Reinforcement Learning for Continuous-Time Mean Variance Portfolio Selection

AUDEGUY, Maximilien
maximilien.audeguy@ensae.fr

CANNARD, Bastien
bastien.cannard@ensae.fr

DJAALEB, Tom
tom.djaaleb@ensae.fr

May 3, 2024

Abstract

This work is a summary of the article *Continuous-time mean-variance portfolio selection: A reinforcement learning framework* from H. Wang and X. Y. Zhou [7]. It was made for the course "Machine Learning in Finance: Theorical Foundations" from J.D. Fermanian and H. Pham at ENSAE Paris.

1 Introduction

Reinforcement learning (RL) applications in quantitative finance, including algorithmic trading and portfolio management, have gained increasing attention. The RL is an active and fast developing subarea in machine learning. An RL agent does not pre-specify a structural model; he learns the best strategies based on trial and error, through interactions with the black-box environment (e.g. the market). This is in direct contrast with econometric methods or supervised/unsupervised learning methods commonly used in quantitative finance research. Agent's actions (controls) serve both as a means to explore (learn) and a way to exploit (optimize). A natural and crucial question is the following: the trade-off between exploration of unknown territory and exploitation of existing knowledge. The availability of extensive microstructure data in today's electronic markets enhances the training and adaptability of RL models, making them increasingly effective for tasks such as optimal order execution [5].

This paper explores the trade-off between exploration and exploitation for RL methods in a continuous-time mean-variance (MV) portfolio optimization setting, addressing the challenges and opportunities in seeking optimal solutions for Markov Decision Processes (MDP) under this framework. It models situations in which agents can interact with markets at ultra-high frequency helped by modern computing resources (e.g. high frequency trading). Interesting and insightful outcomes become attainable when framed in continuous time [8], leveraging the capabilities of stochastic calculus, differential equations, and stochastic control.

It designs a novel RL algorithm approach to continuous-time MV portfolio selection, based on theoretical foundations, and tries to keep it interpretable and implementable. This RL algorithm demonstrates its effectiveness through empirical analyses and comparisons with other state-of-the-art methods for solving the MV problem.

In this overview, following this brief introduction, we will explore the problem with the presentation of the continuous-time mean variance problem (classic case). Next, the focus will shift to the RL tool, the exploratory continuous-time mean-variance (EMV) problem, with a presentation of key concepts such as the entropy that helps regularize the exploratory mean-variance problem and the optimal value function. We will then look at how the algorithm functions, the theorems and policies used to solve it and its relationship with the classical MV problem. The key results of the study will be presented, demonstrating superior performance of the EMV algorithm compared to two other algorithms, which will also be briefly discussed, before concluding with suggestions for future improvements.

2 Problem

2.1 Continuous-time mean-variance problem

Setup The study consider a problem in a one-dimensional framework, with one risky asset and one riskless asset with constant interest rate $r > 0$. For $T > 0$ an investement horizon and $\{W_t, 0 \leq t \leq T\}$ a standard Brownian motion with respect to $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0 \leq t \leq T}, \mathbb{P})$, we can define the dynamic of the risky asset S_t

$$dS_t = S_t(\mu dt + \sigma dW_t), \qquad 0 \leq t \leq T \tag{1}$$

with $S_0 > 0$ the initial value, $\mu \in \mathbb{R}$ the mean parameter and $\sigma > 0$ the volatility. The Sharpe Ratio of the risky asset is then defined by $\rho = \frac{\mu - r}{\sigma}$, with $r > 0$ the constant interest rate of the riskless asset. We can now define $\{x_t^u, 0 \leq t \leq T\}$ the discounted value process of the portfolio and the associated strategy $u = \{u_t, 0 \leq t \leq T\}$. u_t is the discounted value invested in the risky asset at time t. The dynamic of the portfolio is

$$dx_t^u = \sigma u_t(\rho dt + dW_t) \tag{2}$$

with initial value $x_0 \in \mathbb{R}$.

Continuous-time mean-variance problem The continuous time mean-variance problem is the following optimization problem

$$\begin{array}{l} \min_u \mathrm{Var}[x_T^u] \\ \text{s. t. } \mathbb{E}[x_T^u] = z \end{array} \tag{3}$$

where x_t^u satisfies the dynamics (2) under the strategy u and $z \in \mathbb{R}$ is an investement target desired at time T and set at $t = 0$.

The study focus on "pre-committed" strategies of the MV problem, i.e. optimal strategies at $t = 0$ only. To solve the optimization problem, one need to transform it into unconstrained problem using Lagrangian:

$$\min_u \mathbb{E}[(x_T^u)^2] - z^2 - 2w(\mathbb{E}[(x_T^u - w)^2] = \min_u \mathbb{E}[(x_T^u - w)^2] - (w - z)^2 \tag{4}$$

with $w \in \mathbb{R}$ the Lagrange multiplier. The solution $u^* = \{u_t^*, 0 \leq t \leq T\}$ depends on w, which can be determined using the constraints of the problem.

2.2 Exploratory continuous-time mean-variance problem

Classical MV problem has been largely studied, but in order to compute the model, the practitioner needs to estimate parameters from market data, which can be a hard task [4]. Moreover, the model is very sensitive to these parameters and a bad calibration could result in poor performance.

RL algorithms can bypass the estimation of model parameters, by computing optimal weight portfolio directly. It is possible because models are learning (exploring) while optimizing (exploiting). A general framework for the exploratory part was proposed by Wang et al. [6].

Setup In order to propose "exploratory" version of the dynamics of the portfolio (2), the control process $u = \{u_t, 0 \leq t \leq T\}$ is randomized and become distributional with respect to the density $\pi = \{\pi_t, 0 \leq t \leq T\}$. The dynamics of the portfolio become

$$dX_t^\pi = \tilde{b}(\pi_t)dt + \tilde{\sigma}(\pi_t)dW_t, \quad 0 \leq t \leq T \tag{5}$$

with $X_0^\pi = x_0$ and

$$\tilde{b}(\pi) := \int_{\mathbb{R}} \rho\sigma u \pi(u) du \tag{6}$$

$$\tilde{\sigma}(\pi) := \sqrt{\int_{\mathbb{R}} \sigma^2 u^2 \pi(u) du} \tag{7}$$

and $\pi \in \mathcal{P}(\mathbb{R})$, the set of density functions of probability measures on $\mathbb{R}$ that are absolutely continuous with respect to Lebesgue measure. We denote by μ_t and σ_t^2 the mean and variance processes :

$$\mu_t = \int_{\mathbb{R}} u\pi_t(u)du \quad \text{and} \quad \sigma_t^2 = \int_{\mathbb{R}} u^2 \pi_t(u)du - \mu_t^2 \tag{8}$$

We have now the following dynamics

$$dX_t^\pi = \rho\sigma\mu_t dt + \sigma\sqrt{\mu_t^2 + \sigma_t^2}dW_t \tag{9}$$

The process $\pi = \{\pi_t, 0 \leq t \leq T\}$ is used to model exploration. The level of exploration is captured by the accumulative differential entropy

$$\mathcal{H}(\pi) := -\int_0^T \int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))dudt \tag{10}$$

Entropy regularized exploratory mean-variance problem The exploratory MV problem is defined as

$$\min_{\pi\in\mathcal{A}(x_0,0)} \mathbb{E}\left[(X_T^\pi - w)^2 + \lambda\int_0^T\int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))dudt\right] - (w-z)^2 \tag{11}$$

where $\mathcal{A}(x_0, 0)$ is the set of admissible distributional controls on $[0, T]$ and $\lambda > 0$ the "exploration" weight.

The optimization problem will be solved by dynamic programming. Let define the set of admissible controls $\mathcal{A}(s, y)$. Let $(s, y) \in [0, T) \times \mathbb{R}$ and consider the state equation (9) on $[s, T]$ with $X_s^\pi = y$. Let $\mathcal{B}(R)$ be the Borel algebra on $\mathbb{R}$. A control/strategy/portfolio process $\pi = \{\pi_t, s \leq t \leq T\}$ belongs to $\mathcal{A}(s, y)$ if

1. For each $s \leq t \leq T$, $\pi_t \in \mathcal{P}(\mathbb{U})$.
2. For each $A \in \mathcal{B}(\mathbb{R}), \{\int_A \pi_t(u)du, s \leq t \leq T\}$ is $\mathcal{F}_t$-mesurable.
3. $\mathbb{E}[\int_s^T(\mu_t^2 + \sigma_t^2)dt] < \infty$.
4. $\mathbb{E}\left[\left|(X_T^\pi - w)^2 + \lambda\int_s^T\int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))dudt\right| \middle| X_s^\pi = y\right] < \infty$.

From the condition 3., the SDE (9) has a unique strong solution for $s \leq t \leq T$ that satisfies $X_s^\pi = y$.

Optimal value function Controls in $\mathcal{A}(s, y)$ are called open-loop controls which means that the control is independent of the process output. In contrast, feedback controls (policies) are independent of the initial state and depend only of the process output. Feedback policies can generate open-loop controls for any $(s, y) \in [0, T) \times \mathbb{R}$.

A deterministic mapping $\boldsymbol{\pi}(\cdot; \cdot, \cdot)$ is said to be admissible feedback control if

1. $\boldsymbol{\pi}(\cdot; t, x)$ is a density function for each $(t, x) \in [0, T) \times \mathbb{R}$.
2. For each $(s, y) \in [0, T) \times \mathbb{R}$, the following SDE

$$dX_t^{\boldsymbol{\pi}} = \tilde{b}(\boldsymbol{\pi}(\cdot; t, X_t^{\boldsymbol{\pi}}))dt + \tilde{\sigma}(\boldsymbol{\pi}(\cdot; t, X_t^{\boldsymbol{\pi}}))dW_t, \quad t \in [s, T]; X_s^{\boldsymbol{\pi}} = y \tag{12}$$

has a unique strong solution $\{X_t^{\boldsymbol{\pi}}, t \in [s, T]\}$ and the open-loop control $\pi = \{\pi_t, t \in [s, T]\} \in \mathcal{A}(s, y)$ where $\pi_t := \boldsymbol{\pi}(\cdot; t, X_t^\pi)$

For a fixed $w \in \mathbb{R}$, the authors define

$$V(s, y; w) := \inf_{\pi\in\mathcal{A}(s,y)} \mathbb{E}\left[(X_T^\pi - w)^2 + \lambda\int_s^T\int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))dudt \middle| X_s^\pi = y\right] - (w-z)^2 \tag{13}$$

for $(s, y) \in [0, T) \times \mathbb{R}$. The function V is called the optimal value function of the problem, and they define the value function under any given feedback control $\boldsymbol{\pi}$

$$V^{\boldsymbol{\pi}}(s, y; w) = \mathbb{E}\left[(X_T^{\boldsymbol{\pi}} - w)^2 + \lambda\int_0^T\int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))dudt \middle| X_s^{\boldsymbol{\pi}} = y\right] - (w-z)^2 \tag{14}$$

for $(s, y) \in [0, T) \times \mathbb{R}$, where $\pi = \{\pi_t, t \in [s, T]\}$ is the open-loop control generated from $\boldsymbol{\pi}$ with respect to (s, y) and $\{X_t^{\boldsymbol{\pi}}, t \in [s, T]\}$ is the corresponding wealth process.

3 Contributions

3.1 Solution to the EMV problem and equivalence with classical MV

Solving the EMV problem In order to solve the exploratory MV problem (11), the authors apply the classical Bellman's principle of optimality

$$V(t,x;w) = \inf_{\pi\in\mathcal{A}(t,x)} \mathbb{E}\left[V(s,X_s^\pi;w) + \lambda\int_t^s\int_{\mathbb{R}} \pi_v(u)\ln(\pi_v(u))dudv \middle| X_t^\pi = x\right]$$

for $x\in\mathbb{R}$ and $0\le t<s\le T$. V satisfies the Halmiton-Jacobi-Bellman (HJB) equation

$$v_t(t,x;w) + \min_{\pi\in\mathcal{P}(\mathbb{R})}\int_{\mathbb{R}}\left(\frac{1}{2}\sigma^2u^2v_{xx}(t,x;w) + \rho\sigma uv_x(t,x;w) + \lambda\ln(\pi(u))\right)\pi(u)du = 0 \tag{15}$$

with the terminal condition $v(T,x;w) = (x-w)^2 - (w-z)^2$. Using verification technique and the fact that $\pi\in\mathcal{P}(\mathbb{R})$, the optimization problem in the HJB can be solved and the optimal feedback control is given by

$$\boldsymbol{\pi}^*(u;t,x,w) = \mathcal{N}\left(u\middle| -\frac{\rho}{\sigma}\frac{v_x(t,x)}{v_{xx}(t,x;w)}, \frac{\lambda}{\sigma^2v_{xx}(t,x;w)}\right) \tag{16}$$

where $\mathcal{N}(u|\alpha,\beta)$ is the gaussian density with mean α and variance $\beta>0$. Using this result, we obtain the classical solution of the HJB equation

$$v(t,x;w) = (x-w)^2e^{-\rho^2(T-t)} + \frac{\lambda\rho^2}{4}(T^2-t^2) - \frac{\lambda}{2}\left(\rho^2T - \ln\left(\frac{\sigma^2}{\lambda\pi}\right)\right)(T-t) - (w-z)^2 \tag{17}$$

which satisfies $v_{xx}(t,x;w) > 0$ for any $(t,x)\in[0,T]\times\mathbb{R}$. The optimal feedback Gaussian control reduces to

$$\boldsymbol{\pi}^*(u;t,x,w) = \mathcal{N}\left(u\middle| -\frac{\rho}{\sigma}(x-w), \frac{\lambda}{2\sigma^2}e^{\rho^2(T-t)}\right), \quad (t,x)\in[0,T]\times\mathbb{R} \tag{18}$$

Finally, the optimal wealth process under $\boldsymbol{\pi}^*$ is

$$dX_t^* = -\rho^2(X_t^*-w)dt + \sqrt{\rho^2(X_t^*-w)^2 + \frac{\lambda}{2}e^{\rho^2(T-t)}}dW_t, \quad X_0^* = x_0 \tag{19}$$

The optimal value function V (17), the optimal feedback control $\boldsymbol{\pi}^*$ (Gaussian) (18) and the optimal wealth process X_t^* (19) are the main results in the *Theorem 1* of the paper. Moreover, the Lagrange multiplier w is given by $w = \frac{ze^{\rho^2T}-x_0}{e^{\rho^2T}-1}$. Here are some observations that could be relevant :

- Classical and Exploratory MV have same Lagrange multiplier (*Theorem 2* presented later).
- In EMV, the variance of the optimal policy decays in time.
- The variance of the optimal policy decreases as the volatility increases.
- Mean of the optimal policy is independent of the exploration weight λ.
- Variance of the optimal policy is independent of state x.

Solvability equivalence with classical problem and cost of exploration Solvability equivalence means that the solution of one problem will lead to that of the other directly. However, both problems can be solved separately. The optimal value function of the classical problem is

$$V^{\text{cl}}(t,x;w) = (x-w)^2e^{-\rho^2(T-t)} - (w-z)^2 \tag{20}$$

and the optimal feedback control policy is

$$\boldsymbol{u}^*(u;t,x,w) = -\frac{\rho}{\sigma}(x-w) \tag{21}$$

Finally, the optimal wealth process is

$$dx_t^* = -\rho^2(x_t^* - w)dt - \rho(x_t^* - w)dW_t, \quad x_0^* = x_0 \tag{22}$$

We observe that the optimal wealth process for both problems have the same drift which implies the same value of Lagrange multiplier. The *Theorem 2* of the paper propose that the two statements are equivalent :

- (17) is the optimal value function of the exploratory MV and (18) is the associated optimal policy.
- (20) is the optimal value function of the classical MV and (21) is the associated optimal policy.

Theorem 3 of the paper show the convergence of the solution of the exploratory MV to the solution of classical MV when $\lambda \to 0$:

$$\lim_{\lambda \to 0} \boldsymbol{\pi}^*(\cdot; t, x; w) = \delta_{\boldsymbol{u}^*(t,x;w)}(\cdot) \text{ weakly}^1 \tag{23}$$

Moreover

$$\lim_{\lambda \to 0} |V(t, x; w) - V^{\text{cl}}(t, x; w)| = 0 \tag{24}$$

The authors propose to define the cost of exploration as

$$\mathcal{C}^{u^*,\pi^*}(0, x_0, w) := V(0, x_0; w) - \lambda \, \mathbb{E}\left[\int_0^T \int_{\mathbb{R}} \pi_t^*(u) \ln(\pi_t^*(u)) du dt \middle| X_0^{\pi^*} = x_0\right] - V^{\text{cl}}(0, x_0; w) \tag{25}$$

for $x_0 \in \mathbb{R}$ and $\pi^* = \{\pi_t^*, t \in [0, T)\}$ is the optimal strategy generated by the optimal feedback law $\boldsymbol{\pi}^*$. The *Theorem 4* of the paper show that the exploration cost for the MV problem is

$$\mathcal{C}^{u^*,\pi^*}(0, x_0, w) = \frac{\lambda T}{2}, \quad x_0, w \in \mathbb{R} \tag{26}$$

The exploration cost is linear dependent of exploratory weight λ and horizon T. However, it is independent of lagrange multiplier, which means that it do not depends of the profile of investor (aggressive or conservative).

3.2 Policy improvement theorem and RL algorithm

PIT and convergence to optimal policy RL algorithms are composed of two procedures : *policy evaluation* which estimates value function for current policy, and *policy improvements* which updates the current policy in order to improve the value function. *Theorem 5* describes the policy improvement theorem proposed by Wang & Zhou for the exploratory MV problem :

- *Let $w \in \mathbb{R}$ be fixed and $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot; \cdot, \cdot; w)$ be an arbitrarily given admissible feedback control policy. Suppose that value function $V^{\boldsymbol{\pi}}(\cdot, \cdot; w) \in C^{1,2}([0, T) \times \mathbb{R}) \cap C^0([0, T] \times \mathbb{R})$ and satisfies $V_{xx}^{\boldsymbol{\pi}}(t, x; w) > 0$, for any $(t, x) \in [0, T) \times \mathbb{R}$. Suppose that the feedback policy $\tilde{\boldsymbol{\pi}}$ is defined by*

$$\tilde{\boldsymbol{\pi}}(u; t, x, w) = \mathcal{N}\left(u | -\frac{\rho}{\sigma} \frac{V_x^{\boldsymbol{\pi}}(t, x)}{V_{xx}^{\boldsymbol{\pi}}(t, x; w)}, \frac{\lambda}{\sigma^2 V_{xx}^{\boldsymbol{\pi}}(t, x; w)}\right) \tag{27}$$

  is admissible. Then

$$V^{\tilde{\boldsymbol{\pi}}}(t, x; w) \leq V^{\boldsymbol{\pi}}(t, x; w), \quad (t, x) \in [0, T] \times \mathbb{R} \tag{28}$$

This theorem suggests that there are always policies which are Gaussian that improves the value function of any policy. Moreover, *Theorem 1* suggests that a good initial feedback policy may have the form

$$\mathcal{N}(u | a(x - w), c_1 e^{c_2(T-t)})$$

Theorem 6 validate the convergence of the policy and value function to their optimal values, in the case where initial policy is choose carefully :

[1] $\delta_d(x) = \delta(x - d)$ is the Dirac off-center function

- *Let* $\boldsymbol{\pi}_0(u;t,x,w) = \mathcal{N}(u|a(x-w), c_1^{c_2(T-t)})$*, with* $a, c_2 \in \mathbb{R}, c_1 > 0$*. Denote by* $\{\boldsymbol{\pi}_n(u;t,x;w), (t,x) \in [0,T]\times\mathbb{R}, n \geq 1\}$ *the sequence of feedback policies updated by the policy improvement scheme (27), and* $\{V^{\boldsymbol{\pi}_n}(t,x;w), (t,x) \in [0,T]\times\mathbb{R}, n \geq 1\}$ *the sequence of the corresponding value functions. Then*

$$\lim_{n\to\infty} \boldsymbol{\pi}_n(\cdot;t,x;w) = \boldsymbol{\pi}^*(\cdot,t,x;w) \text{ weakly} \tag{29}$$

and

$$\lim_{n\to\infty} V^{\boldsymbol{\pi}_n}(t,x;w) = V(t,x;w) \tag{30}$$

for any $(t,x,w) \in [0,T]\times\mathbb{R}\times\mathbb{R}$

Proofs of *Theorem 5* and *6* can be found in the paper.

EMV Algorithm The EMV Algorithm consists of three procedures : policy evaluation, policy improvement and self correcting scheme for learning w. For policy evaluation, the method in [1] is used for learning $V^{\boldsymbol{\pi}}$. By rearranging Bellman's equation and dividing by $s-t$, we obtain

$$\mathbb{E}\left[\frac{V^{\boldsymbol{\pi}}(s,X_s) - V^{\boldsymbol{\pi}}(t,X_t)}{s-t} + \frac{\lambda}{s-t}\int_s^t \int_{\mathbb{R}} \pi_v(u)\ln(\pi_v(u))dudv \,\middle|\, X_t = x\right] = 0 \tag{31}$$

By taking $s \to t$, we obtain the continuous-time Bellman's error (TD error in [1]) :

$$\delta_t := \dot{V}_t^{\boldsymbol{\pi}} + \lambda \int_{\mathbb{R}} \pi_t(u)\ln(\pi_t(u))du \tag{32}$$

where $\dot{V}_t^{\boldsymbol{\pi}} = \frac{V^{\boldsymbol{\pi}}(t+\Delta t, X_{t+\Delta t}) - V^{\boldsymbol{\pi}}(t,X_t)}{\Delta t}$ is the total derivative and Δt is the discretization step. The objective of the policy evaluation is to minimize δ_t. By denoting V^θ and π^ϕ the parametrized value function and policy, the goal is to minimize

$$C(\theta,\phi) = \frac{1}{2}\mathbb{E}\left[\int_0^T |\delta_t|^2 dt\right] = \frac{1}{2}\mathbb{E}\left[\int_0^T |\dot{V}_t^\theta + \lambda\int_{\mathbb{R}} \pi_t^\phi(u)\ln(\pi_t^\phi(u))du|^2 dt\right] \tag{33}$$

where $\pi^\phi = \{\pi_t^\phi, t \in [0,T]\}$ is generated from $\boldsymbol{\pi}^\phi$. To approximate $C(\theta,\phi)$, one needs to discretize $[0,T]$ into small equal length intervals $[t_i, t_{i+1}]$ and then collect set of samples $\mathcal{D} = \{(t_i, x_i), i = 0, 1..., l+1\}$ with initial sample $(0, x_0)$ for $i = 0$. At each t_i, sample $\pi_{t_i}^\phi$ to obtain allocation u_i in risky asset, and then observe wealth x_{i+1} at next time t_{i+1}. The approximation of $C(\theta,\phi)$ is then

$$C(\theta,\phi) = \frac{1}{2}\sum_{(t_i,x_i)\in\mathcal{D}} \left(\dot{V}^\theta(t_i,x_i) + \lambda\int_{\mathbb{R}} \pi_{t_i}^\phi(u)\ln(\pi_{t_i}^\phi(u))du\right)^2 \Delta t \tag{34}$$

Instead of using deep neural networks to represent V^θ, π^ϕ, authors take advantage of the explicit parametric expressions of *Theorem 1* and *5*. By focusing on Gaussian policy with variance $c_1 e^{c_2(T-t)}$, it leads to entropy parameterized by $\mathcal{H}(\pi_t^\phi) = \phi_1 + \phi_2(T-t), \phi_1 \in \mathbb{R}, \phi_2 > 0$. By *Theorem 1* optimal value function (17), the parameterized V^θ is given by

$$V^\theta(t,x) = (x-w)^2 e^{-\theta_3(T-t)} + \theta_2 t^2 + \theta_1 t + \theta_0, \quad (t,x) \in [0,T]\times\mathbb{R} \tag{35}$$

with $\theta = (\theta_0,\theta_1,\theta_2,\theta_3)'$ From *Theorem 5* result (27), the variance of policy π_t^ϕ is $\frac{\lambda}{2\sigma^2}e^{\theta_3(T-t)}$, resulting the entropy $\frac{1}{2}\ln\left(\frac{\pi e\lambda}{\sigma^2}\right) + \frac{\theta_3}{2}(T-t)$. We obtain

$$\sigma^2 = \lambda\pi e^{1-2\phi_1} \text{ and } \theta_3 = 2\phi_2 = \rho^2 \tag{36}$$

The improved policy becomes

$$\boldsymbol{\pi}(u;t,x,w) = \mathcal{N}\left(u|-\sqrt{\frac{2\phi_2}{\lambda\pi}}e^{\frac{2\phi_1-1}{2}}(x-w), \frac{1}{2\pi}e^{2\phi_2(T-t)+2\phi_1-1}\right) \tag{37}$$

where $\rho > 0$ is assumed. We then obtain

$$C(\theta, \phi) = \frac{1}{2} \sum_{(t_i, x_i) \in \mathcal{D}} \left(\dot{V}^\theta(t_i, x_i) - \lambda(\phi_1 + \phi_2(T - t_i)) \right)^2 \Delta t \tag{38}$$

with $\dot{V}^\theta(t_i, x_i) = \frac{V^\theta(t_{i+1}, x_{i+1}) - V^\theta(t_i, x_i)}{\Delta t}$. In order to learn parameters, a stochastic gradient descent (SGD) algorithm will be implemented, the partial derivatives are

$$\frac{\partial C}{\partial \theta_1} = \sum_{(t_i, x_i) \in \mathcal{D}} \left(\dot{V}^\theta(t_i, x_i) - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t \tag{39}$$

$$\frac{\partial C}{\partial \theta_2} = \sum_{(t_i, x_i) \in \mathcal{D}} \left(\dot{V}^\theta(t_i, x_i) - \lambda(\phi_1 + \phi_2(T - t_i)) \right) (t_{i+1}^2 - t_i^2) \tag{40}$$

$$\frac{\partial C}{\partial \phi_1} = -\lambda \sum_{(t_i, x_i) \in \mathcal{D}} \left(\dot{V}^\theta(t_i, x_i) - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t \tag{41}$$

$$\begin{aligned} \frac{\partial C}{\partial \phi_2} &= \sum_{(t_i, x_i) \in \mathcal{D}} \left(\dot{V}^\theta(t_i, x_i) - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t \\ \times \left(-\frac{2(x_{i+1} - w)^2 e^{-2\phi_2(T - t_{i+1})}(T - t_{i+1}) - 2(x_i - w)^2 e^{-2\phi_2(T - t_i)}(T - t_i)}{\Delta t} - \lambda(T - t_i) \right) \end{aligned} \tag{42}$$

We also have $\theta_3 = 2\phi_2$ and

$$\theta_0 = -\theta_2 T^2 - \theta_1 T - (w - z)^2 \tag{43}$$

Finally the learning scheme for w is

$$w_{n+1} = w_n - \alpha_n (X_T - z) \tag{44}$$

with $\alpha_n > 0, n \geq 1$. For implementation, X_T can be replaced by sample average $\frac{1}{N} \sum_j x_T^j$. It results the following pseudo-code for the EMV Algorithm :

Algorithm 1 EMV: Exploratory Mean-Variance Portfolio Selection

Require: Market simulator, learning rates $\alpha, \eta_\theta, \eta_\phi$, initial wealth x_0, target payoff z, investment horizon T, discretization Δt, exploration rate λ, number of iterations M, sample average size N.
Initialize θ, ϕ and w
for $k = 1$ **to** M **do**
  for $i = 1$ **to** $\lfloor \frac{T}{\Delta t} \rfloor$ **do**
    Sample (t_i^k, x_i^k) from *Market* under π^ϕ.
    Obtain collected samples $\mathcal{D} = \{(t_i^k, x_i^k), 1 \leq i \leq \lfloor \frac{T}{\Delta t} \rfloor\}$.
    Update $\theta \leftarrow \theta - \eta_\theta \nabla_\theta C(\theta, \phi)$ using (39) and (40).
    Update θ_0 using (43) and $\theta_3 \leftarrow 2\phi_2$.
    Update $\phi \leftarrow \phi - \eta_\phi \nabla_\phi C(\theta, \phi)$ using (41) and (42)
  end for
  Update $\pi^\phi \leftarrow \mathcal{N}\left(u| - \sqrt{\frac{2\phi_2}{\lambda \pi}} e^{\frac{2\phi_1 - 1}{2}} (x - w), \frac{1}{2\pi} e^{\phi_2(T-t) + 2\phi_1 - 1} \right)$.
  if $k \bmod N == 0$ **then**
    Update $w \leftarrow w - \alpha \left(\frac{1}{N} \sum_{j=k-N+1}^{k} x^j_{\lfloor \frac{T}{\Delta t} \rfloor} - z \right)$
  end if
end for

4 Simulation and numerical results

In this section, we present the main results of the study in terms of simulations. Indeed, we compare the performance of the EMV algorithm with two other methods that could be used to solve the classical MV problem.

The first one is the traditional maximum likelihood estimation (MLE) that relies on the real-time estimation of the drift μ and the volatility σ in the geometric Brownian motion price model (45). MLE is widely used for estimating parameters such as μ and σ in the geometric Brownian motion model (45). Once the estimators of μ and σ are available using the most recent price time series, the portfolio allocation can be computed using the optimal allocation for the classical MV problem and the Lagrange multiplier formula for investment in risky assets. MLE is particularly effective in adaptive control frameworks, commonly involving an identification phase followed by optimization, and is well-suited for dynamic, non-stationary markets with fluctuating parameters.

Another alternative is based on the deep deterministic policy gradient (DDPG) method. The DDPG method, introduced by Lillicrap et al. [3], is a prominent approach for continuous control reinforcement learning problems. It uses deep neural networks to learn deterministic policies, enhanced by exogenous noise for exploration. To tailor DDPG for the classical MV framework, adjustments include using prioritized experience replay to emphasize terminal experiences in training, and modifying inputs to the actor network for policy determination based on current and historical data. These modifications improve both learning efficiency and overall performance.

4.1 Experiments in different market case

Stationary market case In numerical simulations set, the price model is a process using geometric Brownian motion with constant parameters μ and σ. Over a one-year period, with daily rebalancing, consider μ values from the set $\{-50\%, -30\%, -10\%, 0\%, 10\%, 30\%, 50\%\}$ and σ values from $\{10\%, 20\%, 30\%, 40\%\}$. These parameters reflect typical annualized returns and volatilities assumed for a standard stock in simulation studies. The annualized interest rate is set at $r = 2\%$. The MV problem is aiming for a 40% annualized target return on terminal wealth, beginning from a normalized initial wealth of $x_0 = 1$, thus setting $z = 1.4$ as per equation (3). For the EMV algorithm, set the total number of training episodes to $M = 20000$ and the sample size for learning the Lagrange multiplier w to $N = 10$. The exploration weight is set to $\lambda = 2$. Learning rates are consistently fixed at $\alpha = 0.05$ and $\eta_\theta = \eta_\phi = 0.0005$ across all simulations. To ensure a fair comparison, the same values of M and N are used for the DDPG algorithm. For each method under various market scenarios, the annualized sample mean $\overline{M}$ and sample variance $\overline{V}$ of the last 2000 values of terminal wealth will be compared. Additionally, the corresponding annualized Sharpe ratio $\text{SR} = \frac{\overline{M}-1}{\sqrt{\overline{V}}}$ will be the validation metric.

The full results can be found in the paper. The EMV Algorithm outperform MLE on all the 28 experiments and DDPG on 23 experiments, with a better Sharpe ratio. Training times are the following: EMV and MLE less than 10 seconds; DDPG approximately 3 hours. Moreover, the MLE algorithm is relatively stable and converges faster without exploding.

We have replicate the experiments of the paper and provide further details in Appendix A.

Non stationary market case In this setting, numerical simulations adopt a slowly varying factor within a stochastic factor model, particularly a multi-scale stochastic volatility framework as proposed by Fouque et al. [2]. The dynamic of the risky asset is then

$$dS_t = S_t(\mu dt + \sigma_t dW_t), \qquad 0 \le t \le T \tag{45}$$

where μ_t and σ_t are the drift and volatility processes that vary over each simulation episode. The controlled dynamic of the portfolio over each episode is

$$dx_t^u = \sigma_t u_t(\rho_t dt + dW_t) \tag{46}$$

This reflects time-variant investment opportunities and market conditions. To ensure these stochastic factors change at a much slower scale compared to the learning process, their dynamics are defined by $d\rho_t = \delta dt$ and $d\sigma_t = \sigma_t(\delta dt + \sqrt{\delta} dW_t^1)$, with small $\delta = 0.0001$ to maintain factor values within reasonable ranges over all training episodes. Initial conditions are set with $\rho_0 = -3.2$ and $\sigma_0 = 10$, equivalent to an initial drift $\mu_0 = -30\%$. In this setup, the validation of learning performance utilizes the annualized Sharpe Ratios computed over the last 50 values of terminal wealth, where EMV, MLE, and DDPG achieved scores of 4.43, 3.61, and 6.67 respectively, demonstrating the EMV Algorithm's notable stability even under varying market conditions.

4.2 Decaying exploration weight

Authors also explore the efficacy of a decaying exploration weight λ, arguing that as the RL algorithm iterates, it should increasingly favor exploitation over exploration. By using the outcomes of *Theorem 3*, the paper suggests that the EMV Algorithm could be enhanced by a decaying λ rather than keeping it constant throughout the learning process. To implement this idea, a specific decaying process is introducedtsc for λ, which gradually reduces its value over a series of learning episodes

$$\lambda_k = \lambda_0 \left(1 - \exp\left(\frac{200(k-M)}{M}\right)\right), \text{ for } k = 0, 1, 2, \ldots, M. \tag{47}$$

This approach is empirically validated by comparing the performance of the EMV algorithm using both a constant and a decaying λ. Results indicate a denser distribution and improved robustness when a decaying λ is employed in both market settings (stationary or non stationary), a conclusion that is further substantiated by an increase in the Sharpe Ratio from 3.039 with a constant λ to 3.243 with decaying λ.

In Appendix B, we share our doubts on the formula of the decaying lambda (which is the same in the published paper) and propose to modify it.

5 Conclusion

This research paper presents several strengths both in terms of the model it develops and its overall contribution to the field. The model eliminates the need to estimate parameters, offering a significant advantage over traditional models that require intricate parameter tuning. This characteristic not only makes the model more accessible but also enhances its interpretability compared to black-box models, allowing for a better understanding of how inputs are transformed into outputs.

In terms of performance, the model outperforms established algorithms like DDPG and EMV in both stationary and non-stationary settings, demonstrating robustness and effectiveness across different market conditions. This ability underscores its practical applicability and potential in real-world financial scenarios.

The paper itself is well-structured, beginning with a solid introduction that sets the context and outlines the mean-variance (MV) problem framework. It clearly delineates the challenges and limitations of existing approaches while introducing an innovative reinforcement learning (RL) algorithm designed to tackle the MV problem more effectively. A significant strength of the research paper is its introduction of an algorithm for tackling the mean-variance portfolio problem, even though the algorithm itself is not presented with a detailed implementation guide. This contributes positively to the paper, as it advances theoretical understanding and opens avenues for further development.

This research paper presents an innovative approach to continuous-time mean-variance portfolio selection but encounters specific limitations that might constrain its broader application and clarity. The assumption of log-normal prices in the model represents a significant limitation, as it may not accurately reflect the true dynamics of financial markets where price distributions can exhibit fat tails and skewness.

Additionally, the description of Algorithm 1 lacks detail in certain steps, making the implementation challenging. This lack of clarity in algorithmic steps may prevent effective application or adaptation of the proposed model by practitioners and researchers who might not have access to the authors' specific computational setup or detailed understanding of their methods. These issues highlight critical areas where the paper could improve to enhance its accessibility and applicability in financial modeling and reinforcement learning contexts.

References

[1] Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 2000.

[2] Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, and Knut Solna. Multiscale stochastic volatility asymptotics. *Multiscale Modeling & Simulation*, 2(1):22–42, 2003.

[3] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[4] David G Luenberger. *Investment science*. Oxford university press, 1998.

[5] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, pages 673–680, 2006.

[6] Haoran Wang, Thaleia Zariphopoulou, and Xunyu Zhou. Exploration versus exploitation in reinforcement learning: A stochastic control approach. *arXiv preprint arXiv:1812.01552*, 2018.

[7] Haoran Wang and Xun Yu Zhou. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273–1308, 2020.

[8] Xun Yu Zhou and Duan Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, 42:19–33, 2000.

A Replication in stationary market case

We have replicate the experiments of the paper in the stationary market case. It is not indicate in the article but the formula for calculating the value of the portfolio at t_{i+1} is

$$X_{t_{i+1}} = X_{t_i} \pm u_{t_i} \left(e^{-r\Delta t} \frac{S_{t_{i+1}}}{S_{t_i}} - 1 \right) \tag{48}$$

As ρ is assumed to be strictly positive in the parametrized gaussian density π^ϕ, if $\mu < r$, we need to be careful of the sign of the true u_{t_i}. The fact that the true μ of the market can be unknown is a problem for using the algorithm in real conditions. There will also be an assymetry in the results with symmetric μ (example $\{-50\%, 50\%\}$ due to a non-zero interest rate.

In order to replicate the experiments, it would have been appreciated if the authors had share the initial values fixed for the parameters. The Table 1 summarizes the results we obtained.

First, we used $\eta_\theta = \eta_\phi = 5 \times 10^{-5}$ (instead of 5×10^{-4} in the paper) because the algorithm didn't converge for any market parameters with learning rates in the paper. We noticed that the algorithm do not take less than 10 seconds to be executed as said in the paper but approximatively 15 minutes (run on Intel Core I5-1235U). We also noticed that the convergence can be determined by the initialization of the parameters ϕ, θ, which indicates that sharing parameters used in the paper would be more important.

For experiments that converges, we obtain similar results as in the paper, which is appreciated for us and suggest that we implemented correctly the algorithm. However, we observed that for some parameters (usually small μ), the algorithm doesn't converge and gradient is vanishing. We try to explain this by the formula of updating w : in small μ markets, algorithm do not perform well and so x_T are very far from z (below), w is increasing every 10 iteration and never attain is pseudo-optimal value. As $V(\theta, \phi)$ and other formulas are in w^2, the gradients are vanishing. Some graphical illustrations can be found in the code provided, which will also be available at `https://github.com/Tdjaaleb`.

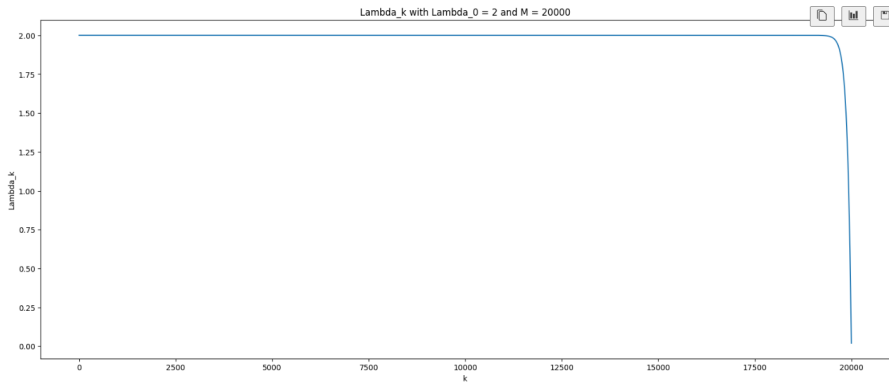
For some reasons presented before, it was an hard task to implement the algorithm, however, we do think that it was a worth task !

Market Scenario	Mean Return	Mean Variance	Sharpe
$\mu = -50\%, \sigma = 10\%$	1.395	0.005	5.181
$\mu = -30\%, \sigma = 10\%$	1.391	0.014	3.195
$\mu = -10\%, \sigma = 10\%$	1.333	0.071	1.249
$\mu = 0\%, \sigma = 10\%$	No convergence	/	/
$\mu = 10\%, \sigma = 10\%$	No convergence	/	/
$\mu = 30\%, \sigma = 10\%$	1.388	0.019	2.785
$\mu = 50\%, \sigma = 10\%$	1.396	0.006	4.739
$\mu = -50\%, \sigma = 20\%$	1.387	0.023	2.538
$\mu = -30\%, \sigma = 20\%$	1.356	0.051	1.579
$\mu = -10\%, \sigma = 20\%$	No convergence	/	/
$\mu = 0\%, \sigma = 20\%$	No convergence	/	/
$\mu = 10\%, \sigma = 20\%$	No convergence	/	/
$\mu = 30\%, \sigma = 20\%$	1.343	0.060	1.401
$\mu = 50\%, \sigma = 20\%$	1.384	0.025	2.383
$\mu = -50\%, \sigma = 30\%$	1.366	0.043	1.751
$\mu = -30\%, \sigma = 30\%$	1.321	0.094	1.044
$\mu = -10\%, \sigma = 30\%$	No convergence	/	/
$\mu = 0\%, \sigma = 30\%$	No convergence	/	/
$\mu = 10\%, \sigma = 30\%$	No convergence	/	/
$\mu = 30\%, \sigma = 30\%$	1.310	0.105	0.956
$\mu = 50\%, \sigma = 30\%$	1.360	0.047	1.649
$\mu = -50\%, \sigma = 40\%$	1.343	0.065	1.346
$\mu = -30\%, \sigma = 40\%$	1.307	0.132	0.845
$\mu = -10\%, \sigma = 40\%$	No convergence	/	/
$\mu = 0\%, \sigma = 40\%$	No convergence	/	/
$\mu = 10\%, \sigma = 40\%$	No convergence	/	/
$\mu = 30\%, \sigma = 40\%$	No convergence	/	/
$\mu = 50\%, \sigma = 40\%$	1.342	0.073	1.261

Table 1: Annualized Sample Mean ($\overline{M}$), Variance ($\overline{V}$), and Sharpe Ratio (SR) for our EMV implementation.

B Some doubts on decaying lambda formula

By using the formula of the paper we obtain this shape for λ_k



We observed that using this

$$\lambda_k = \lambda_0 \left(1 - \exp\left(\frac{(k-M)}{M}\right)\right)$$

draw a good decaying shape :