Chapter 6
Statistical Data Mining of Clinical Data

Ilya Lipkovich, Bohdana Ratitch, and Cristina Ivanescu

6.1 Introduction

6.1.1 What Is Data Mining?

Data mining is understood broadly as a set of analytical tools and methods for extracting nontrivial information from the data so that it can be transformed into useful knowledge and practical tools. Data mining has been evolving and applied in multidisciplinary contexts, and its definitions vary depending on the viewpoint. The following definition reflects the view of the Knowledge Discovery in Databases (KDD):

- Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from large data sets or databases.

A typical statistician's view of data mining expressed succinctly in a textbook by Hand et al. (2001) places more emphasis on the interpretability of discovered "relationships" for decision-makers:

- "Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner."

I. Lipkovich (✉)
Eli Lilly and Company, Indianapolis, IN, USA
e-mail: ilya.lipkovich@lilly.com

B. Ratitch
Bayer Inc., Montreal, Quebec, Canada

C. Ivanescu
IQVIA, Amsterdam, Netherlands

O. V. Marchenko, N. V. Katenka (eds.), *Quantitative Methods in Pharmaceutical Research and Development*, https://doi.org/10.1007/978-3-030-48555-9_6

In Pharma there is no established definition of what data mining is; however, summarizing our experience and observations of the current practices across the industry, we can formulate it broadly as any post hoc analyses:

- "Data mining is any post-hoc analysis of existing clinical data to provide answers to relevant scientific, clinical, and business questions to internal and external stakeholders."

In this review chapter, we take a broad view on data mining in clinical settings as a valuable and principled element in a large cycle of knowledge discovery and confirmation from healthcare data that facilitates a full and efficient use of vast amounts of available data. It is a type of data analytics for problems that have the following common features:

- A large amount of available data in terms of the number of records (patients) and/or the number of features (variables) that has at least some of the following properties:
  - Typically arising from observational studies, or representing "observational elements" embedded within randomized trials
  - Collected for a different purpose than the intended "data mining" analyses
  - Dispersed over different databases
- The relationships that need to be learned from the data may be obscured by
  - Both random and systematic errors
  - Various inconsistencies in data collection and variable construction
  - Missing data (likely "not completely at random")
  - The presence of irrelevant data (noise features) that need to be filtered out
  - Redundancy in relevant data ("overlapping" variables)
  - Time-dependent causal mechanisms with unknown lags
  - The presence of both short-lived and long-term time effects
  - Dynamic dependencies between variables that may change over time
  - Unknown causal relationships among variables
  - Unmeasured confounders and spurious associations between variables

This chapter is organized as follows. In the rest of the introduction section, we present the framework for data mining and machine learning (DMML) and try to connect it with important tasks in drug development. Section 6.2 lays out the key concepts of DMML. Section 6.3 contains a brief overview of selected methods with more emphasis on those that will be featured in our case studies. Section 6.4 summarizes the principles of data mining with clinical data and suggests some elements of the statistical plans for DM. Section 6.5 contains three case studies. Finally, in Sect. 6.6, we provide a brief discussion of the key points of the chapter.

6.1.2 Machine Learning and Data Mining Framework

The fields of data mining and machine learning emerged as a combination of computer science and statistics methods with some additional unique objectives and emphases. As in computer science, one goal of machine learning is to build algorithmic solutions and machines to solve problems; as in statistics, another goal is to do reliable inference from data. The unique objectives of machine learning include the emphasis on how computers can "program themselves" (learning) and how to most effectively capture, store, and retrieve patterns and regularities in data. Data mining is a closely related field, which employs many machine learning and statistics methods. Data mining activities are typically focused on discovering new insights from databases that are often big, heterogeneous, and/or unstructured and which are presumed to contain interesting patterns not known or not sufficiently understood a priori. To name just a few sources, excellent introductions and textbooks in machine learning and data mining are provided by Mitchell (1997), Hand et al. (2001), Hastie et al. (2009), Clarke et al. (2009), Witten et al. (2011), Domingos (2012), and Goodfellow et al. (2016).

Although statistical modeling and machine learning have been developing as separate disciplines, the similarities between the two abound, and they can be used in synergetic ways (Friedman 1997; Hand 1998; Vapnik 2006). Statistical modeling approaches often formulate some assumptions regarding the data distribution and the relationship between the dependent and independent variables and place emphasis on the interpretability of the model and the ability to do inference about the underlying data generation mechanism including the effect of individual predictors on the response.

Somewhat simplifying matters, we can describe classical statistical modeling as largely focusing on estimating a model from which the data arose: $Y = f(\boldsymbol{X}) + \varepsilon$, where ε represents a random error induced either by an experimental procedure, random sampling, or other sources of uncertainty. The error term is modeled with some parametric family of distributions, often with a common assumption that the random errors have the expected value of zero, $E(\varepsilon) = 0$, and is independent of $\boldsymbol{X}$. In statistics, it is common to refer to $\boldsymbol{X}$ as a set of independent or predictor variables and to Y as a response, outcome, or dependent variable. For a continuous outcome variable Y, $f(\boldsymbol{X})$ is the conditional mean $f(\boldsymbol{x}) = E(Y | \boldsymbol{X} = \boldsymbol{x})$ and is referred to as a regression function. For a categorical outcome $Y = \{j : j = 1, .., k\}$, the same representation gives rise to a classification function, where $E(Y = j | \boldsymbol{X} = \boldsymbol{x})$ models the probability of group membership.

For example, in clinical studies, one of the central objectives is to assess whether treatment has a statistically significant effect on a response variable and to estimate the magnitude of the treatment effect. This is often done by estimating a fairly simple statistical model with treatment represented by one of the independent variables and then performing statistical tests about the treatment effect and estimating mean treatment effect based on that model. For example, in the context of continuous outcome variable, it is the mean difference in response under the experimental treatment versus control, possibly adjusted for other independent (pretreatment) variables included in the model, such as baseline patient characteristics. In this case, the goal is to do inference from the data collected so far, without a further objective of predicting responses for future individual patients. As can be further illustrated with this and other applications in healthcare, the classical view focuses on hypothesis testing applied to a single test or a small number of pre-specified tests with clearly defined multiplicity adjustment strategy.

This framework can be contrasted with that of data mining and machine learning (DMML) where the "learning" aspect refers to the ability of a computational system to acquire new knowledge from its environment and data or to organize existing knowledge in a way that facilitates its use. The "machine" aspect emphasizes the automated and algorithmic fashion of the learning, not involving "human intervention." In machine learning, often there is also an objective of creating a "machine" (computational system or tool), which, once trained, can be deployed for future use with new data. This is reflected in a ubiquitous use of the term "training data set" or "training sample" in DMML to designate the data that are available at the learning stage and implying that there will be more data to come.

The differences between classical statistics and machine learning have been a subject of lively debates (see, e.g., Breiman 2001a, b on two modeling cultures within statistics). Unlike classical statistics, DMML methods tend to rely less on formal distributional assumptions and often work with "black box" representations of the target unknown function $f(\boldsymbol{x})$, where the interpretability of the effect of the individual input variables on the output may be limited and not of primary interest and the emphasis is rather on the quality of prediction for future cases.

Classical examples of machine learning for prediction is speech and character (e.g., handwriting) recognition and (more recently) email spam detection where arguably the interpretability of the prediction rules does not play a key role (see, e.g., email spam Example 1 in Hastie et al. 2009). However, the situation is quite different in applications of machine learning in the healthcare such as automated diagnosis of patients where both healthcare providers and patients are not only interested in accurate prediction but would also like to know which features are primarily responsible for discriminating the "events" from "non-events." Here relying on pure "black box" solutions may be less desirable: although a black box model may be entertained as the prediction tool, it then should be followed by various visualizations facilitating the interpretability, such as a decision tree, a variable importance graph, a partial dependence plot, or a low-dimensional projection. This shift from a "black box" to a more transparent and interpretable data mining, reminding us of the exploratory data analysis (EDA, Tukey 1977) with its emphasis on "looking at the data," differentiates the outlook of modern "data miners" from that of "machine learners."

Another distinction can be made between the role of modeling assumptions and model selection in the classical statistics and DMML. In the former, analysis often relies on "standard assumptions" and pre-specified models, while in practical situations the analyst is discouraged from "looking at the data" (even for validating the analysis assumptions) in fear of data dredging, as multiple "looks" may arguably inflate the false positive rates. This outlook is at odds with the discovery nature of the statistical science. Sometimes analysts may act under implicit assumptions that "pre-specified" means "valid," resulting in suboptimal models entertained at "confirmatory stage." While these are not examples of the best application of classical statistics, they often occur in practice, especially in the healthcare settings where pre-specification of analyses required by regulatory agencies played a key role and became a part of the culture. Data visualizations historically did not play an important role in this "traditional" view of data analysis, perhaps because of the fear of "looking at the data" when implementing pre-specified confirmatory analyses. Nevertheless, things are gradually changing, and most large pharmaceutical companies have been creating data mining and visualization groups to facilitate data analysis and presentation in all phases of drug development.

DMML by its nature relies on model selection using data-driven methods with an emphasis on discovery rather than confirmatory analysis. Unlike classical statistics, the emphasis in DMML is not on hypotheses testing but on generating plausible hypotheses that are data-driven ("random"), rather than pre-specified. On the other hand, data-driven model selection inherent in data mining methods may often occur "behind the scenes," and the statistical uncertainty associated with model selection is left unaccounted for in the final analyses and decision-making based on these analyses. Again, perhaps reflecting not the best practices of data mining, the final inference is sometimes based on the findings of a last stage of a complex multistage data mining procedure ignoring the uncertainty associated with all the previous stages. Model validation and incorporation of the uncertainty associated with the entire DMML strategy in the prediction and inference is extremely important for generating useful insights and tools but may be very challenging to implement. Like EDA, data mining (somewhat in contrast with machine learning, having an emphasis on fully automated analysis strategies) encourages various graphical displays and low-dimensional data representations facilitating model selection and interpretability.

Table 6.1 summarizes the above discussion points on the differences and commonalties between data mining/machine learning and traditional statistics.

We conclude this discussion by observing that the distinctions made may oversimplify and overdramatize the situation, and a trend has been emerging for convergence between the "classical" statistics and DMML under a unifying framework where both elements are considered from a common modeling perspective of "statistical learning" emphasizing some general principles such as achieving a trade-off between bias and variance (see Hastie et al. 2009). Many ideas and approaches developed in the two disciplines independently and use different terminologies but share similar concepts and properties. One indication of convergence between the two domains is an increasing interest in developing "classical" inferential procedures for machine learning techniques, such as for inference "after model selection." For example, see Wager et al. (2014) on bagging and random forest, and (Meinshausen et al. 2009, Lockhart et al. 2014, Tian et al. 2016) on post-selection inference in the context of L1 (lasso) penalized regression and related methods. Another example of such blending is procedures that combine classical

Table 6.1 Data mining/machine learning versus "classical" statistics

Classical statistics	Data mining/machine learning
Typically uses relatively small data sets collected from designed experiments or by sampling from well-defined populations	Large and often dispersed and heterogeneous data sets, often collected for (business) purposes other than the data mining
Assumes a data generation mechanism: $y = f(\boldsymbol{X}) + \varepsilon,$ where $f(\boldsymbol{X})$ has relatively simple structure (e.g., a linear model) and the error term(s) are represented by parametric distributions	Often poses its task as recovering unknown function $f(\boldsymbol{X})$ which may be a "black box" (i.e., fairly complex nonlinear relationship) while the presence of statistical uncertainty (noise) is often ignored
The objective is to estimate parameters for the entire population from available sample(s)	The objective is to obtain predictions for new (future) cases or extract useful features that reveal underlying (unknown) structure. The analysis data often represent the entire population
Focus on hypothesis testing applied to a single test or a small number of pre-specified tests with clearly defined multiplicity adjustment strategy	Hypothesis generation (knowledge discovery) rather than formal hypothesis testing, less emphasis on statistical significance (often rather focusing on controlling the false discovery rate)
Interpretability is an important element of modeling culture where the structure of $f(\boldsymbol{X})$ is driven by few pre-selected variables, mainly based on existing domain knowledge or factors of a designed experiment	The "black box" modeling makes interpretability neither important nor easily attainable; however, in data mining applications, the decision-makers often desire to have the decision rules expressed in interpretable form
Modeling relies on "standard assumptions," often discouraging "looking at the data" in fear of data dredging. Underutilizes the discovery element of statistical science	Relies on model selection using data-driven methods with emphasis on discovery rather than confirmatory analysis; incorporation of uncertainty associated with model selection however may be challenging to implement for multistage data mining strategies
Visualization does not play important role, perhaps because of the fear of data dredging when implementing pre-specified confirmatory analyses	Data mining (like EDA and in contrast with machine learning) encourages graphical displays facilitating model selection and interpretability

multiplicity control in hypothesis testing with model averaging for design and analysis of dose-finding studies introduced in (Bretz et al. 2005) and implemented in R package **MCPMod** (Bornkamp et al. 2009).

6.1.3 Machine Learning Tasks for Solving Clinical Problems

For decades, healthcare data have traditionally been analyzed using statistical methods, but the applications of machine learning and data mining have been constantly growing in all areas of health informatics, from molecular biology and genetics, to clinical research, to epidemiology. There are a few major areas in health informatics (Herland et al. 2014): bioinformatics typically focuses on the molecular-level data; neuro-informatics concentrates on analysis of brain imaging data; clinical informatics involves analysis of patient data; public health informatics applies data mining and analytics to population-level data; and translational bioinformatics is an interdisciplinary field that develops techniques for integrating biological and clinical data. In this chapter, we focus on clinical informatics.

Traditional view of the scope of data mining and machine learning in drug development is that its place is primarily in preclinical and early-phase drug discovery (e.g., using machine learning for gene expression analysis). Using data mining in later stages of drug development (Phases 3, 4) is often considered with suspicion as a euphemism of data dredging that sponsors may use to promote favorable views of their products and make unsubstantiated claims (e.g., of enhanced efficacy in subpopulations identified through data mining). Many consider complete pre-specification of analyses in late stages of drug development as the necessary condition of their validity. However, learning from data is a continuous process that does not stop at the beginning of Phase 3. Clearly, not everything is known at the time of new study design, and so not all meaningful analyses can be preplanned; therefore, extracting as much evidence from data as possible, even post hoc, maximizes good use of patient data and resources allocated to a clinical trial. There is indeed a striking contrast between the vast amount of patient-level data (on efficacy and safety) collected in the course of a clinical trial and reporting trial results with a few summaries (ultimately, a single *P*-value for the primary analysis), which suggests large amounts of data collected may be underutilized in the drug development process.

Contrary to this view, we believe that data mining is an integral part on all stages of the drug development process. However, we promote *principled* data mining (as opposed to “data dredging”) and to this end outline some principles and good practices of clinical data mining.

In our review of data mining methodologies, we focus on methods most useful for clinical trial data analysis; however, most of the methodologies equally apply to observational studies where treatment assignments are driven by prescribers’ decisions and not by chance. In fact, as we argue, observational studies and randomized clinical trials (RCTs) have much in common, and this is exactly why data mining (and model selection as its integral element) is needed in both. Often, we can consider clinical trial data as observational study embedded in an RCT. Even in the perfectly designed and conducted RCT, post-randomization events, such as dropouts, effectively break the randomization and make comparison of simple summaries by treatment arm biased and therefore require model-based analysis, even for the assessment of treatment effect under the intention-to-treat (ITT) principle.

Here, we list some general analytic tasks that arise with clinical data (whether originated from a randomized trial or not) that lend themselves to applications of data mining methods, and we group these tasks under more traditional headings of supervised, semi-supervised, or unsupervised learning. Specific examples for some of these tasks will be provided in Sect. 6.5 using case studies.

6.1.3.1 Supervised Learning

Supervised learning occurs when the DMML system is provided both the input and the correct output for a set of training cases and is tasked with learning a function that maps input to output, with the goal of being able to predict the output for future, unlabeled input instances. The initial, labeled set $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, N$, of inputs $\boldsymbol{x}_i$ (a p-dimentional vector) and outputs y_i is referred to as a training set, and the learning algorithm adapts its internal representation of the input-output relationship $\widehat{f}(\boldsymbol{x}_i)$ to minimize some measure of differences between the observed and predicted outputs: y_i and $\widehat{f}(\boldsymbol{x}_i)$, e.g., residual sum of squares $RSS = \sum_{i=1}^{N} \left(y_i - \widehat{f}(\boldsymbol{x}_i) \right)^2$. Supervised learning problems are further grouped into classification when the output variable is a category (e.g., mild, moderate, severe) and regression when the output variable is a real value (e.g., blood pressure or weight).

Some common tasks in the healthcare setting include:

Patient diagnostics. Applications of building diagnostic models informed by various patient-level covariates (symptoms) started to appear decades ago, for example, a simple diagnostic tool was constructed using tree-based decision rules that allowed clinicians of an emergency unit to make a quick assessment whether a patient with non-traumatic chest pain can be diagnosed with a myocardial infraction using ECG and other available markers (Mair et al. 1995). An example of increasing use of diagnostic tools incorporating AI algorithms is a recent approval by FDA of *OsteoDetect*, an image processing device that "analyzes wrist radiographs using machine learning techniques to identify and highlight distal radius fractures during the review of posterior-anterior (PA) and lateral (LAT) radiographs of adult wrists" (FDA 2018).

Building predictive models for patients' future outcomes. Models may be built to predict safety or efficacy outcomes, informed by assigned treatment, biomarkers available prior to treatment initiation, and evolving (early) patient outcomes. Examples of such clinical applications of supervised learning are predicting mortality and readmission after a discharge from an intensive care unit in order to avoid premature discharges from the unit for future patients (Ouanes et al. 2012) and predicting cancer susceptibility, cancer recurrence, and cancer survival (Konstantina et al. 2015).

Modeling intermediate outcomes as part of a treatment evaluation strategy. Supervised learning often arises in clinical applications not as a goal in itself but rather as an intermediate step for obtaining more accurate estimates of treatment effects. This is especially true for evaluating treatment effect in observational trials but also applies to RCTs. For example, to account for missing data, methods of inverse probability weighting can be employed that require modeling the probability of a patient remaining in the trial through specific time. Here the goal is not to predict patient's dropout as such but rather to correct for selection bias in the primary analyses caused by the fact the dropouts may have occurred not completely at random but were associated with patients' covariates and early outcomes. As another example, imputation methods are often used for the same purpose of accounting for selection bias due to dropouts. Constructing an imputation model or a model for inverse probability weighting can often be successfully done using "black box" methods of supervised learning that have an advantage over simple regression methods in that they utilize all available data and do not require preselection of key predictor variables nor assume any specific form of their relationship with the probability of dropout which are typically unknown to the investigators. See Tang and Ishwaran (2017) for comparison of various strategies for imputing missing data via random forest algorithms. In our case study in Sect. 6.5, we will provide an example of using machine learning method to estimate treatment effect under informative treatment switching via inverse probability weighting.

6.1.3.2 Unsupervised Learning

In unsupervised learning, the DMML system is not provided with any "correct answer" such as a training sample where all cases are correctly labeled into target categories or values but rather is designed to discover and model the underlying structure and patterns in the data with the goal of acquiring a better understanding of the data. Unsupervised learning problems are broadly grouped into clustering, where the objective is to discover inherent groupings of similar units described by data (e.g., groups of patients with similar treatment outcomes), and association, where the goal is to discover interesting relations between variables (e.g., co-occurrence of certain diseases or events) which can be also thought of as clustering, although in the variable/feature space.

Some common tasks and examples include:

Clustering to identify patients with similar efficacy outcomes in the absence of a definite single outcome measure determining patient's response to treatment. This is especially relevant for diseases where the patients' well-being is described by a set of variables representing complementary and sometimes conflicting clinical criteria and scales, which is often the case in neuroscience and some other areas. Example of this is clustering patients in treatment of fibromyalgia, as such patients often show great variability in symptoms domains for which a given treatment may be beneficial (Lipkovich et al. 2014; Abtroun et al. 2016). Clustering of patients in the multivariate space of disease symptoms may lead to construction of better criteria for clinical response as well as understating what patient characteristics are driving response in different domains of symptoms.

Identifying patients with distinct response profiles (or trajectories) over time. Response profiles may represent different types of patients, e.g., "early responders who later fail," "relapsers," "gradual responders," "sustained responders," etc. Clustering can be done using traditional statistical methods of analysis of growth curves via finite mixture random effects with categorical latent variables representing class membership (Muthén et al. 2002), as well as by application of multivariate clustering methods, e.g., Lipkovich et al. (2008).

Use of methods for association learning. This objective has been explored in pharmacovigilance to uncover drug-adverse event relationships and drug-drug interactions in spontaneous reporting systems and large healthcare databases such as electronic health records and administrative claims (Harpaz et al. 2012).

Detecting outliers and unusual patterns, often in the context of fraudulent assessment of outcomes. See, e.g., O'Kelly (2004) for a case study illustrating the use of statistical multivariate techniques to identify fraudulent clinical data.

6.1.3.3 Semi-supervised Learning

Note that in many situations learning may need to proceed in an unsupervised manner even in a prediction setting for regression or classification problem where the target variable is entirely missing in the observed (training) data. An interesting case that falls somewhere in between the supervised and unsupervised learning is predicting differences in outcomes for a patient under different treatment regimes (treatment effects) given his/her characteristics. This is not a supervised learning problem because in a typical parallel arm clinical trial, a patient is assigned only to one treatment (experimental or control), and therefore the patient-level treatment differences are unobserved, similar to class labels in the clustering problem. However, because one of the treatment outcomes is observed for every patient, these hypothetical differences can be predicted using methods of traditional supervised learning as building blocks. Here we provide examples of such tasks under the heading of semi-supervised learning:

Subgroup identification. Heterogeneity of treatment effect has been recognized in many therapeutic areas leading to a growing interest in precision medicine (also referred to as personalized medicine) so that therapies can be tailored to characteristics of the patients as well as their environment and lifestyle (Ashley 2015). Much research has been dedicated to identifying genetic traits that are responsible for variations in disease susceptibility and response to treatments, but subgroup identification also extends to other demographic and clinical characteristics that may be predictive of the treatment effect (often referred to as biomarkers). In this setting, the researchers may be presented with a large set of potential biomarkers, and the objective is to determine a small subset that can be used to reliably describe patient profiles with the most beneficial treatment effect or a favorable benefit-risk balance (see Lipkovich et al. 2017). We provide a review of various methods for subgroup identification in Sect. 6.3.3 and illustrate with a case study in Sect. 6.5.1.

Estimating optimal treatment regimes. Another clinical problem closely related to precision medicine which also falls under the semi-supervised learning framework is construction of optimal dynamic treatment regimes (DTRs) utilizing information on patient's characteristics and accumulated patient's outcomes at each decision point. In many health disorders, especially chronic conditions, sequential decision-making is necessary to adapt treatment over time in response to the evolving health status of the patient. This is especially important if there is a high degree of heterogeneity in individual long-term responses to treatment and when treatment may need to be adjusted as a result of emerging side effects. DTRs thus extend the concept of precision medicine to time-varying treatment regimes where therapy (type, dose, and/or timing) may be adjusted over time based on the up-to-date patient information and may be influenced by earlier treatment choices (Murphy 2003, 2005; Chakraborty and Murphy 2014). Development of evidence-based dynamic treatment regimes, just like evidence-based recommendations for the initial choice of treatment, is part of building clinical decision support systems for the entire treatment cycle. Several methods for estimation of optimal DTRs, e.g., Q-learning and A-learning, originate in a subfield of machine learning known as reinforcement learning (Sutton and Barto 1998) where the focus is on decision-making in stochastic dynamic environments. We review the problem and methods of estimation of optimal DTRs in Sect. 6.3.3 and present a case study in Sect. 6.5.2. Although the problem of identifying optimal regimes is a semi-supervised learning problem (in absence of explicit information of what is the optimal regime in training data), it often uses supervised learning approaches as integral components. For example, methods of outcome-weighted learning construct DTRs by casting it in as a series of classification problems (Zhao et al. 2015).

6.1.3.4 Feature Selection and Dimensionality Reduction

A cornerstone of machine learning and data mining methods (whether supervised or unsupervised) is feature selection and dimensionality reduction. Databases often contain a multitude of variables which are potentially related to the problem at hand, but it may not be known in advance which attributes are in fact useful and which are irrelevant or redundant given other attributes. The challenge is compounded by the fact that machine learning often starts with “feature expansion” resulting in transforming the initial set of covariates into a broader set of “features” (e.g., adding variables capturing information on two- and three-way covariate interactions or using feature expansion via radial basis functions). Given this enriched set of features, the DMML system needs to extract useful information to reduce model complexity, improve accuracy, and facilitate interpretation. Feature selection refers to an automatic selection of data attributes that are most useful and relevant for predictive modeling (supervised learning) or identifying patterns in data (unsupervised learning). Examples of such methods for supervised learning range from traditional stepwise model selection techniques to more sophisticated methods of penalized estimation (e.g., lasso method a.k.a. L_1 penalty) and ensemble learning (see Sects. 6.2 and 6.3 for more details). Often feature expansion and selection can be done within a single analytic strategy (e.g., as in support vector machines (SVMs) with a kernel-based feature expansion and an L_1 penalty).

Dimensionality reduction also aims at reducing the number of attributes in a data set, but unlike feature selection, it does so by creating new, fewer combinations of attributes that nevertheless capture the key information in the data (e.g., using methods based on principal components and singular value decomposition). These methods can be used in the context of both supervised and unsupervised learning.

In this chapter we will provide case studies covering some of the above tasks. Clearly, it would be impossible to cover all applications of data mining in clinical research in a single chapter. While we provide some reference to a broader set of applications, we would like to explicitly mention some areas that will not be covered here: applications of data mining/machine learning in molecular biology, genomics, proteomics, microarray data, and medical imaging. While some of case studies will use methods that are applicable to analysis of epidemiological studies and real-world databases (such as claims/electronic medical records), we will not have specific examples here.

6.2 Overview of Key Concepts

The power of machine learning algorithms is in their ability to provide solutions to difficult problems by generalizing from a limited set of examples observed in real life (a training set). This is not unlike statistical inference where, in order for the results to be of practical utility, the inference performed from a finite set of data samples must be generalizable to a population of interest (e.g., finite population as in survey sampling or hypothetical population as in making inference for "future" patients). Therefore, good accuracy/performance on the training data set is typically not the ultimate goal, and performance on new data not included in the training set is of greater importance. In supervised learning, it is a common practice to divide the available data into a training set and a test set so that the solution can be developed on the training set and its performance evaluated on the test set, representing new data not used for learning. In this context, the performance metric applied to the training set while the learning is taking place (e.g., the R-square) often serves as a surrogate for the ultimate performance measure—generalization ability. However, focusing on this surrogate measure, especially when fitting complex models (i.e., with a large number of parameters), may lead to overfitting, so that the model "describes" the random error (noise) in the training data rather than the underlying relationship. Avoiding overfitting and improving generalization performance requires careful consideration, which we review in this section.

6.2.1 Bias-Variance Trade-Off

One important aspect of a machine learning algorithm's performance is the bias-variance trade-off. The generalization error can be decomposed into two main components: bias and variance. For example, as discussed in the previous section, in supervised learning (for continuous outcome), the objective is to find a mapping for the input-output relationship $\widehat{f}(\boldsymbol{x})$ which minimizes the expected prediction error $E\left(y-\widehat{f}(\boldsymbol{x})\right)^2$ which can be decomposed as

$$E\left(y-\widehat{f}(\boldsymbol{x})\right)^2 = \left(Bias\left[\widehat{f}(\boldsymbol{x})\right]\right)^2 + Variance\left[\widehat{f}(\boldsymbol{x})\right] + \sigma^2,$$

where σ^2 is an irreducible error (e.g., due to noise in inputs and outputs). $Bias\left[\widehat{f}(\boldsymbol{x})\right] = E\left[\widehat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right]$ is the method's tendency to consistently produce solutions $\widehat{f}(\boldsymbol{x})$ that deviate from the truth $f(\boldsymbol{x})$. Systematic bias can be introduced, for example, by using an inappropriate model, e.g., using a linear model when the true function is nonlinear, or using optimization algorithms that tend to converge at a local optimum (e.g., greedy search). $Variance\left[\widehat{f}(\boldsymbol{x})\right] = E\left[\widehat{f}(\boldsymbol{x})^2\right] - E\left[\widehat{f}(\boldsymbol{x})\right]^2$ is the method's tendency to produce different solutions (move around its mean) as a result of changes in the training set (even though different training sets are generated by the same underlying process) or randomness that is part of the learning algorithm (e.g., Monte Carlo methods).

There is a trade-off between bias and variance: typically bias decreases as the model complexity increases, while variance increases with model complexity. An increasingly complex model will reach a point where its prediction error on the training set is very small but it overfits the training data and leads to an increase in the error on the test data. This is illustrated in Fig. 6.1 where the training set error, depicted by the light gray line, decreases steadily as the size of the model (a tree-based model in this example) increases, whereas increase in complexity leads to no further gains in the test set error after a certain point (tree size of 5) as depicted by the black solid line. This is why, perhaps counterintuitively at a first thought, a more complex learner (model) is not necessarily better than a more parsimonious one, and there is typically some intermediate model complexity that provides the best performance on the test (and future) data. In Fig. 6.1, the dotted line represents one standard deviation above the best test error, which may be a good target to select model complexity. This leads us to the next topic—model selection.

6.2.2 Model Selection

For reasons discussed above, model selection is thus an integral part of the machine learning process with the ultimate goal of choosing the model that provides the best generalization performance on new data. One aspect in terms of choosing the right model is related to the choice of model class, for example, using a linear vs. nonlinear model. Another aspect relates to choosing which predictor variables to include in the model as this also directly determines model complexity. The squared bias component of the prediction error discussed above can itself be decomposed into two parts: average model (specification) bias and average

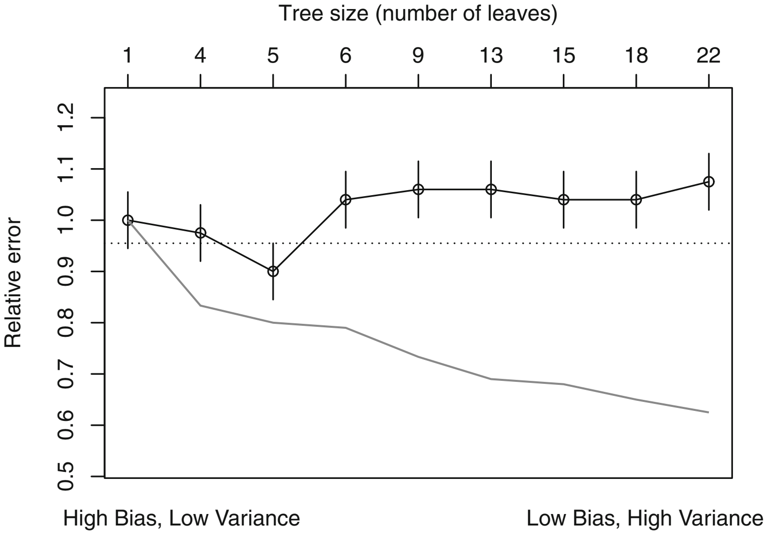

Fig. 6.1 Illustration of bias-variance trade-off using a classification tree example. The x-axis shows the tree size (model complexity); the y-axis shows the relative classification error. The black line is the generalization error (here estimated via tenfold cross-validation), and the gray line is the error estimated from the training set. The error bars are estimates of standard error associated with the cross-validation estimates. The graph suggests that the tree model starts picking up noise when the number of leaves (terminal nodes) exceeds 5

estimation bias. Model bias represents the error between the best-fitting approximation within the chosen class (e.g., a linear model based on a set of chosen predictors) and the true function. The estimation bias is the error between the average estimate of the model parameters and the best-fitting approximation in the class.

For example, for linear models $\widehat{f}(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\beta}$ using a vector of predictor variables $\boldsymbol{X}$ and parameter vector $\boldsymbol{\beta}$, the best-fitting approximation corresponds to the parameter settings $\boldsymbol{\beta}_* = \ arg\ \min_{\boldsymbol{\beta}} E[(f(\boldsymbol{X}) - \boldsymbol{X}^T\boldsymbol{\beta})^2]$. The average squared bias of a specific approximation $\widehat{f}_d(\boldsymbol{x})$ is then decomposed as follows:

$$\left(E_{\boldsymbol{x}}\left[E\left[\widehat{f}_d(\boldsymbol{x})\right] - f(\boldsymbol{x})\right]\right)^2 = \left(E_{\boldsymbol{x}}\left[\boldsymbol{x}^T\boldsymbol{\beta}_* - f(\boldsymbol{x})\right]\right)^2 + \left(E_{\boldsymbol{x}}\left[\boldsymbol{x}^T\boldsymbol{\beta}_* - E\left[\boldsymbol{x}^T\widehat{\boldsymbol{\beta}}_d\right]\right]\right)^2,$$

where the first term on the right-hand side is the model bias and the second term is the estimation bias.

The expectation over the estimated linear predictor $E\left[\boldsymbol{x}^T\widehat{\boldsymbol{\beta}}_d\right]$ is equal to that from the ideal best-fitting linear predictor $\boldsymbol{x}^T\boldsymbol{\beta}_*$ for linear models estimated using the ordinary least squares method, and in this case the estimation bias is zero. For other estimation methods, for example, penalized or ridge regression, the average estimation bias is positive, but then the models obtained with this approach typically reduce the variance component and thus can be used to achieve a desired bias-variance trade-off.

Model selection in terms of choosing the most relevant subset of predictors can be done in many ways. One traditional approach often described in statistics textbooks is based on stepwise variable selection (e.g., forward, backward, or hybrid stepwise selection), where statistical significance of effects associated with each variable are tested in some sequential way and variables are dropped or added to the model one at a time depending on their significance. This approach uses a locally greedy strategy and suffers from several other important drawbacks (e.g., multiple statistical hypothesis testing performed without proper Type I error control, unstable performance, and low prediction accuracy).

In general, data analysis involving a model selection step can be broken down into the following tasks:

1. Choosing a general form of the model (e.g., logistic or linear regression).
2. Specifying the model space (e.g., as defined by original variables $\boldsymbol{X}$, expanding them into main effects and interactions, expanding them using spline basis functions, etc.).
3. Specifying a model search strategy, i.e., a strategy for obtaining a path or multiple paths through the model space that are likely to capture "promising" models (e.g., stepwise model selection produces a sequence of the best models for the number of predictors $k = 1, 2, 3, \ldots$; coefficient paths obtained by varying the amount of penalty in lasso/elastic net; stochastic model search).
4. Specifying model selection criteria in order to identify the final best model(s).
5. Estimating parameters of the final model(s) while taking into account the uncertainty associated with the model selection step (estimation *after* model selection).
6. Predicting outcomes for new data using the selected model(s).

Typically, the general form of the model (task 1) is chosen by the researcher given domain knowledge. Some common approaches to model selection that take into account model selection uncertainly are highlighted below:

- Strategies that build a sequence of possibly overfitted models (e.g., using stepwise selection algorithms or other heuristic methods) and select from that sequence the best model using goodness-of-fit measures (based on penalized likelihood such as AIC (Akaike information criterion), BIC (Bayesian information criterion), cross-validation, or multiple testing procedures).
- An important special case is strategies based on penalized estimation procedures (e.g., lasso, elastic net) produce sparse coefficient paths corresponding to increasing model dimensionality by varying values of the tuning parameter(s) that control the amount of penalty placed on model complexity; the final model is selected by choosing the optimal tuning settings, e.g., by cross-validation (described further below in this section).
- Bayesian and frequentist model averaging where the "final model" is a weighted average over many models that fit data reasonably well (see review papers,

Hoeting et al. 1999; Wang et al. 2009); and other methods of *ensemble* learning such as random forest and boosting (see further in this section).

Although many analytical tools are packaged as "all-in-one" with specific choices for tasks 2 to 6 outlined above, it is useful to evaluate them on individual components. Sometimes a procedure may be reasonable for one aspect but very unsatisfactory for others, and an improved one can be constructed by borrowing approaches from different procedures and recombining their elements.

6.2.3 Variable Importance

A concept closely related to the problem of model selection is that of variable importance (VI)—an integral measure of the relative importance or contribution of a variable in predicting the response. Variable importance is used in many machine learning approaches where a single variable may contribute multiple times in different parts of the model, hence the need to obtain a single score presenting its overall importance. It can be defined in different ways that suit or reflect the construction of specific types of learners. For example, in classification and regression trees (CART, Breiman et al. 1984), variable importance can reflect improvements in the classification error achieved by using this variable to define splitting criteria across all the tree nodes where it is used as a splitter. Another way of determining variable importance (as was first introduced in random forests by Breiman 2001a, b) is to evaluate the reduction in predictive accuracy after a random permutation of the values of a given variable across all training samples. If the variable is strongly associated with response, then after randomly permuting its values, substantial decreases in prediction accuracy can be expected. Other versions of the permutation-based variable importance have been suggested in the literature, e.g., Sandri and Zuccolotto (2008); Strobl (2008); Altmann et al. (2010); and Lipkovich et al. (2017). A recently developed alternative approach to variable importance is based on SHAP values inspired by the Shapley interaction index from game theory (Lundberg and Lee 2017; 2018). The importance of each feature is defined at the level of individual observations by posing an *additive feature attribution model* that decomposes the fitted value into a sum of contributions from each feature (when present in the model). The importance score for a feature reflects its contribution into conditional expectation of the outcome averaged over all possible subsets of other features conditioned upon. Therefore, this approach is different from others in that it summarizes the contributions of a feature into a fitted model (a "black box") irrespective of how good or poor the fit may be. We will discuss several approaches to variable importance in more detail in the context of subgroup identification in Sect. 6.3.3 and in an application of random forest for computing inverse probability of censoring weights in Sect. 6.5.3.

6.2.4 Multiple Testing

The problem of multiple hypothesis testing is well recognized in statistics and relates to the probability of rejecting a null hypothesis when it is in fact true (referred to as Type I error). In machine learning and data mining applications where analysis tends to be more exploratory, it is not uncommon that tens or hundreds of hypothesis tests are performed by the learning algorithm, and thus care must be taken in this context as well. Some methods, both from classical statistics and machine learning, may have some sort of multiple hypothesis testing performed as part of the internal workings of their algorithm. This is true, for example, of model selection methods that rely on significance findings to select predictor variables. But multiple testing can occur in many other contexts as well, e.g., in medical applications in analyses of genomics data to discover genes, among thousands considered, exhibiting significant expression patterns of interest, or in evaluation of clinical safety data based on a multitude of safety tests and types of adverse events. Today there are many approaches for multiplicity control, some being more conservative or powerful than others while being well disciplined, and so some methods may be more appropriate than others in the context of machine learning.

It has been argued that especially in machine learning, where the number of tests can be very large, it is useful to distinguish between the false positive rate and false discovery rate (Glickman et al. 2014). The false positive rate is the probability of rejecting a null hypothesis given that it is true. The false discovery rate (FDR, introduced by Benjamini and Hochberg 1995) is the probability that a null hypothesis is true given that the null hypothesis has been rejected by a test.

A classical Bonferroni procedure that safeguards against *any* false positive findings is very conservative and has a consequence that the power to reject truly false null hypotheses is greatly reduced as the number of hypotheses tested increases. In the context of exploratory analysis where a large number of hypotheses are tested with an intent to generate promising hypotheses for further investigation and confirmation, it may be more relevant to accept a possibility that some discoveries will be false as long as their proportion among all significant findings is acceptably low. This point of view is taken by approaches that control the FDR. Multiplicity control involves establishing an appropriate adjusted significance level against which the *P*-values should be compared or conversely adjusting the raw *P*-values directly. This can also be achieved by resampling/permutation approaches (e.g., Westfall and Young 1993; Westfall and Troendle 2008; Vsevolozhskaya et al. 2015) which can provide empirical distribution of *P*-values. Resampling/permutation-based methods are particularly useful for multiple testing with high-dimensional data as they do not require specific distributional assumptions and utilize the data-based correlation structure among variables which can provide important power advantages. Efron (2010) provided an extensive discussion of issues in large-scale inference, including a novel interpretation of Benjamini and Hochberg's procedure from the empirical Bayes perspective, and introduced the *local FDR* which is defined as posterior probability of false discovery for a single hypothesis given test statistics for tested hypotheses. Another recent advance is developing a very general class of variable selection procedures that control FDR via so-called knockoff variables—a special type of irrelevant or “dummy” variables that mimic the correlation structure in the original variables (Barber and Candès 2015).

6.2.5 Cross-Validation

Cross-validation (Allen 1974; Stone 1974; Geisser 1975) is a method widely used in machine learning for estimation of the true error rate a.k.a. *generalization error* (model assessment) as well as for variable selection and for estimating tuning parameters that control the complexity of the model and machine learning algorithms. In a nutshell, the motivation and general idea behind the cross-validation is as follows. If we had a sufficiently large data set, we could partition it into a training data set, to which a model can be fit, and a validation set, on which performance of the model could be assessed. However, if the size of the available data set is not large, a more efficient use of data, which also would lead to more stable estimates of the model performance, can be achieved using K-fold cross-validation. In this case, the original data set is split into K nonoverlapping data sets (folds) of equal size, typically in a random fashion. For each fold $k = 1, \ldots, K$, a model is fit to a training data set comprised of all data except the k^{th} fold. We will denote such models as $\widehat{f}^{-k}(\boldsymbol{x}), k = 1, \ldots, K$. For each observation $i = 1, \ldots, N$ in the original full data set, let’s denote by $k(i)$ the fold index to which the i^{th} observation was assigned. The cross-validation estimate of the prediction error can be obtained as follows, based on some measure of error or loss $L\left(y, \widehat{f}(\boldsymbol{x})\right)$ defined for any given pair of predictors $\boldsymbol{x}$ and response y:

$$\widehat{Error}^{CV} = \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \widehat{f}^{-k(i)}(\boldsymbol{x}_i)\right).$$

The choice of the number of folds K influences a potential bias of the error estimate (smaller K can result in a larger bias due to smaller sizes of the training data sets) and its variance (larger K leading to higher variance as the training data sets will tend to be more similar, i.e., having more observations in common). A special case when $K = N$ is referred to as leave-one-out (LOO) cross-validation, in which case N different models are fit, each to all data excluding only the i^{th} observation. This estimator is approximately unbiased but can have a high variance. In general, the bias will depend on the size of the original data set and a slope of the error curve versus the size of the training data set. Frequently used choices of the number of folds K are 5 or 10 which attain a good balance between bias and variance in practice (Breiman and Spector 1992; Kohavi 1995).

Cross-validation can be used not only to obtain an estimate of the generalization error of a chosen type of model but also to tune parameters of the fitting method, e.g., the size of the model or the amount of penalty placed on the magnitude of the regression coefficients.

If modeling is carried out using several model selection steps, e.g., variable selection and parameter tuning, cross-validation must be applied across the entire sequence of steps: dividing data into *k* folds at the very beginning, carrying out all modeling steps on all *k*-1 training sets (leaving the k^{th} fold out), and estimating model performance on the k^{th} test fold. Otherwise, steps performed outside of the cross-validation procedure (e.g., variable selection) may have an unfair "advantage" in terms of basing their criteria on all available data, including those that would later be used as new, test examples (see, e.g., Ambroise and McLachlan 2002). While model selection and model assessment tasks both require cross-validation, it may be done using different cross-validation approaches applied in a nested manner (Varma and Simon 2006). One exception to the rule of subjecting the whole learning procedure to cross-validation is that the steps based on unsupervised learning (not involving outcomes) may be performed based on all available data, before creating the folds.

Recent research (Krstajic et al. 2014) also investigated some variants on the basic idea of *K*-fold cross-validation, e.g., with repeated random splits of the data and/or stratification on the outcome variable.

6.2.6 Bootstrap

The bootstrap method was introduced by Efron (1979) and has been used extensively ever since both in statistics and machine learning. In the course of any analysis, some kinds of summaries (statistics) are typically generated to describe the data set, the patterns, characteristics, and relationships underlying its variables. It is useful to characterize the variability and distribution of the estimated statistic induced by the sampling variability, but to do it through gathering many data sets from the population is rarely feasible, and we have to content with having only one data set for analysis. The basic idea of bootstrap is to use the data set at hand as a "surrogate population" and to generate multiple data sets, called bootstrap samples, by resampling with replacement from the original data for the purpose of approximating the distribution of the estimated statistic, which is referred to as the empirical distribution. In the most generic application of bootstrap, one needs to estimate the statistic of interest from each of these bootstrap samples, and these multiple estimates serve as samples from the statistic's distribution. Using these values, one can estimate different characteristics of the underlying sampling distribution, for example, bias, standard error and associated confidence interval, and *P*-values for testing statistical hypotheses, which were the primary goals of bootstrap when it was invented. However, later bootstrap found other "unintended" uses within the realm of machine learning, most notably a bootstrap-based point estimate also known as bagging estimator that plays a key role in algorithms of bagging and random forest. The motivation for bootstrap averaging was that it reduces variability at the expense of small amount of bias for unstable estimation processes where small perturbations in the data may incur substantial differences in estimated models. We will further discuss the application of bootstrap by "bagging" methods in Sect. 6.3.1. Many machine learning procedures belong to this class owing to their inherent instability.

One type of instability is caused by model (variable) selection where bootstrap can be very useful to evaluate model selection uncertainty, as we can fit a model of the same type, such as stepwise selection or lasso, to multiple bootstrap training data sets and obtain different characteristics such as the proportion of times a given variable was selected across all samples which can be plotted against some tuning parameters that control the selection process.

Another use of bootstrap is that it can also help to address the challenge with estimation of the generalization error. Recall the earlier discussion that when constructing a regression or classification model, we are mostly interested in model's predictive accuracy on test data not included in the training data set. Remarkably, bootstrap can be used as a source for generating "test samples," as when we create bootstrap training data sets by resampling with replacement from the original data set, naturally every bootstrap sample will include some observations multiple times, whereas some will not be selected. In fact, it is easy to verify that the probability of an observation to be included in any given bootstrap data set is $1-\left(1-\frac{1}{N}\right)^N$, approximately 0.632. In order to estimate the generalization error on samples that were not included in the training data, we can use the leave-one-out bootstrap approach, similar to the LOO cross-validation approach discussed above. To estimate the LOO generalization error on test data using some error or loss function $L\left(y,\widehat{f}(\boldsymbol{x})\right)$, for each of the N observations $(\boldsymbol{x}_i, y_i)$ in the original full data set, we only look at the predictions from the models that were built with bootstrap training samples where the i^{th} observation was not included, indicated as a subset of indices $J^{(-i)} \subset \{1, .., B\}$, and thus can be designated as a new, test example for this model:

$$\widehat{Error}^{LOOB} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mid J^{(-i)}\mid}\sum_{b\in J^{(-i)}} L\left(y_i,\widehat{f}^b(\boldsymbol{x}_i)\right).$$

Observations that appear in all B bootstrap samples can be omitted from the error calculation.

One drawback of the LOO bootstrap estimator of the generalization error is that the average number of distinct observations in each bootstrap sample is about 0.632 of the original full data set size N and the quality of the model can decline with the reduction of the training set size. In this case, the LOO bootstrap estimate will tend to overestimate the true generalization error. The so-called .632 estimator addresses this issue by estimating the generalization error as a weighted average:

$$\widehat{Error}^{.632} = 0.632 \times \widehat{Error}^{LOOB} + 0.368 \times \widehat{Error}^{train},$$

where the $\widehat{Error}^{train}$ is the training error calculated as the average prediction error over all original training examples when the model is fitted to the full original data set.

The “.632 estimator” corrects bias due to the reduction of the bootstrap training set size, but the second bias-correcting term may be inappropriate if the amount of overfitting is very large and the training error is close to zero, in which case the bias of LOO estimator may be considerable. The “.632+ estimator” (Efron and Tibshirani 1997) improves on this estimator by adjusting the 0.632 and 0.368 weights to reflect the amount of overfitting through the “no-information error rate” estimated as the error rate of the model fit to a data set with no true association between the predictors and outcome. This estimator was shown to outperform the LOO bootstrap and five- and tenfold cross-validation in Efron and Tibshirani (1997), providing low variance and moderate bias.

6.2.7 Ensemble Learning

Several approaches for supervised learning that emerged over the years share a similar basic idea and can be considered as ensemble learning. The general principle of ensemble learning is to build multiple, relatively simple prediction models (referred to as base models or learners, often weak learners, i.e., capable of prediction accuracy at least slightly above random guessing) and combine them into one overall model, which can combine their strengths. As such, ensemble learning consists of two tasks: estimation of a population of base learners from the training data and combining them to produce overall predictions, e.g., by (weighted) voting or averaging. One of the influential works in this area which propelled further research and applications of these methods was done by Hansen and Salamon (1990), who showed that predictions made by a combination of classifiers can be more accurate than predictions from a single classifier as long as each base learner is accurate and the classifiers are diverse. In this context, a classifier is considered accurate if it is better than random guessing. Diversity means that different classifiers make different errors on new data, so that if their errors are uncorrelated, the majority vote or averaging will likely lead to a correct overall classification. We further discuss these ideas in the context of bagging, random forests, and boosting approaches in Sect. 6.3.1.

It should be mentioned that Bayesian model averaging approaches can also be regarded in the framework of ensemble learning, for example, as a large number of models are averaged according to their “credibility”—the posterior distribution of their parameters (see, e.g., Madigan and Raftery 1994; Neal and Zhang 2006). At the same time as pointed out, for example, by Domingos (2000) and Minka (2002), ensemble methods and Bayesian model averaging differ fundamentally in that ensembles change the hypothesis space (e.g., from single decision trees to linear combinations of them), while Bayesian methods weight hypotheses in the original space according to a fixed formula. Bayesian model averaging is implicitly geared towards model selection rather than model combination, so that weights attributed to individual models can get extremely skewed due to overfitting, as too much weight is placed on the maximum likelihood model, to the point where the single highest-weight model usually dominates. In this case, performance of the Bayesian strategies can be worse than that of bagging or boosting. However, if Bayesian method is modified to integrate over combinations of models rather than over individual learners, it can achieve much better results (Monteith et al. 2011; Kim and Ghahramani 2012). These findings also lend support to the view that the power of ensembles lies primarily in the changes in representational and preferential bias inherent in the process of combining several different models.

6.3 Overview of Selected Methods

6.3.1 Supervised Learning

In Sect. 6.1, we discussed supervised machine learning as a counterpart of statistical modeling for regression and classification, where the goal is to approximate a relationship between the dependent variable (outcome) and one or more independent variables (predictors). Regression analysis typically aims at estimating a regression function—the conditional expectation of the outcome Y given the predictors $\boldsymbol{X}$, $E(Y|\boldsymbol{X})$. For classification problems, where the outcome variable represents class labels $k = 1, \ldots, K$, the objective may be to estimate a model of the posterior probabilities $P(Y = k| \boldsymbol{X})$ or define a rule that would assign to each case a class label. Linear regression and logistic regression as well as modeling of other types of outcomes and underlying distributions via generalized linear models and models of time to event are examples of classical approaches widely used in statistics.

6.3.1.1 Penalized Regression

Penalized regression methods have been developed to provide a better prediction accuracy while being computationally efficient and feasible to use even with a large number of predictors. They had been independently proposed by different researches for solving somewhat different tasks: (1) incorporating in the same model a large number of potentially relevant but jointly redundant ("overlapping") predictors (sometimes exceeding the number of observations) without incurring instability in estimated coefficients (multicollinearity) and (2) dealing with a large number of irrelevant (noise) covariates among candidate predictors whose impact on estimation should be minimized (sparsity). These methods estimate model parameters by minimizing the residual sum of squares (more generally, some appropriate loss function, e.g., likelihood-based), but add a constraint (penalty) on the magnitude of the parameters. While this penalty causes the parameter estimates to be biased, it also decreases their variance that may achieve better performance via variance-bias tradeoff. Penalized methods work by shrinking estimated model coefficients to zero. Some methods can shrink a coefficient exactly to zero (effectively eliminating the variable from the model), whereas others shrink all coefficients to some non-zero values. These methods are also referred to as shrinkage or regularization methods. In penalized regression, chosen parameters satisfy the following constrained minimization condition, based on a set of N training samples:

$$\widetilde{\boldsymbol{\beta}} = arg\min_{\boldsymbol{\beta}} \left(\sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta} \right)^2 \right),$$

subject to $Penalty(\boldsymbol{\beta}) < k$.

Various penalized regression methods differ in terms of the penalty $Penalty(\boldsymbol{\beta})$ that they impose. The most popular methods are the ridge regression (Hoerl and Kennard 1970), lasso (Tibshirani 1996), adaptive lasso (Zou 2006), and elastic net (Zou and Hastie 2005).

These methods rely on one or more tuning parameters that determine the amount of shrinkage. Thus, a penalized regression method can produce a set of models, each associated with a specific setting of its tuning parameter(s). For the final model selection, the analyst must employ a tuning method to choose the optimal setting of these parameters. Among widely used approaches are model fit criteria, such as the Mallow's C_p statistic (Gilmour 1996) or Akaike information criterion (AIC) (Akaike 1974), Bayesian information criterion (BIC) (Schwarz 1978), average squared error on the validation data, and cross-validation.

Penalized regression is implemented in commercial statistical packages, including SAS®, as well as in R packages such as **lasso2** (L_1 constrained regression), **lars** (Least Angle Regression [LARS], lasso, and forward stepwise selection), **grplasso** (Group lasso), **glmpath** (L_1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model), **stepPlr** (L_2 penalized logistic regression with a stepwise variable selection), **elasticnet** (elastic net regularization), **glmnet** (lasso and elastic net regularized generalized linear models), and **penalized** (lasso and ridge penalized estimation in generalized linear models and Cox regression model).

6.3.1.2 Classification and Regression Trees

Tree-based models became very popular in data mining solutions since the mid-1980s of the last century and later made their way as building blocks in many modern procedures (e.g., ensemble learning). Therefore, we describe them with more details than others in this review.

Tree-based models can be used both for regression and classification. These models are easily visualized as decision graphs resembling upside-down trees (see

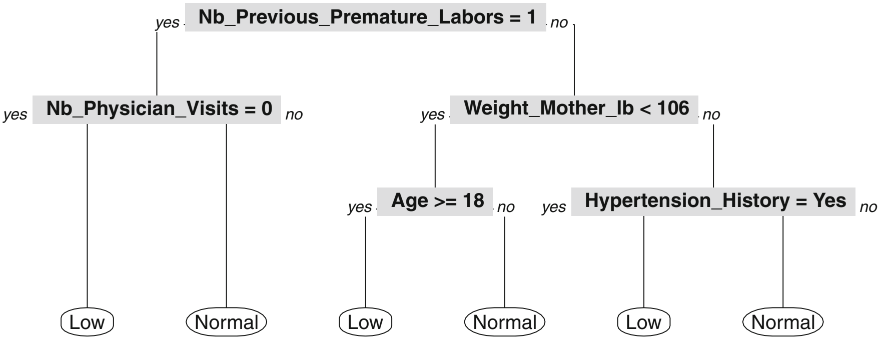

Fig. 6.2 Example of a classification tree for prediction of low/normal birth weight based on mother's characteristics (produced using R package **rpart.plot**)

example in Fig. 6.2): with a single node at the top, called a root node, and where each node can branch out into several (typically, two) child nodes. The nodes at the end of each branch, i.e., nodes that do not have any children, are referred to as terminal or leaf nodes. Each internal (non-leaf) node represents a split of the input space along the axis of one predictor, e.g., Age $\geq$ 18 vs. Age $<$ 18. Each branch culminating at a leaf node—a sequence of internal nodes—specifies a set of conditions with respect to input variables involved in the splits along the branch which define a region in the p-dimentional input space. Therefore, at the end of each branch, a leaf node represents a corresponding region and is assigned a predicted outcome associated with that region—either a numeric constant in the case of regression or a class label in the case of classification. Hence tree-based models are often called piecewise constant.

More specifically, a prediction model represented by a tree with M leaf nodes where each region is denoted as R_m, $m = 1, \ldots, M$ can be described as follows:

$$\widehat{f}(\boldsymbol{x}) = \sum_{m=1}^{M} \widehat{c}_m I\{\boldsymbol{x} \in R_m\},$$

where $I\{\boldsymbol{x} \in R_m\}$ is an indicator function for whether the values of input vector $\boldsymbol{x}$ belong to the region R_m or not and c_m are numerical constants or class labels associated with the regions. For regression, one choice of c_m is the average of outcome values corresponding to training input samples that fall into the corresponding region:

$$\widehat{c}_m = average\{y_i | \boldsymbol{x}_i \in R_{\mathrm{m}}\}.$$

For classification, a class representing majority of y_i values can be chosen as the one determining the prediction in the leaf node:

$$\widehat{c}_m = \arg max_k \frac{1}{N_m} \sum_{\boldsymbol{x}_i \in R_m} I\{y_i = k\},$$

where N_m is the number of training samples that fall into region R_m.

Fitting a tree-based model typically involves a recursive procedure which, starting with the root note, looks for a beneficial split to associate with that node, where a split creates two child nodes (here we focus on binary trees, although procedures with multi-way splits have also been developed, see, e.g., Kim and Loh 2001 and references therein), and so forth until some stopping criterion is satisfied. This construction process includes a number of steps or tasks, and a multitude of procedures have been developed that differ in how they go about them:

- How to choose splits at each node.
- How to decide whether splitting should stop.
- How to choose the optimal size of the tree (model complexity).
- How to assign a prediction value at each leaf (e.g., by averaging/voting as described above).

Each split corresponding to a node in the tree is typically defined based on a single predictor variable (although procedures that form splits based on low-dimensional functions of data have also been proposed). If the variable is quantitative (ordinal), the split condition is of the form "$X_j \leq s$" where s can be any number and is typically chosen among the values of X_j that actually occur in the training data. If the variable is categorical, the split condition is of the form "$X_j \in A$" where A is any subset of classes that can be assumed by X_j. If the condition is satisfied, the branch from that node leads to the left child, otherwise to the right child. The same variable can be used for a split in multiple nodes of the tree.

During the recursive tree-fitting procedure, at each node, the algorithm has to choose the best split across all input variables and their values. This decision is made based on some measure of goodness of split, which is a measure of reduction in node "impurity" due to the split. The most common measures (for classification trees) are the Gini index and entropy (or information gain), while the misclassification error is less frequently used although available in software implementations. Gini impurity index measures how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the tree node, being the sum of p_k $(1 - p_k)$ across all categorical outcomes (labels) $k = 1, \ldots, K$, where p_k is the probability of kth outcome, estimated by the proportion of values $\{y_i = k\}$ in a given node. Statistically, each component is the variance of a Bernoulli random variate associated with the kth outcome category.

Information gain is defined as the reduction in entropy (an information theory measure of uncertainty) due to the split. The entropy associated with each node is a measure of "expected surprise" of the node's outcome and is defined as the sum of $-p_k \log_2(p_k)$ across $k = 1, \ldots, K$, which from the statistical perspective is simply related to the negative multinomial log-likelihood. Gini and entropy measures, unlike the classification error measure, are sensitive to class proportions in a node and can lead to more "pure" splits where one class is largely predominant. Thus, for the node splitting decisions during the tree construction, the former two measures are preferred. Nevertheless, all three measures can be useful for another aspect of the tree optimization—pruning—which we briefly discuss further below.

For regression trees, a popular measure of impurity is variance, and the best split is selected as the one that maximizes the reduction of the total corrected sum of squares due to the split. It is equivalent to choosing the split that maximized the between-group sum of squares in analysis of variance with the candidate split as an independent variable.

Other splitting criteria based on statistical tests have also been developed. For example, the CHAID algorithm (Kass 1980) for classification trees is based on a chi-square statistic that tests for a chance difference of the observed distribution of the categorical outcome across child regions. Similarly, the CHAID method for regression trees uses the F-test from ANOVA models to test the null hypothesis of equality of means between the child regions.

It should be noted that the splitting procedure outlined above tends to suffer from a variable selection bias in that the input variables with more distinct values are favored: the more choices are available for a given variable, the more likely it is to find a good split using that variable for a training set at hand. This may amplify a problem with noise variables if they have more unique values than strong predictors. Several approaches have been developed to alleviate this problem, e.g., FACT (Loh and Vanichsetakul 1988), QUEST (Loh and Shih 1997), CRUISE (Kim and Loh 2001), GUIDE (Loh 2002), and linear discriminant-based approach (Kim and Loh 2003). Another tree-based approach that selects variables in an unbiased way is designed in a conditional inference framework—conditional trees (CTree) by Hothorn et al. (2006). The latter approaches are also based on recursive procedures, but when a split is being selected at a tree node, a splitting variable is selected first, independently of the splitting value (and without an exhaustive search over all splits for all candidate variables).

When values of some predictor variables are missing for some training observations, a question arises how to handle this in the splitting process. In the case of categorical predictors, one can treat missing values as a separate category, which may be beneficial if missingness itself is predictive of outcome. Another approach is based on the use of surrogate variables. This strategy provides trees with a built-in mechanism to deal with missing predictor values in a way that exploits correlations between predictors.

When the best split is identified at the current node, two child nodes are created, and the splitting process is repeated from each of these children, as well as all other leaf nodes. There are several ways to decide when the splitting should stop. One can impose a limit on the minimum number of training samples that fall into a region associated with a leaf node, and when that limit is reached, splitting that node should be stopped. Another possibility is to continue splitting until the reduction of node impurity becomes smaller than some threshold. Yet another option is to stop when

Table 6.2 Summary of approaches for tree-based modeling

Step	Classification trees (categorical response)	Regression trees (quantitative response)	Survival trees (time-to-event response)
Variable selection	Exhaustive search based on splitting criteria (CART)	Exhaustive search based on splitting criteria (CART)	Exhaustive search based on splitting criteria (LeBlanc and Crowley 1992)
	Pre-selection by F-test or χ^2-test (FACT, CHAID, QUEST, GUIDE), association measures (CTree)	Pre-selection by F-test (CHAID) or χ^2 test for sign of residuals vs. predictors (GUIDE), association measures (CTree)	Pre-selection by association measures (CTree)
Splitting criteria	Reduction in Gini index (CART), information gain (CART, C.4.5), change in log-likelihood due to split (test statistic or adjusted *P*-value, JMP), χ^2 test of independence (CHAID)	Reduction in total sum of squares (CART, JMP), adjusted *P*-value from F-test (CHAID, JMP)	Reduction in deviance residual (LeBlanc and Crowley 1993)
Stopping criteria/ pruning	Pruning based on cost-complexity (CART, QUEST, GUIDE), pessimistic pruning (C4.5), reduction in test error; stopping rules based on (adjusted) *P*-values (CHAID, CTree), direct stopping rules (FACT), limits on minimum size of the leaf, number of levels, etc.		

the best split is not statistically significant at a pre-specified level. This approach is referred to as “forward selection” and is implemented in the CHAID algorithm. These approaches will likely result in large trees, susceptible to overfitting, so the initial large tree construction can be followed by pruning—reducing the size of the tree—with the goal of optimizing some cost-complexity criterion involving a penalty parameter interpreted as costs associated with each additional split that determines a trade-off between the goodness of fit to the data and the size of the tree. Tree pruning is considered a “backward elimination” strategy, and one popular approach is known as the weakest link pruning. An appropriate value of the penalty parameter can be found, for example, by cross-validation.

Table 6.2 provides a summary of different approaches available for classification, regression, and survival trees across the three main steps of the tree building procedure.

The key reference for the modern approach to tree-based machine learning is that of Breiman et al. (1984). Books on statistical learning, e.g., by Ripley (1996) and Hastie et al. (2009), and a recent comprehensive review by Loh (2014) can provide further reading.

An early example of applications of tree-based models for clinical data predictive modeling was construction of a diagnostic tool to identify patients with acute myocardial infarction in non-traumatic chest pain patients on admission to the emergency department (Mair et al. 1995). Other notable examples include classification (diagnosis and prognosis) of pulmonary hypertension in mixed connective tissue disease (Kotajima et al. 1997); study of the effects of risk factors on time to hip fracture using tree structures survival analysis (Lu et al. 2003); use of CART as alternatives to logistic regression for the estimation of propensity scores in the context of observational data analysis (Lee et al. 2010); determination of baseline predictors of remission with placebo for patients with major depressive disorder (Nelson et al. 2012); and tools for cancer prognosis and prediction (Konstantina et al. 2015).

The discussion above focused mainly on methods pertaining to the CART (classification and regression tree) methodology. Other tree-based methods developed over the years include ID3, C4.5, and C5.0 (Quinlan 1986, 1993, 2004). The latter, in particular, includes a scheme for deriving rule sets—simplifications of conditions along a tree branch without altering the subset of observations that fall in the branch leaf, which makes them easier to interpret. The multivariate adaptive regression splines (MARS) method (Friedman 1991) can be viewed as a modification of CART designed to improve smoothness of resulting models, the lack of which is inherent in piecewise constant models realized by the trees, as well as to allow fitting additive models, which are difficult to fit with trees. Hierarchical mixture of experts (HME) (Jordan and Jacobs 1994) can also be viewed as a variant of tree-based strategies, where splits are probabilistic functions of a linear combination of multiple inputs. Tree-based models can also be built for survival outcomes (Gordon and Olshen 1985). For example, Therneau et al. (1990) suggested using regression trees with null martingale residuals from a Cox proportional hazards model as the outcome variable. Various splitting criteria for survival trees have been proposed, e.g., a measure of within-node homogeneity based on the negative log-likelihood of the exponential model within a node (Davis and Anderson 1989), deviance residual (LeBlanc and Crowley 1992), weighted impurity based on the observed times and proportions of censored and uncensored subjects in a node (Zhang 1995), and two-sample log-rank statistics for the separation in survival times between child nodes (Segal 1988). Approaches have also been developed to extend the tree-based models to piecewise linear Poisson and logistic regression (e.g., Chaudhuri et al. 1995; Loh 2006) and longitudinal and multi-response variables (e.g., Loh and Zheng 2013) including in a context of identification of subgroup with differential treatment effect (Loh et al. 2016).

One issue with trees is their notorious instability, which is difficult to reduce even with tree pruning strategies. Small changes in the training data set may lead to trees with very different splits. Due to a hierarchical process of tree fitting, any errors in the top layers of the tree propagate all the way down. One way of dealing with this problem is bagging and random forests discussed below. These methods are based on averaging over a collection of trees fitted to different random samples of the data, which substantially reduces variability inherent in individual trees and typically results in improved prediction accuracy.

There are several R packages that implement decision tree methods, for example, the **rpart** (essentially implementing the CART algorithm), **party** (based on conditional tree platform), and **RWeka** packages. Other R packages such as **rattle**, **rpart.plot**, and **RColorBrewer** (a general-purpose color palette package) provide additional functions for visualizing the trees. Other examples of CART commercial implementations include SAS® Enterprise Miner, IBM® SPSS® Decision Trees, and a package by the Salford Systems. The non-R-based package GUIDE implements a number of methods developed by W-Y Loh and colleagues over the last 20 years with a common thread of unbiased variable selection (signified by the "U" letter in the acronym).

6.3.1.3 Bagging

We have mentioned earlier the general idea of ensemble learning as a way to reduce inherent variability in predictive models. Tree-based models, for example, can benefit greatly from this approach as they are flexible enough to represent complex functions (low bias), yet suffer from the high variance. One type of ensemble learning is bagging (Breiman 1996)—a term that is a contraction of *b*ootstrap *agg*regation. As the component terms suggest, the idea is to form B bootstrap data sets from the original data and fit a separate prediction model $\widehat{f}^{b}(\boldsymbol{x}), b = 1, \ldots, B$ to each of them. Then an aggregated (bagged) prediction is obtained as

$$\widehat{f}_{bag}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}\widehat{f}^{b}(\boldsymbol{x}).$$

Bagging reduces the variance component of the generalization error, especially when used with highly unstable models, such as regression trees. Because averaging leaves the bias component unchanged, it improves the predictive accuracy in general. Typically, to ensure the low bias, averaging is applied to full-sized (unpruned) trees.

Analyses using bagging can be carried out using R packages **ipred** and **adabag**.

6.3.1.4 Random Forests

Random forest (Breiman 2001a, b) is another ensemble learning approach that adds to bagging yet another stochastic element: for each candidate split in the learning process, only a random subset of variables is considered ($r \leq p$ of input variables). This is done to reduce the correlation in the base trees leading to enhancing variance-lowering advantages of bagging. When a collection of B trees is obtained on bootstrap samples as in bagging, the trees can exhibit high correlation if there are variables that are strong predictors of outcome, as these same predictors are very likely to be selected as primary splitters in many trees. For bagging to be effective, the base learners should be less dependent (ensuring larger diversity). This is because the average of B identically distributed random variables has the variance of

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2,$$

where ρ is a pairwise correlation and σ^2 is a variance of each variable. The trees estimated from bootstrap samples are identically distributed, and so the second term diminishes with increasing number of bootstrap samples B, but the first term becomes a bottleneck for variance decrease if the correlation ρ is high.

Selecting the best split from a random subset of variables increases the diversity of the ensemble and prevents few "winners" to dominate all the trees. A discussion of how bagging and random subspace projection together improve accuracy can be found in Ho (2002). The smaller the number of variables used to select each split is, the more reduction in correlation could be achieved (thus reduction in variance). At the same time, as the total number of variables grows, and the fraction of relevant variables decreases, the performance of random forests will degrade with a small number of variables sampled for each split because the bias of each tree will increase. A general recommendation is to use $\sqrt{p}$ variables to choose each split in classification problems and $p/3$ (with a minimum of 5) variables in regression problems. This number can also be treated as one of the tuning parameters of the algorithm. The number of trees in the random forest, B, is another tuning parameter, and both can be optimized, for example, using cross-validation. In principle, the depth of the individual trees could also be a tuning parameter, but Segal (2004) demonstrated that only small gains in performance could be achieved by optimizing this aspect, and so full-grown trees are typically used.

After B full-sized (unpruned) trees are constructed using the random forest algorithm, the overall predictor is determined in the same way as in bagging, e.g., as the average of all trees' predictions for regression, and a majority vote across predictions from all trees for classification.

The performance of random forests is often very similar to boosting—another powerful approach that we summarize below—and it often requires relatively little tuning (offer a more "automated" approach compared to boosting).

One of the bonus features of the random forest algorithm is that it provides an estimate of the generalization accuracy based on the out-of-bag (OOB) samples. As previously discussed (Sect. 6.2), test samples not included in the training data play a key role in estimating the generalization error. In the context of random forests, it can be achieved by constructing the overall prediction for each observation $\boldsymbol{x}_i$ in the original full training set by using predictions only from those trees that were estimated from the bootstrap samples where the observation $(\boldsymbol{x}_i, y_i)$ was not included. Although the OOB error estimates are very similar to those produced by cross-validation, with random forests, generalization error estimation can utilize the same bootstrap replicates that were used for model fitting and thus does not create a computational overhead. Wager et al. (2014) extended an earlier work on estimation of the variance for bagged predictors and proposed an efficient approach to variance estimation based on jackknife and infinitesimal jackknife estimators. In this work, Wager et al. addressed potential upward bias in bagged variance estimates due to the Monte Carlo noise resulting from a finite number of bootstrap replicates. They developed bias-corrected versions of the jackknife and infinitesimal jackknife estimators.

The random forest method also provides variable importance scores to rank predictive strengths of all input variables. Variable importance is naturally defined in the context of tree-based models. Any input variable X_j, $j = 1, \ldots, p$ can be involved in multiple splits across the tree and thus contribute to the reduction of node impurity. One way of defining the relative importance of variable X_j in a tree T is as follows:

$$I(j, T) \propto \sqrt{\sum_{\tau \in T} a\left(s_j, \tau\right) \Delta Q(\tau)},$$

where the sum is over nodes τ in the tree T, function $a(s_j, \tau)$ reflects whether variable X_j is involved in a main or surrogate splitting rule s_j in node τ, and $\Delta Q(\tau)$ is a reduction in node impurity due to a split of this node. Variable importance scores estimated within each individual tree are then accumulated over all trees for each variable. Additionally, permutation-based variable importance is also computed by random forests utilizing the OOB samples. This is done by first estimating prediction accuracy on the OOB samples for each tree separately. Then, to estimate the variable importance of variable X_j, values of this variable are randomly permuted across OOB samples, and prediction accuracy is again estimated in those permuted observations for each tree (i.e., without refitting the trees). The decrease in prediction accuracy as a result of permutation is then calculated and averaged across all trees. Although the ranking of variables according to their variable importance scores tends to agree for these two methods, there are some differences in the distribution of the scores.

Other useful features of random forests include computation of proximities between observations that can be used for clustering and locating outliers. Marginal effects of the individual variables on the outcome can also be estimated, which we will illustrate in a case study in Sect. 6.5.3.

The elements of the random forest method as discussed here were introduced by Breiman (2001a, b) following his development of bagging (1996); of note, the term "random forest" was introduced earlier by Ho (1995) in the context of a method based on a consensus of trees estimated in random subspaces of input features. Research and improvement of the random forest methodology continue, e.g., by Xu (2013). Random forests have had many uses in large applications in genomics and proteomics as well as in the analysis of clinical data. The following are just some examples of the latter applications: predicting disease risk from medical diagnosis history using Healthcare Cost and Utilization Project data set (Khalilia et al. 2011), detection and prediction of Alzheimer's disease (Lebedev et al. 2014), multidimensional clinical phenotyping of an adult cystic fibrosis patient population (Conrad and Bailey 2015), and clustering of patients based on tissue marker data (Shi et al. 2005).

Several variants, extensions, and methods related to the random forest methodology have been introduced over the years. A relationship between random forests and the adaptive nearest neighbor algorithm was pointed out by Lin and Jeon (2006)—both approaches can be viewed as so-called weighted neighborhoods schemes. Friedman and Hall (2007) suggested that subsampling without replacement can be used as an alternative to bagging and demonstrated that fitting trees on samples of size $N/2$ achieves approximately the same performance as bagging, and using smaller fractions of N can reduce the variance even further. The "extremely randomized forest" is an extension of the random forest method where both a subset of input variables and their possible split values are selected randomly when considering candidates for each split (Geurts et al. 2006).

Instead of fitting decision trees as base learners in the random forest, other alternatives have been proposed, e.g., multinomial logistic regression and naive Bayes classifiers (Prinzie and Van den Poel 2008) and naïve Bayes models (Aridas et al. 2016). Other variants of random forests include multivariate random forests (Segal and Xiao 2011), enriched random forests (Amaratunga et al. 2008), quantile regression forests (Meinshausen 2006), and random survival forests (Ishwaran et al. 2008).

Random forest software maintained by a collaborator of Leo Breiman, Adele Cutler, is publicly available online (http://www.math.usu.edu/~adele/forests/). There are also several R packages, such as **randomForest**, **randomSurvivalForest**, **extraTrees** (extreme random forest), and **varSelRF** (variable selection using random forest) implementing this methodology.

6.3.1.5 Boosting

Boosting represents another family of ensemble learning algorithms. It was originally introduced for classification as a way to combine many weak learners to produce a powerful "committee." One of the most popular boosting algorithms introduced by Freund and Schapire (1997), called AdaBoost, relies on a set of weak classifiers whose error rate is only slightly better than random guessing. The weak classification algorithm is applied M times to modified—weighted—training data sets. At the first iteration, weights of all data samples are equal $w_i = 1/N$, and a classifier $\widehat{f}_1(\boldsymbol{x})$ is estimated based on a data set with such weights. At subsequent iterations, $m = 2, 3, \ldots, M$, weights are increased for those observations that were misclassified by the classifier from the previous iteration, $\widehat{f}_{m-1}(\boldsymbol{x})$, and decreased for observations that were classified correctly. As the algorithm progresses, successive classifiers are compelled to focus on "difficult" cases missed by previous classifiers. The overall classification at the end is obtained as

$$\widehat{f}_{AdaBoost}(\boldsymbol{x}) = sign\left[\sum_{m=1}^{M} \alpha_m \widehat{f}_m(\boldsymbol{x})\right],$$

where α_m are the weights determining the contribution of each learner based on its weighted training error. Boosting can dramatically increase the accuracy of very weak single classifiers (those that are just slightly better than random guessing) and outperform large single classification trees. Breiman called AdaBoost as the "best off-the-shelf classifier in the world," and the ideas have since been extended to regression as well.

As it was shown later (Friedman et al. 2000), the AdaBoost is a special case of a general class of forward stagewise additive modeling, where the overall predictor consists of an additive model in some basis functions (base learners) $h(\boldsymbol{x}, \beta)$, with each function fitted and added sequentially without changing parameters of the previously fitted functions so that the overall prediction can be improved. After fitting an initial model, at each subsequent stage, $m = 2, 3, \ldots, M$, the overall additive model from the previous stage is expanded as $\widehat{f}_m(\boldsymbol{x}) = \widehat{f}_{m-1}(\boldsymbol{x}) + h_m\left(\boldsymbol{x}, \widehat{\boldsymbol{\beta}}\right)$, so that the additional component $h_m\left(\boldsymbol{x}, \widehat{\boldsymbol{\beta}}\right)$ improves the performance of the previous model. Least Angle Regression is a computationally efficient version of the stagewise approach. Its details and connections to the lasso regression can be found in Efron et al. (2004).

The base learner $h(\boldsymbol{x}, \beta)$ can be selected from a variety of choices: a linear model (e.g., ordinary linear regression with few best selected predictors), a smooth model (e.g., spline), or a shallow tree (perhaps the most common choice). The complexity of the base learner (e.g., the number of terminal nodes of the tree) is controlled by the user.

Gradient boosting (Friedman 2001; Mason et al. 2000) is a generalization of the stagewise modeling approach that allows the optimization of any differentiable loss function and applies to both regression and classification. Stochastic gradient boosting proposed by Friedman (1999) also incorporated some "elements" of bagging, where each successive base learner is fitted on a subset of the training data set drawn at random without replacement. Friedman reported significant improvements with the stochastic gradient boosting when the size of the subsample is between 50% and 80% of the original data set size. The subsampling strategy also allows for estimation of the generalization error using "out-of-sample" observations, similar to as it was described for bootstrap.

Another element that may help prevent overfitting and improve the accuracy of boosting is introduction of regularization (shrinkage) parameter $0 < \nu \leq 1$, so that instead of adding the "full fit" for each successive learner, only a portion of the fit $\nu h_m\left(\boldsymbol{x}, \widehat{\boldsymbol{\beta}}\right)$ is added. The meta-parameter ν is also called "learning rate" parameter because it controls how much is learned from the training data at each step. To summarize, gradient boosting implements several elements of "slow learning" that prevents adapting to the random features of the training data and ensures its ability to generalize well to the "new" data (prediction performance):

- Forward stagewise strategy (not updating previously fitted components of the ensemble when adding new fit)
- Subsampling (fitting a subsequent learner to a random fraction of the training data)
- Shrinkage (adding only a portion of the new fit)

The two most important complexity parameters that control gradient boosting are the number of iterations (fitted models, M) and learning rate (ν), and they are selected by cross-validation. Typically, the smaller the learning rate, the larger is the number of iterations required to achieve a good fit (i.e., until the model starts overfitting training data).

Both random forest and gradient boosting are ensemble methods and have shown comparable performance on a variety of benchmark data sets. However, as boosting is connected with forward stagewise modeling, its theoretical properties are more amenable to analyses compared to random forest that appears more as a "black box" and a highly heuristic method. In particular, the fact that model complexity of the base learner is controlled by the analyst and kept at relatively low level makes boosting a useful tool for understanding the underlying structure of the data. By fitting different boosting models with a tree as the base learner and having varying depths (the number of terminal nodes, K), one can assess the presence of k-order interaction effects in the data. For example, if $K = 2$, only main effects can be captured by a boosting model, as each fitted tree is a "stump" (split on a single variable) capturing only the main effect of that variable; if $K = 3$, two-way interactions can be captured, etc. Friedman and Popescu (1999) developed a procedure based on parametric bootstrap for conducting formal significance testing for the presence of interaction effects.

An excellent discussion of boosting from a statistical perspective for estimating complex parametric and nonparametric models, including generalized linear, additive, and survival analysis models, as well as its application to variable selection, can be found in Bühlmann and Horthorn (2007).

Analyses using boosting approaches can be carried out using various R packages: **adabag** (AdaBoost and bagging), **ada** (AdaBoost with some Friedman's modifications), **gbm** (tree-based gradient boosting), **GAMBoost** (boosting with penalized B-splines), **mboost** (boosting with high-dimensional (generalized) linear or smooth additive models and a possibility to supply own implementation of any negative gradient for general surrogate loss functions), and **xgboost** (Extreme Gradient Boosting, which can automatically do parallel computation on a single machine and includes many common objective functions with the flexibility of allowing for customized objective functions). Two recent additions worth noting are **bujar** (implementing boosting for survival data) and **bst** (implementing a method called twin boosting; Bühlmann and Hothorn 2010).

6.3.1.6 Support Vector Machines

A support vector machine (SVM) is a supervised learning method which was originally developed for classification and later extended to regression. In the context of classification, it is based on a concept of decision boundary or a hyperplane that separates a set of objects in the space of their attributes belonging to different classes. The algorithm tries to learn a parameterized hyperplane that maximizes the margin between the hyperplane and the closest training examples in each class. For example, in a p-dimensional input space and a binary classification problem, a linear SVM classifier is the one that can separate two response classes using a $(p - 1)$-dimensional hyperplane. If data cannot be perfectly separated and classes overlap in the input space, one can define a hinge loss function that allows some samples to fall on the wrong side of the margin (giving zero penalty to samples inside the margin and linearly increasing penalty for those on the wrong side), thus introducing a cost of each misclassification. This leads to a constrained optimization problem, and a classifier of this type can be estimated using a quadratic programming solution with Lagrange multipliers. An extension of this idea is that if a satisfactory linear classifier cannot be obtained, then the input space can be mapped/transformed to a higher-dimensional feature space and then a linear classifier can be built in that feature space. This is reminiscent of other linear methods that expand the model complexity by using, for example, basis expansions such as polynomials or splines. In the case of SVMs, the feature space is allowed to have a very high dimensionality, but the learning algorithm deals with this efficiently by using a hinge loss function and a form of regularization. Indeed, it can be shown (Hastie et al. 2009) that the solution to the constrained optimization posed by SVM is equivalent to estimating a classification model with the hinge loss function and quadratic (ridge) penalty on the coefficients.

The elements that form the foundation of SVMs have been introduced by Vapnik (1996). Other references for introductory reading include a tutorial by Burges (1998) and Evgeniou et al. (2000). Examples of SVM use for clinical data analysis include applications in diabetes research (Kavakiotis et al. 2017), classification of major depressive disorder (Sacchet et al. 2015), breast cancer diagnosis (Zheng et al. 2014), predicting Alzheimer's disease using linguistic deficits and biomarkers (Orimaye et al. 2017), etc.

SVMs are a member of a more general class of kernel methods (Shawe-Taylor and Cristianini 2004) based on the use of kernel functions that operate in a high-dimensional feature space by computing inner products between mappings of data pairs in the feature space. This approach is sometimes referred to as a "kernel trick" because the operations involving inner products in the feature space are computationally cheaper than the ones in the original space. Kernel-based algorithms include Gaussian processes, principal components analysis (PCA), canonical correlation analysis, ridge regression, spectral clustering, and others.

SVM implementations are available in R in the **SVM** and **svmpath** packages, SAS® Enterprise Miner®, a C-based **SVM$^{\mathbf{light}}$** package from Cornell University (available at http://svmlight.joachims.org/), and many other packages publicly available online.

6.3.1.7 Artificial Neural Networks

The roots of neural networks can be traced back both to statistics and machine learning. The basic principle is to create features as linear combinations of inputs and then to fit a predictive function as a nonlinear function of these derived features. In statistics, this idea was used, for example, in the projection pursuit regression, which is based on an additive model of a collection of nonlinear nonparametric functions of the linear combination of the original predictors. The richness of such models in terms of their capacity to represent arbitrary complex functions increases as the number of these feature functions grows, but this also comes at the cost of low interpretability—resulting in a so-called "black box" method.

The term artificial neural network emerged in the fields of artificial intelligence and machine learning, where parallels between the underlying computational model and brain functioning were drawn. A basic artificial neural network model is often represented in the form of a directed graph consisting of several layers of units, the first layer representing the original predictor variables and the last layer the outcome for regression or classification. Values (signals) from one layer are fed into units of the subsequent layer, where units are thought of as representing neurons. Signals pass through the connections representing synapses, which can "weight" the input signals upon entry to the neuron. Each unit then represents a weighted linear combination of its inputs and produces an output signal when the weighted combination exceeds some threshold (i.e., the neuron fires). The outputs from the units in one layer can then be fed as inputs to the units of the subsequent layer. The function regulating "firing" of each neuron can be a step function or a smoother alternative such as the sigmoid function. Fitting such models can be done using a gradient descent approach, referred to in the neural network literature as back-propagation, as it relies on the chain rule for differentiation. Neural networks can be prone to overfitting, and approaches similar to regularization have been developed to deal with the issue. Performance can be sensitive to the initial values of the synaptic weights and to the scale of the input values. The number of units and inner layers typically also needs to be tuned, for example, using cross-validation. The gradient descent-based learning is prone to converge to local optima and benefits from introducing randomness into the starting values of synaptic weights as well as from the use of bagging. Some researches consider artificial neural networks to be a foundation for advances in large-scale machine learning due to their ability to produce highly accurate predictive models in a wide range of applications, including image and sound recognition, text processing, time series analysis, etc. Recently neural networks experienced a revival under the name of "deep learning," to a large extent due to the availability of increased computational power allowing building networks with a larger number of layers than before and some additional improvements in the network architecture (Efron and Hastie 2016). The black box nature of neural networks and the fact that they can be difficult to tune represent some of their disadvantages, which may be particularly important in the context of clinical data mining where the focus is often on generating new, interpretable insights about the data patterns.

Projection pursuit method is due to Friedman and Tukey (1974) and Friedman and Stuetzle (1981). Modern approaches to neural networks are developed by Werbos (1975) and Rumelhart et al. (1986). The book by Ripley (1996) provides an excellent further reading. Applications of neural networks in the clinical data analysis are numerous. Recent examples include a cardiac health prognostic system

(Sunkaria et al. 2014), cancer prognosis and prediction (Konstantina et al. 2015), applications in prostate cancer (Cosma et al. 2017), prediction of pregnancy outcomes in women with systemic lupus erythematous (Paydar et al. 2017), etc.

The original neural networks algorithm has been extended in many ways over the years, giving rise to feed-forward, recurrent, probabilistic, modular, and neuro-fuzzy neural networks and many other variations.

Several R packages offer neural networks learning, such as **nnet**, **neuralnet**, **H2O**, **DARCH**, **deepnet**, and **mxnet**; SAS® Enterprise Miner® provides neural networks functionality, and there are many publicly and commercially available software packages online.

6.3.2 Unsupervised Learning

6.3.2.1 Clustering

Clustering is one of the major applications of unsupervised machine learning. The goal of clustering is to find a grouping (a set of clusters) of a collection of objects, described by their input attributes so that the objects assigned to the same cluster are more similar to each other than to objects in other clusters. Sometimes clusters may need to be arranged in a hierarchy to reflect some natural structures in the data. Once clusters are learned from the data, some descriptive summary attributes may be of interest to describe specific properties of objects within each cluster.

A key concept in clustering is a definition of similarity measure, or conversely dissimilarity measure, based on which the relationship between objects can be evaluated. There are several frequently used measures, but an appropriate choice is often dictated by domain knowledge. In general, the choice of the dissimilarity measure is very important and can have a crucial effect on the resulting clustering (some say that even more so than the choice of the clustering algorithm).

The classical K-means algorithm can be used, for example, when all input variables are quantitative, and the dissimilarity measure is based on the average squared distance—the Euclidean distance. When the Euclidian distance is used as a measure of dissimilarity, the algorithm may be sensitive to outliers, as the observations with the largest distance will have a significant influence on the loss function. For this reason as well as to allow non-quantitative input attributes, the algorithm can be generalized to use other appropriate dissimilarity measures. This more general approach is often referred to as K-medoids method.

The term K-means clustering was first used in MacQueen (1967) although the underlying ideas were introduced by Lloyd (1957) and basically the same algorithm was published by Forgy (1965). The more general K-medoid procedure was described in Kaufman and Rousseeuw (1990). Description of this algorithm can be found in many machine learning textbooks, e.g., Hastie et al. (2009). Some recent examples of applications of K-means clustering in the analysis of clinical data include identifying subgroups of fibromyalgia patients with different forms of disease and outcomes (Docampo et al. 2013; Lipkovich et al. 2014), identifying clinical phenotypes in chronic obstructive pulmonary disease patients with multiple comorbidities (Burgel et al. 2014), phenotyping of severe asthma patients (Wu et al. 2014) and bipolar disorder patients (Wu et al. 2017).

K-means clustering is closely related to the Expectation-Maximization (EM) algorithm. Parallels between the two can be drawn in that the *K*-means clustering approach, for example, in the case of continuous variables, models each cluster by a spherical Gaussian distribution, but assigns each data sample to a single cluster and uses equal weights to mix cluster distributions. Both algorithms are special cases of modeling with Gaussian mixtures. Another approach that can be viewed as a generalization of the *K*-means clustering is the K-SVD algorithm (Aharon et al. 2006). It uses a set of *K* so-called dictionary functions (e.g., wavelets, curvelets, etc.) to create sparse representations of high-dimensional data as linear combinations of dictionary functions. The algorithm then iterates by alternating between sparse coding of the data samples based on the current dictionary and updating the dictionary to fit the data better. Self-organizing maps (Kohonen 1989) is a related method that can be viewed as a constrained variant of *K*-means clustering where cluster centers are placed on one- or two-dimensional manifolds in a feature space constructed from original variables.

K-means clustering is readily available in R **stats** package through the **kmeans** function. The R package **cluster** implements the algorithm for partitioning around medoids. This package also implements a CLARA algorithm specifically designed to work with large data sets. The R package **clustMixType** implements an extension of *K*-means to mixed data types and package **kml** to longitudinal data. SAS® offers multiple procedures for clustering, including FASTCLUS implementing the *K*-means algorithm.

Hierarchical clustering is another approach which produces a hierarchical representation of data groupings, often graphically depicted by a tree-like structure known as dendrogram: individual observations are associated with the lowest level of the hierarchy (leaves), and the entire data set is represented by the highest level—single cluster (root). There are two approaches for hierarchical clustering: agglomerative and divisive based on either recursive merging or partitioning of the data from the previous level of the hierarchy. Hierarchical algorithms work with a measure of dissimilarity between disjoint groups of observations represented by nodes in the hierarchy, which in turn is based on a measure of dissimilarity between individual observations. In agglomerative strategies, the dissimilarity between clusters that are merged from one level to the next is monotone increasing, and the dendrogram is typically drawn such that the height of each node is proportional to the dissimilarity between its two child nodes. Dendrograms provide a graphical summary of the data, but not an obvious description of the clusters, and represent the structure imposed by the algorithm as applied to a particular training sample, which may not necessarily reflect any natural hierarchy in the domain.

Early approaches to agglomerative hierarchical clustering are due to Ward (1963), Macnaughton Smith et al. (1965), Sibson (1973), and Defays (1977). The books by Kaufman and Rousseeuw (1990) and Hastie et al. (2009) provide a good discussion of various clustering algorithms, including hierarchical approaches. Some recent applications of hierarchical clustering in clinical data analysis include finding groups of fibromyalgia patients with similar efficacy outcomes across multiple symptom scales (Abtroun et al. 2016), identification of biomarkers for tuberculosis susceptibility (Luo et al. 2014), identifying distinct hemostatic responses to trauma and key components of the hemostatic system that vary between responses (White et al. 2015), etc.

In the R **stats** package, hierarchical clustering can be carried out using the **hclust** function. Other R packages implementing hierarchical approaches include **cluster**, **fastcluster**, **fastClus**t, **genie**, and **pvclust**. SAS® procedure CLUSTER provides an implementation of hierarchical clustering. Open-source Cluster 3.0 software is available for most operating systems. Most commercial statistical software packages provide hierarchical clustering functionality.

6.3.2.2 Principal Components and Related Methods

Principal component analysis is a dimensionality reduction method where the goal is to find a low-dimensional representation of the data that captures most of the information (variability) of interest in the data. The classical approach relies on the orthogonal linear transformation of the data to a new coordinate system where the principal components are linear manifolds approximating a set of N p-dimensional data points. Some nonlinear generalizations of PCA have also been developed where the principal components are curved manifold approximations. In the linear case, each component is a linear combination of the p original variables. Principal components are constructed as a sequence of components that are mutually uncorrelated and ordered by variance, so that the first principal component accounts for the largest amount of variability in the data, and each subsequent component has the highest variance subject to a constraint that it is orthogonal to the preceding components.

Principal components are typically constructed by eigenvalue decomposition of a data covariance or correlation matrix or a singular value decomposition of a data matrix usually after applying appropriate standardization of variables (e.g., to have 0 mean and standard deviation of 1, corresponding to PCA on the correlation matrix). The decision of whether PCA should be applied to raw or transformed data and the selection of appropriate transformation depends on various subject-matter considerations. For example, PCA on data covariance allows one to capture nontrivial differences in variances among variables, whereas applying PCA to correlations effectively standardizes the data to have the same (unit) variances. The former makes sense when variables are commensurate: for example, reflecting similar rating scales or the same outcomes measured at different time points. However, if the variables are incommensurate, the differences in variance are not meaningful and may reflect trivial differences in measurement scales. As an "extreme case," consider difference in the variance of two variables representing the same variable "body weight" measured first in pounds and then in kilograms. In such cases, of course, data should be standardized before applying PCA. However, one may consider standardizing to unit variances too radical, as it completely washes away any differences in variability among variables, and prefer other standardization procedures, such as transforming data to vary within a unit range. It is also a common practice to remove outliers from the data before applying the PCA if they can be identified, as the results may be quite sensitive to them. Some variants, such as weighted PCA, have been proposed to improve robustness in this respect.

It is a common practice to visualize multivariate data sets by low-dimensional scatter plots using the first 2–3 principal components. This is a special case of a broad class of *multidimensional scaling* procedures (MDS, Kruskal and Wish 1978). A related visualization technique is the *biplot* display, based on singular value decomposition of (appropriately transformed) data matrix (Gower and Hand 1996; Lipkovich and Smith 2002). In biplots, both data columns and rows are represented graphically: as rays and dots, respectively. The cosines of angles between rays roughly reflect the correlations between variables, and the projections of data points onto the rays reflect the data coordinates (values) in the underlying multidimensional space.

The founding ideas behind the PCA date back to 1901 due to Pearson and 1933 due to Hotelling. PCA is covered in many textbooks, e.g., Jolliffe (2002) is devoted entirely to this method, its applications, and many of its variants. PCA was used to identify distinct patterns of coagulopathy after trauma in Kutcher et al. (2013), to uncover important differences in how patients and informants perceive and report Alzheimer's disease symptoms using the Clinical Meaningfulness in Alzheimer Disease Treatment scale (Jacova et al. 2013), and to explore the association between anemia (hepcidin and hemoglobin levels) and clinical disease activity and acute phase response in patients with rheumatoid arthritis (Padjen et al. 2016).

A counterpart of PCA for analysis of nominal categorical data is a multiple correspondence analysis (MCA) (Greenacre 1984). PCA forms principal components as linear combinations of all input variables which may be problematic in sparse domains where $p > N$. Sparse PCA (Zou et al. 2006) addresses this challenge by looking for linear combinations that involve a subset of just a few input variables. PCA is related to the factor analysis (FA, Cattell 1952) and the independent component analysis (Hyvärinen and Oja 2000) that aim to explain joint variations in input variables by unobserved latent variables. Probabilistic PCA (Tipping and Bishop 1999) is a closely related method where principal axes are determined through maximum likelihood estimations of parameters in a latent variable model. Kernel PCA (Schölkopf et al. 1997) uses a "kernel trick" that forms the basis of support vector machines, where data is first nonlinearly mapped into a high-dimensional feature space, and then the PCA is performed in that space, thus generalizing the PCA to a nonlinear setting.

PCA is implemented in many publicly and commercially available packages. In the R's **stats** package, the functions **princomp** and **prcomp** provide the PCA functionality, as well as such packages as **FactoMineR** and **ade4**. In SAS®, procedures PRINCOMP and FACTOR implement the PCA.

6.3.3 *Semi-supervised Learning*

6.3.3.1 Methods for Biomarker and Subgroup Identification from Clinical Trial Data

Various methods for subgroup and biomarker identification have been proposed during the last decade in statistical literature as a response to the need for precision/ personalized medicine: to provide the best treatment for a patient with specific characteristics at a particular time. A comprehensive review can be found in Lipkovich et al. (2017). See also recent review papers focusing on special types of modeling by Lamont et al. (2016), Henderson et al. (2016), Ondra et al. (2016), and Janes et al. (2013).

Broadly, these methods fall within the class of semi-supervised learning, as the goal is predicting treatment contrast for a patient given his or her biomarker profile. Here a biomarker is understood as any patient-specific measure (covariate) taken prior to assigning a treatment. Unlike patient's outcomes, treatment contrasts are not fully observed in the training sample when patients are exposed to only one of the set of possible treatment options (which is typically the case unless a crossover design is entertained). In the literature on the analysis of medical data, biomarkers that are predictive of treatment contrast are called *predictive*, and biomarkers that are predictive of patient's outcomes if left untreated are called *prognostic*. This is somewhat confusing and inconsistent with the general statistical and machine learning terminology where prediction and predictive covariates/variables (predictors) are understood in a broader sense. Nevertheless, we will adopt the above distinction between predictive and prognostic biomarkers that has been accepted by researchers in the area of subgroup analysis. A biomarker can be predictive and prognostic, only prognostic, and only predictive. The latter case corresponds to rare occasions when a biomarker is not predictive of outcomes for the untreated (more broadly, "control") population but is predictive of outcomes in patients who underwent experimental treatment. The distinction depends on the definition of the estimand for measuring treatment contrast: for example, a biomarker may be predictive when measuring the treatment effect in a binary outcome as the difference in proportions but not predictive when using odds ratios and vice versa. Figure 6.3 depicts several situations: when a biomarker is (a) prognostic but not predictive, (b) both prognostic and predictive, (c) predictive but not prognostic, and (d) neither predictive nor prognostic.

A plethora of methods for identifying predictive biomarkers and subgroups (defined in terms of underlying predictive biomarkers) emerged in the recent literature from diverse research areas: machine learning, causal inference, multiple testing, and design and analysis of clinical trials (see Lipkovich et al. 2017). To facilitate our review, we will introduce some minimal notation that will help us in describing common features and differences across subgroup identification methods.

Define a vector of p measurements for candidate biomarkers $X_1, \ldots, X_p$ on the i^{th} subject as $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ and (for the sake of simplicity) two treatment arms

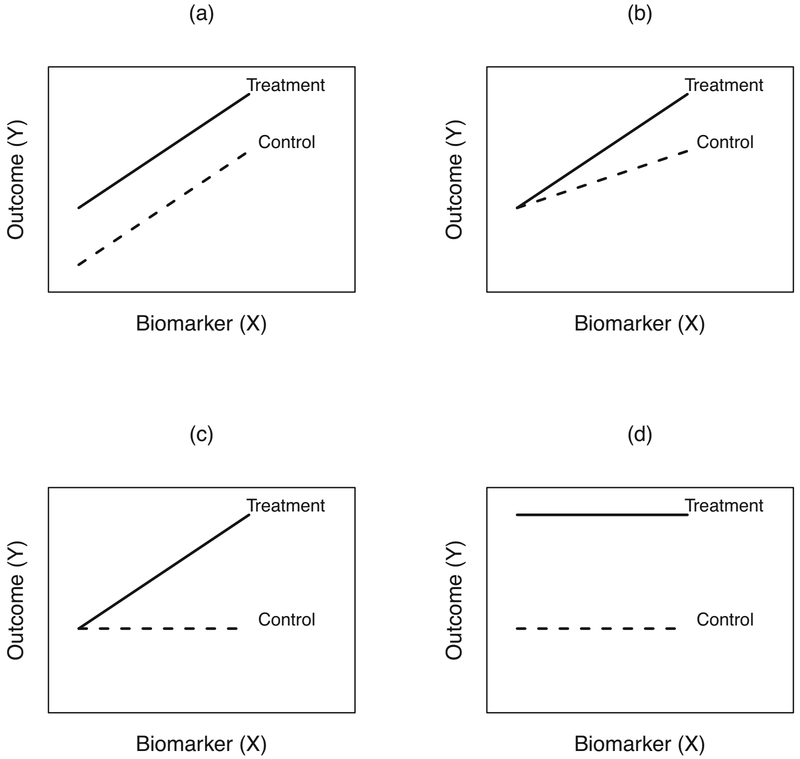

Fig. 6.3 Schematics of predictive and prognostic markers: (**a**) prognostic but not predictive, (**b**) both prognostic and predictive, (**c**) predictive but not prognostic, and (**d**) neither predictive nor prognostic

(control and experimental treatment) indexed by variable $T = \{0, 1\}$ and the outcome variable Y (for simplicity, assume Y is continuous or binary).

Assume the true response function has the following simple representation: $f(\boldsymbol{x}, t) = h(\boldsymbol{x}) + (t - p) \times z(\boldsymbol{x})$, where $f(\boldsymbol{x}, t) \equiv E(Y|\boldsymbol{X} = \boldsymbol{x}, T = t)$; therefore, $z(\boldsymbol{x}) = f(\boldsymbol{x}, 1) - f(\boldsymbol{x}, 0)$ is the "personalized" treatment contrast, and $h(\boldsymbol{x})$ is an unspecified "baseline" function. The constant p is often conveniently used to represent the probability of treatment, $p=$ Pr $(T = 1)$.

Adopting potential outcomes framework (see a review paper by Little and Rubin (2000) and references therein), for each subject we define two potential (hypothetical) outcomes $Y(1)$ and $Y(0)$ with only one being observed. These outcomes are connected with the observed data via the consistency assumption as $Y = Y(1)T + Y(0)(1 - T)$ and with the above quantities as $f(\boldsymbol{x}, t) = E(Y(t)|\boldsymbol{X} = \boldsymbol{x})$, $t = 0$, 1 and $\mathrm{z}(\boldsymbol{x}) = E(Y(1) - Y(0)|\boldsymbol{X} = \boldsymbol{x})$.

Treatment regime is a function, $g(\boldsymbol{x})$, that maps $\boldsymbol{x}$ to treatment 1 or 0. Optimal treatment regime is defined as $g_{opt}(\boldsymbol{x}) = I(f(\boldsymbol{x}, 1) > f(\boldsymbol{x}, 0))$, where $I(a)$ is an indicator function that assumes values 1 or 0 when its argument a is true or false, respectively. In terms of potential outcomes, it is the regime that maximizes the expected potential outcomes assuming everyone follows the regime.

Subgroup $S(\boldsymbol{X})$ is defined by a rule that assigns a subject to the subgroup based on a biomarker vector $\boldsymbol{x}$. For example, $S(\boldsymbol{X}) = \{\boldsymbol{x} : X_1 \leq a, X_2 > b\}$. The subgroup definition is also referred to as subgroup signature. Then treatment effect in the subgroup $S(\boldsymbol{X})$ is defined as $E(z(\boldsymbol{X})|\boldsymbol{X} \in S(\boldsymbol{X}))$, where the expectation is computed with respect to the distribution of $\boldsymbol{X}$. A measure of how interesting a subgroup may be is the excess of treatment effect in the subgroup over that in the overall population: $E(z(\boldsymbol{X})|\boldsymbol{X} \in S(\boldsymbol{X})) - E(z(\boldsymbol{X}))$.

The distinction between predictive and prognostic biomarkers can be formalized as follows: prognostic biomarkers are those that contribute only to $h(\boldsymbol{x})$ (the "main effects"), whereas predictive biomarkers are those that also contribute to $z(\boldsymbol{x})$ (and perhaps to $h(\boldsymbol{x})$ as well).

Based on the above definitions, we can classify different methods that have been proposed recently for selection of predictive biomarkers (and choosing biomarker cutoffs to define subgroups of patients) in terms of what *estimands* (functions or components of these functions) they aim to estimate.

- *Global outcome modeling*: estimating the underlying outcome function $f(\boldsymbol{x}, t)$
- *Global treatment effect modeling*: directly estimating the underlying treatment effect $z(\boldsymbol{x})$
- *Global modeling of treatment regimes:* identifying an optimal treatment assignment rule that produces positive treatment contrast given patients' covariates $g_{opt}(\boldsymbol{x}) = I(z(\boldsymbol{x}) > 0)$
- *Local modeling*: direct search for subgroups with a beneficial treatment effect, i.e., identifying subgroups $S(\boldsymbol{X})$ in the covariate space with large values of treatment effect, such that $z(\boldsymbol{x}) > \delta$, for all $\boldsymbol{x} \in S(\boldsymbol{X})$

As a by-product or an intermediate step of many of these approaches, predictive biomarkers can be identified and ranked; also, optimal cut points associated with biomarkers are often evaluated.

This typology is meant to facilitate the discussion of different methods for subgroup modeling and search and show connections between them. Clearly, these classes are not mutually exclusive as the quantities that different methods estimate are interconnected.

In what follows, we provide a brief description of existing approaches within each class.

Global outcome modeling. Many approaches within this class estimate a single response model that incorporates both main effects (prognostic effects) and treatment by covariate interactions (predictive effects). Alternatively, separate regression models for estimating outcomes within each treatment arm can be entertained. Constructing subgroups typically requires multistage procedures: for example, at the first stage of the Virtual Twins method of Foster et al. (2011), $f(\boldsymbol{x}, t)$, $t = 0, 1$ is estimated using a *black box* model (random forests) fitted to the observed data, which is used to compute hypothetical individual treatment differences $\widehat{z}(\boldsymbol{x}) = \widehat{f}(\boldsymbol{x}, 1) - \widehat{f}(\boldsymbol{x}, 0)$ for each subject with an observed covariate vector $\boldsymbol{x}$. These differences are modeled at the second stage as outcomes via CART procedure. Some researchers advocate for more traditional parametric regression approaches. Since fitting parametric models with a large number of interaction terms poses a lot of problems, methods of penalized regression and their extensions have been proposed to mitigate some of these issues. One challenge in modeling outcomes via penalized regression is that it may fail to detect important treatment-by-biomarker interactions that may be obscured by much stronger main effects (i.e., the effects of prognostic biomarkers), which may require using different penalties for the main and interaction terms (as in FindIt, approach by Imai and Ratkovic 2013). Some methods use a combination of parametric and nonparametric modeling. For instance, Cai et al. (2011) use a combination of a proportional hazards Cox regression at the first stage and nonparametric smoothing at the second; and Dusseldorp et al. (2010) fit simultaneously an additive model (STIMA) for prognostic effects and a tree-based regression model for predictive effects. Bayesian hierarchical modeling of the response function with prognostic and predictive effects includes an early proposal by Dixon and Simon (1991), Smoothing ANOVA by Hodges et al. (2007), and methods based on Bayesian lasso (Gu et al. 2013). Recently, Henderson et al. (2017) have proposed a fully nonparametric Bayesian approach to subgroup evaluation in the context of accelerated failure time models (AFTM) for survival outcomes where the regression function is modeled using Bayesian additive regression trees (BART) and the error function—via a flexible location mixture of normal densities.

Global treatment effect modeling. Approaches in this class obviate the need to model prognostic or "main effects" which "cancel out" in the course of modeling. As a result, procedures in this class may be more robust compared to the global outcome modeling, as they would not be so prone to model misspecification inevitable in global outcome models. For example, in trees, pricewise constant estimates of $z(\boldsymbol{x})$ are obtained simply as treatment effect statistics computed within each terminal node of a tree. The key contributions are Interaction Trees (IT) of Su et al. (2008, 2009) and several new tree-based procedures proposed in Loh et al. (2015, 2016) within the GUIDE recursive partitioning platform. Similarly, Seibold et al. (2015) illustrated how the model-based recursive partitioning (MOB) platform could be adopted for the purpose of subgroup identification by incorporating treatment effect in the models considered within each leaf of the tree. Dusseldorp and Van Mechelen (2014) introduced a tree-based algorithm (QUINT) for subgroup identification that specifically aims at recovering qualitative interactions.

Individualized treatment regimes modeling. Note that the optimal regime can be determined based on the estimated treatment contrast (as in the previous approaches): $g_{opt}(\boldsymbol{x}) = I(z(\boldsymbol{x}) > 0))$. This approach, however, obviates the need of estimating $z(\boldsymbol{x})$ and directly targets the sign of $z(\boldsymbol{x})$ as a binary outcome; it aims at identifying *qualitative* interaction effects. If $z(\boldsymbol{x}) > 0$ for all $\boldsymbol{x}$ (i.e., the drug has beneficial effect for all patients), the potential outcome $Y(1)$ can often be redefined by taking into account a fixed "treatment burden" or "cost." $\delta > 0$. For example, defining $\widetilde{Y}(1) = Y(1) - \delta$ results in shifting the treatment contrast downward: $\widetilde{z}(\boldsymbol{x}) = z(\boldsymbol{x}) - \delta$, so that values become negative for some $\boldsymbol{x}$ and nontrivial optimal regimes can be identified.

Broadly, this class includes any approach that matches patients to one set of candidate treatments, based on available patient-level data. For example, Qian and Murphy (2011) formulated the problem of finding an optimal individual treatment regime (ITR) using traditional outcome modeling ("global outcome modeling," using our terms). They proposed a two-step procedure that first estimates the conditional mean response using penalized regression (with lasso penalty) allowing the inclusion of a large number of candidate biomarkers and associated treatment interactions and at the second stage derives the optimal treatment assignment rule by inverting the model for the conditional mean.

It became apparent, however, that determining optimal treatment regimes does not require estimating the entire mean response function (which is driven by both prognostic and predictive biomarker effects) but critically depends on identifying only predictive biomarkers associated with *qualitative* (as opposed to *quantitative*) treatment-by-covariate interactions. Gunter et al. (2011) proposed some methods for identifying only biomarkers contributing to qualitative interactions with treatment (and therefore to personalized rules) using resampling procedures that ensure family-wise error rate control. Zhang et al. (2012) and Zhao et al. (2012) showed that estimating optimal individualized treatment policies could be cast as a classification problem. For example, in the weighted outcome learning (OWL) methodology of Zhao et al. (2012), the optimal treatment regime $g_{opt}(\boldsymbol{x})$ is found as the one that minimizes the *weighted* misclassification loss: $E\{I(T \neq g(\boldsymbol{X}))w(Y, \boldsymbol{X})\}$, where the expectation is taken with respect to the triple of random variables $\{Y, \boldsymbol{X}, T\}$, and the subject weights $w(Y, \boldsymbol{X})$ are proportional to the outcome Y (here assuming larger values are desirable) and inversely proportional to the probability of patients following the regime prescribed by the rule $g(\boldsymbol{X})$. Then the optimal rule can be found by standard methods of predictive learning aiming at minimizing misclassification loss via appropriate smooth "surrogate" loss functions (e.g., hinge loss resulting in the SVM classifier, well-known in the machine learning community, or negative binomial log-likelihood, familiar to statisticians, resulting in the penalized logistic regression). Intuitively, minimizing the above weighted loss would tend to recover the optimal rule $g_{opt}(\boldsymbol{x})$ that would recommend the actually received treatment for those patients who achieved good outcomes while suggesting switching the treatment for those who failed the treatment they were assigned to in the trial. This method applies to observational trials as well as randomized clinical trials. In the latter case, the inverse weighting by the probability of treatment assignment is trivial and determined by the randomization ratios; in the former case, estimating propensity of the treatment as a function of baseline covariates $\boldsymbol{X}$ needs to be done as a separate modeling step. Other key references include tree-based approaches by Zhang et al. (2015) and Fu et al. (2016) and penalized regression methods by Huang and Fong (2014) and Xu et al. (2015).

We note that from the perspective of optimal treatment regimes, identified subgroups are the two subpopulations based on the sign of the estimated individual treatment difference, $sign(z(\boldsymbol{x}))$. Some researchers would argue that it may be not sufficient to adequately describe the optimal treatment strategy by considering only two subpopulations of patients (“treat” vs. “non-treat”). For example, Dusseldorp and Van Mechelen (2014) consider three groups: patients who benefit from treatment A vs. B, patients who benefit from B vs. A, and the rest allocated to “indifference” zone comprised of patients for whom either treatment may work equally well or not work. On the other hand, the perspective of subgroups defined by the optimal treatment regime may be different from the idea of identifying “natural subgroups,” say, as rectangular regions or “bumps,” which brings us to the last category in our classification of subgroup methods.

Local modeling. The last class of subgroup search methods focuses on the direct search for treatment-by-covariate interactions and selecting subgroups with desirable characteristics, for example, subgroups with enhanced treatment effect. This approach obviates the need to estimate the response function over the entire covariate space and focuses on identifying specific regions with large differential treatment effect. Some of the approaches under this heading (Kehl and Ulm 2006; Chen et al. 2015) were inspired by bump hunting (also known as PRIM, Patient Rule Induction Methods) by Friedman and Fisher (1999) which is a method of predictive modeling that aims at estimating only regions where a target function (here, the treatment contrast, $z(\boldsymbol{x})$) is large. They argued that it may be better to search directly for such “interesting” regions in the covariate space rather than estimating $z(\boldsymbol{x})$ first in the entire space and then discarding the regions that are “uninteresting.” The main goal of bump hunting methods such as PRIM is to define sets of multivariate rectangular regions based on the candidate covariates $X_1, X_2, \ldots, X_p$. The limits of the region are determined in a data-driven manner using a *peeling* technique. Specifically, extreme values of continuous/ordinal covariates or individual levels of nominal covariates are removed. The peeling algorithm is sequentially applied to single covariates, one at a time, and the order of the covariates is determined by the value of an appropriate objective function.

Another strategy for direct subgroup search was first implemented via a recursive partitioning process in the SIDES method (Subgroup Identification based on Differential Effect Search, Lipkovich et al. 2011) and later extended to the SIDEScreen method (Lipkovich and Dmitrienko 2014). In Sect. 6.5.1 of this chapter, we will apply these methods to a case study and provide additional technical details.

Another member of this group of algorithms is Activity Region Finder (ARF), by Amaratunga and Cabrera (2004), that combines algorithms of CART and the bump hunting to search for high or low response (activity) subgroups. Bayesian methods for local modeling were inspired by the idea to treat each subgroup as a model and apply model averaging to a collection of generated subgroups (Berger et al. 2014; Bornkamp et al. 2016). Schnell et al. (2016) implemented a procedure for identifying subgroups as *credible sets* which comprise points in the covariate space with the sufficiently high posterior probability of associated treatment effect $z(\boldsymbol{x})$ exceeding a pre-specified threshold.

Any method within the four groups can be further characterized with respect to the following features, forming a checklist that a user of any subgroup identification method should bear in mind when evaluating whether the method may be appropriate for the problem at hand (Lipkovich et al. 2017).

- Modeling type: frequentist/Bayesian and within either subtype, parametric, semi-parametric, or nonparametric
- Dimensionality of the covariate space that the method can handle: low (1–5), medium (6–15), or high (>15)
- Results produced by the method: selected biomarkers or biomarker ranking that can be used for tailoring, predictive scores for individual treatment effects, optimal treatment assignment, or identified subgroup(s) as biomarker signatures
- Application of complexity control to prevent data overfitting and selection bias when evaluating candidate subgroups
- Evaluation of the Type I error rates/false discovery rates for the *entire* subgroup search strategy
- Availability of "honest" (bias-corrected) estimates of treatment effects in the identified subgroups

The dimensionality of covariate space that can be handled by a proposed method may vary dramatically. Some methods were originally developed for evaluating treatment by covariate interaction in the context of a *single* continuous covariate and later extended to a small number of pre-selected biomarkers (e.g., the method of fractional polynomials by Royston and Sauerbrei (2004, 2013); the STEPP method by Bonetti and Gelber (2000, 2004); Jones et al. (2011)). These can be contrasted with methods developed with the idea of handling high-dimensional covariate vectors and incorporated variable selection as part of the model building strategy (e.g., Virtual Twins by Foster et al. (2011) and many others referenced below). The middle grounds are occupied by methods assuming that a medium-sized set of candidate biomarkers has been specified, e.g., in the statistical analysis plan (SIDES by Lipkovich et al. 2011; Gi method by Loh et al. 2015). Mayer et al. (2015) describe some findings from a survey with respect to the dimensionality of covariate space and other features of subgroup analysis tasks routinely dealt with by Pharma statisticians.

Depending on the method, different results may be produced. For example, some methods are searching for "biomarker signatures." These are often defined as rectangles in covariate space (requiring predictive biomarkers and associated cutoff points), which is motivated by the desire to base clinical decisions on simple and easily interpretable rules, e.g., Foster et al. (2015). Other approaches look for arbitrary biomarker signatures (e.g., additive scoring functions) that would allow ranking all patients by a score reflecting predicted individual treatment effect; and some methods provide selection or scoring for predictive biomarkers (such as variable importance scores) that can be used for tailoring in subsequent clinical development programs.

All subgroup selection methods considered here have data-driven elements. However, the scope of search may vary dramatically from selecting a subgroup based on estimated patient's predictive score as a linear combination of, say, 3 pre-specified continuous biomarkers to identifying a subgroup as a "region" formed by selecting 2 out of 1000 candidate biomarkers with an optimal cut point determined within each of these in a data-driven fashion. Depending on the scope of search in the "covariate space" induced by a specific method, different types of complexity control may be needed. The idea is to ensure that the finally selected biomarker signatures defining subgroups are not unnecessarily "complex" (e.g., by including irrelevant or noise covariates), resulting in spurious findings with little chance to be replicated in the future trials. Such situations occur when the set of candidate subgroups/biomarker signatures includes the elements of different "complexity." For example, of two candidate subgroups, one defined by a single biomarker as $\{\text{Age} \leq 20\}$ and the other defined by two biomarkers, $\{\text{Age} < 20\}$ and $\{\text{Gender} = \text{"Female"}\}$, the latter is more "complex." When using a greedy search over possible signatures, it may appear to fit the observed data better and therefore will look more promising than the former simpler subgroup. As in other applications of machine learning, the chance of spurious findings ("overfitting") increases with model complexity, and to offset that, some forms of complexity penalty are required.

Different approaches of complexity control to prevent data overfitting have been proposed in the context of subgroup/biomarker search including:

- Frequentist penalized regression (e.g., Imai and Ratkovic 2013) and Bayesian shrinkage (e.g., Jones et al. 2011)
- Frequentist ensemble learning methods (e.g., Foster et al. 2011) and Bayesian model averaging (Berger et al. 2014; Bornkamp et al. 2016) that aggregate results over a large number of "learners" (here, subgroups or signatures) to shrink the contribution of noise covariates to zero
- Using "indirect" or less direct criteria for variable/subgroup selection that avoid the exhaustive search for subgroups with desired features (Loh et al. 2015, 2016)

Another example of data overfitting (often called biomarker selection bias) arises when making a choice between subgroups based on biomarkers with widely different sets of candidate cutoff points. For example, a subgroup based on patient's age as a continuous variable with a data-driven cutoff, e.g., $\{\text{Age} \leq 20\}$, has a higher potential for overfitting than a subgroup of seemingly equal complexity based on gender, e.g., $\{\text{Gender} = \text{"Female"}\}$. This is because variable Age has a much larger number of candidate splitting points (basically, all values of Age realized in the database except the extreme ones leading to subgroups not passing the minimal sample size requirement), whereas only two subgroups can be selected based on patient's gender. Therefore, if both biomarkers Age and Gender are irrelevant (noise variables), we would have a higher chance of selecting Age as a promising marker if our selection is based on exhaustive evaluation of all possible subgroups based on Age and Gender. Several approaches were proposed in the literature to deal with variable selection bias (Loh et al. 2015; Seibold et al. 2015).

One may think that "complexity control" would be unnecessary if at the end of the subgroup search, we can correctly evaluate the Type I error or false positive rate associated with the selected subgroups. This would be the case if all candidate subgroups in the covariate space were of the same complexity; then it is straightforward to select the one having the largest apparent treatment effect while controlling for multiplicity (selection bias, winner's curse, etc.) by computing an adjusted *P*-value and a bias-corrected treatment effect. However, as was discussed, the fact that different subgroups may have very different "complexity" requires imposing a penalty for complexity *during* the process of selection, thus preventing us from "chasing noise." Adjusting the finally selected subgroups which may be purely driven by noise *after* selection would be too late, as it would merely suggest that our selected subgroup (after appropriate adjustment) has a very large associated *P*-value and therefore would have little reproducibility in future trials. Indeed, the goal is to avoid making such an unfortunate selection by putting a "constraint jacket" on the selection process.

Of course, even with complexity control, we need to account for multiplicity inherent in subgroup identification by computing adjusted *P*-values associated with treatment effects in the selected subgroups. However, procedures that are less "greedy" would require less adjustment of *P*-values than the "greedier" procedures. This is because a less greedy procedure induces a smaller search space by restricting search to models satisfying complexity constraint, hence less multiplicity burden. For example, the multiplicity adjustment for *P*-values in subgroups selected using a very greedy stepwise selection method (in the context of a linear regression with the main effects and treatment by covariate interactions) would be much harsher than the adjustment of *P*-values when the selection is made using much less greedy methods of penalized regression (e.g., lasso). The analytical expressions for multiplicity-adjusted *P*-values in subgroup search methods are typically not available, and researchers have to resort to approximate *P*-values based on various resampling methods (permutations or bootstrap under null scenarios).

Finally, once the subgroup(s) have been identified, the sponsor would need to make a decision based on anticipated treatment effects in these subgroups (e.g., by computing probabilities of success for different designs involving enriched populations versus the overall population). It is important to understand that even if the identified subgroup may be very close to the true one, the apparent treatment effect computed using the same data that was used for subgroup search (a naive method of data "resubstitution") is likely to overestimate the true treatment effect contained in that subgroup. Like with multiplicity-adjusted *P*-values, the size of the treatment effect can be estimated using resampling methods such as bootstrap or cross-validation (see Foster et al. 2011; Faye et al. 2011; Simon et al. 2011; Loh et al. 2016; Rosenkranz 2016); Bayesian methods implementing shrinkage such as an empirical Bayes correction (Ferguson et al. 2013) or model averaging (Bornkamp et al. 2016) can also be used (see Thomas and Bornkamp 2017 for comparison of several methods for estimating treatment effect in data-driven subgroups). The amount of over-optimism in the naïve estimates of treatment effect computed by resubstitution depends on the richness of the search space and the "greediness" of the search algorithm.

Many applications of subgroup identification to real data sets can be found in the original papers introducing the discussed methods. Here, we provide additional references for applications of existing methods to clinical or observational data. Hardin et al. (2013) applied SIDES to conduct an exploratory analysis of a large multinational, randomized, open-label trial in patients with type 2 diabetes to identify subgroups where the effect of an insulin lispro mix versus insulin glargine was substantially different from that in the overall population. Dmitrienko et al. (2015) applied SIDES methods to the ATTAIN program based on two Phase III multinational trials to evaluate the safety and efficacy of telavancin (test antibiotic) compared to vancomycin (active control antibiotic) for treatment of adults with nosocomial pneumonia. Hou et al. (2015) compared the results of several tree-based subgroup identification methods including Interaction Trees and Virtual Twins to the data from an alcohol dependence pharmacogenetic trial of ondansetron. Patel et al. (2016) analyzed patients with low back pain using data pooled from 19 randomized clinical trials applying Interaction Trees, SIDES, and Indirect Network Meta-analysis to identify subgroups defined by multiple parameters. Doubleday (2016) adopted recursive partitioning methods to evaluate individualized treatment assignment rules from both randomized and observational data and applied it to diabetes data from electronic medical records (see also Fu et al. 2016). Seibold et al. (2016) adopted methods of model-based recursive partitioning to construct individual treatment effect predictions for patients with amyotrophic lateral sclerosis pooled from several randomized clinical trials.

Links to several packages that implemented popular subgroup identification methods could be found at the site maintained by the QSPI (Quantitative Sciences in the Pharmaceutical Industry) Subgroup Analysis Working Group: http://biopharmnet.com/subgroup-analysis-software/. These methods include both R packages available in CRAN (**aVirtualTwins**, **SIDES**, **quint**, **FindIt**, **partykit, model4you, personalized**) and implementations with R and other software provided by the developers for public dissemination: an R package RSIDES implementing SIDES and SIDEScreen methods (Lipkovich and Dmitrienko 2014), R code for ROWSi (Regularized Outcome Weighted Subgroup identification) by Xu et al. (2015), the GUIDE package implementing methods by Loh et al. (2015), and BLASSO by Gu et al. (2013).

The above packages focus on the problem of biomarker/subgroup identification. For situations when one or two markers are pre-selected, Janes et al. (2014) propose a statistical framework for evaluating a candidate treatment selection marker and comparing two continuous markers; an R package developed by the authors implementing these methods is available at http://labs.fhcrc.org/janes/index.html.

6.3.3.2 Q-Learning for Dynamic Treatment Regimes

Q-learning is an approximate dynamic programming algorithm for estimation of optimal dynamic treatment regimes (DTRs). DTRs are the sequences of decision rules, one per decision/intervention stage, that map up-to-date patient information to a recommended treatment. The key is that a patient's treatment at each stage is not known at the start of the treatment sequence, as it depends on time-varying variables that may be influenced by earlier treatment choices. In many health disorders, especially chronic conditions, the sequential decision-making is necessary to adapt treatment over time in response to the evolving health status of the patient. This is especially important if there is a high degree of heterogeneity in an individual response to treatment and when treatment may need to be adjusted as a result of emerging side effects. In such cases, the treatment that appears optimal in the short term may not be best in the long term. Thus, the goal is to optimize a long-term outcome of interest which may be measured at the end of the last treatment stage or may encompass intermediate outcomes, e.g., represent a (weighted) average of clinical outcomes across all intervention stages.

Q-learning can be viewed as an extension of regression to multistage decision problems based on backward induction. It starts with an estimation of the optimal treatment rule at the last stage of treatment based on patient-level data up to the last treatment decision, which may include baseline characteristics, treatment decisions, and intermediate outcomes up to that point. This information is used as “independent variables” for a regression model of the long-term outcome measured after the last treatment decision. Based on this regression model, the last stage optimal treatment is estimated for each patient so that it optimizes the expected long-term outcome. Subsequently, Q-learning performs a similar regression and optimization step for a preceding decision stage to find a treatment that would result in optimal long-term outcome assuming that subsequent, last stage treatment will be determined by the optimal rule constructed in step 1 of the procedure. This backward re-estimation and optimization are performed iteratively until the first decision point, allowing the method to account for future decisions when making treatment choices at earlier stages.

Ideally, DTRs should be estimated from trials with a Sequential Multiple Assignment Randomized Trial (SMART) design (Collins et al. 2007; Almirall et al. 2014), where subjects are randomized multiple times during the course of the trial. At each randomization stage, the set of available treatments may depend on subject-specific characteristics and evolving health status. SMART would be a “gold standard” for determining optimal DTRs as they remove any confounding of treatment assignment with subject characteristics, just like randomized controlled trials are a gold standard for confirmatory clinical trials. For ethical and logistical reasons, SMART are rare in the pharmaceutical industry practice. DTRs can also be constructed using observational data from trials with flexible dosing or evolving treatment assignment, e.g., long-term open-label trials in chronic pain or dynamic second, third, etc. line of treatment selection in cancer. Such studies are more common in practice; however, care must be taken to account for potential confounding.

Maintaining a balance between treatment efficacy and limiting undesirable side effects is an important aspect of successful dynamic treatment regimes, but is a relatively open area of research. Composite scores that integrate measures of treatment efficacy and safety could be used. For example, in Wang et al. (2012), a composite score was constructed by eliciting from the Principal Investigator of the trial subjective numerical values to quantify the clinical desirability of efficacy, toxicity, and progressive disease response to a prostate cancer treatment. In many circumstances, it may be difficult to obtain a single composite measure that encompasses patient and physician preferences across several competing and potentially time-varying outcomes. Laber et al. (2014a) proposed an approach to deal with such challenges by using set-valued functions and recommending a set of possible treatments which are non-inferior across outcomes at each decision point and which can then be considered by individual patients and physicians. Another approach, which avoids using subjective composite measures and works directly with the original efficacy and safety outcomes, is to optimize treatment efficacy using Q-learning under constraints of the risk of adverse events so that DTRs can be designed to achieve specific levels of efficacy tailored to patient's adverse event tolerance limit.

Original ideas behind Q-learning can be found in Watkin (1989) and Watkin and Dayan (1992). These ideas have been further developed and adapted to the context of estimation of dynamic multistage treatment strategies, e.g., by Murphy (2005), Schulte et al. (2014), and Laber et al. (2014b). They have also been extended to survival outcomes by Goldberg and Kosorok (2012) and to discrete utilities (long-term outcomes) and to nonlinear relationships between covariates and outcomes by Moodie et al. (2014). A general overview of SMART design considerations can be found, for example, in Almirall et al. (2014).

Examples of clinical applications of Q-learning for estimation of optimal treatment regimes can be found in a number of recent publications. For example, Wu et al. (2015) applied DTR for treatment of acute bipolar depression; Chakraborty et al. (2013) and Chakraborty and Moodie (2013) for chronic illnesses including major depressive disorder; Laber et al. (2014b) and Nahum-Shani et al. (2012) for attention deficit hyperactivity disorder; Laber et al. (2014a) and Shortreed et al. (2011) for schizophrenia; Moodie et al. (2007) and Sterne et al. (2009) for HIV/AIDS; Strecher et al. (2006) for cigarette addiction; and Lei et al. (2012) for prevention of alcoholism relapse.

The term Q-learning refers to the estimation of a Q-function, which stands for the "quality" associated with a specific treatment choice at each stage given the patient's history up to that point and following the optimal regime thereafter. Having an estimate of the "quality" of each possible treatment decision, we can select the best one at each stage. The challenge, in this case, is to obtain a good unbiased estimate of the Q-function over the entire space of histories and possible treatment choices, which may be difficult to achieve, especially over high-dimensional spaces and with relatively sparse data. A-learning (Blatt et al. 2004) is an alternative method, which estimates an "advantage" for each treatment, i.e., the difference between the quality of a given treatment choice and the optimal treatment at each stage. This approach may be less sensitive to bias introduced by the mismodeling of the Q-function. However, A-learning may have a disadvantage of high variability and require variance reduction techniques, such as bagging or random forests, for successful implementations, and its complexity increases with the number of possible treatment choices.

Both Q-learning and A-learning use regression to estimate some function representing the value of the treatment choice and then obtain the optimal decisions by inverting that function. On the other hand, outcome-weighted learning (OWL) uses a different paradigm by attempting to estimate the optimal DTR directly, casting it as a weighted classification problem. Treatments actually received by patients with good observed outcomes are considered to be "correctly classified," i.e., corresponding to the optimal treatment assignment, whereas treatments actually received by patients with poor outcomes are considered to be "misclassified." The method tries to minimize a weighted misclassification error where the weights are proportional to the observed outcome and inversely proportional to the probability of receiving a given treatment given patient history. Powerful classification methods from machine learning, e.g., support vector machines, can be applied in this context. Two variants of this approach, backward outcome weighted learning (BOWL) and simultaneous outcome weighted learning (SOWL), applicable to multistage treatment regimes were proposed by Zhao et al. (2015).

Publicly available tools for Q-learning applicable to DTRs are limited. The **iqLearn** package in R (Linn et al. 2015) can be used for estimating optimal DTRs from data obtained from a two-stage trial with two treatments at each stage. The **DTRlearn** package in R implements both single- and multiple-stage Q-learning OWL approaches. Proc QLEARN developed for SAS v9.1 or higher for Windows by Ertefaie et al. (2012) at the University of Michigan and the Pennsylvania State University (https://methodology.psu.edu/downloads/procqlearn) can be used with data from a sequential, multiple assignments, randomized trial (SMART) but is limited to situations where the outcome is continuous; there are two decision stages and up to two treatment options at each decision.

6.4 Principles of Data Mining with Clinical Data

Here, we focus on data mining in randomized clinical trials (RCTs). As RCTs are conducted in a highly regulated environment, the interpretation of data mining activities with such data may be considered particularly controversial and therefore calls for clearly defined principles to ensure their validity. It is often argued that data mining with clinical data has limited validity since by its nature it cannot be pre-specified and therefore occupies the lowest rank in the Statistical Analysis Plan.

The relative ranking of the importance of analyses undertaken in a Phase 3 trial can be loosely described as follows: the primary analysis of primary outcome, the primary analysis of secondary outcomes, the secondary analyses of primary outcome, the secondary analyses of secondary outcomes, supportive analyses and sensitivity analyses, and exploratory analyses. We note the striking contrast between the wealth of (often underutilized) data collected in the course of clinical trials and the "minimalistic" focus on the primary analysis as the basis for major study conclusions.

On the other hand, in exploratory Phase 2 studies, it is often felt by the sponsor that any data exploration is allowed as long as it is used only for "internal decision-making." However, when the drug development program is driven by unprincipled and unconstrained data exploration in Phase 2, it often results in a failed Phase 3 study, further contributing to a lack of respect for data mining.

As a result of the described mind-set, the exploratory analysis is felt as belonging to the lowest category of analysis on the above scale and is typically described in the section “Exploratory analysis” which is often a polite word for a “garbage collector” to store various poorly conceived data analysis strategies. This practice blurs the subtler distinction of and the relationship between exploratory and confirmatory data analyses (as originally proposed by J. Tukey in his famous “exploratory data analysis”) where the former paves the road to the latter. The exploratory analysis in this framework is a well-thought activity that forms a continuous process of learning from the data that requires flexible methods of model selection and model fitting (robust to model misspecification) combined with various ways of looking at the data and graphical display. Findings from exploratory analyses may be confirmed in the future trials.

The key idea is that data mining should be understood as a flexible data analytic strategy with various data-driven elements. While “data-driven” means that some elements are not specified in advance, the strategy itself can be pre-specified. This is similar to adaptive clinical trial designs, where the exact trial parameters (e.g., the final sample size or doses that remain under investigation) are not fixed at the design stage, but the adaptation strategy is nevertheless fully pre-specified. Here we list some principles for conducting data mining activities in the context of clinical data mining.

6.4.1 Documenting Business Need and Scientific Rationale for Data Mining

This document may include the following components:

- Statement of hypothesis(es) of interest based on the current understanding of the phenomena (based on relevant literature).
- Scientific assumptions and current relevant scientific theories.
- Relationships of interest and type of research: association/causation/prediction/search for patterns.
- Anticipated findings based on current knowledge (e.g., of biological mechanisms) and a priori considerations of how “unanticipated” findings, if happen, could be further explored. It is not uncommon that when findings are not in line with the current understanding of biologically plausible mechanisms, the researchers come up all too quickly with ad hoc “explanations” of the results. If variables with no a priori-known relationships to the outcome are included in analysis, there should be some plan as to how any potential findings on these predictors can be further explored/investigated/confirmed.
- Definition of “success” and “failure” for the data mining application. Here “success” does not necessarily mean obtaining findings “favorable” to

experimental treatment but rather a success in modeling the data that leads to new insights (e.g., identifying biomarkers predictive of treatment effect). It is important, however, that "failed" analyses are also reported. For example, data mining of an integrated database in depression indicating a lack of reliable predictors of placebo response in itself constitutes an important (negative) finding.
- "Stopping rules" for data mining activity should be specified here (or in the analytics plan section), which helps avoid endless search for "a significant effect."

6.4.2 Developing a Data Mining Analytic Plan

As model selection is an integral component of DM, it is impossible within a DM process to "pre-specify" exactly what statistical models will be used. However, it is important that the analytic strategy is outlined in sufficient detail prior to the beginning of the data analysis.

The scope of data used should be clearly identified; specifically, the following should be defined:

- The target population of interest
- Studies/data sources to be included
- Clinically defined outcomes: e.g., response/relapse/remission criteria
- Outcome variable(s)
- Covariates that potentially may affect the outcome of interest

It is a good practice to list all "data-driven" components and "tuning parameters" of the analytic strategy upfront and explain how they will be identified in the course of the study.

The possibility of replicating findings with additional data sets that were not used in model fitting and selection (test data) should be addressed. If this is not possible due to limited data, other approaches should be used, e.g., bootstrap and cross-validation.

If hypothesis testing is the primary objective, all adjustments for multiplicity and control of Type I error rates should be explained.

If the model selection is a part of the DM strategy (which is almost always the case), the DM Analytic Plan should explain how potential data overfitting would be handled (e.g., via cross-validation, using separate validation data sets).

If inference about causal parameters is the primary objective, all non-randomized covariates that may potentially cause selection bias should be listed and methodology that will be used to overcome it outlined.

If the analytic strategy involves a multistage data analysis (i.e., when selection of an analytic procedure at a later stage may depend on the results at previous stages), the DM plan should contain a discussion on how uncertainty in the multistage process could be accounted for (e.g., via bootstrapping the *entire* multistage analysis sequence, or model averaging). It is often the case that only uncertainty associated with the final stages of such complex analytic strategies is taken into account, while the steps leading to final analyses are left undocumented and obliterated from the "collective memory" of the research team. Also in many exploratory analyses, some stages of analysis involve "human intervention" where decisions are made subjectively which makes it challenging to automate the entire analytic strategy by implementing it in a programming code and prevents the use of resampling methods for evaluating such strategies. When implementing multistage procedures and applying them to resampled data, it is important to account for the fact that on some samples the result may be negative or null, for example, when an empty set is returned for a subgroup search or when no predictive bookmakers are found. In such situation it is important that the analysis strategy should be well-defined in the sense that it is applicable for any data, not only for the specifically observed data set on which it was used.

The DM analysis plan should include "sensitivity analyses" to validate the robustness of the findings to various departures from (often untestable) assumptions. This may relate to assumptions about missing data mechanisms, unmeasured confounders, or possible "structural" changes (e.g., in the relationship between outcomes and predictors) in the future populations that may affect the generalizability of findings. As part of sensitivity analyses, sensitivity to methodology (e.g., frequentist vs. Bayesian) could be explored as well. Visualization tools should be used at all stages of analyses, primarily to investigate potential issues such as outliers, influential observations, etc.

Many data mining techniques are simulation based (e.g., cross-validation, bootstrap); therefore, retaining seed values is recommended to ensure the *reproducibility* of findings.

Finally, we emphasize the importance of proper quality control and validation of analyses, just like in the standard stat analysis of clinical trial data.

6.4.3 Ensuring Data Integrity

Integrating and aggregating information from multiple studies may pose challenges such as:

- Using different clinical outcomes (e.g., rating scales) and different definitions for the same outcomes across multiple data sources.
- The extent of and approach to data cleaning may not be the same across multiple studies. DM often utilizes various patient characteristics and time-dependent covariates that may not be fully cleaned and validated even in locked databases, as they might not have been a focus for analyses intended for clinical study report (e.g., the time of occurrence of certain events or concomitant treatments).
- Many challenges of data aggregation (e.g., using combined regional, ethnic, etc. groups; grouping adverse events, concomitant medications; alignment across common time points) require careful consideration and close collaboration between the statistician and medical team and may require a substantial amount of time in the absence of integrated databases.

When integrated databases are available, it is tempting and often seems reasonable to form initial analysis plan around variables available in the integrated database; however, the absence of important potential predictors in the integrated database cannot serve as a justification for not considering them as potential predictors in the DM analysis plan.

Further, study populations may be somewhat different and/or recruited at significantly different times, and this heterogeneity may need to be accounted for in the model/analysis. This should be done keeping in mind the target population the findings need to be generalized for, possibly leading to re-weighing current data so as to better match the target population.

6.5 Case Studies

In this section, we present three case studies that illustrate several data mining/machine learning methods as applied to clinical trial data. While some of them are taken to a greater level of detail, others are presented in a briefer manner.

6.5.1 Evaluation of Subpopulations Using SIDES Methodology

In this subsection, we illustrate some of the methods introduced in Sect. 6.3.3, specifically variants of the SIDES methodology by applying them to a data set simulated to mimic a realistic data from a Phase 3 study.

6.5.1.1 SIDES Methodology

Here, we provide a brief outline of SIDES method. An interested reader may refer to Lipkovich et al. (2011) and Lipkovich and Dmitrienko (2014) for further details. In our example, SIDES is applied to a binary outcome, and the description is tailored to this type of outcome, although the approach is not limited to it. Also for simplicity, we assume that all covariates are continuous as is the case for our data set, but this is not a requirement.

First, we apply to the data set the SIDES subgroup generation procedure that starts with evaluating a differential splitting criterion at every allowable split of every candidate covariate, which is defined as follows:

$$D = 2\left(1 - \Phi\left(\frac{\left|Z_{left} - Z_{right}\right|}{\sqrt{2}}\right)\right).$$

Here $\Phi(\cdot)$ is the normal CDF; the statistics Z_{left} and Z_{right} are scaled by the pooled standard error of treatment differences between proportions evaluated for the experimental treatment and control in two child groups resulting from splitting a continuous variable X into the *left* $\{X \leq x_0\}$ and the *right* child groups $\{X > x_0\}$, based on a provisional cutoff x_0. Note that the criterion is on the probability scale with smaller values indicating larger differentials.

For each candidate covariate, the best cutoff associated with the smallest value of D is determined. Based on this value, the covariates are ordered from best to worst, and the first M covariates (the *width* parameter) are selected. For any covariate, we have two child groups resulting from splitting at the optimal cutoff, and the child with the largest treatment effect is retained as "promising." Therefore, from M top covariates, M promising subgroups are retained. These are called *promising* subgroups of level 1. Depending on the maximal number of levels L (the *depth* parameter), the process continues recursively by applying the same splitting process to each of the promising groups. The resulting *terminal* groups are considered as final promising subgroups. For example, if $L = 1$, the process stops with the M level 1 groups which will be the terminal groups, whereas if $L = 3$, the process is recursively applied two more times resulting in up to M^3 terminal subgroups. The size of the candidate subgroups is controlled by the user-specified minimal required subgroup size, n_{min}. Only subgroups of size at least n_{min} are considered as allowable splits, and the recursion might stop even before achieving the specified *depth* once no subgroups of required size can be formed.

The above process, which we refer to as base SIDES, results in generating a potentially large pool of subgroups. A greedy approach to subgroup selection would be to simply choose the subgroup from the pool with the largest observed treatment effect (or few subgroups with largest effects). Of course, the observed effect(s) and the associated P-value(s) would be highly overoptimistic. These can be adjusted by using resampling methods. For example, the multiplicity adjusted P-value can be obtained by randomly permuting treatment labels, reapplying the same subgroup search procedure to each null (permuted) set, and computing the smallest P-value over all promising subgroups for each null data set. Based on a large number of null sets, the adjusted P-value can be computed as the proportion of such sets where the minimum P-value is as small as or smaller than the one found in the best subgroup of the actual data set.

However, this greedy process is likely to generate the top subgroups that will be subsequently penalized very severely in terms of having very large adjusted P-values (suggesting that the findings are driven by chance and are not likely to generalize to the future data).

To develop more sensible subgroup search procedures, several methods of restraining the greediness of the search have been proposed. One approach is to introduce a complexity parameter that constrains the search by placing a requirement on how much better the treatment effect in a child group should be compared to that in the parent group at each split. The split is made only if the candidate child group exceeds that threshold.

The second approach, called “SIDEScreen” (pursued in this example), uses a variant of model averaging. From the harvesting process, a variable importance index is evaluated for each covariate that reflects its overall predictive worth. This is defined as the average contribution of a covariate across all generated promising subgroups (counting only terminal subgroups of the harvesting process). Specifically

$$VI(X) = K^{-1} \sum_{i=1}^{K} \nu_i,$$

where $\nu_i = -\log d_i(X)$, if the i^{th} subgroup contains biomarker X, and $\nu_i = 0$ otherwise; K is the number of promising subgroups; and $d_i(X)$ is the splitting criterion evaluated for biomarker X for the selected split.

Thus, computed variable importance scores are screened by applying a screening rule

$$VI(X) > \widehat{E}_0 + k\sqrt{\widehat{V}_0},$$

where $\widehat{E}_0$ and $\widehat{V}_0$ are the mean and variance of the maximal (over all biomarkers), VI score under the null distribution obtained by permuting the treatment labels. These mean and variance are estimated from a large number of such samples. The multiplier k is a free parameter that can be calibrated as $k = \Phi^{-1}(1 - \kappa)$, where κ is interpreted as the probability of selecting at least one noise biomarker in the absence of predictive biomarkers in the data set.

At the second stage, the basic SIDES is applied only to biomarkers selected at the first stage. The final adjusted p-values are computed by replicating the entire two-stage procedure on a large number of additional null sets. Note that the same multiplier k is applied to each null set; therefore, regardless of the value of multiplier at the first stage, the overall Type I error rate of the final subgroup(s) can be controlled at any desired level.

6.5.1.2 Analysis Data

Our example data set *sepsis_ex.csv* is available at QSPI working group site along with the **RSIDES** package: http://biopharmnet.com/subgroup-analysis-software/. The data set is based on a Phase 3 trial conducted to examine the efficacy and safety profiles of a novel treatment for severe sepsis. There are 470 patients (317 patients in the experimental treatment arm and 153 patients in the control arm) with a binary outcome variable *mortality* (the primary endpoint) that represents the survival status of patients after 28 days of treatment: the value of 1 for subjects who died within 28 days and 0 for those who survived. There are eight candidate covariates, including demographic and clinical characteristics listed in Table 6.3, all of which are numerical variables. Note that the results for baseline serum concentration (*il6*)

Table 6.3 Candidate covariates in the severe sepsis data example

Candidate covariates	Description	Median (range)
Time	Time from first sepsis-organ failure to start of treatment (hours)	30.67 (10, 3775.9)
Age	Patient age (years)	59.871 (33.2, 93.3)
Platelets	Baseline local platelets (1000/mm^3)	153 (45, 650)
Sofa	Sum of baseline SOFA scores (cardiovascular, hematologic, hepatic, renal, neurological, and respiratory scores)	8 (3, 17)
Creatinine	Baseline creatinine (mg/dL)	1.5 (1, 20)
Apache	Pre-infusion APACHE-II score	23 (19, 48)
IL6	Baseline serum IL-6 concentration (pg/mL)	406.6 (37.1, 296,550)
Bilirubin	Baseline bilirubin (mg/dL)	1 (0.4, 20.4)

exhibit some extreme values which are not uncommon for this lab measure. We comment that with parametric regression methods, this would be a problem requiring special treatment (e.g., variable transformation), but the tree-based methods are immune to that as they essentially treat a numerical covariate as ordinal and are invariant to any monotone transformation of a covariate.

6.5.1.3 Results

Table 6.4 shows the results of applying base SIDES with parameters

- Width $= 3$
- Depth $= 2$
- $n_{min} = 30$

To illustrate the subgroup generation process, the three intermediate subgroups of the first level (based on variables *time*, *age*, and *il6* with optimal cutoffs 30.67, 59.871, and 519.4) are highlighted in bold. The associated splitting criterion (on the $-\log$ scale, with larger values being better) is shown in the third column. The terminal subgroups are obtained after splitting the above three level 1 groups by the best three variables selected from the candidate list excluding the one selected at the first level. There are eight rather than nine groups because one group based on *time* and *age* occurred twice with the same cutoffs for both variables and was removed as a duplicate. The penultimate column contains the P-value for the overall treatment effect (one-tailed) and unadjusted P-values for promising subgroups. Note that the overall treatment effect is negative while some subgroups show apparently large treatment effect with subgroup $time \leq 30.67$ and age > 59.871 appearing best with an unadjusted P-value of 0.00196. However, the adjusted P-values based on 10,000 sets of randomly permuted treatment labels are hopelessly large. In particular, the adjusted P-value for the above subgroup is 0.5.

Table 6.4 Subgroups generated using base SIDES for sepsis data (*width* $= 3$, *depth* $= 2$, $n_{min} = 30$)

Subgroup	Size	Splitting criterion (−log scale)	*P*-value (unadjusted)	*P*-value (adjusted)
Overall population	470		0.8301	
Time ≤ 30.67	253	5.29	0.0588	0.99
Time ≤ 30.67 and age > 59.871	123	3.37	0.00196	0.50
Time ≤ 30.67 and IL6 > 162.65	171	1.09	0.0136	0.88
Time ≤ 30.67 and bilirubin ≤ 2.5	199	0.74	0.0496	0.99
Age > 59.871	217	4.25	0.0718	0.99
Age > 59.871 and IL6 > 92.8	169	2.29	0.0362	0.97
Age > 59.871 and sofa > 5	183	1.98	0.0172	0.91
IL6 > 519.4	180	2.55	0.1800	1.00
IL6 > 519.4 and age > 56.098	99	4.68	0.0076	0.78
IL6 > 519.4 and creatinine > 1.4	104	1.85	0.0168	0.91
IL6 > 519.4 and time ≤ 30.67	117	1.60	0.0328	0.97

Next, we evaluate subgroups using a less greedy and more powerful Adaptive SIDEScreen approach. First, we apply base SIDES with even less restrictive parameters on generated subgroups than for the first run, specifically setting *width* $= 5$ and *depth* $= 3$ resulting in up to $5^3 = 125$ subgroups based on three covariates. While the resulting subgroups may be picking up a lot of noise and would likely not generalize to the future data, we are not using these subgroups as candidates for the final selection but rather use them as an intermediate step for computing variable importance scores used for covariate screening. Averaging over a broader set of subgroups (models) in general helps to obtain more reliable variable importance scores.

Figure 6.4 contains variable importance and an associated benchmark from 1000 null sets. The threshold rule is $VI(X) > \widehat{E}_0 + \sqrt{\widehat{V}_0}$ with the multiplier $k = 1$. As we can see, two variables, *time* and *age*, stand out having larger scores. Also, they both exceed the threshold based on 1 standard deviation from the null mean.

Table 6.5 shows the results of the second-stage analysis where base SIDES is applied only to variables that passed the screening stage. Predictably, the subgroup is the same as the best one that was found by a greedy base SIDES. However, the adjusted *P*-value is very different from the one computed for the base SIDES.

To understand this seemingly contradictory result, first recall that the adjusted *P*-values are computed for the SIDEScreen procedure by applying to each null set the two-stage procedure, including computing anew the variable importance scores based on subgroups generated from each null set and comparing them with the same null threshold as was applied to the observed data. Of the null sets where some covariates pass the threshold, we identify those having subgroups with *P*-values such as or smaller than the one found in the observed data. Naturally, many null sets would not have any covariates that pass the screening threshold (about 84%, assuming the normal distribution for VI scores under the null and the threshold with $k = 1$). Even if each of the remaining ≈16% of the null sets produced a

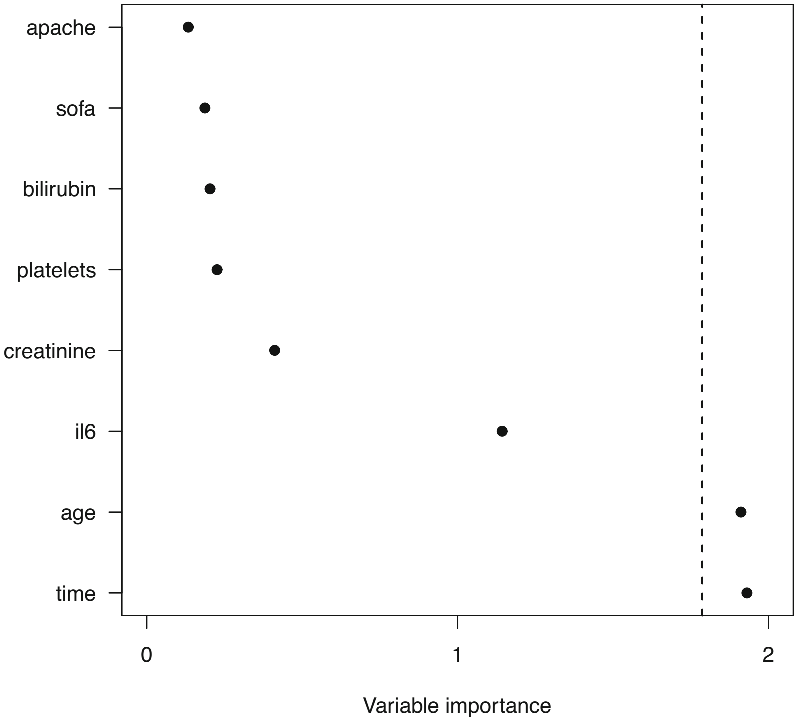

Fig. 6.4 Variable importance scores (shown as filled circles) and the threshold based on null distribution (shown as the dashed line)

Table 6.5 Subgroups generated using Adaptive SIDEScreen for sepsis data

Subgroup	Size	Splitting criterion (–log scale)	P-value (unadjusted)	P-value (adjusted)
Overall population	470		0.8301	
Time $\leq$ 30.67	253	5.29	0.0588	0.112
Time $\leq$ 30.67 and age $>$ 59.871	123	3.37	0.00196	0.036
Age $>$ 59.871	217	4.25	0.0718	0.116

subgroup better than the one found on the observed data, the adjusted *P*-value would be no larger than about 0.16. One might consider the application of this "shrinkage factor" to compute adjusted *P*-values as a kind of cheating. However, note that the selection rules are the same whether we apply them to the observed or null data. Only in the case when at least one covariate will pass the threshold on the observed data would we have the opportunity to adjust an associated *P*-value, which under the true null will amount to the same 0.16. As a result, data with no real predictive marker would likely not proceed to the second stage, thus reducing the probability of spurious findings.

6.5.2 Evaluating Optimal Dynamic Treatment Regimes via Q-Learning

This case study is a brief summary of recently reported analyses of a STEP-BD trial by Wu et al. (2015).

6.5.2.1 The STEP-BD Trial, Analysis Objectives, and Available Data

STEP-BD (Systematic Treatment Enhancement Program for Bipolar Disorder) is a long-term study of bipolar disorder funded by the National Institute of Mental Health (NIMH), the results of which were reported in Sachs et al. (2003). The study enrolled more than 4000 patients from the United States and lasted about 7 years including options for several treatment pathways: an observational trial (standard care pathway, SCP) and several randomized trials (randomized care pathway, RCP). First, all patients entered SCP and then some were eligible to follow one of the RCPs. Within the latter, there were several options (pathways) depending on the depression features. The Wu et al. (2015) analysis is focusing on one of them: acute depression randomized pathway (RAD).

The purpose of RAD was to explore the effectiveness of two antidepressant treatments (bupropion or paroxetine) versus placebo, in addition to a number of mood stabilizers (lithium, valproate, and others) that were used in combination with the two drugs or placebo. Initially, patients were randomly assigned to one antidepressant (150 mg of a sustained release formulation of bupropion or 10 mg of paroxetine) or placebo. After 6 weeks, patients with non-response on the placebo were randomized to either paroxetine or bupropion; patients with non-response on the antidepressant would have the dose of their current antidepressant increased. The schematic of the RAD sub-trial is presented in Fig. 6.5. The reader should bear in mind that patients under active treatments or placebo received mood stabilizers at physician's discretion, which is not reflected in the labels of the figures.

The objective of the analyses in Wu et al. (2015) was estimating optimal DTRs for both stages 1 and 2 to minimize the expected depression score at week 12 (SUMD), based on all available data at the decision time. Note that our ability to search for optimal treatment options is naturally limited by the available (or *feasible*) treatment options restricted by design. Specifically, as we see from Fig. 6.5, the second-stage randomization was only applied to patients who failed on placebo during stage 1. Therefore, the Q-learning algorithm would not be able to "learn" from the data a regime that recommends, for example, to treat with bupropion (at stage 2) those patients who had previously failed on paroxetine (at stage 1).

Patient covariates and intermediate outcomes available for analysis are listed in Table 6.6.

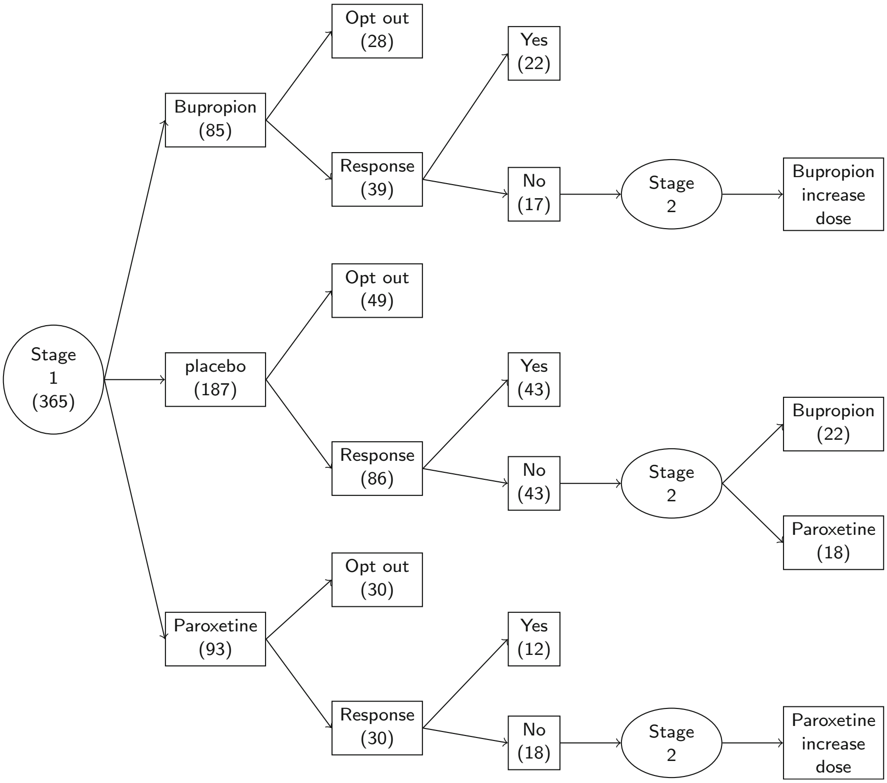

Fig. 6.5 Schematics of the RAD trial

6.5.2.2 Q-Learning Methodology for the RAD Trial

Here we briefly outline the Q-learning method adapted for estimating optimal regimes in the RAD trial of the STEP-BD design. Although the Q-learning applies for a more general *m*-sage learning, our exposition is tailored to the two-stage decision setting and a terminal continuous outcome (here, depression score at week 12).

As described in Sect. 6.3.3, Q-learning is an approximate dynamic programing algorithm that can be viewed as an extension of regression to multistage decision problems. In our case, this amounts to sequentially fitting regressions for the outcome, with pretreatment covariates, earlier treatments, and outcomes fitted as predictors. Starting from the last (i.e., the second stage) decision, Q-learning first finds an optimal decision rule at the second stage as the one maximizing expected outcome after stage 2, given earlier patient outcomes and covariates available prior to treatment decision for stage 2 as well as the treatment choice that has been made at decision stage 1. Then going backward, it regresses the (expected) outcome (that would have resulted if optimal treatment rules at stage 2 were applied) on treatment assigned at stage 1 and covariates available at the decision stage 1. The optimal regime for stage 1 is found as the one maximizing the response of this second regression. The key element is that covariate-by-treatment interactions have to be included in the regressions; otherwise, the estimated optimal regime will be assigning the same treatment to all patients regardless of their covariate values.

The backward induction allows Q-learning to factor in future decisions when making treatment decisions at earlier stages. This can be contrasted with a "myopic" strategy that only looks at intermediate (proximal) outcomes of a current treatment assignment. For example, treatments at stage 1 may lead to temporary alleviation of symptoms and therefore appear beneficial; however, the long-term benefits may become questionable after a later (e.g., second)-stage decisions are factored in.

Several challenges are encountered when applying the Q-learning algorithm to this data, including the need to make model selection given a large number of candidate covariates (Table 6.6) and handling a substantial number of missing data on the outcomes and covariates. These problems, typical of data mining/machine learning applications to clinical data, need to be integrated within estimating the optimal DTR.

Another challenge that appears unique for DTRs (although, more broadly, is present in any "estimation after model selection") is obtaining confidence intervals for outcomes under an estimated DTR. Because the DTR estimator is irregular (non-smooth), the standard bootstrap theory may not apply and other methods (such as *m-out-of-n* bootstrap (Chakraborty et al. 2013)) need to be used.

Finally, even relatively simple rules based on linear regressions with a few selected covariates may appear rather unwieldy for decision-makers (such as prescribing physicians); therefore, more visual and easy-to-use presentation of the rules is desired. This can be accomplished by approximating the estimated DTR with classification trees, in an additional step.

Following Wu et al. (2015), we first describe how the Q-learning would proceed for this case, assuming the correct (e.g., linear) models for Q-functions have been pre-specified, and no data are missing; then we explain how missing data imputation and model selection were integrated within the Q-learning strategy.

First, we define stage 1 and stage 2 Q-functions in terms of available treatment choices, patent-level covariates, and outcomes. Specifically, the *i*th patient in a hypothetical complete data set can be characterized with a trajectory$(X_{1i}, T_{1i}, X_{2i}, T_{2i}, Y_i)$, $i = 1, \ldots, n$, where X_1 denotes a vector of baseline covariates available at decision stage 1, X_2 comprises post-baseline outcomes collected during stage 1 and potentially informing treatment choice at stage 2, and T_1 and T_2 indicate randomized treatment choices at stages 1 and 2, respectively. That is $T_1 = \{Bupropion, Paraxetime, placebo\}$ and $T_2 = \{Bupropion, Paraxetime\}$; Y is the SUMD score at the end of stage 2, with lower values indicating clinically desirable outcome (low depression score).

The Q-functions are essentially the response functions that map patients with particular treatment choices and covariate profiles to expected outcomes, similar to our response functions $f(x, t)$ introduced in Sect. 6.3.3 (in the context of subgroup

Table 6.6 Candidate predictors for regression models in Q-learning (based on Table 5 of Wu et al. 2015)

Variable (label)	Description	Type	Values (range or levels)	Mean (SD) or frequency
AGE	Age at entry (years)	Numerical	18–77	40.59 (11.74)
RACE	Race	Binary	White or Caucasian, non-White	90.4%, 9.6%
GENDER	Gender	Trinary	Male, female, transgender	43%, 56%, 1%
MARSTAT	Marital status	Trinary	Never married, married, separated	35.6%, 33.8%, 30.6%
HINCOME	Annual household income (×\$1000)	Binary	<40, ≥ 40	58.5%, 41.5%
EMPLOY	Employment status	Binary	Employed, unemployed	46.9%, 53.1%
EDUCATE	Education level	Binary	College or below, technical school or above	53%, 47%
MEDINS	Indicator of medical insurance	Binary	Yes, no	72.8%, 27.2%
BITYPE	Bipolar type at entry	Binary	Type I, type II	70.4%, 29.6%
PRONSET	Clinical episode immediately preceding current depressive episode	Trinary	Remission, hypo(manic), mixed	45.9%, 33.2%, 20.9%
SUMD0	Scaled depression at entry	Numerical	0.75–18	7.47 (2.30)
SUMD1[a]	Scaled depression at the end of stage 1	Numerical	0–14	4.49 (3.07)
SUMM0	Scaled mood elevation at entry	Numerical	0–7	1.19 (1.09)
SUMM1[a]	Scaled mood elevation at the end of stage 1	Numerical	0–6.75	0.95 (1.30)
Trt1[a]	Treatment received at stage 1	Trinary	Bupropion, paroxetine, placebo	23.3%, 25.5%, 51.2%
SIDE1	Tremor	Binary	Yes, no	26.9%, 73.1%
SIDE2	Dry mouth	Binary	Yes, no	21.1%, 78.9%
SIDE3	Sedation	Binary	Yes, no	17.1%, 82.9%
SIDE4	Constipation	Binary	Yes, no	5.7%, 94.3%
SIDE5	Diarrhea	Binary	Yes, no	12%, 88%
SIDE6	Headache	Binary	Yes, no	13.7%, 86.3%
SIDE7	Poor memory	Binary	Yes, no	14.3%, 85.7%

[a] Covariates that were available only for the second-stage regression model

identification). The stage-specific Q-functions are defined recursively, starting with the last stage function. In our case

- $Q_2(x_1, t_1, x_2, t_2) = E(Y| X_1 = x_1, T_1 = t_1, X_2 = x_2, T_2 = t_2)$,
- $Q_1(x_1, t_1) = E(\, min_{t_2}\, Q_2(X_1, T_1, X_2, t_2) | X_1 = x_1, T_1 = t_1)$.

The Q_2 is a usual regression, here estimating the "quality" of treatment assignment t_2 for a patient presented with his or her "history" up to that point. Similarly, the function Q_1 measures the quality of assigning treatment t_1 for a patient presented with his/her pretreatment covariates and *assuming optimal decision at subsequent stage* 2, defined by minimizing Q_2 (SUMD score) over t_2.

We assume that $Q_1(\cdot, \theta_1)$ and $Q_2(\cdot, \theta_2)$ are parametrized as linear functions of patient covariate history and prior treatments with vector θ_1 containing regression coefficients associated with X_1, T_1, and X_1 by T_1 interactions and θ_2 containing coefficients for X_1, X_2, T_1, T_2, and (X_1, X_2) by T_2 interactions. The parameters of Q-functions are estimated in three steps:

1. Estimate parameters in θ_2 using only data on *placebo non-responders* who were randomized at the second stage to bupropion or paroxetine, by regressing Y on X_1, X_2, T_1, T_2.
2. Compute new "response" vector $\widetilde{Y}$ to be used for estimating the first stage Q-function, defined as

$$\widetilde{Y} = \begin{cases} \widehat{Q}_2\left(t_2^{opt}\right), & \textit{for placebo nonresponders} \\ Y, & \textit{for the rest of patients} \end{cases},$$

   where $\widehat{Q}_2\left(t_2^{opt}\right)$ is the predicted response from the regression model at the previous step with treatment T_2 set for each patient at the *optimal* value t_2^{opt} corresponding to the minimum of estimated $\widehat{Q}_2$.
3. Estimate parameters in θ_1 using all patients by regressing $\widetilde{Y}$ on X_1, T_1, and compute the optimal first-stage treatment t_1^{opt} by minimizing the estimated $\widehat{Q}_1$.

The variables for modeling Q_1 and Q_2 are selected from 24 potential predictors listed in Table 6.6 using stepwise forward variable selection with the entry and stopping conditions determined by the Bayes information criterion (BIC). This was combined with multiple imputation procedures for missing values. The imputation was done using Fully Conditional Specification (chained equations) procedure available in the R package **mice** (van Buuren 2018). This method imputes missing values using sampling from posterior distributions and does not require explicit specification of joint likelihood. Instead, conditional models are defined for each variable given all the rest. This is especially convenient for data sets of mixed type, combining numerical and categorical variables, where joint distributions are hard to specify. In our case, for continuous variables, Predictive Mean Matching was used, and logistic regression models were used for binary variables.

The stepwise forward selection was conducted in such a way that each candidate variable to be added was evaluated using the BIC averaged across m generated complete data sets. The final model was then selected based on the best average BIC across all models in the list formed by the stepwise selection. First, the optimal model for estimating θ_2 in step 1 of the outlined three-step Q-learning procedure was selected in this fashion. Then the optimal model for Q_1 was selected by applying the same stepwise selection (based on average BIC) to estimating θ_1 (step 3) given $\widehat{\theta}_2$ estimated with the model selected for Q_2. For details of the procedure, see Wu et al. (2015).

Once models for Q_1 and Q_2 have been selected, they were applied to each of the m completed data sets, the resulting m estimates of Q-functions averaged, and optimal treatment regimes found as the minimizers of the averaged Q-functions.

Finally, the optimal treatment assignments t_1^{opt} and t_2^{opt} for each patient were approximated with classification trees using R package *rpart* to provide more easily interpretable rules. To achieve that, classification tree algorithm was applied separately to new variables capturing estimated t_1^{opt} and t_2^{opt} as categorical response variables with covariates, selected for modeling Q_1 and Q_2, as candidate splitting variables. The resulting trees are presented in the left and right panel of Fig. 6.2.

6.5.2.3 Results of Q-Learning

Details of estimated Q-functions and associated regression coefficients can be found in Wu et al. (2015). Here we will briefly discuss the tree representation of the optimal DTR shown in Fig. 6.6. The tree on the left shows assignment rules at the first stage. Interestingly, patients who experienced a (hypo) manic episode immediately preceding the current major depressive episode are not recommended any of the two available antidepressant treatments but rather using only mood stabilizers (note that "placebo" actually refers to treating with mood stabilizers only). For the rest of the patients, bupropion is recommended to younger patients, and paroxetine is

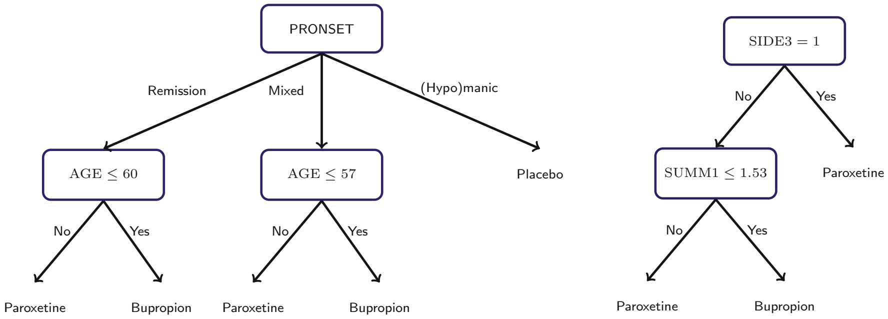

Fig. 6.6 Estimated optimal regimes at stages 1 (left) and 2 (right). Note that the optimal regime at the second stage is evaluated only for those patients who are assigned to placebo at the first stage and fail to show response

Table 6.7 Point estimates and confidence intervals for the expected depression score SUMD at week 12 under estimated DTR and some static regimes (labeled by a pre-specified combination of a first-stage and a second-stage treatment)

Regime	Estimated SUMD	Estimated 90% confidence interval
Estimated optimal DTR	2.13	(1.34, 2.86)
(Bupropion, high-dose bupropion)	6.91	(6.27, 7.71)
(Paroxetine, high-dose paroxetine)	8.25	(7.39, 9.07)
(Placebo, bupropion)	3.71	(3.38, 4.04)
(Placebo, paroxetine)	4.51	(4.10, 4.90)

The lower scores indicate clinically preferred outcome (based on Table 4 from Wu et al. 2015)

recommended to older patients. The tree on the right illustrates the assignment of the second-stage treatment for patients who had failed on placebo during the first stage: variables SUMM1 (mood severity after stage 1) and SIDE3 (presence of sedation side effect) dictate treatment selection; subjects with no sedation side effects and low mood severity are recommended to bupropion, and all others are recommended to paroxetine.

It is also instructive to compare the expected outcomes assuming patients undergo the optimal treatment to outcomes expected under some pre-specified fixed (static) regimes. To estimate expected outcomes under static regimes, an inverse probability weighted estimator was used (Zhang et al. 2013; see also our last case study in Sect. 6.5.3), and confidence intervals were computed using nonparametric bootstrap. The confidence intervals for the optimal DTR estimator were computed using the m-out-of-n bootstrap. Table 6.7 summarizes the results, suggesting some advantages of the estimated dynamic regime.

6.5.3 Estimating Treatment Effect in an Oncology Trial Using Inverse Probability of Censoring Weights

6.5.3.1 Introduction

Demonstrating statistically significant and clinically meaningful gains in overall survival (OS) remains the gold standard to provide evidence of the benefits of new anticancer drugs (Johnson et al. 2015). In clinical trials, in patients with advanced or metastatic cancer, however, it is very common for participants to switch from the treatment to which they were initially randomized to other therapies (Latimer and Abrams 2014), typically after disease progresses on the initially randomized treatment. For both ethical and practical reasons, this option may be built into oncology trial protocols. Switching may also be allowed from the study control treatment to experimental treatment, which is not part of the standard treatment pathway, if no other non-palliative treatments are available.

When patients switch to and benefit from active post-progression therapies, a standard ITT analysis may inaccurately estimate the “true” OS benefit associated with the investigational product the patients were initially randomized to and will affect the cost-effectiveness analyses in the context of economic evaluations that make use of the OS evidence. In general, switching to active post-progression therapies that do not form part of the standard treatment pathway should be adjusted for (Latimer and Abrams 2014).

Several switching adjustment methods that seek to estimate the true treatment effect are available, ranging from simple to complex techniques. Simple or naïve methods such as simple censoring (when data from patients who switched are censored at the point of switching) or exclusion techniques (patients who switched are excluded entirely from the analysis) are highly prone to selection bias and should be avoided (Latimer and Abrams 2014). More complex statistical techniques are classified as *randomization based* (e.g., the rank-preserving structural failure time model or iterative parameter estimation algorithm) or *observational based* (e.g., the two-stage accelerated failure time model [two-stage method] or inverse probability of censoring weights [IPCW]) (Latimer and Abrams 2014). Different switching adjustment methods may be appropriate under certain scenarios, and none is optimal in all circumstances. All of them involve untestable assumptions, as is always the case with causal inference from observational data. Here, for illustration, we will focus on IPCW.

In the IPCW approach, patients are artificially censored at the time of switching, and the weight/influence of uncensored patients with similar prognostic characteristics is increased based on covariate values and a model of the probability of being censored. The key assumption made by the IPCW method is the "no unmeasured confounders" assumption; that is, data must be available for all baseline and time-dependent prognostic factors for mortality that independently predict informative censoring (switching) (Latimer et al. 2014; Robins and Finkelstein 2000). This assumption cannot be tested using the observed data (Robins and Finkelstein 2000). In practice, this is unlikely to be perfectly true, but the method is likely to work adequately if the "no unmeasured confounders" assumption is approximately true; that is, there are no important independent predictors missing (Latimer et al. 2014). Additionally, the method assumes that the model for computing weights is correctly specified and that the probabilities of treatment switching conditional on given covariates are bounded away from zero. The latter would not be the case if physicians were switching patients based on deterministic rules (e.g., all female patients are switched to treatment A and male patients switched to treatment B). As correct model specification plays an important role in implementing the IPCW analysis strategy, modern methods of statistical learning that are free of parametric model assumptions can be very useful because they allow automating the strategy, making it less prone to misspecification error.

6.5.3.2 Example Data Set in Prostate Cancer

The data set used to illustrate the IPCW in this section represents a randomized, double-blind trial with 800 subjects with prostate cancer in each of the two arms—

experimental and placebo. The endpoint of interest for this analysis is overall survival. All subjects were followed up for OS after discontinuing study treatment.

Treatment switching is defined as the switch from the control treatment to the experimental treatment for those subjects randomized to the control arm or from either treatment group to other post-study treatments that are not part of the standard treatment pathway. Treatment switching often occurs upon disease progression or when conclusive evidence accrues about the benefit of the experimental treatment and therefore the study is stopped and unblinded. We assume this typical scenario for our case study.

In our example data set, all subjects in the control group and 27.5% (220/800) of subjects in the experimental arm discontinued the treatment they were randomized to by the data cutoff. In this case study, we are concerned with one type of switch only—when the subject switches from the randomized treatment to another therapy that was not part of the standard treatment pathway, e.g., as per the NICE clinical guideline for prostate cancer (NICE 2014), and we are not concerned with possible multiple switches thereafter. The data set contains a total of 376 switchers, with a larger proportion of switchers in the placebo arm: 18.0% (144/800) and 29.0% (232/800) of subjects in the experimental and placebo groups, respectively.

6.5.3.3 IPCW Methodology

We illustrate herein the IPCW approach for adjusting estimates of a treatment effect in the presence of informative censoring. Censoring is informative when a subject with specific characteristics is more likely to be censored than another (e.g., a subject who has poor prognosis discontinues treatment and is censored because of this). In this case study, we consider treatment switching as the only informative censoring mechanism. All other censoring reasons are modeled as non-informative (as part of the proportional hazard partial likelihood of the Cox regression).

The IPCW method represents a type of Marginal Structural Model (MSM), which was originally developed for use with observational data (Hernán et al. 2001). The IPCW method involves censoring subjects at the time of treatment switch and then controlling for this potentially informative censoring by weighting. Specifically, the follow-up information for subjects who remain at risk for the event is weighted, so that they account not only for themselves but also for subjects with similar characteristics (both baseline and time-dependent) whose follow-up was censored by informative censoring (Robins and Finkelstein 2000).

The IPCW method entails the following general steps. First, for all subjects, follow-up time from randomization until failure (e.g., death) or censoring (informative or otherwise) is partitioned into intervals. At the beginning of each interval, time-dependent variables that may be predictive of informative censoring (switching) or failure are calculated and updated. For each subject and interval, so-called stabilized weights (SW) are then calculated as described by Hernán et al. (2001). The numerator of each weight is the cumulative probability of remaining uncensored by informative censoring from the beginning of follow-up to the end of the interval given only baseline covariates. The denominator of the weight is the cumulative probability of remaining uncensored by informative censoring to the end of the interval given both baseline and time-dependent covariates. In the original formulation of Hernán et al. (2001), an individual's treatment history up until the end of the previous interval is included in both the numerator and denominator. Given that in our case study the cause of informative censoring is the *first* switch from the randomized treatment to another antineoplastic therapy, "past treatment history" is reduced to the initial randomized treatment that is conditioned upon simply by performing computations of weights separately by treatment arms.

Specifically, patient-specific estimates of the stabilized weights at the j^{th} interval, $SW_i(j)$, are obtained as follows (here we drop the patient index from all terms to simplify notation):

$$SW(j) \equiv \frac{\prod_{k=0}^{j} P[C(k)=0|C(k-1)=0, X(0)]}{\prod_{k=0}^{j} P[C(k)=0|C(k-1)=0, X(0), Y(k)]},$$

where

- $C(k)$ is an indicator function representing censoring/treatment switch status at the end of interval k (1, censored due to switching, 0, uncensored).
- $X(0)$ is a vector of subject characteristics measured at baseline (see Table 6.8).
- $Y(k)$ is a vector of time-dependent subject characteristics measured at or prior to the beginning of interval k (see Table 6.8).
- $P[C(k) = 0| \, C(k-1) = 0, X(0)]$ is the probability of remaining uncensored (not switched) at the end of interval k given uncensored at the end of interval $k-1$ and conditioned on baseline characteristics $X(0)$.
- $P[C(k) = 0| \, C(k-1) = 0, X(0), Y(k)]$ is the probability of remaining uncensored (not switched) at the end of interval k given uncensored at the end of interval $k-1$ and conditioned on baseline characteristics $X(0)$ and time-dependent patient characteristics $Y(k)$.

Probabilities of remaining uncensored by informative censoring are unknown and therefore need to be estimated. Here, we use two approaches to illustrate the difference between traditional parametric modeling and methods of machine learning: logistic regression and random forest models (see Sect. 6.3.1). In each case, we fit one model for the denominator and one model for the numerator, with informative censoring (switching) as the dependent variable. Details on how these methods were applied in this case study are provided further below. Both the logistic regression and the random forest models are estimated within each treatment arm separately, to account for potential differences in the reasons that led to switching treatment in each arm. Covariates included in these models represent measurements typically collected in the studies of prostate cancer and are presented in Table 6.8.

A hazard ratio (HR) for the outcome of interest is then estimated using a weighted Cox proportional hazards regression model that includes only baseline variables and the treatment arm indicator (i.e., the indicator of the initial randomized treatment) as

Table 6.8 Covariates for modeling weights in IPCW estimators

Covariates
Baseline covariates:
Age (years, continuous)
Time since diagnosis (categorical; <5 years vs. ≥5 years)
Number of bone metastases at screening (categorical; ≤5 vs. >5)
Presence of visceral disease at baseline (categorical; yes vs. no)
Type of disease progression at study entry (categorical; PSA progression only vs. radiographic progression with or without PSA vs. no disease progression at study entry)
Baseline EQ-5D utility index (continuous)
Baseline FACT-P total score (continuous)
Time-dependent covariates:
ECOG Performance Status (categorical; 0 vs. >0)
History of grade 3/4/5 adverse events (categorical; yes vs. no)
Occurrence of grade 3/4/5 adverse events since last visit (categorical; yes vs. no)
Corticosteroid use (categorical; yes vs. no)
PSA level (continuous)
Laboratory tests: LDH level (categorical; ≤240 IU/mL vs. >240 IU/mL)
EQ-5D utility index (continuous)
FACT-P total score (continuous)
Time since treatment discontinuation (continuous)
Time to treatment discontinuation (continuous)[a]
Disease progression (categorical; yes vs. no)[a]

ECOG Eastern Cooperative Oncology Group, *FACT-P* Functional Assessment of Cancer Therapy-Prostate, *LDH* lactate dehydrogenase, *PSA* prostate-specific antigen

[a] Although disease progression and time to treatment discontinuation do not vary with time, they could be important covariates to be accounted for in the estimation of the weights

covariates. The weights are the subject- and interval-specific stabilized weights as described above.

Because the standard errors for the HRs obtained from the Cox regression analysis do not account for the variability associated with the estimation of the stabilized weights, 95% confidence intervals for HR estimates are obtained by bootstrapping (Hernán et al. 2001; see also Sect. 6.2). This method involves resampling with replacement from the experimental and placebo arms to obtain B (here $B = 100$ for illustration, but we would recommend 2000) bootstrap samples of the original data and repeating all the steps above for each of these samples to calculate B bootstrap estimates of the HR. A 95% CI for the HR is estimated based on the 2.5 and 97.5 percentiles of B bootstrap replicates.

6.5.3.4 Estimating Stabilized Weights with Logistic Regression

To estimate the numerator of the stabilized weights, a logistic regression (model 1) was fitted to the “stacked data” (i.e., with multiple records per patient) from all patient intervals from randomization until treatment switch or failure or censoring, defined as death, withdrawal of consent, or end of study, whichever occurred first. The probability of remaining uncensored was modeled conditional on patient baseline factors listed in Table 6.8 and a time-dependent intercept. The time-dependent intercept was estimated by including a variable indicating the number of days elapsed since randomization at the start of the interval and its quadratic term. The dependent variable in the logistic model was a binary variable (1/0) indicating whether the patient had switched treatment or not during the interval.

To estimate the denominator of the stabilized weights, a similar logistic regression (model 2) was fitted in which the probability of remaining uncensored was modeled conditional on the same baseline factors as above plus patient time-dependent covariates measured at the start of each interval, as listed in Table 6.8. Upon randomized study drug discontinuation, patients are typically followed mainly in terms of their survival status and initiation of new therapies, while other regular study assessments, e.g., ECOG, LDH, SPA, etc., are no longer performed. Therefore, only data as observed at the time of study treatment discontinuation (fixed) and time since treatment discontinuation (time-varying) are used as predictors of treatment switching in our models for the denominator of the weights. In a typical study, the probability of treatment switching prior to study treatment discontinuation is zero by trial design (alternatively, the probability of remaining uncensored is 1). Therefore, the probability of being uncensored was set to 1 for patient intervals prior to study treatment discontinuation, and these observations were not used in the estimation of this logistic model.

For all patient intervals prior to the date at which patients were assumed to be at risk of informative censoring (treatment switching, i.e., the date of study treatment discontinuation), stabilized weights were calculated. The numerator of $SW(j)$ was obtained using the estimates of the first model as described above, and the denominator of $SW(j)$ was set to 1.0 (i.e., the time-dependent probability of switch set equal to zero). Thus, these weights are always less than 1.0. For subsequent intervals, the numerator of $SW(j)$ was calculated using model 1, and the denominator was calculated using model 2. These weights may be greater than 1.0.

6.5.3.5 Estimating Stabilized Weights Using Random Forests

Stabilized weights were also estimated using random forests, in a manner similar as described above for the logistic regression, i.e., fitting separate models within each treatment arm as well as for the numerator and denominator of the weights, using the same baseline and time-dependent covariates, and data from the same patient intervals. This analysis was carried out using the R package **randomForest**. The model can be fit using the following function:

$$model = randomForest(predictors, as.factor(outcome), ntree = 1000),$$

where

- “predictors” contains a matrix with the values of patient covariates included in the model with rows corresponding to patient intervals used to fit the model.
- “outcome” is a vector of binary values representing the switching indicator for each patient interval as previously described.
- “ntree” is a parameter of the random forest algorithm specifying the number of classification trees that are fit as part of the random forest.

A default setting is used for the number of covariates that are randomly chosen as candidates for the splits (the “mtry” parameter) when building the classification trees so that the square root of the number of all available predictors is used. Once the model is estimated, predicted probabilities of not switching (remaining uncensored due to the non-ignorable reason) can be obtained using the function “predict” from the **randomForest** package:

$$pred = as.data.frame\left(predict\left(model, newdata = predictors, type = \text{“}prob\text{”}\right)\right)$$

where

- “model” is a “randomForest” object estimated above.
- “newdata = predictors” specifies that predictions should be provided for the same data set of patient intervals and covariate values.
- type = “prob” argument requests predictions in the form of probabilities as opposed to binary outcomes. These predicted probabilities are used for the calculation of the numerator or denominator of the stabilized weights.

6.5.3.6 Results

When applying the IPCW method, it is important to explore the distributions of the weights estimated in the first part of the method. A necessary condition for the correct model specification is that the stabilized weights have a mean of 1 (Hernán and Robins 2006).

Summary statistics on the stabilized weights for the IPCW analysis are presented in Table 6.9. Irrespective of the method used to estimate the probability of not being informatively censored (logistic regression or random forest), for both treatment

Table 6.9 Descriptive statistics for stabilized weights in IPCW models

Treatment arm	N	Mean	STD	Min	Max	Q1	Median	Q3
Logistic regression								
Placebo	10,692	1.01	0.25	0.87	12.10	0.98	0.99	1.00
Experimental	11,039	1.00	0.07	0.92	2.78	0.98	1.00	1.00
Random forest								
Placebo	10,692	1.02	0.24	0.27	9.60	1.00	1.00	1.00
Experimental	11,039	1.01	0.11	0.28	3.67	1.00	1.00	1.00

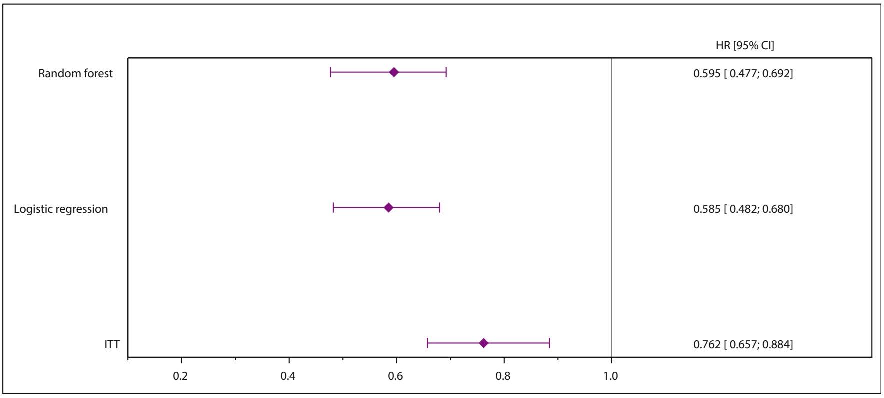

Fig. 6.7 Results of the unadjusted analysis and the IPCW method. The 95% CI are obtained from bootstrapping

arms, the mean of the stabilized weights is very close to 1, as expected. The median of the weights is also close to 1.

The results of the unadjusted analysis and the IPCW method using stabilized weights obtained from both logistic regression and random forest method are provided in Fig. 6.7. The unadjusted results were obtained with the analysis where all ITT subjects were included in the analysis set and no censoring was applied at the point of treatment switching.

Adjusting for the treatment switching, as well as for other baseline characteristics, indicates that the experimental treatment was associated with reduction in the risk of mortality of approximately 41% irrespective of the method used to obtain the stabilized weights (HR = 0.59; 95% CI [0.48; 0.68] using logistic regression and HR = 0.60; 95% CI [0.48; 0.69] using random forest). The unadjusted HR was 0.76, 95% CI [0.66; 0.88]. A smaller HR from the adjusted analysis is expected because there are more switchers in the placebo arm than in the experimental arm which is appropriately accounted for in the adjusted analysis.

As discussed in Sect. 6.3.1, random forests can also provide an insight into which covariates are most predictive of the outcome using the estimated variable importance scores. They can be obtained using the function “importance”:

$$VI = importance(model, type = 1)$$

where the argument “type=1” requests the VI scores estimated based on the mean decrease in accuracy from permuting out-of-bag data (see Sect. 6.3.1). For example, from the treatment-specific models used for the denominators of the weights including baseline and time-dependent covariates, the VI scores are as illustrated in Fig. 6.8. We can see that in both treatment arms, the top four predictors are the time to treatment discontinuation, PSA level, time from randomization, and age.

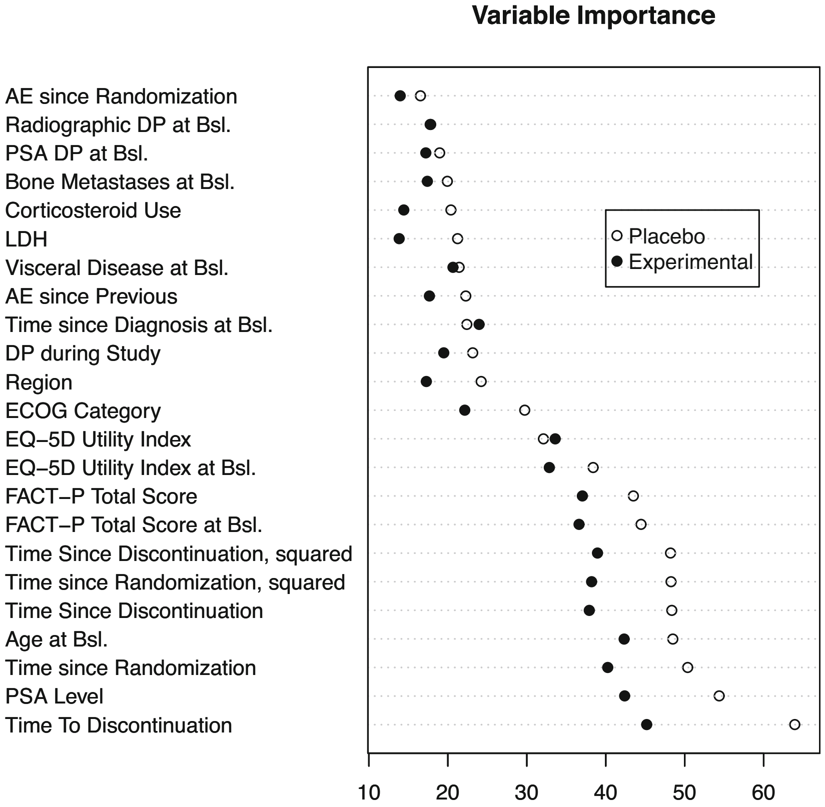

Fig. 6.8 Variable importance scores from random forest models of treatment switching based on baseline and time-dependent covariates (models for denominators of stabilized weights)

To gain further insight into the relationship between the top predictors and the probability of treatment switching, we can obtain partial dependence plots (using the function "partialPlot") that provide a graphical display of the marginal effects of the variables of interest on class probability. Figure 6.9 illustrates such partial dependence plots for the three top predictors in the treatment-specific models of weight denominators.

6.5.3.7 Discussion

The objective of these analyses was to estimate the effect of experimental treatment vs. placebo, adjusting for the potentially confounding effects of receipt of nonstandard anticancer therapy in both treatment groups. This is of particular interest for economic evaluations considering a lifetime horizon where standard ITT analyses are likely to be inappropriate in the presence of treatment switching failing

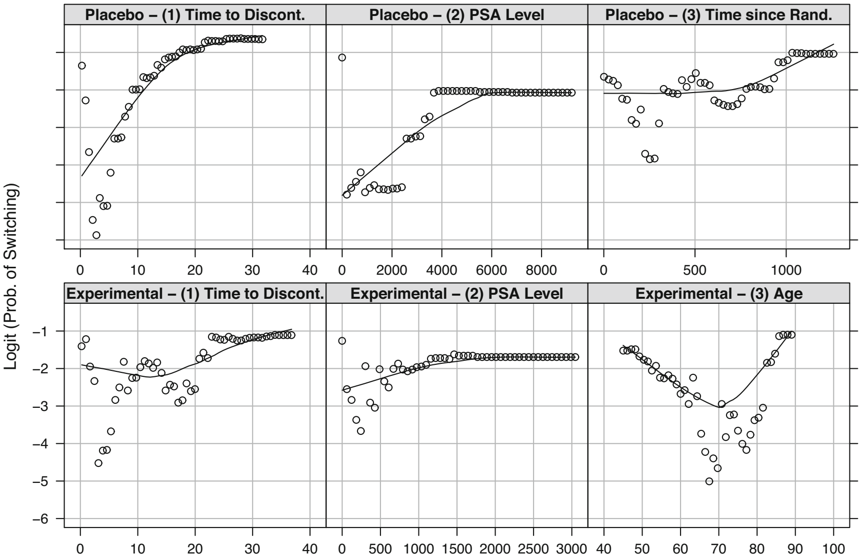

Fig. 6.9 Partial dependence plots for three top predictors from random forest models of treatment switching based on baseline and time-dependent covariates (models for denominators of stabilized weights)

to inform the decision problem (selecting the most effective therapy for a given patient population) from the causal inference perspective.

The IPCW method to adjust for the treatment switching was chosen in this example because a large number of potentially prognostic covariates (that can influence the investigator’s decision to switch treatment for a prostate cancer patient) were available for the analysis. Also, as suggested by the theory and existing evidence, IPCW is best suited for studies where the switching proportions are not very high (Latimer and Abrams 2014), which is the case in our example data set. The IPCW method is reliant on the assumption of “no unmeasured confounders” which is not testable from observed data. One strategy is to include in the analysis a comprehensive set of potentially important confounders identified using expert knowledge (which may include redundant covariates) and rely on powerful machine learning methods to extract useful information in the process of model building. Whether the results could substantially change after including covariates entirely missing in the observed data can be evaluated using sensitivity analyses framework (see Brumback et al. 2004; Klungsøyr et al. 2009).

We have applied the IPCW method where weights were estimated using two approaches: a traditional logistic regression and a modern method of statistical learning, the random forest. In our example, the results using the logistic regression and random forest models of treatment switching provided similar results. The random forest model is of particular interest as it is free of parametric model assumptions, can effectively deal with a large number of predictors without overfitting, is known for its good predictive accuracy, and provides useful insights into the predictive strength of considered covariates and their relationship to the outcome.

6.6 Discussion and Conclusions

DMML methods are becoming now an integral part of data analysis at all stages of clinical drug development, which can be contrasted with its primary use in preclinical stage of "drug discovery" in the past. The need in DMML arises whenever a model selection is entertained, which may occur in different tasks including traditional estimation of the overall treatment effect in the presence of potential confounding due to post-randomization events and novel tasks of treatment optimization in the realm of personalized/precision medicine.

A wealth of patient data collected during the clinical development program may be better utilized with the principled use of DMML that should inform a decision-making process across the entire drug development cycle. We hope that the references to examples of various clinical applications and case studies provided in this chapter will give the reader an appreciation of the breadth of areas where the power of DMML can be leveraged.

Application of DMML to clinical data has some unique features. Unlike more traditional applications of DMML (such as speech and character recognition), with potentially unlimited amount of data that can be used for model training, DMML in clinical settings is dealing with relatively small number of records due to substantial costs and other constraints associated with each patient that can be enrolled in a clinical study. Therefore, a typical application of DMML in the clinical world is within the medium or small "*n*" and medium/large "*p*." Cross-validation and other resampling-based methods, therefore, play a key role.

Modeling of clinical data, whether randomized or based on observational studies, involves methods accounting for different sources of confounding and missing data. This explains the trend of integrating DMML and casual inference methods in some applications.

Another feature of applications of DMML in drug development is the need to control the Type I error or false discovery rate which is a new trend in the area of machine learning that historically considered the concept of statistical significance irrelevant. Typically, the analytical form of the null distribution for many DMML techniques is not available, and one needs to resort to methods of resampling.

It is important to understand that the multiplicity control is interrelated with model complexity control: the latter effectively restricts the model search space and results in a lesser multiplicity burden.

It is a common trend for DMML applications in clinical data that the decision-makers desire interpretable solutions rather than a "black box" which can often be achieved by post-processing the "black box" to produce interpretable graphical displays, such as trees, marginal plots, low dimensional projections, etc.

"Data mining" in the clinical world sometimes was ascribed a negative connotation as "data dredging." However, we argue that using principled DMML strategies and pre-specification of analytic strategies in the data mining plans may help remove the stigma from data mining, making it a valuable set of tools for improved decision-making in the drug development process.

Acknowledgments The authors would like to thank Matthew Rotelli, Qianyi Zhang, and Chakib Battioui for many stimulating discussions on data mining and its role in drug development process, Alex Dmitrienko for long-standing collaboration on developing the SIDES method, Fan Wu, Eric Laber and Emanuel Severus for their generous help with the case study on Q-learning, and Natallia Katenka and Anthony Zagar for reviewing the draft of the chapter and providing many valuable suggestions that led to substantially improved presentation of the material.

References

Abtroun L, Bunouf P, Gendreau RM, Vitton O (2016) Is the efficacy of milnacipran in fibromyalgia predictable? A data-mining analysis of baseline and outcome variables. Clin J Pain 32:435–440

Aharon M, Elad M, Bruckstein A (2006) K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311-4322

Akaike H (1974) A new look at the statistical model identification. IEEE T Automat Contr 19 (6):716–723

Allen D (1974) The relationship between variable selection and data augmentation and a method of prediction. Technometrics 16:125–127

Almirall D, Nahum-Shan I, Sherwood NE, Murphy SA (2014) Introduction to SMART designs for the development of adaptive Interventions: with application to weight loss research. Transl Behav Med 4(3):260-274

Altmann A, Toloşi L, Sander O, Lengauer T (2010) Permutation importance: a corrected feature importance measure. Bioinformatics 26(10):1340-1347

Amaratunga D, Cabrera J. (2004) Mining data to find subsets of high activity. J Stat Plan Inference 122:23-41

Amaratunga D, Cabrera J, Lee Y-S (2008) Enriched random forests. Bio-informatics 24(18):2010-2014

Ambroise C, McLachlan G (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci USA 99:6562–6566

Aridas CK, Kotsiantis SB, Vrahatis MN (2016) Increasing diversity in random forests using Naive Bayes. In Iliadis L, Maglogiannis I (eds) Artificial Intelligence Applications and Innovations, 12th IFIP WG 12.5 International Conference and Workshops, pp. 75–86

Ashley EA (2015) The precision medicine initiative. A national effort. J Am Med Assoc 313 (21):2119-2120

Barber RF, Candès EJ (2015). Controlling the false discovery rate via knockoffs. Ann Stat 43 (5):2055-2085

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. J R Statist Soc Series B 57(1):289-300

Bühlmann P, Hothorn T (2010) Twin Boosting: improved feature selection and prediction, Stat Comput 20:119-138

Berger J, Wang X, Shen L (2014) A Bayesian approach to subgroup identification. J Biopharm Stat 24:110–129

Blatt D, Murphy SA, Zhu J (2004) A-learning for approximate planning. Technical Report 04-63, The Methodology Center, Pennsylvania State Univ., State College, PA

Bonetti M, Gelber RD (2000) A graphical method to assess treatment–covariate interactions using the Cox model on subsets of the data. Stat Med 19:2595–2609

Bonetti M, Gelber RD (2004) Patterns of treatment effects in subsets of patients in clinical trials. Biostatistics 5(3):465–481

Bornkamp B, Pinheiro J, Bretz F. (2009) MCPMod: An R package for the design and analysis of dose-finding studies. J Stat Softw 29(7)1:23

Bornkamp B, Ohlssen D, Magnusson B, Schmidli H (2016) Model averaging for treatment effect estimation in subgroups. Pharm Stat. DOI: https://doi.org/10.1002/pst.179

Breiman L (1996) Bagging predictors. Mach Learn 26:123–140

Breiman L (2001a) Random forests. Mach Learn 45(1):5-32

Breiman L (2001b) Statistical modeling: The two cultures. Stat Sc 16:199–231

Breiman L, Spector P (1992) Submodel selection and evaluation in regression: the X-random case. Int Stat Rev 60:291–319

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and Regression Trees. Chapman & Hall, London

Bretz F, Pinheiro JC, Branson M (2005) Combining multiple comparisons and modeling techniques in dose-response studies. Biometrics 61:738-748

Brumback BA, Hernán MA, Haneuse SJ, Robins JM (2004) Sensitivity analyses for unmeasured confounding assuming a marginal structural model for repeated measures. Stat Med 23(5):749-767

Bühlmann P, Horthorn T (2007) Boosting algorithms: regularization, prediction and model fitting. Stat Sci 22(4):477-505

Burgel PR, Paillasseur JL, Roche N (2014) Identification of clinical phenotypes using cluster analyses in COPD patients with multiple comorbidities. BioMed Res Int Article ID 420134

Burges C (1998) A tutorial on support vector machines for pattern recognition. Knowl Discov Data Min 2(2):121–167

Cai T, Tian L, Wong P, Wei LJ (2011) Analysis of randomized comparative clinical trial data for personalized treatment selections. Biostatistics 12:270–282

Cattell RB (1952) Factor analysis. New York: Harper

Chakraborty B, Moodie EE (2013) Statistical reinforcement learning. Gail M, Krickeberg K, Samet J, Tsiatis A, Wong W (eds) Statistical Methods for Dynamic Treatment Regimes. Springer, New York

Chakraborty B, Murphy SA (2014) Dynamic treatment regimes. Annu Rev Stat Appl 1:447–464

Chakraborty B, Laber EB, Zhao Y (2013) Inference for optimal dynamic treatment regimes using an adaptive m-out-of-n bootstrap scheme. Biometrics 69(3):614-723

Chaudhuri P, Lo W-D, Loh W-Y, Yang C-C (1995) Generalized regression trees. Stat. Sinica 5:641–666

Chen G, Zhong H, Belousov A, Viswanath D (2015) PRIM approach to predictive-signature development for patient stratification. Stat Med 34:317–342

Clarke B, Fokoué E, Zhang HH (2009) Principles and Theory for Data Mining and Machine Learning. Springer, New York

Collins LM, Murphy SA, Strecher V (2007) The multiphase optimization strategy (MOST) and the sequential multiple assignment randomized trial (SMART): New methods for more potent e-health interventions. Am J Prev Med 32(5 Suppl):S112-S118

Conrad DJ, Bailey BA (2015) Multidimensional clinical phenotyping of an adult cystic fibrosis patient population. PLoS One 10(3):e0122705

Cosma G, Brown D, Archer M, Khan M, Pockley AG (2017) A survey on computational intelligence approaches for predictive modeling in prostate cancer, Expert Syst Appl 70:1-19

Davis RB, Anderson JR (1989) Exponential survival trees. Stat Med 8:947-961

Defays D (1977) An efficient algorithm for a complete-link method. Comput J British Comput Soc 20 (4):364–366

Dixon DO, Simon R (1991) Bayesian subset analysis. Biometrics 47:871–882

Dmitrienko A, Lipkovich I, Hopkins A, Li YP, Wang W (2015) Biomarker evaluation and subgroup identification in a pneumonia development program using SIDES. Applied Statistics in Biomedicine and Clinical Trials Design. Chen Z, Liu A, Qu Y, Tang L, Ting N, Tsong Y. (editors). Springer

Docampo E, Collado A, Escaramís G, Carbonell J, Rivera J, Vidal J, Alegre J, Rabionet R, Estivill X (2013) Cluster analysis of clinical data identifies fibromyalgia subgroups. Baradaran HR (ed) PLoS One 8(9):e74873

Domingos P (2000) Bayesian averaging of classifiers and the overfitting problem. In: Proceedings of the 17th International Conference on Machine Learning, pp. 223–230

Domingos P (2012) A few useful things to know about machine learning. Commun ACM 55 (10):78-87

Doubleday K (2016) Generation of Individualized Treatment Decision Tree Algorithm With Application to Randomized Control Trials and Electronic Medical Record Data. Master Theses, The University of Arizona, available at http://arizona.openrepository.com/arizona/bitstream/10150/613559/1/azu_etd_14716_sip1_m.pdf

Dusseldorp E, Van Mechelen I (2014) Qualitative interaction trees: A tool to identify qualitative treatment-subgroup interactions. Stat Med 33:219–237

Dusseldorp E, Conversano C, Van Os BJ (2010) Combining an additive and tree-based regression model simultaneously: STIMA. J Comp Graph Stat 19:514–530

Efron B (1979) Bootstrap methods: another look at the jackknife, Ann Stat 7:1–26

Efron B (2010) Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction. Cambridge University Press

Efron B, Hastie T (2016) Computer Age Statistical Inference: Algorithms, Evidence, and Data Science. Cambridge University Press: New York

Efron B, Tibshirani R (1997) Improvements on crossvalidation: The 0.632+ bootstrap method. J Am Stat Assoc 92:548–560

Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407-499

Ertefaie A, Almiral D, Huang L, Dziak JJ, Wagner AT, Murphy SA (2012) SAS PROC QLEARN users′ guide (Version 1.0.0). University Park: The Methodology Center, Penn State. Available from http://methodology.psu.edu

Evgeniou T, Pontil M, Poggio T (2000) Regularization networks and support vector machines, Adv Comput Math 13(1):1–50

Faye LL, Sun L, Dimitromanolakis A, Bulla SB (2011) A flexible genome-wide bootstrap method that accounts for ranking and threshold-selection bias in GWAS interpretation and replication study design. Stat Med 30(15):1898-912

FDA (U.S. Food and Drug Administration) (2018) “FDA permits marketing of artificial intelligence algorithm for aiding providers in detecting wrist fractures” FDA News Release, May 24, 2018; https://www.fda.gov/newsevents/newsroom/pressannouncements/ucm608833.htm

Ferguson JP, Cho JH, Yang C, Zhao H (2013) Empirical Bayes correction for the Winner’s Curse in genetic association studies. Genet Epidemiol 37(1):60–68

Forgy E (1965) Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. Biometrics 21:768–769

Foster JC, Taylor JMC, Ruberg SJ (2011) Subgroup identification from randomized clinical trial data. Stat Med 30:2867–2880

Foster JC, Taylor JMG, Kaciroti N, Nan B (2015) Simple subgroup approximation to optimal treatment regimes from randomized clinical trial data. Biostatistics 16(2):368-82

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comp Syst Sci 55(1):119-139

Friedman J (1991) Multivariate adaptive regression splines (with discussion). Ann Stat 19(1):1–141

Friedman JH (1997) Data mining and statistics: what's the connection? In: Proceedings of Symposium on the Interface Between Computer Science and Statistics

Friedman J (1999) Stochastic gradient boosting, Technical report, Stanford University

Friedman J (2001) Greedy function approximation: A gradient boosting machine. Ann of Stati 29 (5):1189–1232

Friedman JH, Fisher NI (1999) Bump hunting in high-dimensional data. Stat Comput 9:123–143

Friedman J, Hall P (2007) On bagging and nonlinear estimation. J Stat Plan Inference 137:669–683

Friedman JH, Popescu BE (1999) Predictive Learning via Rule Ensembles. Ann of Appl Stat 2:916–954

Friedman J, Stuetzle W (1981) Projection pursuit regression. J Am Statist Assoc 76:817–823

Friedman JH, Tukey JW (1974) A projection pursuit algorithm for exploratory data analysis. IEEE Transactions on Computers, C–23 (9):881–890

Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting (with discussion). Annals of Statistics 28:337–407

Fu H, Zhou J, Faries DE (2016) Estimating optimal treatment regimes via subgroup identification in randomized control trials and observational studies. Stat Med 35(19):3285-3302

Geisser S (1975) The predictive sample reuse method with applications. J Am Stat Assoc 70 (350):320–328

Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63: 3–42

Gilmour SG (1996) The interpretation of Mallows's Cp-statistic. J R Stat Soc Ser D 45(1):49–56

Glickman ME, Rao SR, Schultz MR (2014) False discovery rate control is a recommended alternative to Bonferroni-type adjustments in health studies. J Clin Epidemiol 67(8):850-857

Goldberg Y, Kosorok, MR (2012) Q-learning with Censored Data. Ann Stat 40(1):529-560

Goodfellow I, Bengio J, Courville A, Bach F (2016) Deep Learning. MIT Press: Cambridge, MA

Gordon L, Olshen RA (1985) Tree-structured survival analysis. Cancer Treat. Rep 69:1065–1069

Gower JC, Hand DJ (1996) Biplots. Chapman and Hall: London

Greenacre MJ (1984) Theory and Applications of Correspondence Analysis. Academic Press: London

Gu X, Yin G, Lee JJ (2013) Bayesian two-step Lasso strategy for biomarker selection in personalized medicine development for time-to-event endpoints. Contemp Clin Trials 36:642–650

Gunter L, Zhu J, Murphy S (2011) Variable selection for qualitative interactions in personalized medicine while controlling the familywise error rate. J Biopharm Stat 21:1063–1078

Hand DJ (1998) Data mining: statistics and more? Am Stat 52(2):112-118

Hand DJ, Mannila H, Smyth P (2001) Principles of Data Mining. The MIT Press: Cambridge.

Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Ana. Mach Intell 12 (10):993–1001

Hardin DS, Rohwer RD, Curtis BH, Zagar A, Chen L, Boye KS, Jiang HH, Lipkovich IA (2013) Understanding heterogeneity in response to antidiabetes treatment: A post hoc analysis using SIDES, a subgroup identification algorithm. J Diab Sci Technol 7:420–429

Harpaz R, DuMouchel W, Shah NH, Madigan D, Ryan P, Friedman C (2012) Novel data mining methodologies for adverse drug event discovery and analysis. Clin Pharmacol Ther 91(6):1010-1021

Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Data Mining, Inference, and Prediction, 2nd Edition. Springer-Verlag: New York

Henderson NC, Louis TA, Wang C, Varadhan R (2016) Bayesian analysis of heterogeneous treatment effects for patient-centered outcomes research. Health Serv Outcomes Res Methodol 16(4):213–233

Henderson NC, Louis TA, Rosner G, Varadhan R (2017) Individualized treatment effects with censored data via fully nonparametric Bayesian accelerated failure time models. Available arXiv preprint arXiv: 1706.06611v1

Herland M, Khoshgoftaar TM, Wald R (2014) A review of data mining using big data in health informatics. J Big Data 1:2

Hernán MA, Robins JM (2006) Estimating causal effects from epidemiological data. J Epidemiol Community Health 60:578–586

Hernán MA, Brumback B, Robins JM (2001) Marginal structural models to estimate the joint causal effect of nonrandomized treatments. J Am Stat Assoc 96(454):440-448

Ho, TK (1995) Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, pp. 278–282

Ho TK (2002) A data complexity analysis of comparative advantages of decision forest constructors. Pattern Anal Appl 5(2):102–112

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999) Bayesian model averaging: A tutorial. Stat Sci 14(4): 382–417

Hodges JS, Cui Y, Sargent DJ, Carlin BP (2007) Smoothing balanced single-error-term analysis of variance. Technometrics 49:12–25

Hoerl AE, Kennard R (1970) Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12:55–67

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol, 24:417–441

Hothorn T, Hornik K, Zeileis A (2006) Unbiased recursive partitioning: A conditional inference framework. J Comp Graph Stat 15(3):651-674

Hou J, Seneviratne C, Su X, Taylor J, Johnson B, Wang XQ, Zhang H, Kranzler HR, Kang J, Liu L (2015) Subgroup identification in personalized treatment of alcohol dependence. Alcohol Clin Exp Res 39(7):1253-1259

Huang Y, Fong Y (2014) Identifying optimal biomarker combinations for treatment selection via a robust kernel method. Biometrics 70:891–901

Hyvärinen A, Oja E (2000) Independent component analysis: Algorithms and applications. Neural Networks 13:411–430

Imai K, Ratkovic M (2013) Estimating treatment effect heterogeneity in randomized program evaluation. Ann Appl Stat 7:443–470

Ishwaran H, Kogalur U, Blackstone E, Lauer M (2008) Random survival forests. Ann Appl Stat 2 (3):841–860

Jacova C, Slack PJ, Hsiung G-YR, Beattie BL, Lee P (2013) Patients' self-reports on function and cognition in Alzheimer's disease are strongly influenced by their affective states: Principal component analysis of the CLIMAT scale. Alzheimers Dement 9(4):650

Janes H, Brown MD, Pepe M, Huang Y (2013) Statistical methods for evaluating and comparing biomarkers for patient treatment selection. UW Biostatistics Working Paper Series. Working Paper 389. http://biostats.bepress.com/uwbiostat/paper389

Janes H, Brown M, Pepe M, Huang Y (2014) An approach to evaluating and comparing biomarkers for patient treatment selection. Int J Biostat 10(1):99-121

Johnson P, Greiner W, Al-Dakkak I, Wagner S (2015) Which metrics are appropriate to describe the value of new cancer therapies? Biomed Res Int 2015:865101

Jolliffe IT (2002) Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer: New York

Jones HE, Ohlssen DI, Neuenschwander B, Racine A, Branson M (2011) Bayesian models for subgroup analysis in clinical trials. Clin Trials 8:129–143

Jordan M, Jacobs R (1994) Hierachical mixtures of experts and the EM algorithm. Neural Comput 6:181–214

Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. App Stat 29:119-127

Kaufman L, Rousseeuw P (1990) Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York

Kavakiotis I, Tsave O, Salifoglou A, Maglaveras N, Vlahavas I, Chouvarda I (2017) Machine learning and data mining methods in diabetes research. Comput Struct Biotechnol J 15:104-116

Kehl V, Ulm K (2006) Responder identification in clinical trials with censored data. Comput Stat Data Anal 50:1338–1355

Khalilia M, Chakraborty S, Popescu M (2011) Predicting disease risks from highly imbalanced data using random forest. BMC Medical Informatics and Decision Making 11:51

Kim H, Loh WY (2001) Classification trees with unbiased multiway splits. J Am Stat Assoc 96:589-604

Kim H, Loh WY (2003) Classification trees with bivariate linear discriminant node models. J Comput and Graph Statsit 12:512-530

Kim H-C, Ghahramani Z (2012) Bayesian classifier combination. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics 22:619–627

Klungsøyr O, Sexton J, Sandanger I, Nygård JF (2009) Sensitivity analysis for unmeasured confounding in a marginal structural Cox proportional hazards model. Lifetime Data Anal 15 (2):278-294

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceeding IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, pp. 1137–1143

Kohonen T (1989) Self-Organization and Associative Memory (3rd edition), Springer: Berlin

Konstantina K, Themis PE, Konstantinos PE, Michalis VK, Dimitrios IF (2015) Machine learning applications in cancer prognosis and prediction. Comput Struct Biotechnol J 13:8–17

Kotajima L, Aotsuka S, Nishimaki T, Kashiwagi H, Kunieda T, Tojo T, Yokohari R (1997) Classification tree criteria of pulmonary hypertension in mixed connective tissue disease. Jpn J Rheumatol 7(4):293-303

Kruskal J B, Wish M. (1978) Multidimensional Scaling. Beverly Hills, California: Sage.

Krstajic D, Buturovic LJ, Leahy DE, Thomas S (2014) Cross-validation pitfalls when selecting and assessing regression and classification models. J Cheminformatics 6:10

Kutcher ME, Ferguson AR, Cohen MJ (2013) A principal component analysis of coagulation after trauma. J Trauma Acute Care Surg 74(5):1223-1230

Laber EB, Lizotte DJ, Ferguson B (2014a) Set-valued dynamic treatment regimes for competing outcomes. Biometrics 70:53–61

Laber EB, Lizotte DJ, Qian M, Pelham WE, Murphy SA (2014b) Dynamic treatment regimes: technical challenges and applications. Electron J Stat 8(1):1225–1272

Lamont A, Lyons MD, Jaki T, Stuart E, Feaster DJ, Tharmaratnam K, Oberski D, Ishwaran H, Wilson DK, Horn MLW (2016). Identification of predicted individual treatment effects in randomized clinical trials. Stat Methods Med Res Mar 17. pii: 0962280215623981

Latimer NR, Abrams KR (2014) NICE DSU Technical Support Document 16: Adjusting survival time estimates in the presence of treatment switching. Available from http://www.nicedsu.org.uk

Latimer NR, Abrams KR, Lambert PC, Crowther MJ, Wailoo AJ, Morden JP, Akehurst RL, Campbell MJ (2014) Adjusting survival time estimates to account for treatment switching in randomized controlled trials-an economic evaluation context: methods, limitations, and recommendations. Med Decis Making 34(3):387-402

Lebedev AV, Westman E, Van Westen GJP, et al. for the Alzheimer's Disease Neuroimaging Initiative and the AddNeuroMed consortium (2014) Random Forest ensembles for detection and prediction of Alzheimer's disease with a good between-cohort robustness. NeuroImage: Clinical 6:115-125

LeBlanc M, Crowley J (1992) Relative Risk Trees for Censored Survival Data. Biometrics 48:411-425

LeBlanc M, Crowley J (1993) Survival trees by goodness of split. J Am Stat Assoc 88:457–467

Lee BK, Lessler J, Stuart EA (2010) Improving propensity score weighting using machine learning. Stat Med 29:337-346

Lei H, Nahum-Shani I, Lynch K, Oslin D, Murphy S (2012) A “Smart” design for building individualized treatment sequences. Annu Rev Clin Psychol 8:21–48

Lin Y, Jeon Y (2006) Random forests and adaptive nearest neighbors. J Am Stat Assoc 101 (474):578-590

Linn KA, Laber EB, Stefanski LA (2015) iqLearn: Interactive Q-learning in R J Stat Softw 64(1): i01

Lipkovich I, Dmitrienko A (2014) Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using SIDES. J Biopharm Stat 24:130–153

Lipkovich I, Dmitrienko A, D'Agostino BR Sr (2017) Tutorial in biostatistics: Data-driven subgroup identification and analysis in clinical trials. Stat Med 36(1):136-196

Lipkovich IA, Houston JP, Ahl J (2008) Identifying patterns in treatment response profiles in acute bipolar mania: a cluster analysis approach. BMC Psychiatry 8:65

Lipkovich I, Dmitrienko A, Denne J, Enas G (2011) Subgroup identification based on differential effect search (SIDES): A recursive partitioning method for establishing response to treatment in subject subpopulations. Stat Med 30:2601–2621

Lipkovich IA, Choy EH, Van Wambeke P, Deberdt W, Sagman D (2014) Typology of patients with fibromyalgia: cluster analysis of duloxetine study patients. BMC Musculoskeletal Disorders 15:450-460

Lipkovich IA, and Smith EP (2002) Biplot and singular value decomposition macros for Excel©. J Stat Softw 7(5)

Little RJ, Rubin DB (2000) Causal effects in clinical and epidemiological studies via potential outcomes. Annu Rev Public Health 21:121–45

Lloyd S (1957) Least squares quantization in PCM. Technical report, Bell Laboratories. Published in 1982 in IEEE Transactions on Information Theory 28:128–137

Lockhart R, Taylor J, Tibshirani RJ, Tibshirani R (2014) A significance test for the lasso. Ann Stat 42:413–463

Loh W-Y (2002) Regression trees with unbiased variable selection and interaction detection. Statistica Sinca 12:361-386

Loh W-Y (2006) Logistic regression tree analysis. Pham H (ed) Handbook of Engineering Statistics, Springer, New York, pp. 537–549

Loh W-Y (2014) Fifty years of classification and regression trees. Int Statist Rev 82(3):329-348

Loh W-Y, Shih YS (1997) Split selection methods for classification trees. Statistica Sinca 7:815-840

Loh W-Y, Vanichsetakul N (1988) Tree-structured classification via generalized discriminant analysis. J Am Stat Assoc 83:715-725

Loh W-Y, Zheng W (2013) Regression trees for longitudinal and multiresponse data. Ann Applied Statist 7:495-522

Loh W-Y, He X, Man M (2015) A regression tree approach to identifying subgroups with differential treatment effects. Stat Med 34:1818-1833

Loh W-Y, Fu H, Man M, Champion V, Yu M (2016) Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables. Stat Med 35(26):4837-4855

Lu Y, Black D, Genant HK, Mathur AK (2003) Study of hip fracture risk using tree structured survival analysis. Journal für Mineralstoffwechsel 10(1):11-16

Luo Q, Mehra S, Golden NA, Kaushal D, Lacey MR (2014) Identification of biomarkers for tuberculosis susceptibility via integrated analysis of gene expression and longitudinal clinical data. Front Genet 5:240

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. Adv in Neural Inf Process Syst 30:4765–4774

Lundberg SM, Lee S-I (2018) Consistent individualized feature attribution for tree ensembles. Available arXiv preprint arXiv:1802.03888v3

Macnaughton Smith P, Williams W, Dale M, Mockett L (1965) Dissimilarity analysis: a new technique of hierarchical subdivision. Nature 202:1034–1035

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. LeCam LM, Neyman J (eds) Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 281–297

Madigan D, Raftery A (1994) Model selection and accounting for model uncertainty using Occam's window. J Am Stat Assoc 89:1535–46

Mair J, Smidt J, Lechleiutner P, Dienstl F, Puschendorf B (1995) A decision tree for the early diagnosis of acute myocardial infarction in nontraumatic chest pain patients at hospital admission. Chest 108:1502-1509

Mason L, Baxter J, Bartlett P, Frean M (2000) Boosting algorithms as gradient descent. Adv Neural Inf Process Syst 12:512–518

Mayer C, Lipkovich I, Dmitrienko A (2015) Survey results on industry practices and challenges in subgroup analysis in clinical trials. Stat Biopharm Res 7:272–282

Meinshausen N (2006) Quantile regression forests. J Mach Learn Res 7:983–999

Meinshausen N, Meier L, Bühlmann P (2009) P-values for high-dimensional regression. J Am Stat Assoc 104:1671–1681

Minka T (2002) Bayesian model averaging is not model combination. MIT Media Lab Note https://tminka.github.io/papers/minka-bma-isnt-mc.pdf

Mitchell T (1997) Machine Learning. The McGraw-Hill Companies

Monteith K, Carroll JL, Seppi K, Martinez T (2011) Turning Bayesian model averaging into Bayesian model combination. In: Proceedings of International Joint Conference on Neural Networks, pp. 2657–2663

Moodie EE, Dean N, Sun YR (2014) Q-learning: Flexible learning about useful utilities. Stat Biosci 6(2):223–243

Moodie EE, Richardson TS, Stephens DA (2007) Demystifying optimal dynamic treatment regimes. Biometrics 63(2):447–455

Murphy SA (2003) Optimal dynamic treatment regimes. J R Stat Soc Ser B 65(part 2):331–366

Murphy SA (2005) An experimental design for the development of adaptive treatment strategies. Stat Med 24(10):1455–1481

Muthén B, Brown CH, Masyn K, Jo B, Khoo ST, Yang CC, Wang CP, Kellam SG, Carlin JB, Liao J (2002) General growth mixture modeling for randomized preventive interventions. Biostatistics 3(4):459-75

Nahum-Shani I, Qian M, Almirall D, Pelham WE, Gnagy B, Fabiano GA, Waxmonsky JG, Yu J, Murphy SA (2012) Q-learning: a data analysis method for constructing adaptive interventions. Psychol Methods 17(4):478–494

Neal R, Zhang J (2006) High Dimensional classification with Bayesian neural networks and Dirichlet diffusion trees. Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) Feature Extraction Foundations and Applications. Springer, New York, pp. 265–296

Nelson JC, Zhang Q, Debert W, Marangell LB, Karamustafalioglu O, Lipkovich IA (2012) Predictors of remission with placebo using an integrated study database from patients with major depressive disorder. Curr Med Res Opin 28(3):325-334

NICE (2014) Clinical guideline 175. Prostate cancer: diagnosis and treatment. January 2014. http://www.nice.org.uk/guidance/cg175

O'Kelly M. (2004) Using statistical techniques to detect fraud: A test case. Pharm Stat 3:237–246

Ondra T, Dmitrienko A, Friede T, Gradf A, Miller F, Stallard N, Posh M (2016) Methods for identification and confirmation of targeted subgroups in clinical trials: a systematic review. J Biopharm Stat 26(1):99-119

Orimaye SO, Wong JS-M, Golden KJ, Wong CP, Soyiri IN (2017) Predicting probable Alzheimer's disease using linguistic deficits and biomarkers. BMC Bioinformatics 18:34

Ouanes I, Schwebel C, Franais A, Bruel C, Philippart F, Vesin A, Soufir L, Adrie C, Garrouste-Orgeas M, Timsit JF, Misset B (2012) A model to predict short-term death or readmission after intensive care unit discharge. J Crit Care 27(4):422.e1–422.e9

Padjen I, Radner H, Öhler L, Smolen J, Aletaha D (2016) Understanding anemia in rheumatoid arthritis: The association of hemoglobin and hepcidin levels with clinical disease activity and acute phase response. Ann Rheum Dis 75:476

Patel S, Hee SW, Mistry D, Jordan J, Brown S, Dritsaki M, Ellard DR, Friede T, Lamb SE, Lord J, Madan J, Morris T, Stallard N, Tysall C, Willis A, Underwood M; the Repository Group. (2016) Identifying back pain subgroups: developing and applying approaches using individual patient data collected within clinical trials. Programme Grants for Applied Research, No. 4.10. Patel S, Hee SW, Mistry D, et al.; the Repository Group. Southampton (UK): NIHR Journals Library

Paydar K, Kalhori SRN, Akbarian M, Sheikhtaheri A (2017) A clinical decision support system for prediction of pregnancy outcome in pregnant women with systemic lupus erythematosus. Int J Med Informatics 97:239-246

Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2 (11):559–572

Prinzie A, Van den Poel D (2008) Random Forests for multiclass classification: Random MultiNomial Logit. Expert Syst Appl 34 (3):1721–1732

Qian M, Murphy SA (2011) Performance guarantees for individualized treatment rules. Ann Stat 39:1180–1210

Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106

Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo

Quinlan JR (2004) C5.0, www.rulequest.com

Ripley BD (1996) Pattern Recognition and Neural Networks. Cambridge University Press

Robins JM, Finkelstein DM (2000) Correcting for noncompliance and dependent censoring in an AIDS clinical trial with inverse probability of censoring weighted (IPCW) log-rank tests. Biometrics 56(3):779-788

Rosenkranz GK (2016). Exploratory subgroup analysis in clinical trials by model selection. Biom J 58(5):1217-1228

Royston P, Sauerbrei W (2004) A new approach to modelling interaction between treatment and continuous covariates in clinical trials by using fractional polynomials. Stat Med 23:2509–2525

Royston P, Sauerbrei W (2013) Interaction of treatment with a continuous variable: simulation study of power for several methods of analysis. Stat Med 32:3788-3803

Rumelhart D, Hinton G, Williams R (1986) Learning internal representations by error propagation. Rumelhart D, McClelland J (eds) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, The MIT Press, Cambridge, MA. pp. 318–362

Sacchet MD, Prasad G, Foland-Ross LC, Thompson PM, Gotlib IH (2015) Support vector machine classification of major depressive disorder using diffusion-weighted neuroimaging and graph theory. Front Psychiatry 6:21

Sachs GS, Thase ME, Otto MW, Bauer M, Miklowitz D, Wisniewski SR, et al. (2003) Rationale, design, and methods of the systematic treatment enhancement program for bipolar disorder (step-bd). Biol Psychiatry 53(11):1028–1042

Sandri M, Zuccolotto P (2008) A bias correction algorithm for the Gini variable importance measure in classification trees. J Comput Graph Stat 17(3):1-18

Schnell PM, Tang Q, Offen WW, Carlin BP (2016) A Bayesian credible subgroups approach to identifying patient subgroups with positive treatment effects. Biometrics 72(4):1026-1036

Schölkopf, B, Smola A, Müller K-R (1997) Kernel principal component analysis. P of International Conference on Artificial Neural Networks: 583–588

Schulte PJ, Tsiatis AA, Laber EB, Davidian M (2014) Q-and A-learning methods for estimating optimal dynamic treatment regimes. Stat Sci 29(4):640-661

Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464

Segal MR (1988) Regression trees for censored data. Biometrics 44(1):35-47

Segal MR (2004) Machine learning benchmarks and random forest regression. Technical report, eScholarship Repository, University of California. https://escholarship.org/uc/item/35x3v9t4

Segal M, Xiao Y (2011) Multivariate random forests. WIREs Data Mining and Knowledge Discovery 1:80–87

Seibold H, Zeileis A, Hothorn T (2015) Model-based recursive partitioning for subgroup analyses. Int J Biostat 12(1)

Seibold H, Zeileis A, Hothorn T (2016) Individual treatment effect prediction for ALS patients. Available arXiv preprint arXiv: 1604.08720

Shawe-Taylor J, Cristianini N (2004) Kernel Methods for Pattern Analysis. Cambridge University Press

Shi T, Seligson D, Belldegrun AS, Palotie A, Horvath S (2005) Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. Mod Pathol 18 (4):547–557

Shortreed SM, Laber E, Lizotte DJ, Stroup TS, Pineau J, Murphy SA (2011) Informing sequential clinical decision-making through reinforcement learning: an empirical study. Mach Learn 84 (1–2):109–36

Sibson R (1973) SLINK: an optimally efficient algorithm for the single-link cluster method. Comput J British Comput Soc 16 (1):30–34

Simon RM, Subramanian J, Li MC, Menezes S (2011) Using cross validation to evaluate the predictive accuracy of survival risk classifiers based on high dimensional data. Briefings in Bioinformatics 1–12

Sterne JA, May M, Costagliola D, De Wolf F, Phillips AN, Harris R, et al. (2009) Timing of initiation of antiretroviral therapy in AIDS-free HIV-1-infected patients: a collaborative analysis of 18 HIV cohort studies. The Lancet 373(9672):1352–63

Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J Roy Stat Soc Series B 36:111–147

Strecher VJ, Shiffman S, West R (2006) Moderators and mediators of a web-based computer-tailored smoking cessation program among nicotine patch users. Nicotine Tob Res 8(S.1):S95-S101

Strobl C (2008) Statistical Issues in Machine Learning – Towards Reliable Split Selection and Variable Importance Measures. Dissertation, Ludwig-maximilians-universität München

Su X, Tsai CL, Wang H, Nickerson DM, Li B (2009) Subgroup analysis via recursive partitioning. J Mach Learn Res 10:141–158

Su X, Zhou T, Yan X, Fan J, Yang S (2008) Interaction trees with censored survival data. Int J Biostat 4(1), 2

Sunkaria RK, Kumar V, Saxena SC, Singhal AM (2014) An ANN-based HRV classifier for cardiac health prognosis. Electron Health 7:315–330

Sutton RS, Barto AG (1998) Reinforcement Learning: An Introduction. MIT Press: Cambridge, MA

Tang F, Ishwaran H (2017) Random forest missing data algorithms. Stat Anal Data Min 00:1–14; DOI: 10.1002/sam.11348; arXiv preprint arXiv: 1701.05305

Therneau TM, Granbsch PM, Fleming TR (1990) Martingale-based residuals for survival models. Biometrika 77:147-160

Thomas M, Bornkamp B (2017) Comparing approaches to treatment effect estimation for subgroups in clinical trials. Stat Biopharm Res 9(2): 160-171

Tian X, Bi N, Taylor J (2016) MAGIC: a general, powerful and tractable method for selective inference. arXiv preprint arXiv: 1607.02630v

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Statist Soc Series B 58 (1):267-288

Tipping ME, Bishop CM (1999) Probabilistic principal component analysis. J R Stat Soc Series B 61(Part 3):611-622

Tukey JW (1977) Exploratory Data Analysis. Pearson

van Buuren S (2018) Flexible Imputation of Missing Data. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC

Vapnik V (1996) The Nature of Statistical Learning Theory. Springer, New York.

Vapnik V (2006) Estimation of Dependences Based on Empirical Data. Empirical Inference Science Afterword of 2006. Springer: New York

Varma S, Simon R (2006) Bias in error estimation when using crossvalidation for model selection. BMC Bioinformatics 7:91

Vsevolozhskaya OA, Greenwood MC, Powell SL, Zaykin DV (2015) Resampling-based multiple comparison procedure with application to point-wise testing with functional data. Environ Ecol Stat 22(1):45–59

Wager S, Hastie T, Efron B (2014) Intervals for Random Forests: The jackknife and the infinitesimal jackknife. J Mach Learn Res 15:1625-1651

Wang L, Rotnitzky A, Lin X, Millikan R, Thal, P (2012) Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer. J Am Stat Assoc 107:493–508

Wang H, Zhang X, Zou G (2009) Frequentist model averaging estimation: A review. Jrl Syst Sci & Complexity 22:732-748

Ward JH (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58 (301):236–244

Watkin CJCH (1989) Learning from Delayed Rewards. Ph.D. Thesis, Cambridge University

Watkin CJCH, Dayan P (1992) Q-Learning. Mach Learn 8:279-292

Werbos PJ (1975) Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, PhD Thesis Harvard University

Westfall PH, Troendle JF (2008) Multiple testing with minimal assumptions. Biometrics J 50 (5):745-55

Westfall PH, Young SS (1993) Resampling-based multiple testing: Examples and methods for p-value adjustment. Wiley: New York

White NJ, Contaifer Jr D, Martin EJ, Newton JC, Mohammed BM, Bostic JL, Brophy GM, Spiess BD, Pusateri AE, Ward KR, Brophy DF (2015) Early hemostatic responses to trauma identified with hierarchical clustering analysis. J Thromb Haemost 13:978–88

Witten IH, Frank E, Hall MA (2011) Data Mining. Practical Machine Learning Tools and Techniques. 3rd Edition. Morgan Kaufmann: Burlington, USA

Wu F, Laber EB, Lipkovich IA, Severus E (2015) Who will Benefit from Antidepressants in the Acute Treatment of Bipolar Depression? A Reanalysis of the STEP-BD Study by Sachs et al. 2007, Using Q-learning. Int J Bipolar Disord 3:7

Wu MJ, Mwangi B, Bauer IE, Passos IC, Sanches M, Zunta-Soares GB, Meyer TD, Hasan KM, Soares JC (2017) Identification and individualized prediction of clinical phenotypes in bipolar disorders using neurocognitive data, neuroimaging scans and machine learning. NeuroImage Part B, 145:254-264

Wu W, Bleecker E, Moore W, Busse WW, Castro M, Chung KF, Calhoun WJ, Erzurum S, Gaston B, Israel E, Curran-Everett D, Wenzel SE (2014) Unsupervised phenotyping of Severe Asthma Research Program participants using expanded lung data. J Allergy Clin Immunol 133 (5):1280-1288

Xu R (2013) Improvements to random forest methodology. PhD thesis, Iowa State University, Iowa, USA

Xu Y, Yu M, Zhao YQ, Li Q, Wang S, Shao J (2015) Regularized outcome weighted subgroup identification for differential treatment effects. Biometrics 71(3):645-53

Zhang B, Tsiatis AA, Laber EB, Davidian M (2013) Robust estimation of optimal dynamic treatment regimes for sequential treatment decisions. Biometrika 100(3):681–94

Zhang B, Tsiatis AA, Davidian M, Zhang M, Laber EB (2012) Estimating optimal treatment regimes from a classification perspective. Statistics 1:103–114

Zhang H (1995) Splitting criteria in survival trees. Seeber GUH, Francis BJ, Hatzinger R, Steckel-Berger G (eds) Statistical Modeling, Proceedings of the 10th International Workshop on Statistical Modeling, Springer, New York.305-314

Zhang Y, Laber EB, Tsiatis A, Davidian M (2015) Using decision lists to construct interpretable and parsimonious treatment regimes. Biometrics 71:895–904

Zhao Y, Zheng D, Rush AJ, Kosorok MR (2012) Estimating individualized treatment rules using outcome weighted learning. J Am Stat Assoc 107:1106–1118

Zhao YQ, Zeng D, Laber EB, Kosorok MR (2015) New statistical learning methods for estimating optimal dynamic treatment regimes. J Am Stat Assoc 110(510):583-598

Zheng B, Yoon SW, Lam SS (2014) Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms. Expert Syst Appl 41(4):1476-1482

Zou H (2006) The adaptive lasso and Its oracle properties. J Am Statist Assoc 101(476):1418-1429

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Statist Soc Series B 67(Part 2):301-320

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comp Graph Stat 15 (2):262–286