Blockhouse Quantitative Research Intern Work Trial Task

Chi-Lin (Jaden) Li

June 2024

1 Optimal Execution of a VWAP Order: A Stochastic Control Approach

1.1 Introduction

The objective of the paper is to develop a strategy for optimal execution of a stock position benchmarked against the Volume Weighted Average Price (VWAP). The goal is to minimize both the mean and variance of order slippage relative to VWAP.

VWAP Definition:

$$\text{VWAP} = \frac{\sum_{i=1}^{N} V_i P_i}{\sum_{i=1}^{N} V_i}$$

where V_i and P_i are the volume and price at the i-th trade.

1.2 Model Formulation

Trading Costs

In the model formulation, the asset price follows an arithmetic Brownian motion, expressed as

$$P(t) = P(0) + \sigma W(t)$$

where σ represents daily volatility, and $W(t)$ is a standard Brownian motion. The actual price paid, including temporary market impact, is modeled as

$$P_u(t) = P(t) + \kappa u(t)$$

where κ is the market impact coefficient and $u(t)$ is the trading rate. The total expenditure for buying Y shares over time is given by

$$TE_u = YP(0) + \sigma \int_0^T W(t)u(t)\,dt + \kappa \int_0^T u^2(t)\,dt$$

The VWAP benchmark is approximated in continuous time as

$$\text{VWAP} \approx \frac{\int_0^T P(t)\,d\tilde{V}(t)}{\tilde{V}(T)}$$

with the relative volume curve modeled by a gamma bridge

$$\gamma(t) = \frac{L(t)}{L(T)}$$

1.3 Optimization Problem

The paper formulates the optimization problem as a mean-variance trade-off to minimize expected slippage and its variance. Slippage is defined as

$$\text{slip}_u = TE_u - \text{VWAP}$$

The objective function is formulated to balance trading costs and the variance of deviation from the volume curve:

$$\min_u \left[\kappa E\left[\int_0^T u^2(t)\, dt \right] + \lambda\sigma^2 E\left[\int_0^T (\gamma(t) - X_u(t))^2\, dt \right] \right]$$

This formulation incorporates a quadratic trading cost term and a variance term to capture the uncertainty in the execution process.

1.4 Hamilton-Jacobi-Bellman (HJB) Equation

The value function

$$v(t, x, \gamma) = a(t)x^2 + b(t)\gamma x + c(t)x + d(t)\gamma^2 + f(t)\gamma + g(t)$$

represents the expected cost of executing an order from time t to the end of the trading period. The optimal trading rate

$$u^*(t) = -\frac{v_x(t, x, \gamma)}{2\kappa} = \frac{-2a(t)x + b(t)\gamma + c(t)}{2\kappa}$$

minimizes the value function. The optimal holdings

$$X_u^*(t) = xe^{-\frac{1}{\kappa}\int_t^T a(s)\, ds} - \frac{1}{2\kappa}\int_t^T (b(s)\gamma(s) + c(s))e^{-\frac{1}{\kappa}\int_s^T a(r)\, dr}\, ds$$

describe the optimal share position over time.

1.5 Parameter Estimation

To fit the model to empirical data, the paper uses a method of moments estimator for the relative volume curve

$$V(t) = at^b(1-t)^c$$

Parameters a, b, and c are estimated by minimizing the difference between empirical and theoretical moments:

$$\hat{a}, \hat{b}, \hat{c} = \arg\min_{a,b,c} \left[(M_1^{\text{empirical}} - M_1^{\text{theoretical}})^2 + (M_2^{\text{empirical}} - M_2^{\text{theoretical}})^2 \right]$$

where

$$M_1 = \int_0^1 V(t)\, dt$$

and

$$M_2 = \int_0^1 V(t)^2\, dt$$

This approach ensures the model accurately reflects observed trading volumes.

1.6 Simulation and Results

The paper validates the model using a 60-day rolling window for parameter estimation and assesses the goodness-of-fit with statistical tests like the z-test and Kolmogorov-Smirnov test. The simulation results demonstrate the model's accuracy in fitting empirical data and its effectiveness in minimizing slippage and trading costs. By providing a robust framework for optimal execution, the paper offers practical insights for improving trading strategies benchmarked against VWAP.

1.7 Summary of Key Equations

1. **Gamma Bridge**:
$$\gamma(t) = \frac{L(t)}{L(T)}$$
2. **Time Transformation**:
$$G(t) = at^3 + bt^2 + ct$$
3. **Value Function**:
$$v(t, x, \gamma) = a(t)x^2 + b(t)\gamma x + c(t)x + d(t)\gamma^2 + f(t)\gamma + g(t)$$
4. **Optimal Control**:
$$u^*(t) = \frac{-2a(t)x + b(t)\gamma + c(t)}{2\kappa}$$
5. **Optimal Holdings**:
$$X_u^*(t) = xe^{-\frac{1}{\kappa}\int_t^T a(s)\,ds} - \frac{1}{2\kappa}\int_t^T (b(s)\gamma(s) + c(s))e^{-\frac{1}{\kappa}\int_s^T a(r)\,dr}\,ds$$

2 Hierarchical Deep Reinforcement Learning for VWAP Strategy Optimization

In the dynamic world of financial markets, optimizing trade execution strategies is crucial for minimizing costs and maximizing returns. One of the commonly used execution strategies is the Volume Weighted Average Price (VWAP) strategy, which aims to execute trades in line with the market's volume distribution. This project explores the application of Hierarchical Deep Reinforcement Learning (HDRL) to enhance VWAP strategy optimization. By leveraging a hierarchical approach, we decompose the complex decision-making process into high-level and low-level controllers. The high-level controller sets strategic sub-goals, while the low-level controller executes specific trading actions to achieve these sub-goals. Through this structured methodology, the HDRL model can efficiently navigate the complexities of the trading environment, adapt to changing market conditions, and achieve optimal trade execution that closely aligns with the VWAP target. This innovative approach not only improves execution performance but also provides a scalable solution for real-time trading systems.

2.1 Problem Definition

Let T be the total trading period divided into N time intervals, with each interval t representing a decision point. The goal is to minimize the deviation from the VWAP, defined as:

$$\text{VWAP} = \frac{\sum_{t=1}^{N} P_t V_t}{\sum_{t=1}^{N} V_t} \tag{1}$$

where P_t is the price at time t and V_t is the volume traded at time t.

2.2 High-Level Controller

The high-level controller (meta-controller) sets sub-goals for the low-level controller based on the overall trading objective. It operates at a coarser time scale T_H.

State

$$S_H : \text{Cumulative volume traded, remaining time, market conditions, etc.}$$

Action

$$A_H : \text{Sub-goal, such as the target volume to be traded over the next } T_H \text{ intervals.}$$

Policy

$$\pi_H(S_H) = A_H$$

2.3 Low-Level Controller

The low-level controller executes actions to achieve the sub-goals set by the high-level controller. It operates at a finer time scale T_L.

State

$$S_L : \text{Immediate market features such as current price, volume, and order book information.}$$

Action

$$A_L : \text{Specific trading decisions, like the volume to trade in the next interval.}$$

Policy

$$\pi_L(S_L) = A_L$$

2.4 Reward Function

The reward function evaluates the performance of both controllers. For VWAP optimization, the reward can be defined as the negative deviation from the VWAP target.

High-Level Controller

$$R_H = -\left|\text{VWAP} - \frac{\sum_{t=1}^{N} P_t V_t}{\sum_{t=1}^{N} V_t}\right|$$

Low-Level Controller

$$R_L = -\left|\text{VWAP}_{target} - \frac{\sum_{t=t_0}^{t_1} P_t V_t}{\sum_{t=t_0}^{t_1} V_t}\right|$$

where t_0 to t_1 are the intervals within the sub-goal period.

2.5 Training the Model

Training involves learning the policies π_H and π_L using Reinforcement Learning algorithms such as Deep Q-Learning (DQN) or Proximal Policy Optimization (PPO).

High-Level Policy Update

$$\theta_H \leftarrow \theta_H + \alpha \nabla_{\theta_H} E[R_H]$$

Low-Level Policy Update

$$\theta_L \leftarrow \theta_L + \alpha \nabla_{\theta_L} E[R_L]$$

where θ_H and θ_L are the parameters of the high-level and low-level policies, respectively, and α is the learning rate.

2.6 Implementation Details

Feature Engineering

Extract relevant features for state representation, including historical price and volume data, order book depth, and market indicators.

Neural Network Architecture

Design separate neural networks for high-level and low-level policies, ensuring they can capture the complexities of their respective tasks.

Exploration vs. Exploitation

Implement strategies like epsilon-greedy or entropy regularization to balance exploration and exploitation during training.