Reinforcement learning for portfolio optimization

Kristoffer Andersson

Centrum Wiskunde & Informatica

Bologna,
September 29, 2022

Table of contents

1. Motivation

2. Proposed reinforcement learning method

3. Summary

What are the benefits with the proposed method

Very general asset dynamics, for example:

- Financial time series from real world;
- Data from complex multi-factor scenario generator;
- Any SDE (jumps, transaction costs, different lending and borrowing rates, high dimensional, driven by fBM etc.).

Does not rely on the dynamic programming principle to hold - Very general target function:

- Mean-variance type problems;
- Beyond MV, *e.g.*, only penalize downside risk.

Portfolio optimization from a stochastic control perspective

Wealth dynamics described by a controlled SDE

$$X_t^u = x_0 + \int_0^t b(t, X_t^u, u_t)\mathrm{d}t + \int_0^t \sigma(t, X_t^u)\mathrm{d}W_t.$$

$X^u = (X_t^u)_{t\in[0,T]}$ state of the system, $u = (u_t)_{t\in[0,T]}$ control of the system, taking on values in $\mathbb{R}^d$ and $U \subset \mathbb{R}^\ell$, respectively.

To measure performance of the control, a **cost functional** is used

$$J^u(t,x) = \mathbb{E}\big[g(X_T^u)\,|\,X_t = x\big].$$

The **control problem** is to find a control $u \in \mathcal{U}_{[0,T]}$($:=$ set of admissible controls) such that the cost functional is minimized.

Assuming the infimum is attainable, we define a **value function** as

$$V(t,x) = \inf_{u\in\mathcal{U}_{[t,T]}} J(t,x;u),$$

and we consider u^* to be optimal if for $t \in [0, T]$

$$J(t,x;u^*) = V(t,x).$$

Dynamic Programming Principle

Recall

$$V(t,x) = J(t,x;u^*) = \mathbb{E}\big[g(X_T^{u^*})\,|\,X_t^{u^*} = x\big].$$

Then (under conditions on g) by the law of iterated expectations we have

$$\mathbb{E}\big[g(X_T^{u^*})\,|\,X_t^{u^*} = x\big] = \mathbb{E}\Big[\mathbb{E}\big[g(X_T^{u^*})\,|\,X_{t+\Delta t}^{u^*}\big]\,\big|\,X_t^{u^*} = x\Big]$$

and in turn we have (one version of) the **Dynamic programming principle (DPP)**:

$$V(t,x) = \mathbb{E}\Big[V\big(t+\Delta t, S_{t+\Delta t}^{u^*}\big)\,|\,X_t^{u^*} = x\Big],$$

for $\Delta t \in [0, T-t]$[1].

Question: Why is it important that the DPP holds?

[1] From now on skip superscript and denote $X_t = X_t^{u^*}$

Dynamic programming - backward induction

Answer: When the DPP holds, it is possible to start with the terminal wealth of the portfolio and solve for the optimal strategy backwards in time.

Recall

$$V(tx) = \mathbb{E}\Big[V\big(t+\Delta t, X^{u^*}_{t+\Delta t}\big) \,|\, X^{u^*}_t = x\Big].$$

Therefore, by setting $t = T - \Delta t$ we have

$$V(T-\Delta t, x) = \mathbb{E}\big[V(T, X_T) \,|\, X_{T-\Delta t} = x\big] = \mathbb{E}\big[g(X_T) \,|\, X_{T-\Delta t} = x\big].$$

The problem then boils down to computing conditional expectations which can be done with *e.g.*, regression, Fourier methods, PDE or FBSDE approaches etc..

In summary: When the DPP holds, there are many available methods to solve the problem at hand.

Requirements for the DPP

Question: When does the DPP hold?
The terminal wealth $g(X_T)$ should be linear in the sense of conditional expectations.

DPP **holds**:

- (logarithmic utility) $g(X_T) = -\ln(X_T)$;
- (exponential utility) $g(X_T) = e^{-aX_T}$.

DPP **fails to hold** (here we allow g to depend also on the law of X_T):

- (mean-variance) $g(X_T, \mathcal{L}[X_T]) = -\mathbb{E}[X_T] + \lambda \mathrm{Var}[X_T]$;
- (mean-quantile) $g(\mathcal{L}[X_T]) = -\mathbb{E}[X_T] + \mathrm{Quantile}_q(X_T)$;
- Any general function depending on the law of X_T (except a linear combination of expectations).

When the DPP fails to hold almost all existing methods breaks down since backward induction can no longer be used.

Idea based on

- *Deep learning approximation for stochastic control problems* J Han, W E. Advances in Neural Information Processing Systems, Deep Reinforcement learning workshop
- *Convergence of a robust deep FBSDE method for stochastic control* K Andersson, A Andersson, CW Oosterlee arXiv preprint arXiv:2201.06854

Problem setup

Recall (generalized) the problem

$$\begin{cases} \inf\limits_{u \in \mathcal{U}} g\big(X_T, \mathcal{L}[X_T]\big), \quad \text{subject to} \\ X_T = x_0 + \int_0^T b(t, X_t, u_t)\mathrm{d}t + \int_0^T \sigma(t, X_t)\mathrm{d}W_t. \end{cases}$$

Three problems to solve, preferably in the following order:

(1) Approximation of controlled SDE (or any other process);

(2) Approximation of the law of X_T;

(3) Optimize objective function.

Sketch of method

Assuming for now that the asset process is described by an SDE.

(1) Euler–Maruyama is used to discretize the SDE. Leads to a **discretization error**;

(2) The law of X_T is approximated by the empirical law of a discrete approximation of X_T. Leads to an **approximation error**.

These two errors can easily be controlled as long as we can prove that the continuous control problem converges to the discrete counterpart (**highly non-trivial**).

Done in the context of FBSDEs without deoendency on the law of X_T in: *Convergence*

of a robust deep FBSDE method for stochastic control
K Andersson, A Andersson, CW Oosterlee arXiv preprint arXiv:2201.06854

(3) Here we would rely on a neural network which would result in a reinforcement learning algorithm of. This would lead to two errors; one optimization error, and one approximation error (approximation capacity of a neural network).

These errors are difficult to treat especially the optimization error. For the approximation error we need rely on **the universal approximation theorem**.

Summary

- Motivate why a method for general portfolio optimization is desirable - Existing methods are very limited in terms of asset dynamics and target;
- Based on experience, good chances that such algorithm would be stable and accurate;
- Numerical analysis could be carried out, very challenging and interesting from a mathematical perspective.

Thanks for your attention!