Blockhouse Task

Name: Yasasvi Naidu Vennela

Modeling the Temporary Impact Function $g_t(x)$

In modern electronic markets, executing large orders can move prices unfavorably due to limited liquidity — this phenomenon is referred to as market impact or slippage. The aim of this task is to quantify the temporary market impact function $g_t(x)$ using limit order book (LOB) data, and understand how this function can inform optimal execution strategies.

Using one-minute snapshot data for three stock tickers (AMZN, MSFT, GOOG), I analyzed how the cost of immediate execution changes as a function of order size. The function $g_t(x)$ measures the additional cost per share over the mid-price when placing a market buy order of size x at time t.

Slippage Definition

Slippage quantifies the market impact of an order, reflecting how much the execution price deviates from the mid-price due to limited liquidity at the best quotes. At time t, the mid-price is calculated as:

$$\text{Mid}_t = \frac{\text{Best Ask}_t + \text{Best Bid}_t}{2} \tag{1}$$

For a buy order of x shares, one must consume available shares on the ask side of the LOB. Let the LOB have prices p_i with volumes v_i at each level. The total cost of purchasing x shares is:

$$\text{Cost}_t(x) = \sum_{i=1}^{k} p_i \cdot v_i \quad \text{where} \quad \sum_{i=1}^{k} v_i = x \tag{2}$$

Then the temporary impact function becomes:

$$g_t(x) = \frac{\text{Cost}_t(x)}{x} - \text{Mid}_t \tag{3}$$

Convexity and Empirical Behavior

The function $g_t(x)$ is empirically convex — larger orders access worse prices deeper in the book, increasing the average execution cost. This shape was evident in all three stocks, although the steepness varied. For highly liquid stocks, the slippage curve grows gradually; for thinly traded ones, slippage increases sharply even at moderate order sizes.

Modeling and Validation

To model $g_t(x)$, I first considered a linear approximation:

$$g_t(x) = \beta_t x \tag{4}$$

However, this failed to capture the convexity seen in real LOBs. A better alternative was the power-law form:

$$g_t(x) = \alpha_t x^\delta \tag{5}$$

I fitted this model at each time t using slippage values computed from simulated market orders of varying sizes (e.g., 10–300 shares). The parameters α_t and δ_t were estimated using least-squares curve fitting. Typically, I observed δ_t between 1.2 and 1.5, confirming convex behavior.

Visualization Insights

Plots of $g_t(x)$ vs x confirmed strong convexity, particularly for larger trades. For AMZN and MSFT, which are more liquid, the increase in slippage was smoother. For GOOG, steeper curves were observed, indicating more sensitivity to trade size. This validates that a single global model is insufficient — market impact must be estimated dynamically and per stock.

Conclusion and Link

This modeling framework captures how trade size influences execution cost and forms the foundation for optimal allocation. Parameters α_t and δ_t were estimated from the LOB data at each minute, enabling construction of a temporally resolved impact model.

GitHub Link:

https://github.com/YasasviNaidu/Blockhouse-impact-model

Mathematical Framework for Share Allocation

We are required to purchase S shares across $N = 390$ intervals, such that total temporary impact is minimized.

Let $x = [x_1, x_2, ..., x_N]$ be our allocation vector. The goal is to solve:

$$\min_{x_1+\cdots+x_N=S,\, x_i\geq 0} \sum_{i=1}^{N} g_i(x_i) \cdot x_i \tag{6}$$

Objective Interpretation

The term $g_i(x_i) \cdot x_i$ represents the total slippage cost for executing x_i shares at time i. Minimizing the sum ensures that total execution cost stays as close to the mid-price as possible.

Convex Optimization Formulation

Assuming $g_i(x) = \alpha_i x^\delta$, we get:

$$\min_{x_i\geq 0,\, \sum x_i=S} \sum_{i=1}^{N} \alpha_i x_i^{1+\delta} \tag{7}$$

This is a convex optimization problem. The parameters α_i and δ can be estimated for each interval by fitting historical LOB data.

Heuristic Algorithm

If future LOBs are unknown, we adopt an online heuristic:

- At each time t, estimate $g_t(x)$ by simulating the execution against current LOB.
- Allocate x_t proportionally to favorable liquidity (e.g., inverse of $g_t'(x)$).
- Adjust future allocations based on remaining volume $S_t = S - \sum_{i<t} x_i$.

Conclusion

This framework offers a data-driven, flexible approach for minimizing market impact while ensuring execution. Future improvements could include volatility-aware models, inventory risk terms, or reinforcement learning-based dynamic policies.