Discrete-Time Portfolio Optimization under Maximum Drawdown Constraint with Partial Information and Deep Learning Resolution

Carmine de Franco, Johann Nicolle, and Huyên Pham

In Memory of Mark H Davis

Abstract We study a discrete-time portfolio selection problem with partial information and maximum drawdown constraint. Drift uncertainty in the multidimensional framework is modeled by a prior probability distribution. In this Bayesian framework, we derive the dynamic programming equation using an appropriate change of measure, and obtain semi-explicit results in the Gaussian case. The latter case, with a CRRA utility function is completely solved numerically using recent deep learning techniques for stochastic optimal control problems. We emphasize the informative value of the learning strategy versus the non-learning one by providing empirical performance and sensitivity analysis with respect to the uncertainty of the drift. Furthermore, we show numerical evidence of the close relationship between the non-learning strategy and a no short-sale constrained Merton problem, by illustrating the convergence of the former towards the latter as the maximum drawdown constraint vanishes.

1 Introduction

This paper is devoted to the study of a constrained allocation problem in discrete time with partial information. We consider an investor who is willing to maximize the expected utility of her terminal wealth over a given investment horizon. The

Carmine de Franco
OSSIAM, 80, Avenue de la Grande Armée 75017 Paris, France,
e-mail: carmine.de-franco@ossiam.com

Johann Nicolle
LPSM-OSSIAM, 80, Avenue de la Grande Armée 75017 Paris, France,
e-mail: johann.nicolle@ossiam.com

Huyên Pham
Université de Paris, Bâtiment Sophie Germain, Case courrier 7012, 75205 Paris Cedex 13, France,
e-mail: pham@lpsm.paris

G. Yin, T. Zariphopoulou (eds.), *Stochastic Analysis, Filtering, and Stochastic Optimization*,
https://doi.org/10.1007/978-3-030-98519-6_5

risk-averse investor is looking for the optimal portfolio in financial assets under a maximum drawdown constraint. The maximum drawdown is a common metric in finance and represents the largest drop in the portfolio value. Our framework incorporates this constraint by setting a threshold representing the proportion of the current maximum of the wealth process that the investor is willing to keep.

The expected rate of assets' return (drift) is unknown, but information can be learnt by progressive observation of the financial asset prices. The uncertainty about the rate of return is modeled by a probability distribution, i.e., a prior belief on the drift. To take into account the information conveyed by the prices, this prior will be updated using a Bayesian learning approach.

An extensive literature exists on parameters uncertainty and especially on filtering and learning techniques in a partial information framework. To cite just a few, see [18], [20], [5], [16], [2], and [6]. Somme articles deal with risk constraints in a portfolio allocation framework. For instance, paper [19] tackles dynamic risk constraints and compares the continuous and discrete time trading while some papers especially focus on drawdown constraints, see in particular seminal paper [11] or [4]. More recently, the authors of [8] study infinite-horizon optimal consumption-investment problem in continuous-time, and in paper [3], authors use forecasts of the mean and covariance of financial returns from a multivariate hidden Markov model with time-varying parameters to build the optimal controls.

As it is not possible to solve analytically our constrained optimal allocation problem, we have applied a machine learning algorithm developed in [13] and [1]. This algorithm, called *Hybrid-Now*, is particularly suited for solving stochastic control problems in high dimension using deep neural networks.

Our main contributions to the literature is twofold: a detailed theoretical study of a discrete-time portfolio selection problem including both drift uncertainty and maximum drawdown constraint, and a numerical resolution using a deep learning approach for an application to a model of three risky assets, leading to a five-dimensional problem. We derive the dynamic programming equation (DPE), which is in general of infinite-dimensional nature, following the change of measure suggested in [9]. In the Gaussian case, the DPE is reduced to a finite-dimensional equation by exploiting the Kalman filter. In the particular case of constant relative risk aversion (CRRA) utility function, we reduce furthermore the dimensionality of the problem. Then, we solve numerically the problem in the Gaussian case with CRRA utility functions using the deep learning *Hybrid-Now* algorithm. Such numerical results allow us to provide a detailed analysis of the performance and allocations of both the learning and non-learning strategies benchmarked with a comparable equally-weighted strategy. Finally, we assess the performance of the learning compared to the non-learning strategy with respect to the sensitivity of the uncertainty of the drift. Additionally, we provide empirical evidence of convergence of the non-learning strategy to the solution of the classical Merton problem when the parameter controlling the maximum drawdown vanishes.

The paper is organized as follows: Section 2 sets up the financial market model and the associated optimization problem. Section 3 describes, in the general case,

the change of measure and the Bayesian filtering, the derivation of the dynamic programming equation and details some properties of the value function. Section 4 focuses on the Gaussian case. Finally, Section 5 presents the neural network techniques used, and shows the numerical results.

2 Problem setup

On a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a discrete filtration $(\mathcal{F}_k)_{k=0,\ldots,N}$ satisfying the usual conditions, we consider a financial market model with one riskless asset assumed normalized to one, and d risky assets. The price process $(S_k^i)_{k=0,\ldots,N}$ of asset $i \in [\![1, d]\!]$ is governed by the dynamics

$$S_{k+1}^i = S_k^i e^{R_{k+1}^i}, \quad k = 0, \ldots, N-1, \tag{1}$$

where $R_{k+1} = (R_{k+1}^1, \ldots, R_{k+1}^N)$ is the vector of the assets log-return between time k and $k+1$, and modeled as:

$$R_{k+1} = B + \epsilon_{k+1}. \tag{2}$$

The drift vector B is a d-dimensional random variable with probability distribution (prior) μ_0 of known mean $b_0 = \mathbb{E}[B]$ and finite second order moment. Note that the case of known drift B means that μ_0 is a Dirac distribution. The noise $\epsilon = (\epsilon_k)_k$ is a sequence of centered i.i.d. random vector variables with covariance matrix $\Gamma = \mathbb{E}[\epsilon_k \epsilon_k']$, and assumed to be independent of B. We also assume the fundamental assumption that the probability distribution ν of ϵ_k admits a strictly positive density function g on $\mathbb{R}^d$ with respect to the Lebesgue measure.

The price process S is observable, and notice by relation (1) that R can be deduced from S, and vice-versa. We will then denote by $\mathbb{F}^o = \left\{\mathcal{F}_k^o\right\}_{k=0,\ldots,N}$ the observation filtration generated by the process S (hence equivalently by R) augmented by the null sets of $\mathcal{F}$, with the convention that for $k = 0$, $\mathcal{F}_0^o$ is the trivial algebra.

An investment strategy is an $\mathbb{F}^o$-progressively measurable process $\alpha = (\alpha_k)_{k=0,\ldots,N-1}$, valued in $\mathbb{R}^d$, and representing the proportion of the current wealth invested in each of the d risky assets at each time $k = 0, \ldots, N-1$. Given an investment strategy α and an initial wealth $x_0 > 0$, the (self-financed) wealth process X^α evolves according to

$$\begin{cases} X_{k+1}^\alpha = X_k^\alpha \left(1 + \alpha_k' \left(e^{R_{k+1}} - \mathbb{1}_d\right)\right), & k = 0, \ldots, N-1, \\ X_0^\alpha = x_0. \end{cases} \tag{3}$$

where $e^{R_{k+1}}$ is the d-dimensional random variable with components $\left[e^{R_{k+1}}\right]_i = e^{R_{k+1}^i}$ for $i \in [\![1, d]\!]$, and $\mathbb{1}_d$ is the vector in $\mathbb{R}^d$ with all components equal to 1.

Let us introduce the process Z_k^α, as the maximum up to time k of the wealth process X^α, i.e.,

$$Z_k^\alpha := \max_{0\le \ell \le k} X_\ell^\alpha, \qquad k = 0, \ldots, N.$$

The maximum drawdown constraints the wealth X_k^α to remain above a fraction $q \in (0,1)$ of the current historical maximum Z_k^α. We then define the set of *admissible* investment strategies $\mathcal{A}_0^q$ as the set of investment strategies α such that

$$X_k^\alpha \ge q Z_k^\alpha, \quad \text{a.s.}, \quad k = 0, \ldots, N.$$

In this framework, the portfolio selection problem is formulated as

$$V_0 := \sup_{\alpha \in \mathcal{A}_0^q} \mathbb{E}\left[U\left(X_N^\alpha\right)\right], \tag{4}$$

where U is a utility function on $(0,\infty)$ satisfying the standard Inada conditions: continuously differentiable, strictly increasing, concave on $(0,\infty)$ with $U'(0) = \infty$ and $U'(\infty) = 0$.

3 Dynamic programming system

In this section, we show how Problem (4) can be characterized from dynamic programming in terms of a backward system of equations amenable for algorithms. In a first step, we will update the prior on the drift uncertainty, and take advantage of the newest available information by adopting a Bayesian filtering approach. This relies on a suitable change of probability measure.

3.1 Change of measure and Bayesian filtering

We start by introducing a change of measure under which R_1,..., R_N are mutually independent, identically distributed random variables and independent from the drift B, hence behaving like a noise. Following the methodology detailed in [9] we define the σ-algebras

$$\mathcal{G}_k^0 := \sigma(B, R_1, \ldots, R_k), \quad k = 0, \ldots, N,$$

and $\mathbb{G} = (\mathcal{G}_k)_k$ the corresponding complete filtration. We then define a new probability measure $\overline{\mathbb{P}}$ on $(\Omega, \bigvee_{k=1}^N \mathcal{G}_k)$ by

$$\left.\frac{\mathrm{d}\overline{\mathbb{P}}}{\mathrm{d}\mathbb{P}}\right|_{\mathcal{G}_k} := \Lambda_k, \quad k = 0, \ldots, N,$$

with

$$\Lambda_k := \prod_{\ell=1}^{k} \frac{g(R_\ell)}{g(\epsilon_\ell)}, \quad k = 1, \ldots, N, \quad \Lambda_0 = 1.$$

The existence of $\overline{\mathbb{P}}$ comes from the Kolmogorov's theorem since Λ_k is a strictly positive martingale with expectation equal to one. Indeed, for all $k = 1, ..., N$,

- $\Lambda_k > 0$ since the probability density function g is strictly positive
- Λ_k is $\mathcal{G}_k$-adapted,
- As $\epsilon_k \perp\!\!\!\perp \mathcal{G}_{k-1}$, we have

$$\begin{aligned}\mathbb{E}[\Lambda_k|\mathcal{G}_{k-1}] &= \Lambda_{k-1}\mathbb{E}\Big[\frac{g(B+\epsilon_k)}{g(\epsilon_k)}|\mathcal{G}_{k-1}\Big] \\ &= \Lambda_{k-1}\int_{\mathbb{R}^d}\frac{g(B+e)}{g(e)}g(e)de = \Lambda_{k-1}\int_{\mathbb{R}^d} g(z)dz = \Lambda_{k-1}.\end{aligned}$$

Proposition Under $\overline{\mathbb{P}}$, $(R_k)_{k=1,...,N}$, is a sequence of i.i.d. random variables, independent from B, having the same probability distribution ν as ϵ_k. □

Proof. See Appendix 6.1. □

Conversely, we recover the initial measure $\mathbb{P}$ under which $(\epsilon_k)_{k=1,...,N}$ is a sequence of independent and identically distributed random variables having probability density function g where $\epsilon_k = R_k - B$. Denoting by $\overline{\Lambda}_k$ the Radon-Nikodym derivative $\mathrm{d}\mathbb{P}/\mathrm{d}\overline{\mathbb{P}}$ restricted to the σ-algebra $\mathcal{G}_k$:

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\overline{\mathbb{P}}}\Big|_{\mathcal{G}_k} = \overline{\Lambda}_k,$$

we have

$$\overline{\Lambda}_k = \prod_{i=1}^{k}\frac{g(R_i - B)}{g(R_i)}.$$

It is clear that, under $\mathbb{P}$, the return and wealth processes have the form stated in equations (2) and (3). Moreover, from Bayes formula, the posterior distribution of the drift, i.e. the conditional law of B given the asset price observation, is

$$\mu_k(db) := \mathbb{P}[B \in db|\mathcal{F}_k^o] = \frac{\pi_k(db)}{\pi_k(\mathbb{R}^d)}, \quad k = 0, \ldots, N, \tag{5}$$

where π_k is the so-called unnormalized conditional law

$$\pi_k(db) := \overline{\mathbb{E}}[\overline{\Lambda}_k \mathbb{1}_{\{B\in db\}}|\mathcal{F}_k^o], \quad k = 0, \ldots, N.$$

We then have the key recurrence linear relation on the unnormalized conditional law.

Proposition We have the recursive linear relation

$$\pi_\ell = \bar{g}(R_\ell - \cdot)\pi_{\ell-1}, \quad \ell = 1, \dots, N, \tag{6}$$

with initial condition $\pi_0 = \mu_0$, where

$$\bar{g}(R_\ell - b) = \frac{g(R_\ell - b)}{g(R_\ell)}, \quad b \in \mathbb{R}^d,$$

and we recall that g is the probability density function of the identically distributed ϵ_k under $\mathbb{P}$. □

Proof. See Appendix 6.2 . □

3.2 The static set of admissible controls

In this subsection, we derive some useful characteristics of the space of controls which will turn out to be crucial in the derivation of the dynamic programming system.

Given time $k \in [\![0, N]\!]$, a current wealth $x = X_k^\alpha > 0$, and current maximum wealth $z = Z_k^\alpha \geq x$ that satisfies the drawdown constraint $qz \leq x$ at time k for an admissible investment strategy $\alpha \in \mathcal{A}_0^q$, we denote by $A_k^q(x, z) \subset \mathbb{R}^d$ the set of static controls $a = \alpha_k$ such that the drawdown constraint is satisfied at next time $k + 1$, i.e. $X_{k+1}^\alpha \geq qZ_{k+1}^\alpha$. From the relation (3), and noting that $Z_{k+1}^\alpha = \max[Z_k^\alpha, X_{k+1}^\alpha]$, this yields

$$A_k^q(x, z) = \left\{ a \in \mathbb{R}^d : 1 + a'(e^{R_{k+1}} - \mathbb{1}_d) \;\geq\; q \max\left[\frac{z}{x}, 1 + a'(e^{R_{k+1}} - \mathbb{1}_d)\right] \text{a.s.} \right\}. \tag{7}$$

Recalling from Proposition 1, that the random variables $R_1, ..., R_N$ are i.i.d. under $\overline{\mathbb{P}}$, we notice that the set $A_k^q(x, z)$ does not depend on the current time k, and we will drop the subscript k in the sequel, and simply denote by $A^q(x, z)$.

Remembering that the support of ν, the probability distribution of ϵ_k, is $\mathbb{R}^d$, the following lemma characterizes more precisely the set $A^q(x, z)$.

Lemma 1 *For any $(x, z) \in \mathcal{S}^q := \{(x, z) \in (0, \infty)^2 : qz \leq x \leq z\}$, we have*

$$A^q(x, z) = \left\{ a \in \mathbb{R}_+^d : |a|_1 \leq 1 - q\frac{z}{x} \right\},$$

where $|a|_1 = \sum_{i=1}^d |a_i|$ for $a = (a_1, \dots, a_d) \in \mathbb{R}_+^d$.

Proof. See Appendix 6.3. □

Let us prove some properties on the admissible set $A^q(x, z)$.

Lemma 2 *For any $(x, z) \in \mathcal{S}^q$, the set $A^q(x, z)$ satisfies the following properties:*

1. It is decreasing in q: $\forall q_1 \leq q_2$, $A^{q_2}(x, z) \subseteq A^{q_1}(x, z)$,

2. It is continuous in q,
3. It is increasing in x: $\forall x_1 \le x_2$, $A^q(x_1, z) \subseteq A^q(x_2, z)$,
4. It is a convex set,
5. It is homogeneous: $a \in A^q(x, z) \Leftrightarrow a \in A^q(\lambda x, \lambda z)$, *for any* $\lambda > 0$.

Proof. See Appendix 6.4. □

3.3 Derivation of the dynamic programming equation

The change of probability detailed in Subsection 3.1 allows us to turn the initial partial information Problem (4) into a full observation problem as

$$\begin{aligned} V_0 := \sup_{\alpha \in \mathcal{A}_0^q} \mathbb{E}[U(X_N^\alpha)] &= \sup_{\alpha \in \mathcal{A}_0^q} \overline{\mathbb{E}}[\overline{\Lambda}_N U(X_N^\alpha)] \\ &= \sup_{\alpha \in \mathcal{A}_0^q} \overline{\mathbb{E}}\Big[\overline{\mathbb{E}}[\overline{\Lambda}_N U(X_N^\alpha) | \mathcal{F}_N^o]\Big] \\ &= \sup_{\alpha \in \mathcal{A}_0^q} \overline{\mathbb{E}}\Big[U(X_N^\alpha)\pi_N(\mathbb{R}^d)\Big], \end{aligned} \tag{8}$$

from Bayes formula, the law of conditional expectations, and the definition of the unnormalized filter π_N valued in $\mathcal{M}_+$, the set of nonnegative measures on $\mathbb{R}^d$. In view of Equation (3), Proposition 1, and Proposition 2, we then introduce the dynamic value function associated to Problem (8) as

$$v_k(x, z, \mu) = \sup_{\alpha \in \mathcal{A}_k^q(x,z)} J_k(x, z, \mu, \alpha), \quad k \in [\![0, N]\!], \ (x, z) \in \mathcal{S}^q, \ \mu \in \mathcal{M}_+,$$

with

$$J_k(x, z, \mu, \alpha) = \overline{\mathbb{E}}\Big[U(X_N^{k,x,\alpha})\pi_N^{k,\mu}(\mathbb{R}^d)\Big],$$

where $X^{k,x,\alpha}$ is the solution to Equation (3) on $[\![k, N]\!]$, starting at $X_k^{k,x,\alpha} = x$ at time k, controlled by $\alpha \in \mathcal{A}_k^q(x, z)$, and $(\pi_\ell^{k,\mu})_{\ell=k,\dots,N}$ is the solution to (6) on $\mathcal{M}_+$, starting from $\pi_k^{k,\mu} = \mu$, so that $V_0 = v_0(x_0, x_0, \mu_0)$. Here, $\mathcal{A}_k^q(x, z)$ is the set of admissible investment strategies embedding the drawdown constraint: $X_\ell^{k,x,\alpha} \ge qZ_\ell^{k,x,z,\alpha}$, $\ell = k, \dots, N$, where the maximum wealth process $Z^{k,x,z,\alpha}$ follows the dynamics: $Z_{\ell+1}^{k,x,z,\alpha} = \max[Z_\ell^{k,x,z,\alpha}, X_{\ell+1}^{k,x,\alpha}]$, $\ell = k, \dots, N-1$, starting from $Z_k^{k,x,z,\alpha} = z$ at time k. The dependence of the value function upon the unnormalized filter μ means that the probability distribution on the drift is updated at each time step from Bayesian learning by observing assets price.

The dynamic programming equation associated to (8) is then written in backward induction as

$$\begin{cases} v_N(x,z,\mu) = U(x)\mu(\mathbb{R}^d), \\ v_k(x,z,\mu) = \sup\limits_{\alpha\in\mathcal{A}_k^q(x,z)} \overline{\mathbb{E}}\Big[v_{k+1}(X_{k+1}^{k,x,\alpha}, Z_{k+1}^{k,x,z,\alpha}, \pi_{k+1}^{k,\mu})\Big], \quad k = 0,\dots,N-1. \end{cases}$$

Recalling Proposition 2 and Lemma 1, this dynamic programming system is written more explicitly as

$$\begin{cases} v_N(x,z,\mu) = U(x)\mu(\mathbb{R}^d), \quad (x,z)\in\mathcal{S}^q,\ \mu\in\mathcal{M}_+, \\ v_k(x,z,\mu) = \sup\limits_{a\in A^q(x,z)} \overline{\mathbb{E}}\Big[v_{k+1}\Big(x(1+a'(e^{R_{k+1}}-\mathbb{1}_d)), \\ \qquad \max\big[z, x(1+a'(e^{R_{k+1}}-\mathbb{1}_d))\big], \bar{g}(R_{k+1}-\cdot)\mu\Big)\Big], \end{cases} \tag{9}$$

for $k = 0,\dots,N-1$. Notice from Proposition 1 that the expectation in the above formula is only taken with respect to the noise R_{k+1}, which is distributed under $\overline{\mathbb{P}}$ according to the probability distribution ν with density g on $\mathbb{R}^d$.

3.4 Special case: CRRA utility function

In the case where the utility function is of CRRA (Constant Relative Risk Aversion) type, i.e.,

$$U(x) = \frac{x^p}{p}, \quad x > 0, \ \text{ for some } 0 < p < 1, \tag{10}$$

one can reduce the dimensionality of the problem. For this purpose, we introduce the process $\rho = (\rho_k)_k$ defined as the ratio of the wealth over its maximum up to current as:

$$\rho_k^\alpha = \frac{X_k^\alpha}{Z_k^\alpha}, \quad k = 0,\dots,N.$$

This ratio process lies in the interval $[q,1]$ due to the maximum drawdown constraint. Moreover, recalling (3), and observing that $Z_{k+1}^\alpha = \max[Z_k^\alpha, X_{k+1}^\alpha]$, together with the fact that $\frac{1}{\max[z,x]} = \min[\frac{1}{z}, \frac{1}{x}]$, we notice that the ratio process ρ can be written in inductive form as

$$\rho_{k+1}^\alpha = \min\Big[1, \rho_k^\alpha(1+\alpha_k'(e^{R_{k+1}}-\mathbb{1}_d))\Big], \quad k = 0,\dots,N-1.$$

The following result states that the value function inherits the homogeneity property of the utility function.

Lemma 3 *For a utility function U as in* (10)*, we have for all $(x,z)\in\mathcal{S}^q$, $\mu\in\mathcal{M}_+$, $k\in[\![0,N]\!]$,*

$$v_k(\lambda x, \lambda z, \mu) = \lambda^p v_k(x,z,\mu), \quad \lambda > 0.$$

Proof. See Appendix 6.5. □

In view of the above Lemma, we consider the sequence of functions w_k, $k \in [\![0, N]\!]$, defined by

$$w_k(r, \mu) = v_k(r, 1, \mu), \quad r \in [q, 1], \ \mu \in \mathcal{M}_+,$$

so that $v_k(x, z, \mu) = z^p w_k(\frac{x}{z}, \mu)$, and we call w_k the reduced value function. From the dynamic programming system satisfied by v_k, we immediately obtain the backward system for $(w_k)_k$ as

$$\begin{cases} w_N(r, \mu) = \frac{r^p}{p} \mu(\mathbb{R}^d), \quad r \in [q, 1], \ \mu \in \mathcal{M}_+, \\ w_k(r, \mu) = \sup\limits_{a \in A^q(r)} \overline{\mathbb{E}}\Big[w_{k+1}(\min\left[1, r(1 + a'(e^{R_{k+1}} - \mathbb{1}_d))\right], \bar{g}(R_{k+1} - \cdot)\mu)\Big], \end{cases} \tag{11}$$

for $k = 0, \ldots, N-1$, where

$$A^q(r) = \Big\{a \in \mathbb{R}^d_+ : a'\mathbb{1}_d \leq 1 - \frac{q}{r}\Big\}.$$

We end this section by stating some properties on the reduced value function.

Lemma 4 *For any $k \in [\![0, N]\!]$, the reduced value function w_k is nondecreasing and concave in $r \in [q, 1]$.*

Proof. See proof in Appendix 6.6. □

4 The Gaussian case

We consider in this section the Gaussian framework where the noise and the prior belief on the drift are modeled according to a Gaussian distribution. In this special case, the Bayesian filtering is simplified into the Kalman filtering, and the dynamic programming system is reduced to a finite-dimensional problem that will be solved numerically. It is convenient to deal directly with the posterior distribution of the drift, i.e. the conditional law of the drift B given the assets price observation, also called normalized filter. From (5) and Proposition 2, it is given by the inductive relation

$$\mu_k(db) = \frac{g(R_k - b)\mu_{k-1}(db)}{\int_{\mathbb{R}^d} g(R_k - b)\mu_{k-1}(db)}, \quad k = 1, \ldots, N. \tag{12}$$

4.1 Bayesian Kalman filtering

We assume that the probability law ν of the noise ϵ_k is Gaussian: $\mathcal{N}(0, \Gamma)$, and so with density function

$$g(r) = (2\pi)^{-\frac{d}{2}} |\Gamma|^{-\frac{1}{2}} e^{-\frac{1}{2} r' \Gamma^{-1} r}, \quad r \in \mathbb{R}^d. \tag{13}$$

Assuming also that the prior distribution μ_0 on the drift B is Gaussian with mean b_0, and invertible covariance matrix Σ_0, we deduce by induction from (12) that the posterior distribution μ_k is also Gaussian: $\mu_k \sim \mathcal{N}(\hat{B}_k, \Sigma_k)$, where $\hat{B}_k = \mathbb{E}[B|\mathcal{F}_k^o]$ and Σ_k satisfy the well-known inductive relations:

$$\hat{B}_{k+1} = \hat{B}_k + K_{k+1}(R_{k+1} - \hat{B}_k), \quad k = 0, \ldots, N-1 \tag{14}$$

$$\Sigma_{k+1} = \Sigma_k - \Sigma_k(\Sigma_k + \Gamma)^{-1}\Sigma_k, \tag{15}$$

where K_{k+1} is the so-called Kalman gain given by

$$K_{k+1} = \Sigma_k(\Sigma_k + \Gamma)^{-1}, \quad k = 0, \ldots, N-1. \tag{16}$$

We have the initialization $\hat{B}_0 = b_0$, and the notation for Σ_k is coherent at time $k = 0$ as it corresponds to the covariance matrice of B. While the Bayesian estimation $\hat{B}_k$ of B is updated from the current observation of the log-return R_k, notice that Σ_k (as well as K_k) is deterministic, and is then equal to the covariance matrix of the error between B and its Bayesian estimation, i.e. $\Sigma_k = \mathbb{E}[(B - \hat{B}_k)(B - \hat{B}_k)']$. Actually, we can explicitly compute Σ_k by noting from Equation (12) with g as in (13) and $\mu_0 \sim \mathcal{N}(b_0, \Sigma_0)$ that

$$\mu_k \sim \frac{e^{-\frac{1}{2}\left(b - \left(\Sigma_0^{-1} + \Gamma^{-1}k\right)^{-1}\left(\Gamma^{-1}\sum_{j=1}^k R_j + \Sigma_0^{-1} b_0\right)\right)\left(\Sigma_0^{-1} + \Gamma^{-1}k\right)\left(b - \left(\Sigma_0^{-1} + \Gamma^{-1}k\right)^{-1}\left(\Gamma^{-1}\sum_{j=1}^k R_j + \Sigma_0^{-1} b_0\right)\right)}}{(2\pi)^{\frac{d}{2}} |(\Sigma_0^{-1} + \Gamma^{-1}k)^{-1}|^{\frac{1}{2}}}.$$

By identification, we then get

$$\Sigma_k = (\Sigma_0^{-1} + \Gamma^{-1}k)^{-1} = \Sigma_0(\Gamma + \Sigma_0 k)^{-1}\Gamma. \tag{17}$$

Moreover, the innovation process $(\tilde{\epsilon}_k)_k$, defined as

$$\tilde{\epsilon}_{k+1} = R_{k+1} - \mathbb{E}[R_{k+1}|\mathcal{F}_k^o] = R_{k+1} - \hat{B}_k, \quad k = 0, \ldots, N-1, \tag{18}$$

is a $\mathbb{F}^o$-adapted Gaussian process. Each $\tilde{\epsilon}_{k+1}$ is independent of $\mathcal{F}_k^0$ (hence $\tilde{\epsilon}_k$, $k = 1, \ldots, N$ are mutually independent), and is a centered Gaussian vector with covariance matrix:

$$\tilde{\epsilon}_{k+1} \sim \mathcal{N}(0, \tilde{\Gamma}_{k+1}), \quad \text{with } \tilde{\Gamma}_{k+1} = \Sigma_k + \Gamma.$$

We refer to [15] and [14] for these classical properties about the Kalman filtering and the innovation process.

Remark 1 From (14), and (18), we see that the Bayesian estimator $\hat{B}_k$ follows the dynamics

$$\begin{cases} \hat{B}_{k+1} = \hat{B}_k + K_{k+1}\tilde{\epsilon}_{k+1}, & k = 0, \ldots, N-1 \\ \hat{B}_0 = b_0, \end{cases}$$

which implies in particular that $\hat{B}_k$ has a Gaussian distribution with mean b_0, and covariance matrix satisfying

$$\mathrm{Var}(\hat{B}_{k+1}) = \mathrm{Var}(\hat{B}_k) + K_{k+1}(\Sigma_k + \Gamma)K'_{k+1} = \mathrm{Var}(\hat{B}_k) + \Sigma_k(\Sigma_k + \Gamma)^{-1}\Sigma_k.$$

Recalling the inductive relation (15) on Σ_k, this shows that $\mathrm{Var}(\hat{B}_k) = \Sigma_0 - \Sigma_k$. Note that, from Equation (15), $(\Sigma_k)_k$ is a decreasing sequence which ensures that $\mathrm{Var}(\hat{B}_k)$ is positive semi-definite and is nondecreasing with time k. ♢

4.2 Finite-dimensional dynamic programming equation

From (18), we see that our initial portfolio selection Problem (4) can be reformulated as a full observation problem with state dynamics given by

$$\begin{cases} X^{\alpha}_{k+1} = X^{\alpha}_k\big(1 + \alpha'_k(e^{\hat{B}_k + \tilde{\epsilon}_{k+1}} - \mathbb{1}_d)\big), \\ \hat{B}_{k+1} = \hat{B}_k + K_{k+1}\tilde{\epsilon}_{k+1}, \quad k = 0, \dots, N-1. \end{cases} \tag{19}$$

We then define the value function on $[\![0, N]\!] \times \mathcal{S}^q \times \mathbb{R}^d$ by

$$\tilde{v}_k(x, z, b) = \sup_{\alpha \in \mathcal{A}^q_k(x,z)} \mathbb{E}\big[U(X^{k,x,b,\alpha}_N)\big], \quad k \in [\![0, N]\!],\ (x, z) \in \mathcal{S}^q,\ b \in \mathbb{R}^d,$$

where the pair $(X^{k,x,b,\alpha}, \hat{B}^{k,b})$ is the process solution to (19) on $[\![k, N]\!]$, starting from (x, b) at time k, so that $V_0 = \tilde{v}_0(x_0, x_0, b_0)$. The associated dynamic programming system satisfied by the sequence $(\tilde{v}_k)_k$ is

$$\begin{cases} \tilde{v}_N(x, z, b) = U(x), \quad (x, z) \in \mathcal{S}^q,\ b \in \mathbb{R}^d, \\ \tilde{v}_k(x, z, b) = \sup_{a \in A^q(x,z)} \mathbb{E}\Big[\tilde{v}_{k+1}\Big(x(1 + a'(e^{b+\tilde{\epsilon}_{k+1}} - \mathbb{1}_d)), \\ \qquad\qquad \max\big[z, x(1 + a'(e^{b+\tilde{\epsilon}_{k+1}} - \mathbb{1}_d))\big], b + K_{k+1}\tilde{\epsilon}_{k+1}\Big)\Big], \end{cases}$$

for $k = 0, \dots, N-1$. Notice that in the above formula, the expectation is taken with respect to the innovation vector $\tilde{\epsilon}_{k+1}$, which is distributed according to $\mathcal{N}(0, \tilde{\Gamma}_{k+1})$.

Moreover, in the case of CRRA utility functions $U(x) = x^p/p$, and similarly as in Section 3.4, we have the dimension reduction with

$$\tilde{w}_k(r, b) = \tilde{v}_k(r, 1, b), \quad r \in [q, 1],\ b \in \mathbb{R}^d,$$

so that $\tilde{v}_k(x, z, b) = z^p \tilde{w}_k(\frac{x}{z}, b)$, and this reduced value function satisfies the backward system on $[q, 1] \times \mathbb{R}^d$:

$$\begin{cases} \tilde{w}_N(r, b) = \frac{r^p}{p}, \quad r \in [q, 1],\ b \in \mathbb{R}^d, \\ \tilde{w}_k(r, b) = \sup_{a \in A^q(r)} \mathbb{E}\Big[\tilde{w}_{k+1}\Big(\min\big[1, r(1 + a'(e^{b+\tilde{\epsilon}_{k+1}} - \mathbb{1}_d))\big], b + K_{k+1}\tilde{\epsilon}_{k+1}\Big)\Big], \end{cases}$$

for $k = 0, \ldots, N-1$.

Remark 2 (No short-sale constrained Merton problem) In the limiting case when $q = 0$, the drawdown constraint is reduced to a non-negativity constraint on the wealth process, and by Lemma 1, this means a no-short selling and no borrowing constraint on the portfolio strategies. When the drift B is also known, equal to b_0, and for a CRRA utility function, let us then consider the corresponding constrained Merton problem with value function denoted by v_k^M, $k = 0, \ldots, N$, which satisfies the standard backward recursion from dynamic programming:

$$\begin{cases} v_N^M(x) = \dfrac{x^p}{p}, \quad x > 0, \\ v_k^M(x) = \sup\limits_{\substack{a' \mathbb{1}_d \le 1 \\ a \in [0,1]^d}} \mathbb{E}\Big[v_{k+1}^M \big(x(1 + a'(e^{b_0 + \epsilon_{k+1}} - \mathbb{1}_d)) \big) \Big], \quad k = 0, \ldots, N-1. \end{cases} \tag{20}$$

Searching for a solution of the form $v_k^M(x) = K_k x^p / p$, with $K_k \ge 0$ for all $k \in [\![0, N]\!]$, we see that the sequence $(K_k)_k$ satisfies the recursive relation:

$$K_k = S K_{k+1}, \quad k = 0, \ldots, N-1,$$

starting from $K_N = 1$, where

$$S := \sup_{\substack{a' \mathbb{1}_d \le 1 \\ a \in [0,1]^d}} \mathbb{E}\Big[\Big(1 + a'(e^{b_0 + \epsilon_1} - \mathbb{1}_d) \Big)^p \Big],$$

by recalling that $\epsilon_1, \ldots, \epsilon_N$ are i.i.d. random variables. It follows that the value function of the constrained Merton problem, unique solution to the dynamic programming system (20), is equal to

$$v_k^M(x) = S^{N-k} \frac{x^p}{p}, \quad k = 0, \ldots, N,$$

and the constant optimal control is given by

$$a_k^M = \operatorname*{argmax}_{\substack{a' \mathbb{1} \le 1 \\ a \in [0,1]^d}} \mathbb{E}\left[\left(1 + a' \left(e^{R_1} - \mathbb{1}_d \right) \right)^p \right] \quad k = 0, \ldots, N-1.$$

$\Diamond$

5 Deep learning numerical resolution

In this section, we exhibit numerical results to promote the benefits of learning from new information. To this end, we compare the learning strategy (Learning) to the non-learning one (Non-Learning) in the case of the CRRA utility function and

the Gaussian distribution for the noise. The prior probability distribution of B is the Gaussian distribution $\mathcal{N}(b_0, \Sigma_0)$ for Learning while it is the Dirac distribution concentrated at b_0 for Non-Learning.

We use deep neural network techniques to compute numerically the optimal solutions for both Learning and Non-Learning. To broaden the analysis, in addition to the learning and non-learning strategies, we have computed an "admissible" equally weighted (EW) strategy. More precisely, this EW strategy will share the quantity $X_k - qZ_k$ equally among the d assets. Eventually, we show numerical evidence that the Non-Learning converges to the optimal strategy of the constrained Merton problem, when the loss aversion parameter q vanishes.

5.1 Architectures of the deep neural networks

Neural networks (NN) are able to approximate nonlinear continuous functions, typically the value function and controls of our problem. The principle is to use a large amount of data to train the NN so that it progressively comes close to the target function. It is an iterative process in which the NN is tuned on a training set, then tested on a validation set to avoid over-fitting. For more details, see for instance [12] and [10].

The algorithm we use, relies on two dense neural networks: the first one is dedicated to the controls (A_{NN}) and the second one to the value function (VF_{NN}). Each NN is composed of four layers: an input layer, two hidden layers and an output layer:

(i) The input layer is $d+1$-dimensional since it embeds the conditional expectations of each of the d assets and the ratio of the current wealth to the current historical maximum ρ.

(ii) The two hidden layers give the NN the flexibility to adjust its weights and biases to approximate the solution. From numerical experiments, we see that, given the complexity of our problem, a first hidden layer with $d + 20$ neurons and a second one with $d + 10$ are a good compromise between speed and accuracy.

(iii) The output layer is d-dimensional for the controls, one for each asset representing the weight of the instrument, and is one-dimensional for the value function. See Figures 1 and 2 for an overview of the NN architectures in the case of $d = 3$ assets.

Parameter	$\mathbf{A_{NN}}$	$\mathbf{VF_{NN}}$
Initializer	uniform(0, 1)	He_uniform
Regularizers	L2 norm	L2 norm
Activation functions	Elu and Sigmoid for output layer	Elu and Sigmoid for output layer
Optimizer	Adam	Adam
Learning rates: step N-1	5e-3	1e-3
steps k = 0,...,N-2	6.25e-4	5e-4
Scale	1e-3	1e-3
Number of elements in a training batch	3e2	3e2
Number of training batches	1e2	1e2
Size of the validation batches	1e3	1e3
Penalty constant	3e-1	NA
Number of epochs: step N-1	2e3	2e3
steps k = 0,...,N-2	5e2	5e2
Size of the training set: step N-1	6e7	6e7
steps k = 0,...,N-2	1.5e7	1.5e7
Size of the validation set: step N-1	2e6	2e6
steps k = 0,..., N-2	5e5	5e5

Table 1 Parameters for the neural networks of the controls $\mathbf{A_{NN}}$ and the value function $\mathbf{VF_{NN}}$.

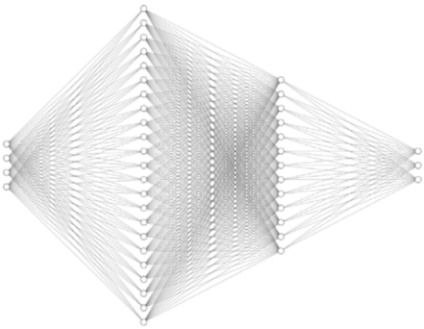

Fig. 1 A_{NN} architecture with $d = 3$ assets

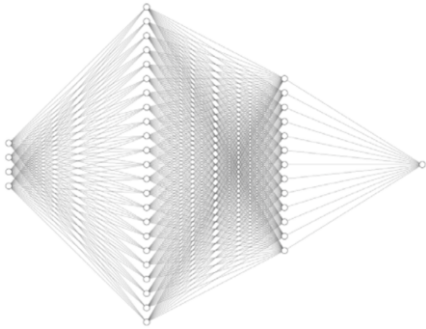

Fig. 2 VF_{NN} architecture with $d = 3$ assets

We follow the indications in [10] to setup and define the values of the various inputs of the neural networks which are listed in Table 1.

To train the NN, we simulate the input data. For the conditional expectation $\hat{B}_k$, we use its time-dependent Gaussian distribution (see Remark 1): $\hat{B}_k \sim \mathcal{N}(b_0, \Sigma_0 - \Sigma_k)$, with Σ_k as in Equation (17). On the other hand, the training of ρ is drawn from the uniform distribution between q and 1, the interval where it lies according to the maximum drawdown constraint.

5.2 *Hybrid-Now* algorithm

We use the *Hybrid-Now* algorithm developped in [1] in order to solve numerically our problem. This algorithm combines optimal policy estimation by neural networks

and dynamic programming principle which suits the approach we have developped in Section 4.

With the same notations as in Algorithm 1 detailed in the next insert, at time k, the algorithm computes the proxy of the optimal control $\hat{\alpha}_k$ with A_{NN}, using the known function $\hat{V}_{k+1}$ calculated the step before, and uses V_{NN} to obtain a proxy of the value function $\hat{V}_k$. Starting from the known function $\hat{V}_N := U$ at terminal time N, the algorithm computes sequentially $\hat{\alpha}_k$ and $\hat{V}_k$ with backward iteration until time 0. This way, the algorithm loops to build the optimal controls and the value function pointwise and gives as output the optimal strategy, namely the optimal controls from 0 to $N-1$ and the value function at each of the N time steps.

The maximum drawdown constraint is a time-dependent constraint on the maximal proportion of wealth to invest (recall Lemma 1). In practice, it is a constraint on the sum of weights of each asset or equivalently on the output of A_{NN}. For that reason, we have implemented an appropriate penalty function that will reject undesirable values:

$$G_{Penalty}(A, r) = K \max\left(|A|_1 \leq 1 - \frac{q}{r}, 0\right), \quad A \in [0,1]^d, \quad r \in [q, 1].$$

This penalty function ensures that the strategy respects the maximum drawdown constraint at each time step, when the parameter K is chosen sufficiently large.

Algorithm 1: *Hybrid-Now*

Input: the training distributions μ_{Unif} and μ^k_{Gauss};

▷ $\mu_{Unif} = \mathcal{U}(q, 1)$

▷ $\mu^k_{Gauss} = \mathcal{N}(b_0, \Sigma_0 - \Sigma_k)$

Output:

- estimate of the optimal strategy $(\hat{a}_k)_{k=0}^{N-1}$;
- estimate of the value function $\left(\hat{V}_k\right)_{k=0}^{N-1}$;

Set $\hat{V}_N = U$;

for $k = N-1, \ldots, 0$ **do**

Compute:

$$\hat{\beta}_k \in \underset{\beta \in \mathbb{R}^{2d^2+56d+283}}{\operatorname{argmin}} \mathbb{E}\left[G_{Penalty}(A_{NN}(\rho_k, \hat{B}_k; \beta), \rho_k) - \hat{V}_{k+1}\left(\rho^{\beta}_{k+1}, \hat{B}_{k+1}\right)\right]$$

where $\rho_k \sim \mu_{Unif}$, $\hat{B}_k \sim \mu^k_{Gauss}$,
$\hat{B}_{k+1} = \tilde{H}_k(\hat{B}_k, \tilde{\epsilon}_{k+1})$ and $\rho^{\beta}_{k+1} = F\left(\rho_k, \hat{B}_k, A_{NN}\left(\rho_k, \hat{B}_k; \beta\right), \tilde{\epsilon}_{k+1}\right)$;

▷ $F(\rho, b, a, \epsilon) = \min\left(1, \rho\left(1 + \sum_{i=1}^{d} a^i\left(e^{b^i+\epsilon^i} - 1\right)\right)\right)$

▷ $\tilde{H}_k(b, \epsilon) = b + \Sigma_0(\Gamma + \Sigma_0 k)^{-1}\epsilon$

Set $\hat{a}_k = A_{NN}\left(.; \hat{\beta}_k\right)$;

▷ $\hat{a}_k$ is the estimate of the optimal control at time k.

Compute:

$$\hat{\theta}_k \in \underset{\theta \in \mathbb{R}^{2d^2+54d+261}}{\operatorname{argmin}} \mathbb{E}\left[\left(\hat{V}_{k+1}\left(\rho^{\hat{\beta}_k}_{k+1}, \hat{B}_{k+1}\right) - VF_{NN}\left(\rho_k, \hat{B}_k; \theta\right)\right)^2\right]$$

Set $\hat{V}_k = VF_{NN}\left(., \hat{\theta}_k\right)$;

▷ $\hat{V}_k$ is the estimate of the value function at time k.

A major argument behind the choice of this algorithm is that, it is particularly relevant for problems in which the neural network approximation of the controls and value function at time k, are close to the ones at time $k+1$. This is what we expect in our case. We can then take a small learning rate for the Adam optimizer which enforces the stability of the parameters' update during the gradient-descent based learning procedure.

5.3 Numerical results

In this section, we explain the setup of the simulation and exhibit the main results. We have used Tensorflow 2 and deep learning techniques for Python developped in [10]. We consider $d = 3$ risky assets and a riskless asset whose return is assumed

Parameter	Value
Number of risky assets d	3
Investment horizon in years T	1
Number of steps/rebalancing N	24
Number of simulations/trajectories $\tilde{N}$	1000
Degree of the CRRA utility function p	0.8
Parameter of risk aversion q	0.7
Annualized expectation of the drift B	$\begin{bmatrix} 0.05 & 0.025 & 0.12 \end{bmatrix}$
Annualized covariance matrix of the drift B	$\begin{bmatrix} 0.2^2 & 0 & 0 \\ 0 & 0.15^2 & 0 \\ 0 & 0 & 0.1^2 \end{bmatrix}$
Annualized volatility of ϵ	$\begin{bmatrix} 0.08 & 0.04 & 0.22 \end{bmatrix}$
Correlation matrix of ϵ	$\begin{bmatrix} 1 & -0.1 & 0.2 \\ -0.1 & 1 & -0.25 \\ 0.2 & -0.25 & 1 \end{bmatrix}$
Annualized covariance matrix of the noise ϵ	$\begin{bmatrix} 0.0064 & -0.00032 & 0.00352 \\ -0.00032 & 0.0016 & -0.0022 \\ 0.00352 & -0.0022 & 0.0484 \end{bmatrix}$

Table 2 Values of the parameters used in the simulation.

to be 0, on a 1-year investment horizon for the sake of simplicity. We consider 24 portfolio rebalancing during the 1-year period, i.e., one every two weeks. This means that we have $N = 24$ steps in the training of our neural networks. The parameters used in the simulation are detailed in Table 2.

First, we show the numerical results for the learning and the non-learning strategies by presenting a performance and an allocation analysis in Subsection 5.3.1. Then, we add the admissible constrained EW to the two previous ones and use this neutral strategy as a benchmark in Subsection 5.3.2. Ultimately, in Subsection 5.3.3, we illustrate numerically the convergence of the non-learning strategy to the constrained Merton problem when the loss aversion parameter q vanishes.

5.3.1 Learning and non-learning strategies

We simulate $\tilde{N} = 1000$ trajectories for each strategy and exhibit the performance results with an initial wealth $x_0 = 1$. Figures 3 illustrates the average historical level of the learning and non-learning strategies with a 95% confidence interval. Learning outperforms significantly Non-Learning with a narrower confidence interval revealing that less uncertainty surrounds Learning performance, thus yielding less risk.

An interesting phenomenon, visible in Fig. 3, is the nearly flat curve for Learning between time 0 and time 1. Indeed, whereas Non-Learning starts investing immediately, Learning adopts a safer approach and needs a first time step before allocating a significant proportion of wealth. Given the level of uncertainty surrounding b_0, this first step allows Learning to fine-tune its allocation by updating the prior belief with

the first return available at time 1. On the contrary, Non-Learning, which cannot update its prior, starts investing at time 0.

Fig. 4 shows the ratio of Learning over Non-Learning. A ratio greater than one means that Learning outperforms Non-Learning and underperforms when less than one. It shows the significant outperformance of Learning over Non-Learning except during the first period where Learning was not significantly invested and Non-Learning had a positive return. Moreover, this graph reveals the typical increasing concave curve of the value of information described in [17], in the context of investment decisions and costs of data analytics, and in [6] in the resolution of the Markowitz portfolio selection problem using a Bayesian learning approach.

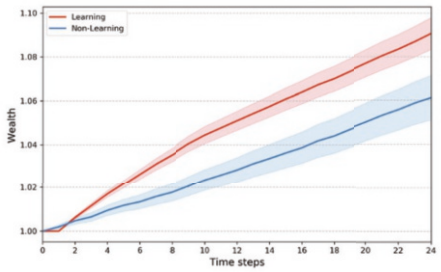

Fig. 3 Historical Learning and Non-Learning levels with a 95% confidence interval.

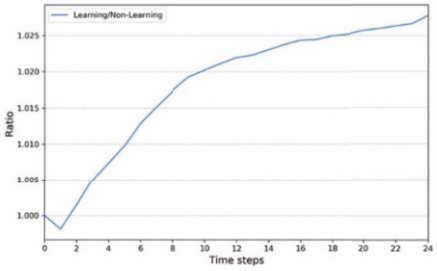

Fig. 4 Historical ratio of Learning over Non-Learning levels.

Table 3 gathers relevant statistics for both Learning and Non-Learning such as: average total performance, standard deviation of the terminal wealth X_T, Sharpe ratio computed as average total performance over standard deviation of terminal wealth. The maximum drawdown (MD) is examined through two statistics: noting $MD_\ell^{\tilde{s}}$ the maximum drawdown of the ℓ-th trajectory of a strategy $\tilde{s}$, the average MD is defined as,

$$\text{Avg MD}^{\tilde{s}} = \frac{1}{\tilde{N}} \sum_{\ell=1}^{\tilde{N}} \text{MD}_\ell^{\tilde{s}},$$

for $\tilde{N}$ trajectories of the strategy $\tilde{s}$, and the worst MD is defined as,

$$\text{Worst MD}^{\tilde{s}} = \min \left(MD_1^{\tilde{s}}, \ldots, MD_{\tilde{N}}^{\tilde{s}} \right).$$

Finally, the Calmar ratio, computed as the ratio of the average total performance over the average maximum drawdown, is the last statistic exhibited.

With the simulated dataset, Learning delivered, on average, a total performance of 9.34% while Non-Learning only 6.40%. Integrating the most recent information yielded a 2.94% excess return. Moreover, risk metrics are significantly better for Learning than for Non-Learning. Learning exhibits a lower standard deviation of

Statistic	Learning	Non-Learning	Difference
Avg total performance	9.34%	6.40%	2.94%
Std dev. of X_T	11.88%	16.67%	-4.79%
Sharpe ratio	0.79	0.38	104.95%
Avg MD	-1.53%	-6.54%	5.01%
Worst MD	-11.74%	-27.18%	15.44%
Calmar ratio	6.12	0.98	525.26%

Table 3 Performance metrics: Learning and Non-Learning. The difference for ratios are computed as relative improvement.

terminal wealth than Non-Learning (11.88% versus 16.67%), with a difference of 4.79%. More interestingly, the maximum drawdown is notably better controlled by Learning than by Non-Learning, on average (−1.53% versus −6.54%) and in the worst case (−11.74% versus −27.18%). This result suggests that learning from new observations, helps the strategy to better handle the dual objective of maximizing total wealth while controlling the maximum drawdown. We also note that learning improves the Sharpe ratio by 104.95% and the Calmar ratio by 525.26%.

Fig. 5 and 6 focus more precisely on the portfolio allocation. The graphs of Fig. 5 show the historical average allocation for each of the three risky assets. First, none of the strategies invests in Asset 2 since it has the lowest expected return according to the prior, see Table 2. Whereas Non-Learning focuses on Asset 3, the one with the highest expected return, Learning performs an optimal allocation between Asset 1 and Asset 3 since this strategy is not stuck with the initial estimate given by the prior. Therefore, Learning invests little at time 0, then balances nearly equally both Assets 1 and 3, and then invests only in Asset 3 after time step 12. Instead, Non-Learning is investing only in Asset 3, from time 0 until the end of the investment horizon.

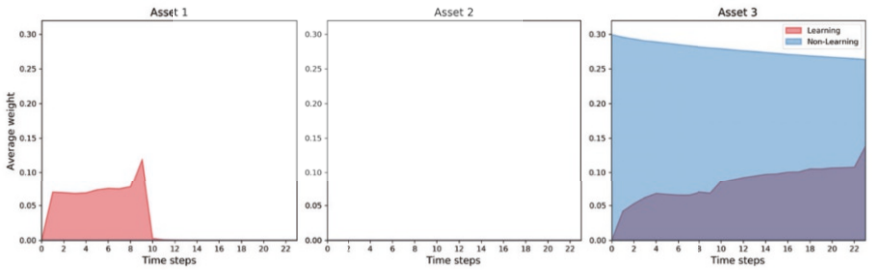

Fig. 5 Historical Learning and Non-Learning asset allocations.

The curves in Fig. 6 recall each asset's optimal weight, but the main features are the colored areas that represent the average historical total percentage of wealth invested by each strategy. The dotted line represents the total allocation constraint they should satisfy to be admissible. To satisfy the maximum drawdown constraint, admissible strategies can only invest in risky assets the proportion of wealth that, in theory, could be totally lost. This explains why the non-learning strategy invests

at full capacity on the asset that has the maximum expected return according to the prior distribution.

We clearly see that both strategies satisfy their respective constraints. Indeed, looking at the left panel, Learning is far from saturating the constraint. It has invested, on average, roughly 10% of its wealth while its constraint was set around 30%. Non-learning invests at full capacity saturating its allocation constraint. Remark that this constraint is not a straight line since it depends on the value of the ratio: current wealth over current historical maximum, and evolves according to time.

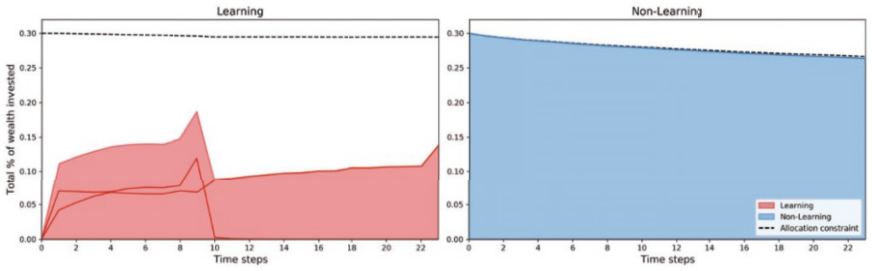

Fig. 6 Historical Learning and Non-Learning total allocations.

5.3.2 Learning, non-learning and constrained equally-weighted strategies

In this section, we add a simple constrained equally-weighted (EW) strategy to serve as a benchmark for both Learning and Non-Learning. At each time step, the constrained EW strategy invests, equally across the three assets, the proportion of wealth above the threshold q.

Fig. 7 shows the average historical levels of the three strategies: Learning, Non-Learning and constrained EW. We notice Non-Learning outperforms constrained EW and both have similar confidence intervals. It is not surprising to see that Non-Learning outperforms constrained EW since Non-Learning always bets on Asset 3, the most performing, while constrained EW diversifies the risks equally among the three assets.

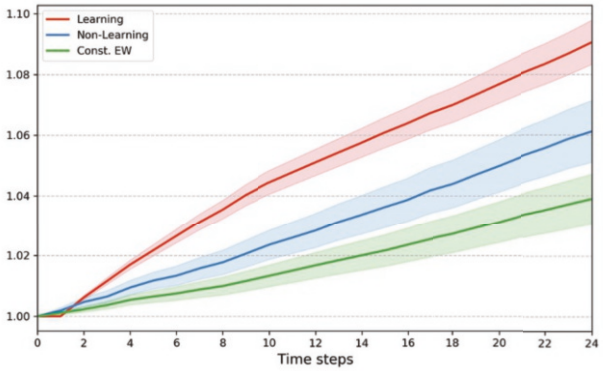

Fig. 7 Historical Learning, Non-Learning and constrained EW (Const. EW) levels with a 95% confidence interval.

Fig. 8 shows the ratio of Learning over constrained EW: it depicts the same concave shape as Fig. 4. The outperformance of Non-Learning with respect to constrained EW is plot in Fig. 9 and confirms, on average, the similarity of the two strategies.

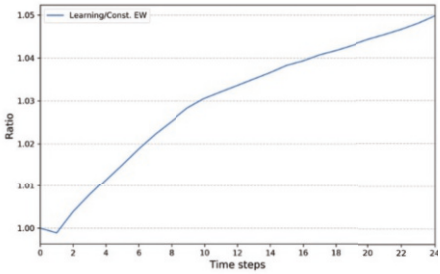

Fig. 8 Ratio Learning over constrained EW (Const. EW) according to time.

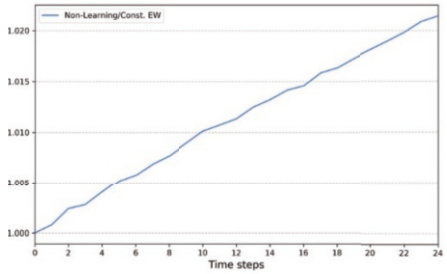

Fig. 9 Ratio Non-Learning over constrained EW (Const. EW) according to time.

Table 4 collects relevant statistics for the three strategies. Learning clearly surpasses constrained EW: it outperforms by 5.49% while reducing uncertainty on terminal wealth by 1.92% resulting in an improvement of 182.08% of the Sharpe ratio. Moreover, it better handles maximum drawdown regarding both the average and the worst case, exhibiting an improvement of 3.17% and 10.09% respectively, enhancing the Calmar ratio by 647.56%.

The Non-Learning and the constrained EW have similar profiles. Even if Non-Learning outperforms constrained EW by 2.5%, it has a higher uncertainty in terminal wealth (+2.87%). This results in similar Sharpe ratios. Maximum drawdown, both on average and considering the worst case are better handled by constrained EW (−4.70% and −21.83% respectively) than by Non-Learning (−6.54% and −27.18% respectively) thanks to the diversification capacity of constrained EW. The better per-

Statistic	Const. EW	L	NL	L - Const. EW	NL - Const. EW
Avg total performance	3.85%	9.34%	6.40%	5.49%	2.55%
Std dev. of X_T	13.80%	11.88%	16.67%	-1.92%	2.87%
Sharpe ratio	0.28	0.79	0.38	182.08%	37.63%
Avg MD	-4.70%	-1.53%	-6.54%	3.17%	-1.84%
Worst MD	-21.83%	-11.74%	-27.18%	10.09%	-5.34%
Calmar ratio	0.82	6.12	0.98	647.56%	-19.56%

Table 4 Performance metrics: Constrained EW (Const. EW) vs Learning (L) and Non-Learning (NL). The difference for ratios are computed as relative improvement.

formance of Non-Learning compensates the better maximum drawdown handling of constrained EW, entailing a better Calmar ratio for Non-Learning 0.98 versus 0.82 for constrained EW.

5.3.3 Non-learning and Merton strategies

We numerically analyze the impact of the drawdown parameter q, and compare the non-learning strategies (assuming that the drift is equal to b_0), with the constrained Merton strategy as described in Remark 2. Fig. 10 confirms that when the loss aversion parameter q goes to zero, the non-learning strategy approaches the Merton strategy.

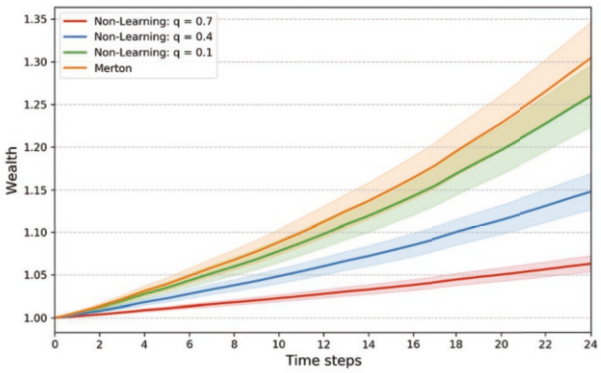

Fig. 10 Wealth curves resulting from the Merton strategy and the non-learning strategy for different values of q.

In terms of assets' allocation, the Merton strategy saturates the constraint only by investing in the asset with the highest expected return, Asset 3, while the non-learning strategy adopts a similar approach and invests at full capacity in the same asset. To illustrate this point, we easily see that the areas at the top and bottom-left corner converge to the area at the bottom-right corner of Fig. 11.

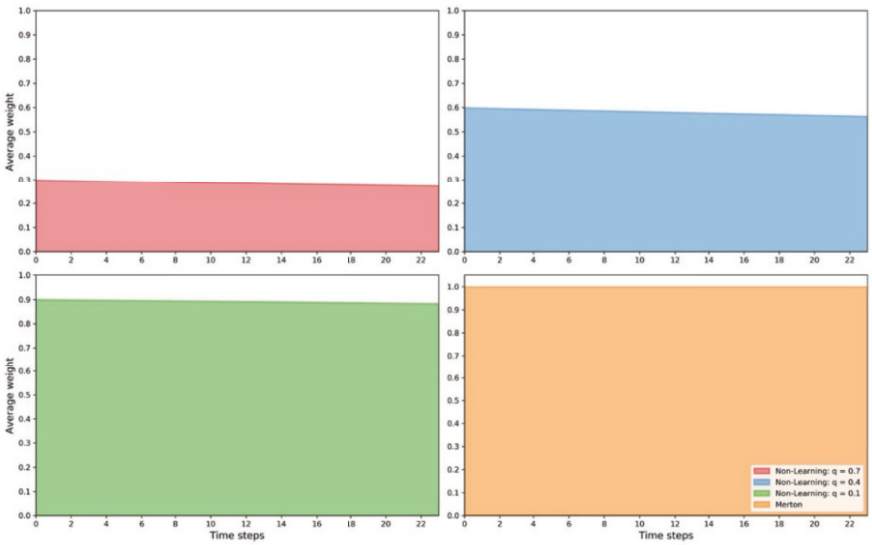

Fig. 11 Asset 3 average weights of the non-learning strategies with $q \in \{0.7, 0.4, 0.1\}$ and the Merton strategy.

As q vanishes, we observe evidence of the convergence of the Merton and the non-learning strategies, materialized by a converging allocation pattern and resulting wealth trajectories. It should not be surprising since both have in common not to learn from incoming information conveyed by the prices.

5.4 Sensitivities analysis

In this subsection, we study the effect of changes in the uncertainty about the beliefs of B. These beliefs take the form of an estimate b_0 of B, and a degree of uncertainty about this estimate, the covariance of Σ_0 of B. For the sake of simplicity, we design Σ_0 as a diagonal matrix whose diagonal entries are variances representing the confidence the investor has in her beliefs about the drift. To easily model a change in Σ_0, we define the modified covariance matrix $\tilde{\Sigma}$ as

$$\tilde{\Sigma}_{unc} := unc * \Sigma_0,$$

where $unc > 0$. From now on, the prior of B is $\mathcal{N}(b_0, \tilde{\Sigma}_{unc})$.

A higher value of unc means a higher uncertainty materialized by a lower confidence in the prior estimate of the expected return of B, b_0. We consider learning strategies with values of $unc \in \{1/6,\ 1,\ 3,\ 6,\ 12\}$. The value $unc = 1$ was used for Learning in Subsection 5.3.

Equation (2) implies that the returns' probability distribution depends upon *unc*. It implies that for each value of *unc*, we need to compute both Learning and Non-Learning on the returns sample drawn from the same probability law to make relevant comparisons.
Therefore, from a sample of a thousand returns paths' draws, we plot in Fig. 12 the average curves of the excess return of Learning over its associated Non-Learning, for different values of the uncertainty parameter *unc*.

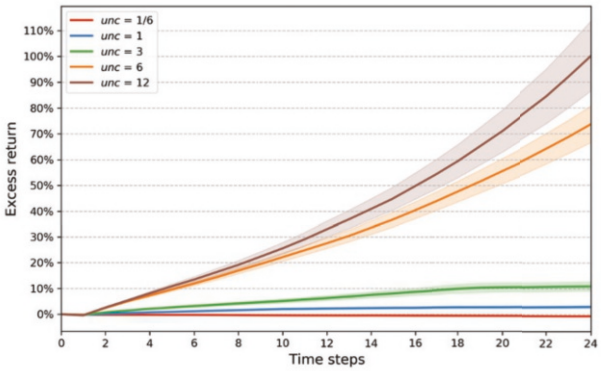

Fig. 12 Excess return of Learning over Non-Learning with a 95% confidence interval for different levels of uncertainty.

Looking at Fig. 12, we notice that when uncertainty about b_0 is low, i.e. $unc = 1/6$, Learning is close to Non-Learning and unsurprisingly the associated excess return is small. Then, as we increase the value of *unc* the curves steepen increasingly showing the effect of learning in generating excess return.

Table 5 summarises key statistics for the ten strategies computed in this section. When $unc = 1/6$, Learning underperforms Non-Learning. This is explained by the fact that Non-Learning has no doubt about b_0 and knows Asset 3 is the best performing asset acoording to its prior, whereas Learning, even with low uncertainty, needs to learn it generating a lag which explains the underperformance on average. For values of $unc \geq 1$ Learning outperforms Non-learning increasingly, as can be seen on Fig. 13, at the cost of a growing standard deviation of terminal wealth.

The Sharpe ratio of terminal wealth is higher for Learning than for Non-Learning for any value of *unc*. Nevertheless, an interesting fact is that the ratio rises from $unc = 1/6$ to $unc = 1$, then reaches a level close to 0.8 for values of $unc = 1, 3, 6$ then decreases when $unc = 12$.
This phenomenon is more visible on Fig. 14 that displays the Sharpe ratio of terminal wealth of Learning and Non-Learning according to the values of *unc*, and the associated relative improvement. Clearly, looking at Figures 13 and 14, we remark

	unc = 1/6		*unc* = 1		*unc* = 3		*unc* = 6		*unc* = 12	
Statistic	L	NL	L	NL	L	NL	L	NL	L	NL
Avg total performance	3.87%	4.35%	9.45%	6.00%	19.96%	10.25%	90.03%	16.22%	130.07%	30.44%
Std dev. of X_T	5.81%	9.22%	12.10%	17.28%	25.01%	28.18 %	113.69%	41.24%	222.77%	70.84%
Sharpe ratio	0.67	0.47	0.78	0.35	0.80	0.36	0.79	0.39	0.58	0.43
Avg MD	-2.51%	-5.21%	-1.40%	-6.78%	-1.90%	-8.40%	-2.68%	-10.14%	-3.58%	-11.35%
Worst MD	-7.64%	-17.88%	-5.46%	-24.01%	-7.99%	-26.68%	-15.62%	-29.22%	-16.98%	-29.47%
Calmar ratio	1.54	0.83	6.77	0.89	10.49	1.22	33.65	1.60	36.32	2.68

Table 5 Performance and risk metrics: Learning (L) vs Non-Learning (NL) for different values of uncertainty *unc*.

that while increasing *unc* gives more excess return, too high values of *unc* in the model turn out to be a drag as far as Sharpe ratio improvement is concerned.

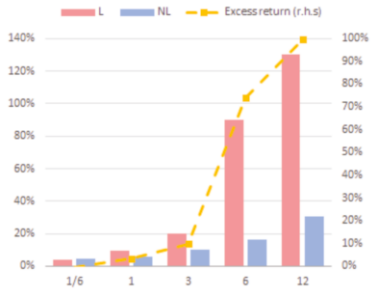

Fig. 13 Average total performance of Learning (L) and Non-Learning (NL), and excess return, for *unc* $\in \{1/6, 1, 3, 6, 12\}$.

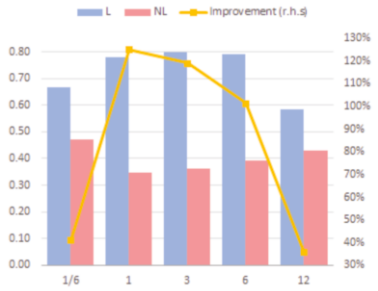

Fig. 14 Sharpe ratio of terminal wealth of Learning (L) and Non-Learning (NL), and relative improvement, for *unc* $\in \{1/6, 1, 3, 6, 12\}$.

For any value of *unc*, Learning handles maximum drawdown significantly better than Non-Learning whatever it is the average or the worst. This results in a better performance per unit of average maximum drawdown (Calmar ratio), for Learning. We also see that the maximum drawdown constraint is satisfied for every strategies of the sample and for any value of *unc* since the worst maximum drawdown is always above -30%, the lowest admissible value with a loss aversion parameter q set at 0.7. Fig. 15 reveals how the average maximum drawdown behaves regarding the level of uncertainty. Non-Learning maximum drawdown behaves linearly with uncertainty: the wider the range of possible values of B the higher the maximum drawdown is on average. It emphasizes its inability to adapt to an environment in which the returns have different behaviors compared to their expectations. Learning instead, manages to keep a low maximum drawdown for any value of *unc*. Given the previous remarks, it is obvious that the gain in maximum drawdown from learning grows with the level of uncertainty.

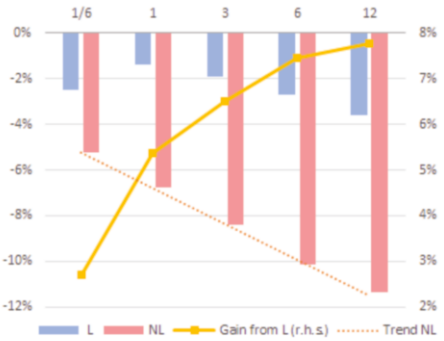

Fig. 15 Average maximum drawdown of Learning (L) and Non-Learning (NL) and the gain from learning for *unc* $\in \{1/6, 1, 3, 6, 12\}$.

Figures 16-20 represent portfolio allocations averaged over the simulations. They depict, for each value of the uncertainty parameter *unc*, the average proportion of wealth invested, in each of the three assets, by Learning and Non-Learning. The purpose is not to compare the graphs with different values of *unc* since the allocation is not performed on the same sample of returns. Rather, we can identify trends that are typically differentiating Learning from Non-Learning allocations.

Since the maximum drawdown constraint is satisfied by the capped sum of total weights that can be invested, the allocations of both Learning and Non-Learning are mainly based on the expected returns of the assets.
Non-Learning, by definition, does not depend on the value of the uncertainty parameter. Hence, no matter the value of *unc*, its allocation is easy to characterize since it saturates its constraint investing in the asset that has the best expected return according to the prior. In our setup, Asset 3 has the highest expected return, so Non-Learning invests only in it and saturates its constraint of roughly 30% during all the investment period. The slight change of the average weight in Asset 3 comes from ρ, the ratio wealth over maximum wealth, changing over time.

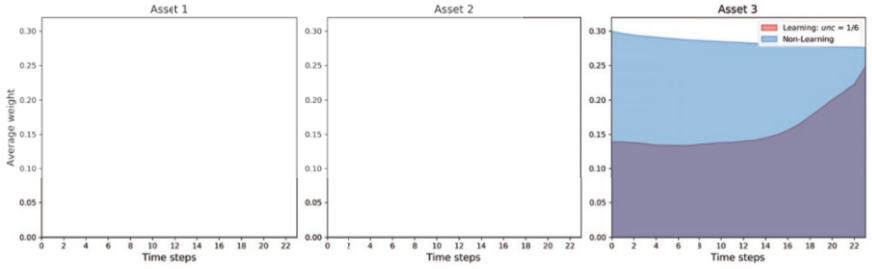

Fig. 16 Learning and Non-Learning historical assets' allocations with $unc = 1/6$.

Unlike Non-Learning, depending of the value of *unc*, Learning can perform more sophisticated allocations because it can adjust the weights according to the incoming information. Nonetheless, in Fig. 16, when *unc* is low, Learning and Non-Learning look similar regarding their weights allocation since both strategies invest, as of time 0, a significant proportion of their wealth only in Asset 3.
On the right panel of Fig. 16, the progressive increase in the weight of Asset 3 illustrates the learning process. As time goes by, Learning progressively increases the weight in Asset 3 since it has the highest expected return. It also explains why Learning underperforms Non-Learning for low values of *unc*; contrary to Non-Learning which invests at full capacity in Asset 3, Learning needs to learn that Asset 3 is the optimal choice.

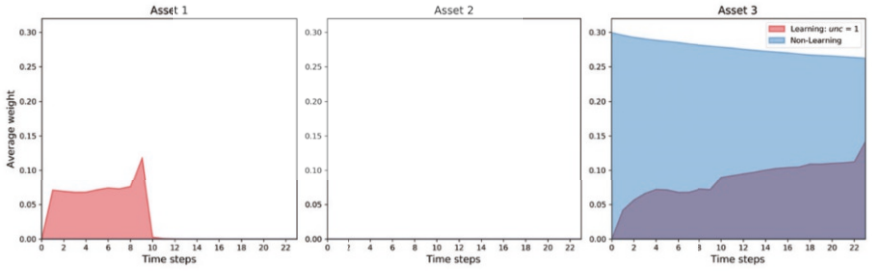

Fig. 17 Learning and Non-Learning historical assets' allocations with $unc = 1$.

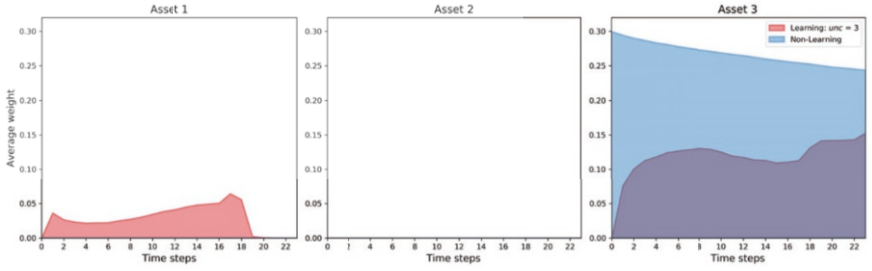

Fig. 18 Learning and Non-Learning historical assets' allocations with $unc = 3$.

However, as uncertainty increases, Learning and Non-Learning strategies start differentiating. When $unc \geq 1$, Learning invests little, if any, at time 0. In addition, an increase in *unc* allows the inital drift to lie in a wider range and generates investment opportunities for Learning. This explains why Learning invests in Asset 1 when $unc = 1, 3, 6, 12$ although the estimate b_0 for this asset is lower than for Asset 3. In Fig. 19, we see that Learning even invests in Asset 2 which has the lowest expected drift.

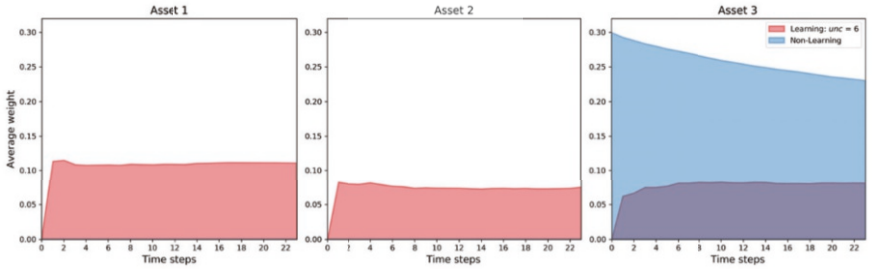

Fig. 19 Learning and Non-Learning historical assets' allocations with $unc = 6$.

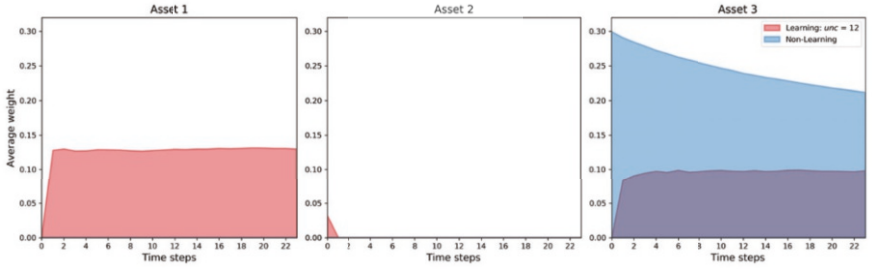

Fig. 20 Learning and Non-Learning historical assets' allocations with $unc = 12$.

Figures 21-25 illustrate the historical total percentage of wealth allocated for Learning and Non-Learning with different levels of uncertainty. As seen previously, Non-Learning has fully invested in Asset 3 for any value of unc.

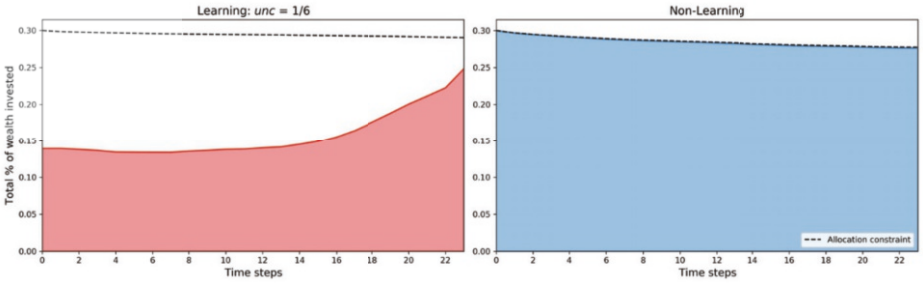

Fig. 21 Historical total allocations of Learning and Non-Learning with $unc = 1/6$.

Moreover, Learning has always less investment that Non-Learning for any level of uncertainty. It suggests that Learning yields a more cautious strategy than Non-Learning. This fact, in addition to its wait-and-see approach at time 0 and its ability

to better handle maximum drawdown, makes Learning a safer and more conservative strategy than Non-Learning. This can be seen in Fig. 21, where both Learning and Non-Learning have invested in Asset 3, but not at the same pace. Non-Learning goes fully in Asset 3 at time 0, whereas Learning increments slowly its weight in Asset 3 reaching 25% at the final step. When *unc* is low, there is no value added to choose Learning over Non-Learning from a performance perspective. Nevertheless, Learning allows for a better management of risk as Table 5 exhibits.

As *unc* increases, in addition to being cautious, Learning mixes allocation in different assets, see Figures 22-25, while Non-Learning is stuck with the highest expected return asset.

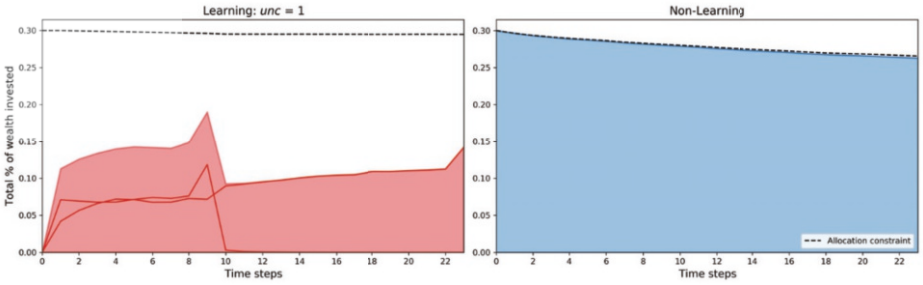

Fig. 22 Historical total allocations of Learning and Non-Learning with $unc = 1$.

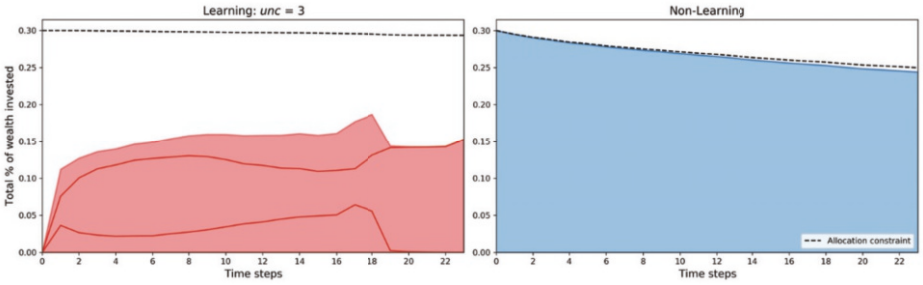

Fig. 23 Historical total allocations of Learning and Non-Learning with $unc = 3$.

Learning is able to be opportunistic and changes its allocation given the prices observed. For example in Fig. 22, Learning starts investing in Asset 1 and 3 at time 1 and stops at time 12 to weigh Asset 1 while keeping Asset 3. Similar remarks can be made for Fig. 23, where Learning puts non negligeable weights in all three risky assets for $unc = 6$ in Fig. 24.

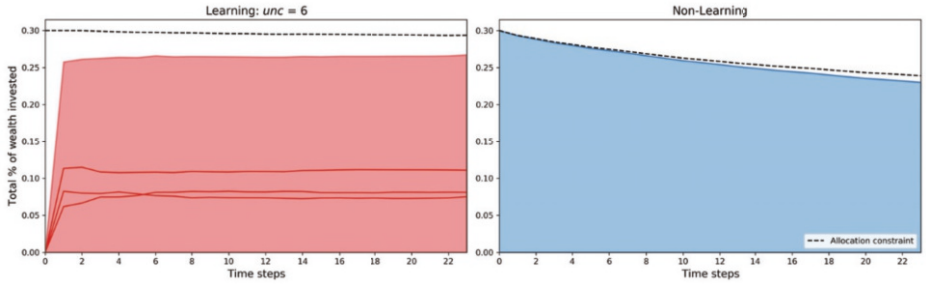

Fig. 24 Historical total allocations of Learning and Non-Learning with $unc = 6$.

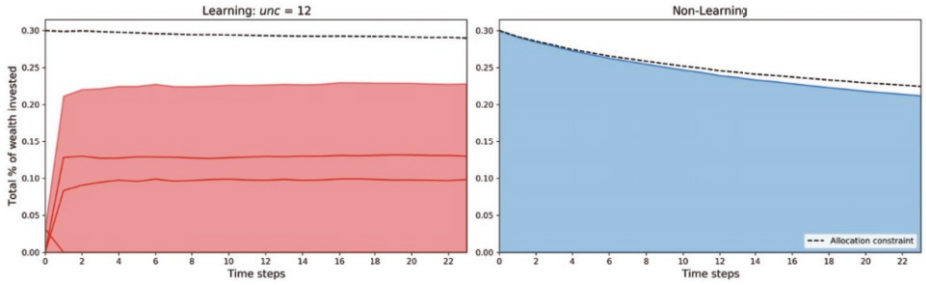

Fig. 25 Historical total allocations of Learning and Non-Learning with $unc = 12$.

6 Conclusion

We have studied a discrete-time portfolio selection problem by taking into account both drift uncertainty and maximum drawdown constraint. The dynamic programming equation has been derived in the general case thanks to a specific change of measure. More explicit results have been provided in the Gaussian case using the Kalman filter. Moreover, a change of variable has reduced the dimensionality of the problem in the case of CRRA utility functions. Next, we have provided extensive numerical results in the Gaussian case with CRRA utility functions using recent deep neural network techniques. Our numerical analysis has clearly shown and quantified the better risk-return profile of the learning strategy versus the non-learning one. Indeed, besides outperforming the non-learning strategy, the learning one provides a significantly lower standard deviation of terminal wealth and a better controlled maximum drawdown. Confirming the results established in [7], this study exhibits the benefits of learning in providing optimal portfolio allocations.

Appendix

6.1 Proof of Proposition 1

For all $k = 1, ..., N$, the law under $\overline{\mathbb{P}}$, of R_k given the filtration $\mathcal{G}_{k-1}$ yields the unconditional law under $\mathbb{P}$ of ϵ_k. Indeed, since $(\Lambda_k)_k$ is a $(\mathbb{P}, \mathbb{G})$-martingale, we have from Bayes formula, for all Borelian $F \subset \mathbb{R}^d$,

$$\begin{aligned}
\overline{\mathbb{P}}[R_k \in F|\mathcal{G}_{k-1}] = \overline{\mathbb{E}}[\mathbb{1}_{\{R_k \in F\}}|\mathcal{G}_{k-1}] &= \frac{\mathbb{E}[\Lambda_k \mathbb{1}_{\{R_k \in F\}}|\mathcal{G}_{k-1}]}{\mathbb{E}[\Lambda_k|\mathcal{G}_{k-1}]} \\
&= \mathbb{E}[\frac{\Lambda_k}{\Lambda_{k-1}} \mathbb{1}_{\{R_k \in F\}}|\mathcal{G}_{k-1}] = \mathbb{E}\left[\frac{g(B+\epsilon_k)}{g(\epsilon_k)} \mathbb{1}_{\{R_k \in F\}}|\mathcal{G}_{k-1}\right] \\
&= \int_{\mathbb{R}^d} \frac{g(B+e)}{g(e)} \mathbb{1}_{\{B+e \in F\}} g(e)de = \int_{\mathbb{R}^d} g(z)\mathbb{1}_{\{z \in F\}}dz \\
&= \mathbb{P}[\epsilon_k \in F].
\end{aligned}$$

This means that, under $\overline{\mathbb{P}}$, R_k is independent from B and from $R_1, .., R_{k-1}$ and that R_k has the same probability distribution as ϵ_k. □

6.2 Proof of Proposition 2

For any borelian function $f : \mathbb{R}^d \mapsto \mathbb{R}$ we have, on one hand, by definition of π_{k+1}:

$$\overline{\mathbb{E}}[\overline{\Lambda}_{k+1} f(B)|\mathcal{F}^o_{k+1}] = \int_{\mathbb{R}^d} f(b)\pi_{k+1}(db),$$

and, on the other hand, by definition of $\overline{\Lambda}_k$:

$$\begin{aligned}
\overline{\mathbb{E}}[\overline{\Lambda}_{k+1} f(B)|\mathcal{F}^o_{k+1}] &= \overline{\mathbb{E}}\left[\overline{\Lambda}_k f(B) \frac{g(R_{k+1}-B)}{g(R_{k+1})}\middle|\mathcal{F}^o_{k+1}\right] \\
&= \overline{\mathbb{E}}\left[\overline{\Lambda}_k f(B) g(R_{k+1}-B)\middle|\mathcal{F}^o_{k+1}\right](g(R_{k+1}))^{-1} \\
&= \int_{\mathbb{R}^d} f(b) \frac{g(R_{k+1}-b)}{g(R_{k+1})} \pi_k(db),
\end{aligned}$$

where we use in the last equality the fact that R_{k+1} is independent of B under $\overline{P}$ (recall Proposition 1). By identification, we obtain the expected relation. □

6.3 Proof of Lemma 1

Since the support of the probability distribution ν of ϵ_k is $\mathbb{R}^d$, we notice that the law of the random vector $Y_k := e^{R_k} - \mathbb{1}_d$ has support equal to $(-1,\infty)^d$. Recall from (7) that $a \in A_k^q(x,z)$ iff

$$1 + a'Y_{k+1} \geq q \max\Big[\frac{z}{x}, 1 + a'Y_{k+1}\Big], \quad a.s. \tag{21}$$

(i) Take some $a \in A_k^q(x,z)$, and assume that $a^i < 0$ for some $i \in [\![1,d]\!]$. Let us then define the event $\Omega_M^i = \{Y_{k+1}^i \geq M, Y_{k+1}^M \in [0,1], j \neq i\}$, for $M > 0$, and observe that $\mathbb{P}[\Omega_M^i] > 0$. It follows from (21) that

$$1 + a_i M + \max_{j\neq i} |a_j| \geq q\frac{z}{x}, \quad \text{on } \Omega_M^i,$$

which leads to a contradiction for M large enough. This shows that $a^i \geq 0$ for all $i \in [\![1,d]\!]$, i.e. $A_k^q(x,z) \subset \mathbb{R}_+^d$.

(ii) For $\varepsilon \in (0,1)$, let us define the event $\Omega_\varepsilon = \{Y_{k+1}^i \leq -1 + \varepsilon, i = 1,\dots,d\}$, which satisfies $\mathbb{P}[\Omega_\varepsilon] > 0$. For $a \in A^q(x,z)$, we get from (21), and since $a \in \mathbb{R}_+^d$ by Step (i):

$$1 - (1-\varepsilon)a'\mathbb{1}_d \geq q\frac{z}{x}, \quad \text{on } \Omega_\varepsilon.$$

By taking ε small enough, this shows by a contradiction argument that

$$A_k^q(x,z) \subset \Big\{a \in \mathbb{R}_+^d : 1 - a'\mathbb{1}_d \geq q\frac{z}{x}\Big\}. \;=:\; \tilde{A}^q(x,z). \tag{22}$$

(iii) Let us finally check the equality in (22). Fix some $a \in \tilde{A}^q(x,z)$. Since the random vector Y_{k+1} is valued in $(-1,\infty)^d$, it is clear that

$$1 + a'Y_{k+1} \geq 1 - a'\mathbb{1}_d \;\geq\; q\frac{z}{x} \;\geq\; 0, \quad a.s.,$$

and thus

$$1 + a'Y_{k+1} \geq q[1 + a'Y_{k+1}], \quad a.s.,$$

which proves (21), hence the equality $A^q(x,z) = \tilde{A}(x,z)$. □

6.4 Proof of Lemma 2

1. Fix $q_1 \leq q_2$ and $(x,z) \in \mathcal{S}^{q_2} \subset \mathcal{S}^{q_1}$. We then have

$$a \in A^{q_2}(x,z) \;\Rightarrow\; a \in \mathbb{R}_+^d \text{ and } a'\mathbb{1}_d \;\le\; 1 - q_2\frac{z}{x} \;\le\; 1 - q_1\frac{z}{x} \;\Longrightarrow\; a \in A^{q_1}(x,z),$$

which means that $A^{q_2}(x,z) \subseteq A^{q_1}(x,z)$.
2. Fix $q \in (0,1)$, and consider the decreasing sequence $q_n = q + \frac{1}{n}$, $n \in \mathbb{N}^*$. For any $(x,z) \in \mathcal{S}^{q_n}$, we then have $A^{q_n}(x,z) \subseteq A^{q_{n+1}}(x,z) \subset A^{a}(x,z)$, which implies that the sequence of increasing sets $A^{q_n}(x,z)$ admits a limit equal to

$$\lim_{n\to\infty} A^{q_n}(x,z) = \underset{n\ge 1}{\cup} A^{q_n}(x,z) \;=\; A^{q}(x,z),$$

since $\lim_{n\to\infty} q_n = q$. This shows the right continuity of $q \mapsto A^q(x,z)$. Similarly, by considering the increasing sequence $q_n = q - \frac{1}{n}$, $n \in \mathbb{N}^*$, we see that for any $(x,z) \in A^q(x,z)$, the sequence of decreasing sets $A^{q_n}(x,z)$ admits a limit equal to

$$\lim_{n\to\infty} A^{q_n}(x,z) = \underset{n\ge 1}{\cap} A^{q_n}(x,z) \;=\; A^{q}(x,z),$$

since $\lim_{n\to\infty} q_n = q$. This proves the continuity in q of the set $A^q(x,z)$.
3. Fix $q \in (0,1)$, and (x_1,z), $(x_2,z) \in \mathcal{S}^q$ s.t. $x_1 \le x_2$. Then,

$$a \in A^{q}(x_1,z) \;\Longrightarrow\; a \in \mathbb{R}_+^d \text{ and } a'\mathbb{1}_d \le 1 - q\frac{z}{x_1} \le 1 - q\frac{z}{x_2} \;\Longrightarrow\; a \in A^{q}(x_2,z),$$

which shows that $A^q(x_1,z) \subseteq A^q(x_2,z)$.

4. Fix $q \in (0,1)$, $(x,z) \in A^a(x,z)$. Then, for any a_1, a_2 of the set $A^q(x,z)$, and $\beta \in (0,1)]$, and denoting by $a_3 = \beta a_1 + (1-\beta)a_2 \in \mathbb{R}_+^d$, we have

$$a_3'\mathbb{1}_d \;=\; \beta a_1'\mathbb{1}_d + (1-\beta)a_2'\mathbb{1}_d \le \beta(1 - q\frac{z}{x}) + (1-\beta)(1-q\frac{z}{x}) \;=\; 1 - q\frac{z}{x}.$$

This proves the convexity of the set $A^q(x,z)$.
4. The homogeneity property of $A^q(x,z)$ is obvious from its very definition. □

6.5 Proof of Lemma 3

We prove the result by backward induction on time k from the dynamic programming equation for the value function.

• At time N, we have for all $\lambda > 0$,

$$v_N(\lambda x, \lambda z, \mu) = \frac{(\lambda x)^p}{p} = \lambda^p v_N(x,z,\mu),$$

which shows the required homogeneity property.
• Now, assume that the homogeneity property holds at time $k+1$, i.e $v_{k+1}(\lambda x, \lambda z, \mu) = \lambda^p v_{k+1}(x,z,\mu)$ for any $\lambda > 0$. Then, from the backward relation (9), and the

homogeneity property of $A^q(x,z)$ in Lemma 2, it is clear that v_k inherits from v_{k+1} the homogeneity property. □

6.6 Proof of Lemma 4

1. We first show by backward induction that $r \mapsto w_k(r,\cdot)$ is nondecreasing in on $[q,1]$ for all $k \in [\![0,N]\!]$.
• For any $r_1, r_2 \in [q,1]$, with $r_1 \le r_2$, and $\mu \in \mathcal{M}_+$, we have at time N

$$w_N(r_1,\mu) \;=\; U(r_1)\mu(\mathbb{R}^d) \le U(r_2)\mu(\mathbb{R}^d) \;=\; w_N(r_2,\mu).$$

This shows that $w_N(r,\cdot)$ is nondecreasing on $[q,1]$.
• Now, suppose by induction hypothesis that $r \mapsto w_{k+1}(r,\cdot)$ is nondecreasing. Denoting by $Y_k := e^{R_k} - \mathbb{1}_d$ the random vector valued in $(-1,\infty)^d$, we see that for all $a \in A^q(r_1)$

$$\min\Big[1, r_1(1+a'Y_{k+1})\Big] \le \min\Big[1, r_2(1+a'Y_{k+1})\Big], \quad a.s.$$

since $1 + a'Y_{k+1} \ge 1 - a'\mathbb{1}_d \ge q\frac{1}{r_1} \ge 0$. Therefore, from backward dynamic programming Equation (11), and noting that $A^q(r_1) \subset A^q(r_2)$, we have

$$\begin{aligned} w_k(r_1,\mu) &= \sup_{a\in A^q_{(}r_1)} \overline{\mathbb{E}}\Big[w_{k+1}\big(\min\big[1, r_1(1+a'Y_{k+1})\big], \bar{g}(R_{k+1}-\cdot)\mu\big)\Big] \\ &\le \sup_{a\in A^q(r_2)} \overline{\mathbb{E}}\Big[w_{k+1}\big(\min\big[1, r_2(1+a'Y_{k+1})\big], \bar{g}(R_{k+1}-\cdot)\mu\big)\Big] \;=\; w_k(r_2,\mu), \end{aligned}$$

which shows the required nondecreasing property at time k.

2. We prove the concavity of $r \in [q,1] \mapsto w_k(r,\cdot)$ by backward induction for all $k \in [\![0,N]\!]$. For $r_1, r_2 \in [q,1]$, and $\lambda \in (0,1)$, we set $r = \lambda r_1 + (1-\lambda)r_2$, and for $a_1 \in A^q(r_1)$, $a_2 \in A^q(r_2)$, we set $a = (\lambda r_1 a_1 + (1-\lambda)r_2 a_2)/r$ which belongs to $A^q(r)$. Indeed, since $a_1, a_2 \in \mathbb{R}^d_+$, we have $a \in \mathbb{R}^d_+$, and

$$a \;=\; \Big(\frac{\lambda r_1 a_1 + (1-\lambda)r_2 a_2}{r}\Big)'\mathbb{1}_d \le \frac{\lambda r_1}{r}\big(1-\frac{q}{r_1}\big) + \frac{(1-\lambda)r_2}{r}\big(1-\frac{q}{r_2}\big) \;=\; 1-\frac{q}{r}.$$

• At time N, for fixed $\mu \in \mathcal{M}_+$, we have

$$\begin{aligned} w_N(\lambda r_1 + (1-\lambda)r_2, \mu) &= U(\lambda r_1 + (1-\lambda)r_2) \\ &\ge \lambda U(r_1) + (1-\lambda)U(r_2) \;=\; \lambda w_N(r_1,\mu) + (1-\lambda)w_N(r_2,\mu), \end{aligned}$$

since U is concave. This shows that $w_N(r,\cdot)$ is concave on $[q,1]$.
• Suppose now the induction hypothesis holds true at time $k+1$: $w_{k+1}(r,\cdot)$ is concave on $[q,1]$. From the backward dynamic programming relation (11), we then have

$$
\begin{aligned}
&\lambda w_k(r_1,\mu) + (1-\lambda) w_k(r_2,\mu) \\
&\le \lambda \mathbb{E}\Big[w_{k+1}\big(\min[1, r_1(1 + a_1' Y_{k+1})], \bar{g}(R_{k+1} - \cdot)\mu \big) \Big] \\
&\qquad\qquad + (1-\lambda)\mathbb{E}\Big[w_{k+1}\big(\min[1, r_2(1 + a_2' Y_{k+1})], \bar{g}(R_{k+1} - \cdot)\mu \big) \Big] \\
&\le \mathbb{E}\Big[w_{k+1}\big(\lambda \min[1, r_1(1 + a_1' Y_{k+1})] + (1-\lambda) \min[1, r_2(1 + a_2' Y_{k+1})], \bar{g}(R_{k+1} - \cdot)\mu \big) \Big] \\
&= \mathbb{E}\Big[w_{k+1}\big(\min[1, r(1 + a' Y_{k+1})], \bar{g}(R_{k+1} - \cdot)\mu \big) \Big] \;\le\; w_k(r,\mu),
\end{aligned}
$$

where we used for the second inequality, the induction hypothesis joint with the concavity of $x \mapsto \min(1, x)$, and the nondecreasing monotonicity of $r \mapsto w_{k+1}(r, \cdot)$. This shows the required inductive concavity property of $r \mapsto w_k(r, \cdot)$ on $[q, 1]$. □

References

[1] Achref Bachouch, Côme Huré, Nicolas Langrené, and Huyên Pham. Deep neural networks algorithms for stochastic control problems on finite horizon, part 2: numerical applications. *Methodology and Computing in Applied Probability*, https://doi.org/10.1007/s11009-019-09767-9, 2021.

[2] Alexis Bismuth, Olivier Guéant, and Jiang Pu. Portfolio choice, portfolio liquidation, and portfolio transition under drift uncertainty. *Mathematics and Financial Economics*, 13(4):661–719, 2019.

[3] Stephen Boyd, Erik Lindström, Henrik Madsen, and Peter Nystrup. Multiperiod portfolio selection with drawdown control. *Annals of Operations Research*, 282(1-2):245–271, 2019.

[4] Jaksa Cvitanić and Ioannis Karatzas. On portfolio optimization under "drawdown" constraints. *Constraints, IMA Lecture Notes in Mathematics & Applications 65*, 1994.

[5] Jakša Cvitanić, Ali Lazrak, Lionel Martellini, and Fernando Zapatero. Dynamic portfolio choice with parameter uncertainty and the economic value of analysts? recommendations. *The Review of Financial Studies*, 19(4):1113–1156, 2006.

[6] Carmine De Franco, Johann Nicolle, and Huyên Pham. Bayesian learning for the markowitz portfolio selection problem. *International Journal of Theoretical and Applied Finance*, 22(07), 2019.

[7] Carmine De Franco, Johann Nicolle, and Huyên Pham. Dealing with drift uncertainty: a bayesian learning approach. *Risks*, 7(1):5, 2019.

[8] Romuald Elie and Nizar Touzi. Optimal lifetime consumption and investment under a drawdown constraint. *Finance and Stochastics*, 12(3):299, 2008.

[9] Robert J Elliott, Lakhdar Aggoun, and John B Moore. *Hidden Markov Models: Estimation and Control*. Springer, 2008.

[10] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.

[11] Sanford J Grossman and Zhongquan Zhou. Optimal investment strategies for controlling drawdowns. *Mathematical finance*, 3(3):241–276, 1993.

[12] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[13] Côme Huré, Huyên Pham, and Nicolas Bachouch, Achref a Langrené. Deep neural networks algorithms for stochastic control problems on finite horizon, part 1: convergence analysis. *SIAM Journal of Numerical Analysis*, 59(1):525–557, 2021.

[14] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, 03 1961.

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.

[16] Ioannis Karatzas and X. Zhao. Bayesian Adaptative Portfolio Optimization. In *Option Pricing, Interest Rates and Risk Management*. Cambridge University Press, 2001.

[17] Jussi Keppo, Hong Ming Tan, and Chao Zhou. Investment decisions and falling cost of data analytics. 2018.

[18] Peter Lakner. Optimal trading strategy for an investor: the case of partial information. *Stochastic Processes and their Applications*, 76(1):77–97, 1998.

[19] Imke Redeker and Ralf Wunderlich. Portfolio optimization under dynamic risk constraints: Continuous vs. discrete time trading. *Statistics & Risk Modeling*, 35(1-2):1–21, 2018.

[20] L Chris G Rogers. The relaxed investor and parameter uncertainty. *Finance and Stochastics*, 5(2):131–154, 2001.