Optimal Execution via Multi-Objective Multi-Armed Bandits (Student Abstract)

Francois Buet-Golfouse[1], **Peter Hill** [2]

[1] University College London
[2] Independent Researcher
ucahfbu@ucl.ac.uk, peterhill96@gmail.com

Abstract

When trying to liquidate a large quantity of a particular stock, the price of that stock is likely to be affected by trades, thus leading to a reduced expected return if we were to sell the entire quantity at once. This leads to the problem of optimal execution, where the aim is to split the sell order into several smaller sell orders over the course of a period of time, to optimally balance stock price with market risk. This problem can be defined in terms of difference equations. Here, we show how we can reformulate this as a multi-objective problem, which we solve with a novel multi-armed bandit algorithm.

Introduction

We consider the *optimal execution* problem. Contrary to (Almgren and Chriss 2001), we consider optimal execution as a *multi-objective* problem, whereby an agent sells or buys a large amount of shares (thus maximising their profit), while minimising the adverse price movements that are a consequence of their own trades, over T steps. Here, the agent starts with an inventory Q_0 (to be liquidated) of a stock whose price is given by S_t at time t and chooses an amount a_t of shares to trade at each time-step, leading to a (running) cash profit X_t. We wish to find an optimal trading strategy trading off both objectives. Reformulating this as a multi-armed bandit (MAB) problem (Cannelli et al. 2020)[1]. allows us to include risk appetites in the model. [2]

[1] We consider this as a MAB problem as this is an *online learning* algorithm, which is important for trading during the course of the day. (Cannelli et al. 2020) also shows that bandits can outperform RL in practice.

[2] This paper was prepared for informational purposes by the authors, and is not a product of any institution's Research Department. The views expressed therein are solely those of the authors and do not reflect those of any institution or employer, past and present. The authors make no representation and warranty whatsoever and disclaim all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

Objectives

The *expected return*, $\mathbb{E}[R_{0:T}]$, to be maximised, consists of the proceeds of the liquidation and the present value of the remaining inventory at the time horizon T: $R_{0:T} := X_T + Q_T S_T$. The *expected cost*, $\mathbb{E}[C_{0:T}]$, to be minimised, penalises a non-zero inventory, both at T and intermediate steps t: $C_{0:T} := Q_T^2 + \phi \sum_{t=0}^{T-1} Q_t^2$, with $\phi > 0$.

Model Dynamics

At each time-step $t = 0, \cdots, T-1$, the change in quantities of interest is given by:

$$Q_{t+1} - Q_t := \Delta Q_t = -a_t \quad (1)$$
$$X_{t+1} - X_t := \Delta X_t = S_t a_t - k a_t^2 \quad (2)$$
$$S_{t+1} - S_t := \Delta S_t = -b a_t + \sigma \epsilon_{t+1}, \quad (3)$$

with $k, b, \sigma > 0$ model parameters representing, respectively, temporary, permanent impacts, and volatility risk; ϵ_{t+1} represents a zero-mean random shock happening between times t and $t+1$ (note they need not be iid).

Temporal Credit Assignment

A key step in our approach is to design step-wise rewards and costs, rather than focusing on terminal values. To do so, we write $R_{0:T} = \sum_{t=0}^{T-1} r_t$ and $C_{0:T} = \sum_{t=0}^{T-1} c_t$, where

$$r_t = (b-k)\, a_t^2 - b a_t Q_t + \sigma\, (Q_t - a_t)\, \epsilon_{t+1}$$
$$c_t = a_t^2 - 2 Q_t a_t + \phi Q_t^2.$$

Multi-armed Bandits

In a MAB problem, we aim to maximise a reward function $f : \mathcal{X} \to \mathbb{R}$ by learning the optimal action $\mathbf{x} = \text{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$, based on noisy samples $y_t = f(\mathbf{x_t}) + \epsilon_t$ of the reward function. We consider $\mathcal{X} = \mathcal{A} \times \mathcal{Z}$, where $\mathcal{A}$ is the action space and $\mathcal{Z}$ is space of contextual information.

Multi-objective In a multi-objective setting, we consider a reward function, $f : \mathcal{X} \to \mathbb{R}^J$, where J is the number of different objectives that we include. We observe $\mathbf{y}_t = f(\mathbf{a}_t, \mathbf{z}_t) + \boldsymbol{\epsilon}_t$, where $\mathbf{y}_t = (y_t^{[1]}, \cdots, y_t^{[J]})^T \in \mathbb{R}^J$.

Approach	Single Step	Multi-Step
Action Space	$\{[a_t]\}$	$\{[a_1, \cdots, a_T]\}$
Reward Function $f : \mathcal{X} \to \mathbb{R}^2$	$\mathcal{X} = \mathbb{R} \times Z \times U$ $f(x_t) = [r_t, -c_t]^T$	$\mathcal{X} = \mathbb{R}^T \times Z \times U$ $f(x_t) = [R_{0:T}, -C_{0:T}]$
Cumulative Reward	$\sum_{t=5(d-1)}^{5d} \tilde{f}(x_t, u_t)$	$\tilde{f}(x_d, u_d)$

Table 1: Single-Step vs Multi-Step Approach

Preferences We consider the space of *user preferences*, $\mathcal{U}$, in which the user considers the importance of different objectives. An example could be $\mathcal{U} = \mathbb{S}_{k-1} := \{\mathbf{a} \in \mathbb{R}^k : a_j \geq 0, \sum_j a_j = 1\}$. We define a *surrogate reward function* $\tilde{f} : \mathcal{A} \times \mathcal{U} \to \mathbb{R}$, which is a preference-dependent single valued reward. For example, $\tilde{f}(a, u) = a^T u$.

Observability We assume that $\tilde{y}_t = \tilde{f}(x_t) + \epsilon_t$ is always observable, but we could also observe extra information. We wish to use all the information we have to update our model.

Gaussian Processes

Gaussian Processes (GP's) (Rasmussen 2003) are a flexible, non-parametric way to model the reward function, f, i.e., $f \sim GP(\boldsymbol{\mu}(\cdot), \mathbf{K}(\cdot, \cdot))$. The predictive distribution of a Gaussian process regression model has a closed-form solution, thus we have simple closed form expressions for updating $\boldsymbol{\mu}$ and $\mathbf{K}$ when new information is available.

CGP-UCB-MO

The Multi-Objective Contextual Gaussian Process Upper Confidence Bound (CGP-UCB-MO) algorithm adapts the existing CGP-UCB algorithm (Krause and Ong 2011) to the multi-objective setting. We embed the space $\mathcal{X} = \mathcal{A} \times \mathcal{Z}$ into a larger space $\bar{\mathcal{X}} = \mathcal{X} \times \mathcal{U} \times \mathcal{J}$, where $\mathcal{J}$ is a space allowing for extra information to be included. We can then define a new function $\bar{f} : \bar{\mathcal{X}} \to \mathbb{R}$ and model this reward function using a Gaussian Process, as in the single-objective setting. We can choose our next action based on the following:

$$\mathbf{a}_t := \text{argmax}_{\mathbf{a} \in \mathcal{A}} \mu_{t-1}(\mathbf{a}, \mathbf{z}_t, \mathbf{u}_t) + \sqrt{\beta_t} \sigma_{t-1}(\mathbf{a}, \mathbf{z}_t, \mathbf{u}_t).$$

Bandits for Optimal Execution

We trade off expected return $R_{0:T}$ with market risk $C_{0:T}$ through multi-objective MABs. We can thus account for the user's risk appetite, through their preferences.

Single step vs multi step In the *single step* approach, we sequentially observe our reward at each discrete time step. In the *multi-step* approach, we consider a global optimisation over all time steps up until time T. Table 1 shows a comparison of the approaches, and Algorithm 1 shows the single-step approach. In both cases, we consider a surrogate reward function of the form $\tilde{f}(x_t, u_t) = f(x_t)^T u_t$. For each day, we also consider the *cumulative reward*, $R^{[d]}$, across day d, in order to allow us to compare the techniques.

Preference	$[1, 0]$	$[0.75, 0.25]$	$[0.5, 0.5]$	$[0.25, 0.75]$	$[0, 1]$
Mean	0.202	0.095	0.088	0.055	-0.039
Median	0.222	0.088	0.064	0.072	-0.024

Table 2: Mean reward difference between multi-step and single step approach over final 10 days.

Algorithm 1: Optimal Execution Single Step Algorithm

Require: Q_0, S_0, context $\mathbf{z_t}$, preference vector $\mathbf{u}_t$, Discrete time intervals per day T.
for $t = 1, \cdots, T$ **do**
  Run CGP-UCB-MO to find action a_t and observe surrogate reward $\tilde{f}(a_t, u_t)$ subject to $a_t < Q_t$.
  Update $Q_{t+1} = Q_t - a_t$
end for
Calculate $R^{[d]}$ based on Table 1
Reset Q_t to Q_0 and S_t to S_0

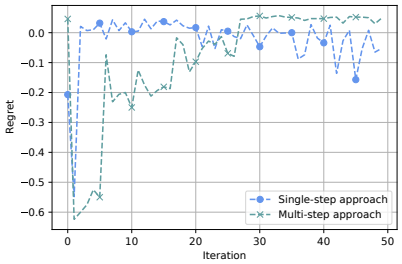

Figure 1: Reward over time for $u = [0.75, 0.25]$

Experimentation

We consider 5 discrete time steps in a day ($T = 5$ in the multi-step approach). Each day, $Q_0 = 1$, $S_0 = 1,000$. We consider different preference vectors, but fixed parameters ($b = 0.1, k = 1, \alpha = 1, \phi = 0.1$). Figure 1 shows the change in $R^{[d]}$ for each approach ($u = [0.75, 0.25]$). The multi-step approach reaches a better solution but is slower. Table 2 also shows this with different preference vectors.

References

Almgren, R.; and Chriss, N. 2001. Optimal execution of portfolio transactions. *Journal of Risk*, 3: 5–40.

Cannelli, L.; Nuti, G.; Sala, M.; and Szehr, O. 2020. Hedging using reinforcement learning: Contextual k-Armed Bandit versus Q-learning. *arXiv preprint arXiv:2007.01623*.

Krause, A.; and Ong, C. 2011. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24.

Rasmussen, C. E. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*, 63–71. Springer.