs

While in computer science, a vast literature on estimating optimal policies over an increasing time horizon exists (Szepesvári, 2010; Sugiyama, 2015), this scenario is rare in the DTR literature. By adopting backward induction, most of the existing methods cannot extrapolate beyond the time horizon in the observed data. However, for some chronic conditions, or those with very short time steps, the time horizon is not definite, in the sense that treatment decisions are made continually throughout the life of the patient, with no fixed time point for the final treatment decision.

To the best of our knowledge, only two proposals (Ertefaie and Strawderman, 2018; Luckett et al., 2020) have been advanced in DTR literature for indefinite-horizon tasks. We now illustrate these methods; they are both developed under a time-homogeneous Markov behavior, and, while the *V-learning* technique of Luckett et al. (2020) directly maximizes the policy (direct RL), the alternative *Greedy-Gradient Q-learning* (GGQ) of Ertefaie and Strawderman (2018) uses indirect methods.

**Greedy-Gradient Q-Learning** The first extension of DTRs estimation in indefinite-horizon problems was introduced by Ertefaie and Strawderman (2018). Motivated by the original GGQ algorithm of Maei et al. (2010), they proposed a generalization of the GGQ imposing a time-homogeneous Markov assumption on the state-action sequences for each subject. Although not imposed by other DTR methods, this assumption exemplifies estimation and inference by working with time-independent Q-functions and optimal regimes, and avoiding the need for backward induction, which has time-horizon limitations.

We adopt similar notation as in the previous sections, with the introduction of an absorbing state $c$ representing, for instance, a death event. We assume that at each time $t$ patients' covariates $X_t$ take values in the state space $\mathcal{X}^\star \doteq \mathcal{X} \cup \{c\}$, with $\mathcal{X} \cap \{c\} = \emptyset$. We remind that in time-homogeneous Markov Decision Processes (MDPs), transition probabilities $\{p_t\}_{t\geq 1}$, states and actions spaces are time-independent. Let also the state and action spaces be finite, with the action space $\mathcal{A}_x$ defined by the covariates' information. $\mathcal{A}_x$ consists of $0 < m_x \leq m$ treatments, with $m$ the total number of available treatments during all the steps. For any $t$ such that $X_t = c$, let $A_x = \mathcal{A}_c = \{u\}$, where $u$ denotes "undefined"; this implies that $p(X_{t+1} = c, A_{t+1} = u | X_t = c, A_t = y) = 1$.

Now, denoted with $\widetilde{T} \doteq \inf\{t > 0 : X_t = c\}$ a stopping time (death, for example), individual trajectories, including also the last final state, will be given by $(X_1, A_1, Y_2, \ldots, X_{\widetilde{T}-1}, A_{\widetilde{T}-1}, Y_{\widetilde{T}}, X_{\widetilde{T}})$. Note that $\mathbb{P}(\widetilde{T} < \infty | X_1, A_1) = 1$, regardless of $(X_1, A_1)$.

Based on these specifications, under the standard causal inference assumptions, one can define the infinite time horizon stage $t$ action-value function for a specified deterministic regime $\pi(\mathbf{h_t}) = \pi(x_t) = \pi(x)$, for $x \in \mathcal{X}$, as

$$Q(x,a) \doteq \mathbb{E}_{\boldsymbol{\pi}}\left[\mathbf{R_t}|X_t = x_t, A_t = a_t\right] = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{\tau=1}^{\infty}\gamma^{\tau-t}Y_{\tau+1}\middle|X_t = x_t, A_t = a_t\right].$$

For estimating an optimal regime, Ertefaie and Strawderman (2018) proposed to estimate the optimal Q-function $Q^*(x,a)$ with linear FA (as illustrated in Sect. 15.4.4.1). Let $Q(x,a;\theta^*)$ be a parametric model for $Q^*(x,a)$ indexed by $\theta^* \in \Theta \subseteq \mathbb{R}^q$, and suppose a linear model with interactions, i.e., $Q(x,a;\theta^*) = \theta^{*T}\psi(x,a)$, with $\psi(x,a)$ being a known $q$-dimensional vector of features summarizing the state and treatment pair. To ensure $Q^*(c,a) = 0$, we also need $\psi(c,a) = 0$. Now, the Bellman optimality equation suggests and motivates the following unbiased estimating function for $\theta^*$

$$\hat{D}(\theta^*) = \mathbb{P}_N\left\{\sum_{t=1}^{T}\left(Y_{t+1} + \gamma \max_{a' \in \mathcal{A}_{X_{t+1}}} Q(X_{t+1}, a';\theta^*) - Q(X_t, A_t;\theta^*)\right)\psi(X_t, A_t)\right\}, \tag{15.23}$$

where $\mathbb{P}_N$ denote the empirical average on $N$ i.i.d. trajectories, and $\psi(X_t, A_t) \doteq \nabla_{\theta^*} Q(X_t, A_t;\theta^*)$.

**V-Learning** In the GGQ method of Ertefaie and Strawderman (2018), the estimated policy is based on the estimating equation in (15.23), which contains a non-smooth max operator that makes estimation difficult without large amounts of data (Laber et al., 2014a), and, depending directly on $\hat{\theta}^*$, it requires modeling the transition probabilities. Motivated by a mobile health application, where policy estimation is continuously updated in real time as data accumulate (starting with small sample sizes), an alternative method, which directly maximizes estimated values over a class of policies, was proposed in Luckett et al. (2020).

Under the same causal inference and time-homogeneous MDP assumptions of Ertefaie and Strawderman (2018), and provided interchange of the sum and integration is justified, the targeted state-value function of policy $d$ in state $x_t$ is

$$V(x_t) = \sum_{\tau \geq t}\mathbb{E}\left[\gamma^{\tau-t}Y_{\tau+1}\left(\prod_{v=1}^{\tau}\frac{d(A_v|X_v)}{\pi_v(A_v|S_v)}\middle|X_t = x_t\right)\right],$$

where $\pi_v$ is and exploration policy, which can be seen as the randomization probability in an RCT, and $d$ an arbitrary policy which we want to learn about.

In light of the Bellman equation in (15.11) for the value function, it follows that, for any function $\psi$ defined on the state space $\mathcal{X}_t$, the state-value function satisfies

$$0 = \mathbb{E}\left[\frac{d(A_t|X_t)}{\pi_t(A_t|S_t)}\left(Y_{t+1} + \gamma V(X_{t+1}) - V(X_t)\right)\psi(X_t)\right],$$

which represents an importance-weighted variant of the Bellman optimality (Sutton and Barto, 2018).

Now let $V(x;\theta)$, with $\theta \in \Theta \subseteq \mathbb{R}^q$, denote a model for $V(x)$. Assuming that $V(x;\theta)$ is differentiable everywhere in $\theta$, for fixed $x$ and $d$, let $\psi(x)$ be the gradient of $V(x;\theta)$, i.e., $\psi(x) \doteq \nabla_\theta V(x;\theta)$, and define the alternative estimating equation function as

$$\hat{\Lambda}(\theta) = \mathbb{P}_N \left[ \sum_{t=1}^{T} \frac{d(A_t|X_t)}{\pi_t(A_t|S_t)} \left(Y_{t+1} + \gamma V(X_{t+1};\theta) - V(X_t;\theta)\right) \nabla_\theta V(X_t;\theta) \right].$$

V-learning only requires modeling the policy and the value function, rather than the data-generating process. In addition, by directly maximizing the estimated value over a class of policies (Luckett et al., 2020) it avoids the non-smooth max operator in (15.23). The developed RL method is applicable over indefinite horizons and is suitable for both offline and online learning.

## 15.5 Inference in DTRs

Statistical inference plays a key role in a wide range of problems arising in DTRs. These include, for instance: (i) comparing two or more pre-specified and/or estimated regimes; (ii) evaluating the performance and potential benefits of an estimated optimal treatment regime; or (iii) identifying key tailoring variables that may matter in making high-quality treatment recommendations. In these problems one can think of inference for mainly two types of quantities: (1) inference for the parameters indexing the theoretically optimal regime, which helps understand the relevance of different predictors on making an optimal decision, and (2) inference for the value function of a regime, either a regime that was pre-specified, or one that was estimated. The latter helps in evaluating alternative regimes and comparing them with a gold standard, i.e., a regime with a maximally achievable expected performance (or value).

Although there exists a rich literature on development and estimation of optimal DTRs, the associated inference problem has received less or only secondary attention, with a main focus on confidence intervals (CIs). An effervescent interest in the topic characterized the novel DTRs literature around a decade ago, highlighting how inference represents an open problem with several technical challenges, with a major one caused by the phenomenon of *non-regularity* (Robins, 2004). Briefly, by non-regularity we mean the lack of locally uniform convergence; as discussed in Lizotte and Tahmasebi (2017), it can be a result of the sampling distributions of corresponding estimators changing abruptly as a function of the true underlying parameters. We refer to Chakraborty and Moodie (2013) for a rich discussion on the topic and the inclusion of several examples. In DTRs, this may occur, for instance, when two or more treatments produce (nearly) the same mean optimal outcome.

We now aim to provide an overview of the different aspects of inference in DTRs. Because the problem involves asymptotic theoretical results, which makes the nature

of its discussion admittedly more technical, our goal is to only introduce, our goal is to only introduce the salient features of inference in DTRs, particularly in presence of non-regularity. All readers interested to cover this topic more in depth may refer to Chakraborty and Moodie (2013), Laber et al. (2014b) and the recent book of Tsiatis et al. (2019).

### *15.5.1 Inference for Parameters Indexing the Optimal Regime*

Standard approaches to perform inference in many statistical problems rely on known asymptotic approximations to the sampling distribution of an estimator for the targeted estimand. However, as introduced above, in DTRs, inference is complicated by the presence of non-regularity. This phenomenon has several practical implications for both of the two statistical inference areas, i.e., estimation (point and interval) and hypothesis testing. First, estimators of these quantities are necessarily non-regular and asymptotically biased (Van Der Vaart, 1991; Robins, 2004; Hirano and Porter, 2012). Second, traditional asymptotic theory for approximating the sampling distributions of non-regular estimators, such as normal approximations or the bootstrap, can be quite poor. Thus, traditional tools may not be used directly to derive reliable CIs or to guarantee desirable properties in the context of hypothesis testing. This means that any inference technique that aims to provide good frequentist properties such as nominal Type-I error rate and/or nominal coverage rate of CIs in small samples has to seriously address the problem of non-regularity.

**Asymptotic Bias** As shown by a large number of authors, presence of asymptotic bias may be indicative of bias in small samples and may influence nominal Type-I error levels in hypothesis testing and coverage rates of CIs (Blumenthal and Cohen, 1968; Casella and Strawderman, 1981; Robins, 2004; Chakraborty et al., 2010; Moodie and Richardson, 2010). Thus, much attention has been given to characterize and reduce it. To cite, Moodie and Richardson (2010) studied the bias problem in the context of the indirect G-estimation, proposing a method called *Zeroing Instead of Plugging In* for reducing it. This is referred to as the *hard-threshold* estimator by Chakraborty et al. (2010), who proposed an alternative version, named *soft-threshold* estimator, in the context of Q-learning. Both techniques were demonstrated to reduce bias in small samples.

In a similar spirit, Song et al. (2015) and Goldberg et al. (2013) proposed minimizing a penalized version of the objective in the first step of a two-stage Q-learning analysis. Indeed, in many indirect methods, first Robins (2004) for G-estimation, and later Chakraborty et al. (2010) for Q-learning, pointed out that the treatment effect parameters at any stage prior to the last can be non-regular under certain longitudinal distributions of the data. For instance, if we consider again the two-stage two-treatment model for the Q-functions proposed in (15.15), we have that the optimal DTR is given by

$$d_t^*(\mathbf{H}_t) = \underset{a_t \in \mathcal{A}_t}{\arg\max}(\psi_t^T \mathbf{H_{t1}}) A_t = sign(\psi_t^T \mathbf{H_{t1}}), \quad t = 1, 2,$$

where $sign(x) = 1$ if $x > 0$ and $-1$ otherwise.

Inference for $\psi_2$, the stage 2 parameters, is straightforward since this falls in the framework of standard linear regression. In contrast, inference for $\psi_1$, the stage 1 parameters, is complicated by the non-regularity resulting from the underlying non-smooth maximization operation in the estimation procedure. More specifically, the inferential problem arises when the quantity $\psi_t^T \mathbf{H_{t1}}$ is close to zero with positive probability (i.e., for at least some subjects with history $\mathbf{H_{t1}}$), as non-differentiable in that point. Under mild assumptions, Laber et al. (2014b) characterized the asymptotic bias of the first-stage estimator, which is indeed non-zero when the second-stage treatment effect has a positive probability of being zero.

More recently, Fan et al. (2019) proposed the *smoothed Q-learning*, dictated by the use of a modified version of $\hat{\psi}_t^T \mathbf{H_{t1}}$ in the above model. This is given by $\left(\hat{\psi}_t^T \mathbf{H_{t1}}\right) K_\alpha(\hat{\psi}_t^T \mathbf{H_{t1}})$, with $K(\cdot)$ a kernel function that admits a probability density function defined as $K_\alpha(x) \doteq K(x/\alpha)$, where $\alpha > 0$ is the smoothing parameter.

**Confidence Intervals** The practical use of optimal DTRs for informing clinical decision making or future research needs to be accompanied by reliable measures of uncertainty. Thus, CIs have received a remarkably central attention in the DTR literature. Indeed, if CIs of parameters associated with some of the tailoring variables included in the statistical models contain zero, then those variables may need not be collected, and, alternatively, the length of a CI may indicate the extent of variability, thus uncertainty around the estimate of an important variable. Such CIs can be useful in exploratory data analysis from observational data when trying to interactively find a suitable model for, say, the Q-functions, before starting a SMART, with consequent improvement of the data collection burden and cost in a future implementation. Thus, CIs can also be viewed as a tool, albeit one that is not very sophisticated, for doing variable selection (Chakraborty and Moodie, 2013).

Notably, estimators and methods mentioned above (Chakraborty et al., 2010; Song et al., 2015; Goldberg et al., 2013) have been originally suggested for constructing high-quality CIs. However, in general, despite the direct connection between CIs and estimators, one should notice that: (1) there is no strict requirement of unbiasedness in the estimators for obtaining CIs that deliver the desired level of confidence and allow for valid inference, and (2) there are no guarantees that an asymptotically unbiased estimator will lead to a high-quality CI. Indeed, bias only reflects the mean of the sampling distribution whereas CIs require estimation of the tails of the sampling distribution; thus, reducing asymptotic bias is not sufficient for having reliable inference. To illustrate, the estimators proposed in Chakraborty et al. (2010), Song et al. (2015), Goldberg et al. (2013), by involving additional non-smooth operations of the data, lead to inconsistent CIs under local alternatives.

Construction of confidence sets for parameters indexing the optimal DTR has received its due attention from a broad DTR literature. Orellana et al. (2010) studied the problem in the context of direct IPTW-based methods; in this case, confidence

sets are based on standard Taylor series arguments, and are asymptotically valid under a set of smoothness assumptions. Even in the case of OWL methods (Zhao et al., 2012b, 2015, 2019; Chen et al., 2016; Zhou et al., 2017; Liu et al., 2018), which originally involved a (non-convex) 0–1 loss function, by replacing the latter with a convex surrogate for solving the optimization problem, an automatic solution for inference was provided as well. Under specific surrogates (some examples are discussed in Tsiatis et al., 2019), the OWL estimator is shown to be consistent, normally distributed and with superior performance as compared to the standard hinge loss in some contexts (Zhao et al., 2019; Jiang et al., 2019).

For indirect methods such as Q-learning, proposals for constructing reliable CIs include the *adjusted projection confidence intervals* of Robins (2004), where a joint CI for all of the parameters (of all stages) is formed based on inverting hypothesis tests; the *m-out-of-n bootstrap* of Chakraborty et al. (2013), where a practically convenient adaptive method for bootstrapping under non-regularity is presented, and the novel locally consistent *adaptive confidence intervals* of Laber et al. (2014b). We refer to Chakraborty and Murphy (2014) for a discussion on these techniques. Alternative proposals include the *interactive Q-learning* method of Laber et al. (2014a), where the maximization step in Q-learning is delayed, enabling the estimation to be performed before the non-smooth, non-monotone transformation; and more recently, the LASSO-based procedure of Jeng et al. (2018). The latter, in the context of A-learning, proposes an asymptotically unbiased estimator and derives its limiting distribution in presence of a high number of covariates and interactions as well.

**Hypothesis Testing** While a broad DTRs literature focusing on estimation (particularly CIs) exists, less attention has been given to hypothesis testing. One plausible reason for this could lie in the primary role of DTRs studies, more specifically SMARTs, in which estimation and inference about an optimal treatment regime is usually a secondary or exploratory analysis intended to generate new hypotheses for subsequent research and not to directly dictate treatment to new patients (Murphy, 2005a; Zhang et al., 2018). Consequently, hypothesis testing around estimation of an optimal regime is not generally central to sample size, as we will also see in Sect. 15.6.

### *15.5.2 Inference for the Value Function of a Regime*

**Estimation Under Non-regularity** Similar to inference for model parameters, inference for the optimal value is challenging due to non-regularity. In general non-regularity arises in the case of *exceptional laws* (Robins, 2004), that is, probability distributions where there exists a strata of history covariates that occurs with positive probability and for which treatment is neither beneficial nor harmful, thus when the probability that the optimal rule is not unique is positive.

A first solution for performing inference for value functions of DTRs was proposed by Zhang et al. (2012b), based on which inference is restricted to a class of regimes indexed by a finite-dimensional vector. At non-exceptional laws, the authors showed that their estimator is (up to a negligible term) equal to the estimator under regularity conditions. In the same regular setting, Wu and Wang (2021) developed a smoothed robust estimator with asymptotically normal distribution, suitable for both model parameters and value function inference. Although the theoretical background is based on regularity conditions, the authors show that their bootstrap CIs for the optimal value function displays a fair degree of robustness when non-regularity occurs.

However, restricting inference to non-exceptional laws is limiting as often a zero-treatment effect may characterize patients of some strata of history variables. Chakraborty et al. (2014) proposed using the *m-out-of-n double bootstrap* to obtain inference for the value of an estimated DTR in a more general setting. When the treatment mechanism is known or is estimated according to a correctly specified parametric model, valid inference could be performed with IPTW or AIPTW (see Sect. 15.4). In non-regular problems, under certain conditions, the *m-out-of-n bootstrap* yields valid inference, however, at the cost of wider CIs.

More recently, Luedtke and Van Der Laan (2016) developed interesting theory for inference for the value function under exceptional laws, and proposed an alternative approach based on an online one-step estimator and split sampling. Although asymptotically valid, the resulting CI for the value function, similar to Chakraborty et al. (2014), can be wide due to using partial sample for inference. For correcting this issue, and at the same time allowing for high number of covariates to be included in the model, Zhu et al. (2019) proposed a hard-thresholding *high-dimensional Q-learning*. This method allows simultaneously estimating the optimal DTR and selecting the variables that have an important contribution to the individual outcome. The asymptotic properties of the optimal value function estimator as well as the parameter estimators are then established by adjusting the bias by thresholding.

**Hypothesis Testing** In SMARTs, hypothesis testing is generally conducted for comparing different embedded DTRs in a trial. Methods have been proposed for both: (i) superiority tests, i.e., testing whether one embedded regime yields better primary outcome on average than another (Nahum-Shani et al., 2012b); and (ii) non-inferiority and equivalence tests, i.e., testing whether one embedded regime yields benefits that are non-inferior or equivalent to those produced by (active) standard of care (Ghosh et al., 2020).

To illustrate, an important scientific question motivating the SMART for weight loss management study (Pfammatter et al., 2019) introduced in Sect. 15.3, concerns the comparison of say $DTR_1$, which recommends initiating treatment with App, and then to augment it vigorously (App + TXT + Coaching) as soon as the individual exhibits early signs of non-response, and continue with App alone as long as the individual is responding; and $DTR_2$, which recommends to initiate treatment with App + Coaching, and then to vigorously augment it (App + Coatching + TXT + MR) as soon as the individual exhibits early signs of non-response, and to continue

with App + Coaching as long as the individual is responsive. The rationale for this comparison relates to cost and burden. Because $DTR_2$ recommends coaching throughout, it is likely to be effective, yet relatively costly and burdensome. $DTR_1$, on the other hand, offers coaching only to those individuals who seem to need it most (i.e., early non-responders), hence it is hypothesized to be non-inferior, namely no less effective than $DTR_2$. If $DTR_1$ is equally or more beneficial in terms of ultimate weight loss compared to $DTR_2$, then the former should be selected for real-world implementation.

Ghosh et al. (2020) discuss the problem of hypothesis testing for non-inferiority and equivalence testing. The authors propose a test statistic based on the means of the embedded DTRs and a pre-specified non-inferiority margin (in the case of non-inferiority tests), considering a single continuous primary outcome. By design, the mean of each DTR is a weighted average of primary outcomes of patients having treatment trajectories consistent with that regime (Nahum-Shani et al., 2012b; Oetting and Levy, 2007). The weighted average derives from a structural imbalance between responders and non-responders; thus, the IPTW method is a natural technique to be used for accounting for this imbalance. The test statistic has asymptotic normal distribution, which can thus be used for rejecting the null hypothesis and performing power analysis.

## 15.6 Practical Considerations and Final Remarks

In the previous sections we focused on the methodological aspects for developing DTRs, including data sources, existing techniques and inferential aspects. These arguments were mainly introduced and studied within the statistical and ML literature, and generally evaluated through simulations. Now, we want to provide a more concrete idea of the study of optimal DTR in real-world settings, as found in clinical literature. At the same time, we want to illustrate the main practical challenges that clinical researchers face in applying these methods, and some of the proposed solutions, if any.

In Mahar et al. (2021), a detailed overview of how DTRs are optimized with observational data in practice is provided. Using the PubMed database, the authors identified 63 eligible studies, mostly published after 2005 (up to October 2020), and almost half (25, 45%) in the last 5 years), showing thus that the practical field development is quite recent. Identified studies are most concentrated in the chronic disease area: HIV/AIDS (27, 43%), followed by cancer (8, 13%), and diabetes (6, 10%). Common statistical approaches illustrated in Sect. 15.4 were implemented. IPTW-based methods were the most commonly used, followed by parametric G-formula related methods and Q-learning. Yet, there was a lack of transparency regarding some of the specific methodological approaches used across many studies, particularly in relation to either missing data, model evaluation, model selection, or model sensitivity, and only eight studies described all four methodological approaches. The most commonly used statistical software were R

**Table 15.2** Some of the existing R packages for developing Dynamic Treatment Regimes

| R package name | Functions and methods |
|---|---|
| `DynTxRegime` (Holloway et al., 2020) | `owl` (Outcome Weighted Learning; (Zhao et al., 2012a)); `bowl` (Backwards Outcome Weighted Learning; (Zhao et al., 2015); `rwl` (Residual Weighted Learning; (Zhou et al., 2017)); `qLearn` (Q-Learning Algorithm; (Murphy, 2005b)); `iqLearn` (Interactive Q-Learning; (Laber et al., 2014a)); `optimalSeq` (Augmented Inverse Probability Weighting; (Zhang et al., 2012b, 2013)) |
| `DTRreg` (Wallace et al., 2020) | `method = "gest"` (G-estimation; (Robins, 2004)); `method = "dwol"` (Dynamic Weighted Ordinary Least Squares; (Wallace and Moodie, 2015); `method = "qlearn"` (Q-learning; (Murphy, 2005b)) |
| `iqLearn` (Linn et al., 2015) | Interactive Q-Learning (Laber et al., 2014a) |
| `qLearn` (Xin et al., 2012) | Q-Learning (Murphy, 2005b) |
| | GGQ (Ertefaie and Strawderman, 2018)—code in authors' Supplementary material |
| | V-learning (Luckett et al., 2020)—code based on `optim` function |
| | Bayesian Machine Learning (Murray et al., 2018)—code in authors' Supplementary material |

and SAS, with only 21 studies providing the code used for performing data analysis. In Table 15.2, we report a list of existing R packages (and respective functions) associated with the methods illustrated in Sect. 15.4, some of which were also employed by the reviewed papers.

We believe that the lack of reporting methodological details in relation to the above mentioned problems may be caused by either an underestimation of the problem or a lack of existing (recognized) statistical tools for the specific DTR context. Thus, we acknowledge that there is an enormous research-practice gap in the area, and we encourage readers interested in advancing methodological tools to support this endeavor to advocate strongly to domain science collaborators. In what follows we want to report some of the few existing work in the field for which enormous space for improvement exists. This may help both clinicians in carrying our estimation of optimal DTR more transparently, and methodological researchers in developing and/or improving existing techniques with specific considerations about the DTR area.

### *15.6.1 Model Choice and Variable Selection*

A first practical challenge when estimating optimal regimes is related to the choice of the statistical model, including the form of dependency of the outcome variables

on the independent covariates (e.g., linear vs non-linear) and all the relevant patient information that should be considered (and their relationship; e.g., interactions). Most of the methods discussed in Sect. 15.4 focus on estimation, and implicitly assume that the models upon which they rely are correctly (or over-) specified. Thus, they are not designed for variable selection with the objective of optimizing treatment decisions. In clinical trials and more important in observational studies, numerous variables are collected and variable selection is essential for guaranteeing stability end reliability. The alternative technique of deep learning, as discussed in Sect. 15.3, may overcame this problem and automatically perform variable selection. However, the interpretability limitation of the latter represents a major barrier for performing inference and drawing conclusions.

In statistics, the identification of the so-called tailoring variables can be viewed as a problem of variable selection. However, despite the vast literature on variable selection in regression (which may be useful for Q-learning), these methods are largely focused on minimizing prediction error; that is, identifying predictive variables, i.e., those that are important for high-quality predictions. In DTR, in addition to the prediction-quality, the interest is also in identifying prescriptive variables, i.e., those that have a qualitative interaction with treatment and thus are critical for high-quality decision making (Peto, 1982). Thus, in the framework of optimal DTRs estimation, methods incorporating this selection criterion are an important adjunct to standard variable selection methods.

Existing works that focused on variable selection in DTR are limited but cover several interesting DTR frameworks discussed in Sect. 15.4. These include ranking methods (Gunter et al., 2011a), regression based methods (Fan et al., 2016), weighted classification type learning methods (Zhang and Zhang, 2018b), to mention a few. Notably, a few works also proposed formal hypothesis testing procedures that take variable selection into account. Here we briefly illustrate some of the existing methods (Shi et al., 2019; Qian et al., 2021). In the latter, the aim is to identify (among a large set of candidates) covariates that interact with treatment, via a sequential testing procedure. As our focus is on prescriptive variables in general, we now briefly illustrates some of the existing techniques.

One of the first works in the DTR context (Biernot and Moodie, 2010), is motivated by a real-world randomized study, the STAR*D, with a binary outcome (remission) and only discrete covariates. Here, the authors compare some already existing strategies, i.e., the *S-scores* proposed in Gunter et al. (2007) and the *reducts* (Swiniarski and Skowron, 2003), adopted form computer science, showing the unsatisfactory performance of the latter. Other works include Gunter's contributions (Gunter et al., 2011a,b,c), which integrate Lasso, the *Bayesian Information Criterion* (BIC), bootstrap sampling and thresholding, and S-score ranking, leading to a complex and computationally intensive approach, but subsequently also propose a more simple stepwise selection method that surprisingly seems to work well compared to the more complex one. The authors also discuss and define a new method for variable selection that is able to control for the number of false discoveries (Gunter et al., 2011c).

Building on the work of Gunter et al. (2011b), an alternative approach based on a modified S-score is proposed by Fan et al. (2016). Their method selects qualitatively interacted variables sequentially (based on the notion of *sequential advantage*), and hence excludes marginally important but jointly unimportant variables or vice versa. The authors show that this method can handle a large amount of covariates even if sample size is small, and by introducing a stopping criteria that tunes the cut-off can efficiently achieve the goal of maximizing the expected outcome of the treatment regime. This stopping criteria, based on proportion of incremental sequential advantages, is used to decide how many variables to be included for decision making.

In a different framework, i.e., the classification framework, more specifically C-learning (Zhang and Zhang, 2018a), Zhang and Zhang (2018b) propose a new method to select important prescriptive variables (among a high number) for estimating the optimal treatment regimes. The general procedure is based on forward sequentially minimizing the weighted misclassification error rate and, as Fan et al. (2016), in each step it takes into account previously selected variables. In a two-stage decision problem setting, with linear models assumed for the outcome, this novel proposal selects less variables yet with better value than the other methods evaluated in Fan et al. (2016).

An alternative idea, focused on the model choice is presented in Wallace et al. (2019). In model choice the goal is to select a model among a set of candidate models (which may differ in terms of the included variables) based on a certain performance measure. In the context of SNNMs and G-estimation, Wallace et al. (2019) adapt the performance metric of *quasi-likelihood information criterion* (QIC) and compares it with common Wald-type selection methods. The authors assume linear treatment models, and always correctly specified in terms of the proposed linear regression. In their simulation study, comparing eight different blip models corresponding to possible combinations of three predictors at each stage (using none, one, two, or all three), the authors show that QIC, with its guarantee of consistency, performs at least as well than simpler Wald-type approaches for continuous outcomes, particularly when the sample or effect sizes are small, or there is correlation between candidate covariates.

### *15.6.2 Sample Size Considerations and Power Analysis*

Ensuring an adequate power to detect statistical significance, by determining the optimal sample size, is a critical step in the design of a planned research protocol for an experimental trial, such as a SMART. Most of the power analysis and sample size considerations have indeed focused on SMART designs and pilot SMARTs (Almirall et al., 2012), whose primary analysis is generally related to hypothesis-generating analyses such as estimation of an optimal treatment regime. Tsiatis et al. (2019) extensively discuss different ways of performing sample size calculations and empowering a SMART design based on the different goals of the

study. In general, SMARTs can be more sample size efficient than standard RCTs. However, the sample size can drastically inflate with the increase of the number of treatment components.

Depending on the primary analyses of a trial, sample size and power considerations could be more or less straightforward. For example, for many SMARTs, the primary analysis specified in the protocol consists in the comparison of response rates among first-stage treatment options or comparison of fixed treatment regimes embedded in the SMART. In this case the sample size formulae are known or quite simple, and rely on asymptotic theory; while empower of a SMART is based on the Bonferroni correction (Aickin and Gensler, 1996) when multiple comparisons are made. An extensive overview of these computations are provided in Chapter 9.3 of Tsiatis et al. (2019). Alternatively, the original works of Murphy (2005b) and Kidwell and Wahed (2013) illustrate sample size calculation for the terminal outcome comparison of two fixed regimes that are "non-overlapping" or "overlapping," i.e., if regimes differ in the first-stage treatment option or not, respectively. In most cases, these fixed regimes will be embedded in the SMART; however, this does not affect the sample size calculations. The additional works of Ertefaie et al. (2016) and Artman et al. (2020) also give an idea on sample size methodology on identifying an optimal embedded regime(s) from among those represented in the SMART.

Although estimation of an optimal DTR typically is considered a secondary and hypothesis-generating analysis (thus, not factored into sample size calculations), recently the practical interest of precision medicine has placed the estimation of an optimal regimes in the primary analysis of a SMART (Laber et al., 2016). As a result, the need of sample size calculation has brought a surge of attention within the DTR literature. The problem is principally tackled to ensure: (i) sufficient power for detecting a clinically meaningful difference in the values under an optimal treatment regime and under some comparator regime; and (ii) estimated optimal DTR value within a pre-specified tolerance (of that under a true optimal regime) with high probability. In such a setting, because of the non-smooth functional of the underlying generative data model (as illustrated in Sect. 15.5) standard large sample methods for inference cannot be applied without modification or strong assumptions. Two procedures for sample size calculations can be found: (1) normality-based procedures, which rely of assumptions that are sufficiently strong to appeal to standard inference; and (2) non-parametric projection-based methods. We refer again to the general work of Tsiatis et al. (2019) for an overview of formal derivations of sample sizes in this setting; and to the more specific work of Rose et al. (2019) for further discussion of evaluation of performance and implications for practice.

Despite increasing popularity, the concept of DTR is still relatively novel for being used in experimental SMART designs. In addition, there is also a considerably fewer literature, compared to the observational counterpart, on the use and practical application of existing methodologies in this setting. It was thus suggested that researchers could benefit from conducting a pilot SMART to assess the feasibility and acceptability concerns of the DTRs and consequently the full-scale SMART

trial (Almirall et al., 2012; Kim et al., 2016; Yan et al., 2021). Although the primary feasibility and acceptability goal of pilot studies, the authors also proposed different ways to operationalize sample size calculation specifically for the pilot SMART. For example, in Almirall et al. (2012), the team sizes the pilot SMART from an operational perspective, by ensuring a minimal number of participants in the smallest subgroup of a SMART design, so that at least all subgroups in a SMART can be observed. An alternative proposal suggests the use of a precision-based approach that allows the researchers to observe the response or non-response rate, by ensuring the rate is confined within a pre-specified margin of error (Kim et al., 2016). We make it clear that the precision-based sample size does not relates to power (and effect sizes), but is based on the width of the CI (the precision). More recently, this idea was extended by Yan et al. (2021) for different outcome types as well. Their approach has the benefit of information on the marginal mean estimates of the DTR within a meaningful pre-specified margin of error and has performances comparable to those of Almirall et al. (2012).

### *15.6.3 Missing Data*

Missing data and dropout are central problems in almost any healthcare domain when it comes to data analyses. In the context of DTRs, they complicate statistical analysis and results' generalizability because participants who drop out do not experience the entire sequence of decision points of an observational or experimental trial. Specifically focusing on SMARTs, in Shortreed et al. (2014) a multiple imputation technique under the assumption that missingness is at random is proposed, while Liu et al. (2017b) discusses a stage-wise enrichment design for two-stage SMARTs that may address problems of attrition after the first stage. However, in general, there is relatively little literature on methodology for handling missing information in DTRs estimation.

### *15.6.4 Additional Issues and Final Remarks*

The content above highlights and summarizes an increasing progress and interest in the development of DTRs. However, despite remarkable theoretical results, their application in real life is very limited. The majority of existing studies are methodological investigations and typically only tackle a simplified real-world setting, or not even mention a potential setting. This obviously causes a number of practical limitations when it comes to develop DTR in real life. To illustrate, an exciting open problem is related to the formalization of suitable and context specific relationships between patients' covariates, interventions, and outcomes variables. This is particularly true for the outcome function, whose formalization sometimes requires combining multiple and possibly competing outcomes. This task may

only be performed under adequate prior knowledge of the specific domain, e.g., with clinicians, and/or patients' preference, elicitation. See Butler et al. (2018) and Luckett et al. (2021) for a discussion on the topic.

In addition, despite the simpler setting of proposed methods compared to the complex real-world scenarios, many of the proposed methodological strategies may be difficult for clinical readers and researchers to employ and interpret, due to, e.g., complex estimating equations, Monte Carlo simulations, and/or ML tools such as neural networks. While, for instance, some software packages exist for implementing many of the reviewed algorithms (as reported in Table 15.2), they may require both adaptation to the setting under study (e.g., from categorical to continuous outcome variables) and users' knowledge about the specific software. To this end, we believe that user-friendly and readily usable by non-experts software (e.g., *Shiny apps*) would make the real-world implementation easier for applied scientists and improve the real-world development of DTRs, with a consequent practical benefit to patients, clinicians, and healthcare systems.

We conclude by noticing that this chapter aims to provide an overview of the study of DTRs, including its fundamental mathematical framework, the proposed methodologies for developing DTRs (to which a major focus was dedicated), existing data sources, the issue of inference, and other practical considerations. We do not cover in detail the recent advances for all the specific problems, providing only some useful references for interested readers. We hope to endow the reader with a foundation for further study of the expansive literature on this topic, acknowledging that there is an enormous research-practice gap to be bridged.

## References

Aickin, M., & Gensler, H. (1996). Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. *American Journal of Public Health*, *86*(5), 726–728.

Ajzen, I., & Madden, T. J. (1986). Prediction of goal-directed behavior: Attitudes, intentions, and perceived behavioral control. *Journal of Experimental Social Psychology*, *22*(5), 453–474.

Almirall, D., Compton, S. N., Gunlicks-Stoessel, M., Duan, N., & Murphy, S. A. (2012). Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in Medicine*, *31*(17), 1887–1902.

Almirall, D., Nahum-Shani, I., Sherwood, N. E., & Murphy, S. A. (2014). Introduction to smart designs for the development of adaptive interventions: with application to weight loss research. *Translational Behavioral Medicine*, *4*, 260–274.

Arjas, E., & Saarela, O. (2010). Optimal dynamic regimes: presenting a case for predictive inference. *The International Journal of Biostatistics*, *6*(2): Article 10.

Artman, W. J., Nahum-Shani, I., Wu, T., Mckay, J. R., & Ertefaie, A. (2020). Power analysis in a smart design: sample size estimation for determining the best embedded dynamic treatment regime. *Biostatistics*, *21*(3), 432–448.

Atan, O., Jordon, J., & van der Schaar, M. (2018). Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Banks, H. T., Jang, T., & Kwon, H. D. (2011). *Feedback control of HIV antiviral therapy with long measurement time*. Tech. rep., North Carolina State University. Center for Research in Scientific Computation.

Bekiroglu, K., Russell, M. A., Lagoa, C. M., Lanza, S. T., & Piper, M. E. (2017). Evaluating the effect of smoking cessation treatment on a complex dynamical system. *Drug and Alcohol Dependence*, *180*, 215–222.

Bellman, R. (1965). *Dynamic programming* (Vol. 1 ed.). Princeton University Press.

Bennett, C. C., & Hauser, K. (2013). Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach. *Artificial Intelligence in Medicine*, *57*(1), 9–19.

Berry, D. A. (2001). Adaptive trials and Bayesian statistics in drug development. *Biopharmaceutical Report*, *9*(2), 1–11.

Berry, D. A. (2004). Bayesian statistics and the efficiency and ethics of clinical trials. *Statistical Science*, *19*(1), 175–187.

Bertsekas, D. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Bhatt, D. L., & Mehta, C. (2016). Adaptive designs for clinical trials. *New England Journal of Medicine*, *375*(1), 65–74.

Biernot, P., & Moodie, E. E. (2010). A comparison of variable selection approaches for dynamic treatment regimes. *The International Journal of Biostatistics*, *6*(1): Article 6.

Blatt, D., Murphy, S. A., & Zhu, J. (2004). A-learning for approximate planning. *Ann Arbor*, *1001*, 48109–2122.

Blumenthal, S., & Cohen, A. (1968). Estimation of the larger of two normal means. *Journal of the American Statistical Association*, *63*(323), 861–876.

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Burnett, T., Mozgunov, P., Pallmann, P., Villar, S. S., Wheeler, G. M., & Jaki, T. (2020). Adding flexibility to clinical trial designs: an example-based guide to the practical use of adaptive designs. *BMC Medicine*, *18*(1), 1–21.

Butler, E. L., Laber, E. B., Davis, S. M., & Kosorok, M. R. (2018). Incorporating patient preferences into estimation of optimal individualized treatment rules. *Biometrics*, *74*(1), 18–26.

Cain, L. E., Robins, J. M., Lanoy, E., Logan, R. W., Costagliola, D., & Hernán, M. A. (2010). When to start treatment? a systematic approach to the comparison of dynamic regimes using observational data. *The International Journal of Biostatistics*, *6*(2): Article 18.

Casella, G., & Strawderman, W. E. (1981). Estimating a bounded normal mean. *The Annals of Statistics*, *9*(4), 870–878.

Chakraborty, B., & Moodie, E. E. M. (2013). *Statistical methods for dynamic treatment regimes: Reinforcement learning, causal inference, and personalized medicine*. Springer.

Chakraborty, B., & Murphy, S. A. (2014). Dynamic treatment regimes. *Annual Review of Statistics and Its Application*, *1*, 447–464.

Chakraborty, B., Murphy, S., & Strecher, V. (2010). Inference for non-regular parameters in optimal dynamic treatment regimes. *Statistical Methods in Medical Research*, *19*(3), 317–343.

Chakraborty, B., Laber, E. B., & Zhao, Y. (2013). Inference for optimal dynamic treatment regimes using an adaptive m-out-of-n bootstrap scheme. *Biometrics*, *69*(3), 714–723.

Chakraborty, B., Laber, E. B., & Zhao, Y. Q. (2014). Inference about the expected performance of a data-driven dynamic treatment regime. *Clinical Trials*, *11*(4), 408–417.

Chen, G., Zeng, D., & Kosorok, M. R. (2016). Personalized dose finding using outcome weighted learning. *Journal of the American Statistical Association*, *111*(516), 1509–1521.

Chen, M. H., Müller, P., Sun, D., Ye, K., & Dey, D. K. (2010). *Frontiers of statistical decision making and Bayesian analysis: In Honor of James O. Berger*. Springer Science & Business Media.

Cheung, Y. K., Chakraborty, B., & Davidson, K. W. (2015). Sequential multiple assignment randomized trial (smart) with adaptive randomization for quality improvement in depression treatment program. *Biometrics*, *71*(2), 450–459.

Collins, L. M., Murphy, S. A., & Bierman, K. L. (2004). A conceptual framework for adaptive preventive interventions. *Prevention Science*, *5*, 185–196.

Cotton, C. A., & Heagerty, P. J. (2011). A data augmentation method for estimating the causal effect of adherence to treatment regimens targeting control of an intermediate measure. *Statistics in Biosciences*, *3*, 28–44.

Dawson, R., & Lavori, P. W. (2012). Efficient design and inference for multistage randomized trials of individualized treatment policies. *Biostatistics*, *13*(1), 142–152.

Dehejia, R. H. (2005). Program evaluation as a decision problem. *Journal of Econometrics*, *125*(1–2), 141–173.

Dugdale, A., & Payne, P. (1977). Pattern of lean and fat deposition in adults. *Nature*, *266*(5600), 349–351.

Ertefaie, A., & Strawderman, R. L. (2018). Constructing dynamic treatment regimes over indefinite time horizons. *Biometrika*, *105*(4), 963–977.

Ertefaie, A., Wu, T., Lynch, K. G., & Nahum-Shani, I. (2016). Identifying a set that contains the best dynamic treatment regimes. *Biostatistics*, *17*(1), 135–148.

Fan, A., Lu, W., & Song, R. (2016). Sequential advantage selection for optimal treatment regime. *The Annals of Applied Statistics*, *10*(1), 32.

Fan, Y., He, M., Su, L., & Zhou, X. H. (2019). A smoothed q-learning algorithm for estimating optimal dynamic treatment regimes. *Scandinavian Journal of Statistics*, *46*(2), 446–469.

Ghosh, P., Nahum-Shani, I., Spring, B., & Chakraborty, B. (2020). Noninferiority and equivalence tests in sequential, multiple assignment, randomized trials (smarts). *Psychological Methods*, *25*(2), 182.

Goldberg, Y., Song, R., & Kosorok, M. R. (2013). Adaptive q-learning. In *From probability to statistics and back: High-dimensional models and processes–A Festschrift in honor of Jon A. Wellner* (pp. 150–162). Institute of Mathematical Statistics.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gunter, L., Zhu, J., & Murphy, S. (2007). Variable selection for optimal decision making. In *Conference on Artificial Intelligence in Medicine in Europe* (pp. 149–154). Springer.

Gunter, L., Chernick, M., & Sun, J. (2011a). A simple method for variable selection in regression with respect to treatment selection. *Pakistan Journal of Statistics and Operation Research*, *7*, 363–380.

Gunter, L., Zhu, J., & Murphy, S. (2011b). Variable selection for qualitative interactions. *Statistical Methodology*, *8*(1), 42–55.

Gunter, L., Zhu, J., & Murphy, S. (2011c). Variable selection for qualitative interactions in personalized medicine while controlling the family-wise error rate. *Journal of Biopharmaceutical Statistics*, *21*(6), 1063–1078.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hirano, K., & Porter, J. R. (2009). Asymptotics for statistical treatment rules. *Econometrica*, *77*(5), 1683–1701.

Hirano, K., & Porter, J. R. (2012). Impossibility results for nondifferentiable functionals. *Econometrica*, *80*(4), 1769–1790.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, *81*(396), 945–960.

Holloway, S., Laber, E., Linn, K., Zhang, B., Davidian, M., & Tsiatis, A. (2020). *Dyntxregime: Methods for estimating optimal dynamic treatment regimes*. R package version 49 3.

Horowitz, M. (2008). The role of registries in facilitating clinical research in bmt: examples from the center for international blood and marrow transplant research. *Bone Marrow Transplantation*, *42*(1):S1–S2.

Jeng, X. J., Lu, W., & Peng, H. (2018). High-dimensional inference for personalized treatment decision. *Electronic Journal of Statistics*, *12*(1), 2074.

Jennison, C., & Turnbull, B. W. (2000). *Group sequential methods with applications to clinical trials*. Chapman & Hall/CRC Press.

Jennison, C., & Turnbull, B. W. (2013). Interim monitoring of clinical trials: Decision theory, dynamic programming and optimal stopping. *Kuwait Journal of Science*, *40*(2), 43–49.

Jiang, B., Song, R., Li, J., & Zeng, D. (2019). Entropy learning for dynamic treatment regimes. *Statistica Sinica*, *29*(4), 1633.

Jonsson, A. (2019). Deep reinforcement learning in medicine. *Kidney Diseases*, *5*(1), 18–22.

Keys, A., Brožek, J., Henschel, A., Mickelsen, O., & Taylor, H. L. (1950). *The biology of human starvation* (2 Vols.). Univ. of Minnesota Press.

Kidwell, K. M. (2015). Chapter 2: DTRs and SMARTs: Definitions, designs, and applications. In *Adaptive treatment strategies in practice: Planning trials and analyzing data for personalized medicine* (pp. 7–23). SIAM.

Kidwell, K. M., & Wahed, A. S. (2013). Weighted log-rank statistic to compare shared-path adaptive treatment strategies. *Biostatistics*, *14*(2), 299–312.

Kim, H., Ionides, E. L., & Almirall, D. (2016). A sample size calculator for smart pilot studies. *SIAM Undergraduate Research Online*, *9*, 229–250.

Krakow, E. F., Hemmer, M., Wang, T., Logan, B., Arora, M., Spellman, S., Couriel, D., Alousi, A., Pidala, J., Last, M., et al. (2017). Tools for the precision medicine era: how to develop highly personalized treatment recommendations from cohort and registry data using q-learning. *American Journal of Epidemiology*, *186*(2), 160–172.

Kwon, H. D., Lee, J., & Yoon, M. (2014). An age-structured model with immune response of HIV infection: Modeling and optimal control approach. *Discrete & Continuous Dynamical Systems-B*, *19*(1), 153.

Laber, E. B., & Zhao, Y. Q. (2015). Tree-based methods for individualized treatment regimes. *Biometrika*, *102*(3), 501–514.

Laber, E. B., Linn, K. A., & Stefanski, L. A. (2014a). Interactive model building for q-learning. *Biometrika*, *101*(4), 831–847.

Laber, E. B., Lizotte, D. J., Qian, M., Pelham, W. E., & Murphy, S. A. (2014b). Dynamic treatment regimes: Technical challenges and applications. *Electronic Journal of Statistics*, *8*(1), 1225.

Laber, E. B., Zhao, Y. Q., Regh, T., Davidian, M., Tsiatis, A., Stanford, J. B., Zeng, D., Song, R., & Kosorok, M. R. (2016). Using pilot data to size a two-arm randomized trial to find a nearly optimal personalized treatment strategy. *Statistics in Medicine*, *35*(8), 1245–1256.

Lavori, P. W., & Dawson, R. (2004). Dynamic treatment regimes: practical design considerations. *Clinical Trials*, *1*, 9–20.

Lavori, P. W., & Dawson, R. (2008). Adaptive treatment strategies in chronic disease. *Annual Review of Medicine*, *59*, 443–453.

Lavori, P. W., Dawson, R., & Rush, A. J. (2000). Flexible treatment strategies in chronic disease: clinical and research implications. *Biological Psychiatry*, *48*, 605–614.

Lei, H., Nahum-Shani, I., Lynch, K., Oslin, D., & Murphy, S. A. (2012). A "smart" design for building individualized treatment sequences. *Annual Review of Clinical Psychology*, *8*, 21–48.

Linn, K. A., Laber, E. B., & Stefanski, L. A. (2015). iqlearn: Interactive q-learning in r. *Journal of Statistical Software*, *64*(1), 1–25.

Liu, N., Liu, Y., Logan, B., Xu, Z., Tang, J., & Wang, Y. (2019). Learning the dynamic treatment regimes from medical registry data through deep q-network. *Scientific Reports*, *9*(1), 1–10.

Liu, Y., Logan, B., Liu, N., Xu, Z., Tang, J., & Wang, Y. (2017a). Deep reinforcement learning for dynamic treatment regimes on medical registry data. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 380–385). IEEE.

Liu, Y., Wang, Y., & Zeng, D. (2017b). Sequential multiple assignment randomization trials with enrichment design. *Biometrics*, *73*(2), 378–390.

Liu, Y., Wang, Y., Kosorok, M. R., Zhao, Y. Q., & Zeng, D. (2018). Augmented outcome-weighted learning for estimating optimal dynamic treatment regimens. *Statistics in Medicine*, *37*(26), 3776–3788.

Lizotte, D. J., & Tahmasebi, A. (2017). Prediction and tolerance intervals for dynamic treatment regimes. *Statistical Methods in Medical Research*, *26*(4), 1611–1629.

Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., & Kosorok, M. R. (2020). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*, *115*(530), 692–706.

Luckett, D. J., Laber, E. B., Kim, S., & Kosorok, M. R. (2021). Estimation and optimization of composite outcomes. *Journal of Machine Learning Research*, *22*(167), 1–40.

Luedtke, A. R., & Van Der Laan, M. J. (2016). Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *Annals of Statistics*, *44*(2), 713.

Lunceford, J. K., Davidian, M., & Tsiatis, A. A. (2002). Estimation of survival distributions of treatment policies in two-stage randomization designs in clinical trials. *Biometrics*, *58*(1), 48–57.

MacKinnon, D. P., Cheong, J., & Pirlott, A. G. (2012). *Statistical mediation analysis*. American Psychological Association.

Maei, H. R., Szepesvári, C., Bhatnagar, S., & Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*.

Mahar, R. K., McGuinness, M. B., Chakraborty, B., Carlin, J. B., IJzerman, M. J., & Simpson, J. A. (2021). A scoping review of studies using observational data to optimise dynamic treatment regimens. *BMC Medical Research Methodology*, *21*(1), 1–13.

Manski, C. F. (2000). Identification problems and decisions under ambiguity: Empirical analysis of treatment response and normative analysis of treatment choice. *Journal of Econometrics*, *95*, 415–442.

Manski, C. F. (2002). Treatment choice under ambiguity induced by inferential problems. *Journal of Statistical Planning and Inference*, *105*(1), 67–82.

Manski, C. F. (2004). Statistical treatment rules for heterogeneous populations. *Econometrica*, *72*(4), 1221–1246.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Moodie, E. E., & Richardson, T. S. (2010). Estimating optimal dynamic regimes: Correcting bias under the null. *Scandinavian Journal of Statistics*, *37*(1), 126–146.

Moodie, E. E., Richardson, T. S., & Stephens, D. A. (2007). Demystifying optimal dynamic treatment regimes. *Biometrics*, *63*(2), 447–455.

Moodie, E. E. M., Platt, R. W., & Kramer, M. S. (2009). Estimating response-maximized decision rules with applications to breastfeeding. *Journal of the American Statistical Association*, *104*, 155–165.

Murphy, S. A. (2003). Optimal dynamic treatment regimes. *Journal of The Royal Statistical Society Series B-statistical Methodology*, *65*, 331–355.

Murphy, S. A. (2005a). An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, *24*(10), 1455–1481.

Murphy, S. A. (2005b). A generalization error for q-learning. *Journal of Machine Learning Research*, *6*, 1073–1097.

Murphy, S. A., van der Laan, M., & Robins, J. M. (2001). Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, *96*, 1410–1423.

Murphy, S. A., Lynch, K. G., Oslin, D. W., McKay, J. R., & Tenhave, T. R. (2007). Developing adaptive treatment strategies in substance abuse research. *Drug and Alcohol Dependence*, *88*(Suppl 2), S24–30.

Murray, T. A., Yuan, Y., & Thall, P. F. (2018). A Bayesian machine learning approach for optimizing dynamic treatment regimes. *Journal of the American Statistical Association*, *113*(523), 1255–1267.

Nahum-Shani, I., Qian, M., Almirall, D., Pelham, W. E., Gnagy, B., Fabiano, G. A., Waxmonsky, J. G., Yu, J., & Murphy, S. A. (2012a). Experimental design and primary data analysis methods for comparing adaptive interventions. *Psychological Methods*, *17*(4), 457–477.

Nahum-Shani, I., Qian, M., Almirall, D., Pelham, W. E., Gnagy, B., Fabiano, G. A., Waxmonsky J. G, Yu, J., & Murphy, S. A. (2012b). Experimental design and primary data analysis methods for comparing adaptive interventions. *Psychological Methods*, *17*(4), 457.

Nahum-Shani, I., Smith, S. N., Spring, B. J., Collins, L. M., Witkiewitz, K. A., Tewari, A., & Murphy S. A. (2018). Just-in-time adaptive interventions (jitais) in mobile health: Key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine: A Publication of the Society of Behavioral Medicine*, *52*, 446–462.

Navarro-Barrientos, J. E., Rivera, D. E., & Collins, L. M. (2011). A dynamical model for describing behavioural interventions for weight loss and body composition change. *Mathematical and Computer Modelling of Dynamical Systems*, *17*(2), 183–203.

Neyman, J. S. (1923). On the application of probability theory to agricultural experiments. Essay on principles. section 9. (translated and edited by DM Dabrowska and TP speed, statistical science (1990), 5, 465–480). *Annals of Agricultural Sciences*, *10*, 1–51.

Oetting, A. I., & Levy, J. A. (2007). Statistical methodology for a smart design in the development of adaptive treatment strategies. In *Causality and Psychopathology*. Oxford University Press.

Ogunnaike, B. A., & Ray, W. H. (1994). *Process dynamics, modeling, and control*. Oxford University Press.

Orellana, L., Rotnitzky, A., & Robins, J. M. (2010). Dynamic regime marginal structural mean models for estimation of optimal dynamic treatment regimes, part I: main content. *The International Journal of Biostatistics*, *6*(2): Article 8.

Pearl, J. (2000). Chapter 6: Simpson's paradox, confounding, and collapsibility. In *Causality: Models, reasoning and inference* (pp. 173–200). Cambridge University Press.

Pelham, W. E., Hoza, B., Pillow, D. R., Gnagy, E. M., Kipp, H. L., Greiner, A. R., Waschbusch, D. A., Trane, S. T., Greenhouse, J. B., Wolfson, L. J., & FitzPatrick, E. R. (2002). Effects of methylphenidate and expectancy on children with ADHD: Behavior, academic performance, and attributions in a summer treatment program and regular classroom settings. *Journal of Consulting and Clinical Psychology*, *70*(20), 320–335.

Peto, R. (1982). Statistical aspects of cancer trials. In *Treatment of cancer* (pp. 867–871). Chapman and Hall.

Pfammatter, A. F., Nahum-Shani, I., DeZelar, M., Scanlan, L., McFadden, H. G., Siddique, J., Hedeker, D., & Spring, B. (2019). Smart: Study protocol for a sequential multiple assignment randomized controlled trial to optimize weight loss management. *Contemporary Clinical Trials*, *82*, 36–45.

Qian, M., & Murphy, S. A. (2011). Performance guarantees for individualized treatment rules. *Annals of Statistics*, *39*(2), 1180–1210.

Qian, M., Chakraborty, B., Maiti, R., & Cheung, Y. K. (2021). A sequential significance test for treatment by covariate interactions. *Statistica Sinica*, *31*, 1–22.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., & Ghassemi, M. (2017). Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In *Machine Learning for Healthcare Conference* (pp. 147–163). PMLR.

Rivera, D. E., Pew, M. D., & Collins, L. M. (2007). Using engineering control principles to inform the design of adaptive interventions: A conceptual introduction. *Drug and Alcohol Dependence*, *88*, S31–S40.

Robins, J., Orellana, L., & Rotnitzky, A. (2008). Estimation and extrapolation of optimal treatment and testing strategies. *Statistics in Medicine*, *27*(23), 4678–4721.

Robins, J. M. (1986). A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, *7*, 1393–1512.

Robins, J. M. (1989). The analysis of randomized and non-randomized aids treatment trials using a new approach to causal inference in longitudinal studies. In *Health service research methodology: A focus on AIDS* (pp. 113–159).

Robins, J. M. (1994). Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics-Theory and Methods*, *23*(8), 2379–2412.

Robins, J. M. (1997). Causal inference from complex longitudinal data. In *Latent variable modeling and applications to causality* (pp. 69–117). Springer.

Robins, J. M. (2000). Marginal structural models versus structural nested models as tools for causal inference. In *Statistical models in epidemiology, the environment, and clinical trials* (pp. 95–133). Springer.

Robins, J. M. (2004). Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second Seattle Symposium in Biostatistics* (pp. 189–326). Springer.

Rose, E. J., Laber, E. B., Davidian, M., Tsiatis, A. A., Zhao, Y. Q., & Kosorok, M. R. (2019). *Sample size calculations for smarts*. NC State University Department of Statistics Technical Report 1, 1–30.

Rosenbaum, P. R. (1991). Discussing hidden bias in observational studies. *Annals of Internal Medicine*, *115*(11), 901–905.

Rosenberg, E. S., Davidian, M., & Banks, H. T. (2007). Using mathematical modeling and control to develop structured treatment interruption strategies for HIV infection. *Drug and Alcohol Dependence*, *88*, S41–S51.

Rosenberger, W. F., & Lachin, J. M. (2015). *Randomization in clinical trials: Theory and practice*. John Wiley & Sons.

Rosthøj, S., Fullwood, C., Henderson, R., & Stewart, S. (2006). Estimation of optimal dynamic anticoagulation regimes from observational data: a regret-based approach. *Statistics in Medicine*, *25*, 4197–215.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, *66*, 688–701.

Rubin, D. B. (1978). Bayesian inference for causal effects: The role of randomization. *Annals of Statistics*, *6*, 34–58.

Rubin, D. B. (1980). Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, *75*(371), 591–593.

Schulte, P. J., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2014). Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science: A Review Journal of the Institute of Mathematical Statistics*, *29*(4), 640.

Schwartz, J., Wang, W., & Rivera, D. (2006). Optimal tuning of process control-based decision policies for inventory management in supply chains. *Automatica*, *42*, 1311–1320.

Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Doyle III, F. J. (2016). *Process dynamics and control*. John Wiley & Sons.

Shi, C., Song, R., & Lu, W. (2019). On testing conditional qualitative treatment effects. *Annals of Statistics*, *47*(4), 2348–2377.

Shortreed, S. M., Laber, E., Scott Stroup, T., Pineau, J., & Murphy, S. A. (2014). A multiple imputation strategy for sequential multiple assignment randomized trials. *Statistics in Medicine*, *33*(24), 4202–4214.

Song, R., Wang, W., Zeng, D., & Kosorok, M. R. (2015). Penalized q-learning for dynamic treatment regimens. *Statistica Sinica*, *25*(3), 901.

Stevens, L. M., Linstead, E., Hall, J. L., & Kao, D. P. (2021). Association between coffee intake and incident heart failure risk: A machine learning analysis of the FHS, the ARIC study, and the CHS. *Circulation: Heart Failure*, *14*(2), e006799.

Sugiyama, M. (2015). *Statistical reinforcement learning: modern machine learning approaches*. CRC Press.

Sun, Y., & Wang, L. (2021). Stochastic tree search for estimating optimal dynamic treatment regimes. *Journal of the American Statistical Association*, *116*(533), 421–432.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, *24*(6), 833–849.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *4*(1), 1–103.

Tao, Y., & Wang, L. (2017). Adaptive contrast weighted learning for multi-stage multi-treatment decision-making. *Biometrics*, *73*(1), 145–155.

Tao, Y., Wang, L., & Almirall, D. (2018). Tree-based reinforcement learning for estimating optimal dynamic treatment regimes. *The Annals of Applied Statistics*, *12*(3), 1914.

Thall, P. F. (2015). Chapter 4: Smart design, conduct, and analysis in oncology. In *Adaptive treatment strategies in practice: Planning trials and analyzing data for personalized medicine* (pp. 41–54). SIAM.

Thall, P. F., Millikan, R. E., & Sung, H. G. (2000). Evaluating multiple treatment courses in clinical trials. *Statistics in Medicine*, *19*(8), 1011–1028.

Thall, P. F., Sung, H. G., & Estey, E. H. (2002). Selecting therapeutic strategies based on efficacy and death in multicourse clinical trials. *Journal of the American Statistical Association*, *97*(457), 29–39.

Thall, P. F., Logothetis, C., Pagliaro, L. C., Wen, S., Brown, M. A., Williams, D., & Millikan, R. E. (2007a). Adaptive therapy for androgen-independent prostate cancer: a randomized selection trial of four regimens. *Journal of the National Cancer Institute*, *99*(21), 1613–1622.

Thall, P. F., Wooten, L. H., Logothetis, C. J., Millikan, R. E., & Tannir, N. M. (2007b). Bayesian and frequentist two-stage treatment strategies based on sequential failure times subject to interval censoring. *Statistics in Medicine*, *26*(26), 4687–4702.

Tsiatis, A. A., Davidian, M., Holloway, S. T., & Laber, E. B. (2019). *Dynamic treatment regimes: Statistical methods for precision medicine*. Chapman & Hall/CRC Press.

van der Laan, M., & Petersen, M. (2007a). Causal effect models for realistic individualized treatment and intention to treat rules. *The International Journal of Biostatistics*, *3*(1): Article 3.

van der Laan, M., & Petersen, M. (2007b). Statistical learning of origin-specific statically optimal individualized treatment rules. *The International Journal of Biostatistics*, *3*(1): Article 6.

Van Der Vaart, A. (1991). On differentiable functionals. *Annals of Statistics, 19* (1), 178–204.

Vansteelandt, S., Joffe, M., et al. (2014). Structural nested models and g-estimation: The partially realized promise. *Statistical Science*, *29*(4), 707–731.

Voils, C. I., Chang, Y., Crandell, J. L., Leeman, J., Sandelowski, M. J., & Maciejewski, M. L. (2012). Informing the dosing of interventions in randomized trials. *Contemporary Clinical Trials*, *33*(6), 1225–1230.

Wagner, E. H., Austin, B. T., Davis, C., Hindmarsh, M. F., Schaefer, J. K., & Bonomi, A. E. (2001). Improving chronic illness care: Translating evidence into action. *Health Affairs*, *20*(6), 64–78.

Wahed, A. S., & Tsiatis, A. A. (2004). Optimal estimator for the survival distribution and related quantities for treatment policies in two-stage randomization designs in clinical trials. *Biometrics*, *60*(1), 124–133.

Wallace, M., Moodie, E., Stephens, D., & Simoneau, G. (2020). *DTRreg: DTR estimation and inference via g-estimation, dynamic WOLS, q-learning, and dynamic weighted survival modeling (DWSurv)*. R package version 17.

Wallace, M. P., & Moodie, E. E. (2014). Personalizing medicine: a review of adaptive treatment strategies. *Pharmacoepidemiology and Drug Safety*, *23*(6), 580–585.

Wallace, M. P., & Moodie, E. E. (2015). Doubly-robust dynamic treatment regimen estimation via weighted least squares. *Biometrics*, *71*(3), 636–644.

Wallace, M. P., Moodie, E. E., & Stephens, D. A. (2019). Model selection for g-estimation of dynamic treatment regimes. *Biometrics*, *75*(4), 1205–1215.

Wang, L., Rotnitzky, A., Lin, X., Millikan, R. E., & Thall, P. F. (2012). Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer. *Journal of the American Statistical Association*, *107*, 493–508.

Wang, L., Yu, W., He, X., Cheng, W., Ren, M. R., Wang, W., Zong, B., Chen, H., & Zha, H. (2020). Adversarial cooperative imitation learning for dynamic treatment regimes. In *Proceedings of The Web Conference 2020* (pp. 1785–1795).

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292.

Wu, Y., & Wang, L. (2021). Resampling-based confidence intervals for model-free robust inference on optimal treatment regimes. *Biometrics*, *77*(2), 465–476.

Xin, J., Chakraborty, B., & Laber, E. (2012). *qlearn: Estimation and inference for q-learning*. R package version 10 1.

Xu, Y., Müller, P., Wahed, A. S., & Thall, P. F. (2016). Bayesian nonparametric estimation for dynamic treatment regimes with sequential transition times. *Journal of the American Statistical Association*, *111*(515), 921–950.

Yan, X., Ghosh, P., & Chakraborty, B. (2021). Sample size calculation based on precision for pilot sequential multiple assignment randomized trial (smart). *Biometrical Journal*, *63*(2), 247–271.

Zajonc, T. (2012). Bayesian inference for dynamic treatment regimes: Mobility, equity, and efficiency in student tracking. *Journal of the American Statistical Association*, *107*(497), 80–92.

Zhang, B., & Zhang, M. (2018a). C-learning: A new classification framework to estimate optimal dynamic treatment regimes. *Biometrics*, *74*(3), 891–899.

Zhang, B., & Zhang, M. (2018b). Variable selection for estimating the optimal treatment regimes in the presence of a large number of covariates. *The Annals of Applied Statistics*, *12*(4), 2335–2358.

Zhang, B., Tsiatis, A. A., Davidian, M., Zhang, M., & Laber, E. (2012a). Estimating optimal treatment regimes from a classification perspective. *Stat*, *1*(1), 103–114.

Zhang, B., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2012b). A robust method for estimating optimal treatment regimes. *Biometrics*, *68*(4), 1010–1018.

Zhang, B., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2013). Robust estimation of optimal dynamic treatment regimes for sequential treatment decisions. *Biometrika*, *100*(3), 681–694.

Zhang, C., Chen, J., Fu, H., He, X., Zhao, Y. Q., & Liu, Y. (2020). Multicategory outcome weighted margin-based learning for estimating individualized treatment rules. *Statistica Sinica*, *30*, 1857.

Zhang, Y., Laber, E. B., Davidian, M., & Tsiatis, A. A. (2018). Interpretable dynamic treatment regimes. *Journal of the American Statistical Association*, *113*(524), 1541–1549.

Zhao, Y., Kosorok, M. R., & Zeng, D. (2009). Reinforcement learning design for cancer clinical trials. *Statistics in Medicine*, *28*(26), 3294–3315.

Zhao, Y., Zeng, D., Rush, A. J., & Kosorok, M. R. (2012a). Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, *107*, 1106–1118.

Zhao, Y. Q., Zeng, D., Rush, A. J., Kosorok, M. R. (2012b). Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, *107*(499), 1106–1118.

Zhao, Y. Q., Zeng, D., Laber, E. B., & Kosorok, M. R. (2015). New statistical learning methods for estimating optimal dynamic treatment regimes. *Journal of the American Statistical Association*, *110*(510), 583–598.

Zhao, Y. Q., Laber, E. B., Ning, Y., Saha, S., & Sands, B. E. (2019). Efficient augmentation and relaxation learning for individualized treatment rules using observational data. *The Journal of Machine Learning Research*, *20*(1), 1821–1843.

Zhou, X., Mayer-Hamblett, N., Khan, U., & Kosorok, M. R. (2017). Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, *112*(517), 169–187.

Zhu, W., Zeng, D., & Song, R. (2019). Proper inference for value function in high-dimensional q-learning for dynamic treatment regimes. *Journal of the American Statistical Association*, *114*(527), 1404–1417.