Adaptive Execution with Online Price Impact Learning

Beomsoo Park
Electrical Engineering, Stanford University
beomsoo@stanford.edu

December 11, 2009

1. Introduction

Buying or selling a large block of security is often followed by unfavorable movement of price which is called price impact. One reason for the impact is that the block execution causes abrupt imbalance between supply and demand and the other is that it might convey to other investors information about fundamental value of the security that will be reflected on their future investment decisions. Thus, when submitting large orders, it is important to take into account the price impact in order to minimize the amount of value lost by it.

We propose an efficient multi-period execution algorithm to minimize transaction cost incurred by the price impact when buying or selling a block order. We assume that we have incomplete knowledge about the price impact which is indeed the case in practice. A good execution algorithm strikes a balance between "exploration" and "exploitation." That is, on one hand, it needs to learn unknown characteristics of the price impact for better future trading decisions in exchange of losing optimality at present. On the other hand, it should make the best of current knowledge about the price impact for making an optimal decision that is not necessarily effective for identifying remaining uncertainty about the impact.

More precisely, this problem can be formulated as a dynamic program that has random variables representing uncertainty on the price impact. But it is quite challenging to solve the corresponding Bellman equation due mainly to need for incorporating probability distributions on the random variables into a state space. Also, difficulty comes from the fact that both state and action spaces are continuous and that in most practical cases a trading horizon is finite, say a few days or a week. Therefore, we seek to propose a reasonably good, simple heuristic strategy that captures a good balance between exploration and exploitation and compare its performance with that of a naive baseline strategy to be defined later and an upper bound derived through information relaxation. To this end, we propose a *linearized least squares with regularization* that is a modified version of least squares with regularization dealing efficiently with nonlinear relationship between observations and model parameters.

2. Problem Formulation

Consinder a trader who wants to liquidate a large long position or to recover a large short position of a stock over a finite time horizon T. Let x_t denote the size of his ex-trade position at period t with an initial position x_0 such that a positive value implies a long position. He requires that his final position x_T be zero. He is assumed to be risk averse such that he seeks to maximize the objective function of the form

$$\mathsf{E}\left[\sum_{t=1}^{T}\left\{\Delta p_t x_{t-1} - \rho\sigma_\epsilon x_{t-1}^2\right\}\Big|\, x_0\right]$$

where $\Delta p_t \equiv p_t - p_{t-1}$ is defined as the increment of a per-share transaction price, which will be defined later. The first term in the sum represents the change in book value that can be viewed as a per-stage revenue. The second term reflects a holding cost, with $\rho\sigma_\epsilon$ expressing the extent to which the trader would execute sooner rather than later. Note that the risk aversion coefficient is proportional to the volatility of the stock price σ_ϵ.

We model a natural price movement, defined as evolution over time of the price without the trader's transactions, as a Gaussian random walk, i.e. $\tilde{p}_t = \tilde{p}_{t-1} + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$. In practice, due to presence of the price impact is the trader faced with a less favorable per-share transaction price at which he sells or buys a share. In order to capture this unfavorable movement of the price, we propose the following price impact model with n "time constants".

$$p_t = \tilde{p}_t + \mathbf{1}^\top y_t,\ \ y_t = Ay_{t-1} + \gamma u_t \in \mathbb{R}^n,\ \ A = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n \end{bmatrix},\ \ \gamma = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{bmatrix},\quad 0 \le \alpha_i \le 1,\ \ \gamma_i > 0\ \ \forall i$$

where u_t represents the size of market order submitted by the trader at period t and the pair (α_i, γ_i) characterizes one market impact component. Note that the per-share transaction price p_t is a function of u_t. For $\alpha_i = 0$, trading u_t affects the transaction price only at stage t and we call this impact an immediate impact. For $0 < \alpha_i < 1$, the excitation triggered by u_t dies away as time goes by and we call this impact a temporary impact. For $\alpha_i = 1$, the effect of u_t on the transaction price persists and we call this impact a permanent impact. In this paper, we consider the case in which the coefficients α_i's and γ_i's are unknown to the trader. Instead, we assume that he has (usually inaccurate) estimates for the parameters through data analysis on historic transaction records.

To sum up, the trader solves the following finite horizon control problem:

$$\begin{array}{ll} \max\limits_{\pi} & \mathsf{E}\left[\sum\limits_{t=1}^{T} \left\{\Delta p_t x_{t-1} - \rho\sigma_\epsilon x_{t-1}^2\right\} \Big| x_0\right] \\ \text{subject to} & \tilde{p}_t = \tilde{p}_{t-1} + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2), \quad p_t = \tilde{p}_t + \mathbf{1}^\top y_t, \\ & y_t = Ay_{t-1} + \gamma u_t,\ y_0 = \vec{0} \in \mathbb{R}^n, \quad x_t = x_{t-1} + u_t,\ x_T = 0. \end{array}$$

where the policy $\pi = (\pi_1, \pi_2, \ldots)$ is a collection of functions π_t each of which maps history up to time t to a trading decision u_t. For notational simplicity, define $z_t^i = \sum_{j=1}^{t} \alpha_i^{t-j} u_j$ and $z_0^i = 0$ for all i. Then, it is easy to see that z_t^i satisfies the recursion $z_t^i = \alpha_i z_{t-1}^i + u_t$ and $y_t^i = \gamma_i z_t^i$ for all i.

3. Analysis

3.1. Bellman Equation with Augmented State

Let $\mathcal{H}_t$ be history up to period t. From Bayesian perspective, together with some prior distributions on α and γ, Bellman equation for this problem can be written as

$$V_t(x_{t-1}, \mathcal{H}_t) = \max_{u_t}\ \mathsf{E}\left[\Delta p_t x_{t-1} - \rho\sigma_\epsilon x_{t-1}^2 + V_{t+1}(x_t, \mathcal{H}_{t+1}) \Big| \mathcal{H}_t\right].$$

But it is quite challenging to get a closed-form solution for V_t. Therefore, we aim to find an approximate solution via policy parameterization. One might want to think of value function approximation as an alternative solving methodology. In this particular problem, we prefer the former to the latter by the following reasons: first, a class of well-structured policies are readily available from analysis of clairvoyant case which will be done in the subsequent section. Moreover, it is impossible to simulate price impact without actually trading and to get samples.[1] Finally, we need to learn the model parameters in an online fashion over a relatively short time horizon and thus we should take advantage of special structural properties of this execution model that the clairvoyant policy can capture effectively.

[1] Some people call this Heisenberg Uncertainty Principle of Price Impact

3.2. Policy Parameterization

In order to derive an upper bound for the trader's profit and a class of policies parameterized by α and γ, consider the case in which both α and γ are known. We seek to find reward-to-go functions $V_t(x_{t-1}, z_{t-1}; \alpha, \gamma)$ that satisfy the following Bellman equation.

$$V_t(x_{t-1}, z_{t-1}; \alpha, \gamma) = \max_{u_t \in \mathbb{R}} \left\{ \gamma^\top (\mathbf{1} u_t - (I - A) z_{t-1}) x_{t-1} - \rho \sigma_\epsilon x_{t-1}^2 + V_{t+1}(x_{t-1} + u_t, A z_{t-1} + \mathbf{1} u_t; \alpha, \gamma) \right\}$$

with the terminal condition $V_T(x_{T-1}, z_{T-1}; \alpha, \gamma) = -(\gamma^\top \mathbf{1} + \rho\sigma_\epsilon) x_{T-1}^2 - \gamma^\top (I - A) z_{T-1} x_{T-1}$. It is natural to conjecture that $V_t(x_{t-1}, z_{t-1}; \alpha, \gamma)$ is quadratic in the two arguments x_{t-1} and z_{t-1}, that is, $V_t(x_{t-1}, z_{t-1}; \alpha, \gamma) = B_t x_{t-1}^2 + C_t^\top x_{t-1} z_{t-1} + z_{t-1}^\top D_t z_{t-1}$. In order to determine the coefficients $B_t \in \mathbb{R}$, $C_t \in \mathbb{R}^n$, and $D_t \in \mathbb{R}^{n \times n}$, we need to solve the following equation for them.

$$\begin{aligned} B_t x_{t-1}^2 + C_t^\top z_{t-1} x_{t-1} + z_{t-1}^\top D_t z_{t-1} = \max_{u_t} \Big\{ & \gamma^\top (\mathbf{1} u_t - (I - A) z_{t-1}) x_{t-1} - \rho\sigma_\epsilon x_{t-1}^2 \\ & + B_{t+1}(x_{t-1} + u_t)^2 + C_{t+1}^\top (A z_{t-1} + \mathbf{1} u_t)(x_{t-1} + u_t) + (A z_{t-1} + \mathbf{1} u_t)^\top D_{t+1} (A z_{t-1} + \mathbf{1} u_t) \Big\} \end{aligned}$$

The second-order optimality condition is $B_{t+1} + C_{t+1}^\top \mathbf{1} + \mathbf{1}^\top D_{t+1} \mathbf{1} < 0$. Comparing the coefficients on both sides, we obtain the following set of three recursive equations:

$$\begin{aligned} B_t &= B_{t+1} - \rho\sigma_\epsilon - \frac{(2B_{t+1} + C_{t+1}^\top \mathbf{1} + \gamma^\top \mathbf{1})^2}{4(B_{t+1} + C_{t+1}^\top \mathbf{1} + \mathbf{1}^\top D_{t+1} \mathbf{1})}, \\ C_t^\top &= C_{t+1}^\top A - \gamma^\top (I - A) - \frac{(2B_{t+1} + C_{t+1}^\top \mathbf{1} + \gamma^\top \mathbf{1})(C_{t+1}^\top + 2(\mathbf{1}^\top D_{t+1}))A}{2(B_{t+1} + C_{t+1}^\top \mathbf{1} + \mathbf{1}^\top D_{t+1} \mathbf{1})}, \\ D_t &= A^\top \left(D_{t+1} - \frac{(C_{t+1} + 2(D_{t+1}\mathbf{1}))(C_{t+1}^\top + 2(\mathbf{1}^\top D_{t+1}))}{4(B_{t+1} + C_{t+1}^\top \mathbf{1} + \mathbf{1}^\top D_{t+1} \mathbf{1})} \right) A. \end{aligned}$$

with terminal condition $B_T = -(\gamma^\top \mathbf{1} + \rho\sigma_\epsilon)$, $C_T^\top = -\gamma^\top (I - A)$ and $D_T = \vec{0}$. Note that each B_t, C_t and D_t are functions of α and γ. The upper bound for the profit is given by $V_1(x_0, z_0; \alpha, \gamma)$.

3.3. Parameter Estimation

Two primary difficulties are present for estimating the price impact parameters α and γ. One is the absence of offline training examples and the other is highly nonlinear, nonconvex relationship between observations and the model parameters. Especially, due to the second reason we cannot directly apply ordinary least squares into the estimation problem of interest. In light of this, we propose *linearized least squares with regularization* to address the technical difficulties. Let us highlight two important features of this algorithm. One is that this algorithm converts the nonlinear observation process into a linear one using iterates in the previous stage so that it preserves efficiency of ordinary least squares. The other is that it deals effectively with the constraints for α and γ, i.e. $0 \le \alpha \le 1$ and $\gamma > 0$, using regularization while the solution of ordinary least squares usually violates the range constraints. Algorithm 1 gives the details of the linearized least squares with regularization.

Let us finish this section with comparison of Algorithm 1 with two possible alternatives: extended Kalman filtering and nonlinear least squares. The extended Kalman filtering suffers from the "out-of-range" problem related to the range constraints for α and γ and requires a strong probabilistic assumption that the prior distributions on α and γ be Gaussian. Meanwhile, nonlinear least squares can avoid these problems but it turns out that it converges much more slowly than Algorithm 1.

4. Numerical Experimentation

For numerical experimentation, we use the following setting for the model parameters.

- $T = 100$ for 1 trading day (6.5 hours), the lentgh of one trading interval ≈ 4 mins
- Daily volatility = \$0.25 (two ticks), i.e. $\sigma_\epsilon = \$0.25/\sqrt{T}$
- Risk aversion coefficient: $\rho = 5 \times 10^{-5}$
- Initial position $x_0 = 100,000$ shares
- True parameter values: $\alpha^* = (0.259, 0.509, 0.713, 0.874)$, $\gamma^* = (3, 4, 5, 5) \times 10^{-5}$
- Initial estimates: $\alpha^{(0)} = (0.406, 0.763, 0.874, 0.973)$, $\gamma^{(0)} = (6, 7, 8, 10) \times 10^{-5}$

The choice of (α^*, γ^*) and $(\alpha^{(0)}, \gamma^{(0)})$ represents the case where today's market is more liquid than we expected yesterday. For the purpose of comparison, we define a naive baseline policy as the one using the same initial estimates for α and γ over the entire time horizon, i.e. no adaptation for α and γ. It is based on the hypothetical notion that daily change of α and γ is so small that it can be ignored.

We evaluate performances of our adaptive execution policy(AE) and the baseline policy(BL) by two performance measures: percentage performance loss relative to the clairvoyant case(CL) and percentage performance gain relative to the baseline policy. We carry out 40 simulation runs for averaging. The following summerizes the results.

- Average performance percentage loss of BL relative to CL: -11.4%
- Average performance percentage loss of AE relative to CL: -0.62%
- Average performance percentage gain of AE over BL: 9.68%
- Percentage sample deviation from the CL value function = -0.24%

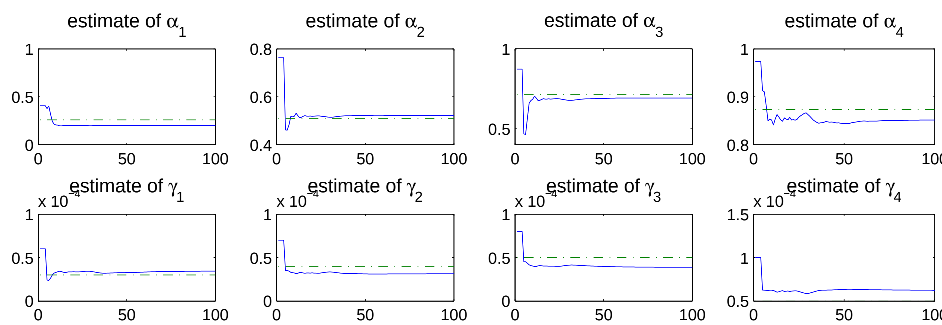

Figure 1: Evolution over time of estimates for α (above) and γ (below)

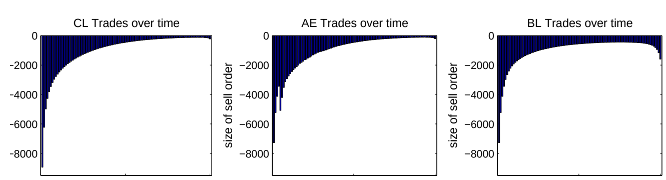

Figure 2: Evolution over time of size of order: (i) clairvoyant case(left), (ii) adaptive execution(middle), (iii) baseline policy (right)

5. Conclusion

We quantify contribution of the adaptive execution policy adopting online learning of the unknown price impact coefficients relative to the naive baseline policy with no adaptation and we show that the performance gain from adaptation is significant. This execution model fits for intraday execution in the situation where price impact remains unchanged within a single trading day but changes daily. One might want to argue the presence of intraday change of price impact but it is quite challenging to track the time-varying aspect because of lack of data set for training. Hopefully, the price impact model adopted here can capture practical intraday price impact pattern reasonably well *on average* and our adaptive execution policy can make a contribution to reduce transaction costs incurred by price impact. Future work includes (i) theoretical justification for choice of regularization coefficient and (ii) exploration for a better estimate over an uncertainty ellipsoid centered at the linearized least squares estimate that has been studied in multi-armed bandit literature.

References

1. D. Bertsimas and A. W. Lo, 1998. Optimal control of execution costs. Journal of Financial Markets

2. A. Obizhaeva and J. Wang, 2005. Optimal trading strategy and supply/demand dynamics. Revised and resubmitted. Journal of Financial Markets

3. C. C. Moallemi, B. Park, and B. Van Roy, 2009. Strategic execution in the presence of an uninformed arbitrageur. submitted for publication.

4. D. Bertsekas, 2007. Dynamic programming and optimal control.

5. R. Sutton and A. Barto, 1998. Reinforcement learning: an introduction.

Algorithm 1 Linearized Least Squares with Regularization

Initialize x_0, y_0, z_0, $\alpha^{(0)}$, $\gamma^{(0)}$
// Wait until coefficient matrices for OLS are full-rank
for $t = 1 : n-1$ **do**
  Choose greedy u_t w.r.t. $V_t(x_{t-1}, z_{t-1}; \alpha^{(t-1)}, \gamma^{(t-1)})$
  $x_t := x_{t-1} + u_t$; $z_t := A^{(t-1)} z_{t-1} + \mathbf{1} u_t$; $\alpha^{(t)} := \alpha^{(t-1)}$; $\gamma^{(t)} := \gamma^{(t-1)}$
end for
// Once the matrices are full-rank, perform ordinary least squares with regularization on the linearized model
for $t = n : T$ **do**
  Initialize $\alpha^{(t)} := \alpha^{(t-1)}$, $\gamma^{(t)} := \gamma^{(t-1)}$
  while $\alpha^{(t)}$ and $\gamma^{(t)}$ do not converge yet **do**
    Compute $y_1, \ldots, y_{t-1}$ using $\alpha^{(t)}$, $\gamma^{(t)}$.

$$G_t^\alpha := \begin{bmatrix} y_1^\top \\ \vdots \\ y_t^\top \end{bmatrix}, \; h_t^\alpha := \begin{bmatrix} \Delta p_1 + \mathbf{1}^\top (y_0 - \gamma u_1) \\ \vdots \\ \Delta p_t + \mathbf{1}^\top (y_{t-1} - \gamma u_t) \end{bmatrix}$$

    $\alpha^{(t)} := \text{argmin}_\alpha \, ||G_t^\alpha \alpha - h_t^\alpha||_2^2 + \lambda_t^\alpha ||\alpha - \alpha^{(t-1)}||_2^2$
    Compute $z_1, \ldots, z_{t-1}$ using $\alpha^{(t)}$.

$$G_t^\gamma := \begin{bmatrix} u_1 \mathbf{1}^\top - z_0^\top (I - A) \\ \vdots \\ u_t \mathbf{1}^\top - z_{t-1}^\top (I - A) \end{bmatrix}, \; h_t^\gamma := \begin{bmatrix} \Delta p_1 \\ \vdots \\ \Delta p_t \end{bmatrix}$$

    $\gamma^{(t)} := \text{argmin}_\gamma \, ||G_t^\gamma \gamma - h_t^\gamma||_2^2 + \lambda_t^\gamma ||\gamma - \gamma^{(t-1)}||_2^2$
  end while
  Choose greedy u_{t+1} w.r.t. $V_t(x_{t-1}, z_{t-1}; \alpha^{(t)}, \gamma^{(t)})$; $x_{t+1} := x_t + u_{t+1}$; $z_{t+1} := A^{(t)} z_t + \mathbf{1} u_{t+1}$
end for