Mid-Price Prediction in a Limit Order Book

Deepan Palguna and Ilya Pollak

***Abstract*—We propose several nonparametric predictors of the mid-price in a limit order book, based on different features constructed from the order book data observed contemporaneously and in the recent past. We evaluate our predictors in the context of an order execution task by constructing order execution strategies that incorporate these predictors. In our evaluations, we use a large dataset of historical order placements, cancellations, and trades over a five-month period in 2013-2014 for liquid stocks traded on NASDAQ. We show that some of the features achieve statistically significant improvements compared to some standard strategies that do not incorporate price forecasting. For the two features that achieve the best performance, the trading cost improvement is on the order of one basis point, which can be economically very significant for asset managers with large portfolio turnovers and for brokers with considerable trading volumes.**

I. Introduction

Many securities markets are organized as double auctions where each incoming *limit order*—i.e., an order to buy or sell at a specific price—is stored in a data structure called the *limit order book*. The order flow is visible to every market participant in real time, with a delay determined by the speed of the data connection between the entity maintaining the orders and the market participant.[1] A trade happens whenever a *marketable order* arrives—i.e., an order to buy or sell at the best currently available price. To construct a trade, the incoming marketable order is matched with the best possible entry (or, if necessary, multiple entries) of the opposite side of the order book.

Accurately forecasting the dynamics of the order book is crucial in a variety of application scenarios, for example:

(1) *A broker charged with executing a trade.* The broker's ability to forecast the behavior of the order book would help him optimize the trade execution schedule—i.e., decide how to partition the total number of units to be traded into small amounts, how to time each small order, and how aggressively to trade it [19], [1], [24].

(2) *An asset manager contemplating a portfolio rebalancing.* A predictor of the order book dynamics would help forecast the cost incurred through the price impact of the trades required to perform the rebalancing [25], [23], [7].

(3) *An exchange or a regulator charged with maintaining orderly markets.* Order book forecasting tools would help exchanges and regulators quickly detect the possibility of abnormal events (such as the Flash Crash of May 6, 2010 [15], [11]) and take timely measures to prevent such events or to dampen their effects.

The importance of order book forecasting has motivated academic research in modeling the order book dynamics, developing prediction algorithms, and empirically studying the behavior of order books [6].

In the problem of optimal trade execution, order book dynamics is a critical element for studying the price impact of order sizes. Much previous work on optimal execution relies on macro-level modeling of the price impact of a trade [4], [2], [13]. The execution algorithms proposed in this literature do not model the state or the dynamics of the order book, and hence produce trading decisions that do not take into account the order book information.

A number of more recent papers on optimal execution methods do take the order book information into account. In [19], [1], [24], it is assumed that the order book has an underlying simple shape to which it returns in the absence of large orders, a property called *resilience*. While the assumption of order book resilience has some grounding in empirical observations [16], [5], the assumption of a single steady-state shape is not supported by empirical studies and appears to be too strong and too simplistic to enable accurate modeling of price and size dynamics in practice. Order book information in the form of order book imbalance is used in [26] to solve an optimal stopping problem for deciding the time to liquidate a single share. In [18], the authors take a reinforcement learning approach to the problem of optimal order execution and compare their approach to a simple submit and leave strategy.

Other recent contributions model order arrivals as marked point processes [12], [25], [9], [8]. These models are used to estimate the price impact of trades to help in making order placement decisions. Such marked point process models have also been used to formulate optimal order execution and market making strategies, which have been explored in [3], [14]. Due to the high dimensionality of limit order books, these methods necessitate reliable estimation of many parameters, which can be a challenge to perform.

In the present paper, we propose nonparametric approaches to short-term forecasting of the mid-price change (i.e., the change in the average of the best offer and the best bid prices). We introduce a state describing the configuration of the order book at each time, similar to the state used in [9]. We construct four state-dependent features which get updated during the course of a trading day, as new order flow information arrives. Based on the feature values, we predict whether the mid-price would increase, decrease or remain the same δ seconds into the future. We also propose simpler price predictors, based on the contemporaneous share imbalance and the order flow imbalance.

Authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907-2035. Contact e-mail: dpalguna@purdue.edu. Earlier versions of this paper have been published in [20] [21] and [22].

[1] We do not consider here hidden or partially hidden orders which are allowed by some markets.

In order to evaluate our predictors, we incorporate them into an order execution strategy. Through simulations on a set of liquid US stocks from NASDAQ's ITCH dataset, we show that several of our predictors result in statistically significant average gains as compared to strategies whose execution schedules do not depend on the observations of the order book. One of the three state-based features and the share imbalance achieve the highest performance gains, followed by the order flow imbalance whose performance gain is also statistically significant. The remaining three state-based features do not produce statistically significant performance improvements.

We start our exposition in Section II by introducing some terminology. Section III describes our prediction algorithm using a three-dimensional state vector and four state-based features. In Section IV, we describe the predictors that use the instantaneous share imbalance and order flow imbalance. Section V presents our evaluation method that uses order execution to test the effectiveness of our predictors. In Section VI, we describe the execution strategy derived in [19], which assumes very simple order book dynamics. We use this strategy as one of our benchmarks in Section VII, where we present experimental results using real order book data from NASDAQ's ITCH data.

II. TERMINOLOGY

A *limit order* is an order to buy or sell a desired number of units of an asset at a desired price. The order book at any time t is the collection of all outstanding limit orders at all prices at that time. The highest price among all the buy limit orders at time t is called the *best bid price* $(V_b(t))$ at time t. The lowest price among all the sell limit orders at time t is called the *best ask price* $(V_a(t))$ or *best offer price* at time t. The difference between the best ask and the best bid prices is called the *bid-ask spread*. The average of the best ask and the best bid is called the *mid-price*, denoted using $V(t)$.

A *marketable order* is an order to buy or sell a desired number of units of an asset at the best currently available price. When a buy marketable order comes in, it gets executed against the current limit orders at the best ask price. If there are not enough units offered by the sellers at the best ask price to fill the entire marketable order, the remainder of the marketable order gets executed at the next best price, etc., until the entire marketable order is executed. In the same way, incoming sell marketable orders get executed against the bid side of the order book. We call all those orders that result in an execution and take liquidity *market* orders.

We assume that only three kinds of events can change the configuration of an order book: limit orders, limit order cancellations, and market orders. Figs. 1 and 2 illustrate the order book and its changes in response to incoming orders.

III. FEATURE-BASED PREDICTION

A. Description of three-dimensional states

We describe the state of the order book at time t with a three-dimensional vector $\mathbf{S}(t)$ whose entries are:

$$S_1(t) = \left\lfloor \frac{\text{Number of shares at the best bid price at time t}}{100} \right\rfloor;$$

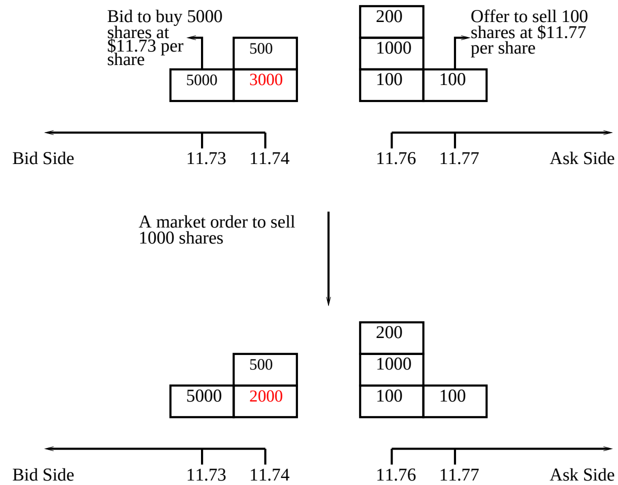

Fig. 1. Incoming market order

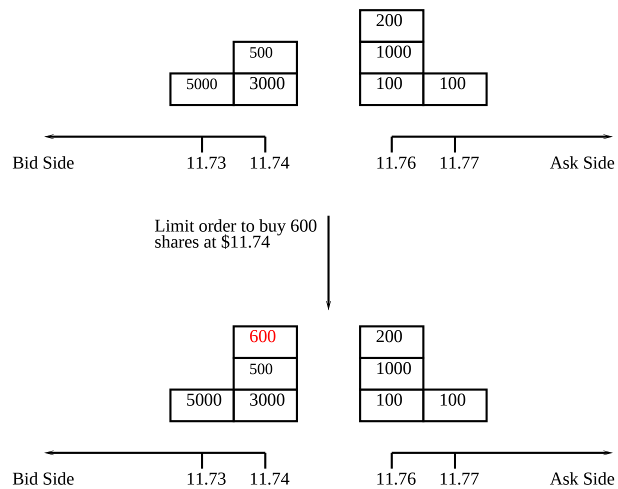

Fig. 2. Incoming limit order

$$S_2(t) = \left\lfloor \frac{\text{Number of shares at the best ask price at time t}}{100} \right\rfloor;$$
$$S_3(t) = \text{the bid-ask spread (in cents) at time } t.$$

These states are similar to those used in [9]; however, in [9] it is assumed that all orders have equal sizes whereas we do not make this assumption and use shares instead of orders. The mid-price change between time t and time $t+\delta$ is denoted as

$$D(t, \delta) = V(t+\delta) - V(t).$$

In our data set, every price is an integer number of cents. Therefore, the mid-prices and their changes are integer multiples of 0.5 cents. We assume that the price changes $D(t, \delta)$ are temporally stationary over our estimation windows (up to one trading day), in particular, that the marginal distribution of $D(t, \delta)$ does not depend on t and that the conditional distribution of $D(t, \delta)$ given the state $\mathbf{S}(t)$ does not depend on t. We use $\tau(t)$ to denote the earliest time after t at which the mid-price is different from $V(t)$.

B. Features

For any state s and any time interval δ, we define four *conditional features*:

- The conditional probability of non-zero mid-price change,
$$\mathbb{P}[D(t,\delta) \neq 0 | \mathbf{S}(t) = \mathbf{s}] \tag{1}$$
- The conditional expectation of the mid-price change
$$\mathbb{E}[D(t,\delta) | \mathbf{S}(t) = \mathbf{s}] \tag{2}$$
- The conditional expectation of mid-price change at the time of first mid-price change
$$\mathbb{E}[D(t,\tau(t)) | \mathbf{S}(t) = \mathbf{s}] \tag{3}$$
- The conditional expectation of the time to first change
$$\mathbb{E}[\tau(t) | \mathbf{S}(t) = \mathbf{s}] \tag{4}$$

C. Estimating conditional features from data

We assume ergodicity and empirically estimate these features as temporal averages. Our temporal discretization step is $\Delta t = 1\text{ms}$. We compute all our empirical estimates within a single trading day, and set $t = 0$ at the beginning of the trading day. We assume that δ in Eqs. (1,2) contains an integer number of discretization steps. Let $\mathbf{s}$ be a particular state. At any time t—which is an integer multiple of Δt—during the trading day, we define
$$\#_{\{\mathbf{S}(n)=\mathbf{s}\}}[0,t]$$
to be the number of occurrences of the event $\mathbf{S}(n) = \mathbf{s}$ among the times $n = 0, \Delta t, 2\Delta t, \ldots, t - \delta$. We similarly define
$$\#_{\{\mathbf{S}(n)=\mathbf{s} \text{ and } D(n,\delta)\neq 0\}}[0,t]$$
to be the number of occurrences of both $\{\mathbf{S}(n) = \mathbf{s}\}$ and $\{D(n,\delta) \neq 0\}$ among the times $n = 0, \Delta t, 2\Delta t, \ldots, t - \delta$. We estimate the conditional probability in Eq. (1) as the ratio
$$\frac{\#_{\{\mathbf{S}(n)=\mathbf{s} \text{ and } D(n,\delta)\neq 0\}}[0,t]}{\#_{\{\mathbf{S}(n)=\mathbf{s}\}}[0,t]}$$
We estimate the conditional expectation in Eq. (2) as
$$\frac{\sum\limits_{n:\mathbf{S}(n)=\mathbf{s}} D(n,\delta)}{\#_{\{\mathbf{S}(n)=\mathbf{s}\}}[0,t]},$$
where the summation in the numerator is performed over all the times $n = 0, \Delta t, 2\Delta t, \ldots, t - \delta$ when event $\mathbf{S}(n) = \mathbf{s}$ occurs. The conditional expectations in Eq. (3,4) are similarly estimated as
$$\frac{\sum\limits_{n:\mathbf{S}(n)=\mathbf{s}} D(n,\tau(n))}{\#_{\{\mathbf{S}(n)=\mathbf{s}\}}[0,t]} \text{ and } \frac{\sum\limits_{n:\mathbf{S}(n)=\mathbf{s}} \tau(n)}{\#_{\{\mathbf{S}(n)=\mathbf{s}\}}[0,t]} \text{ respectively.}$$

Using the three-dimensional state can result in there being too many states for features conditioned on each state to be estimated reliably. Therefore, at the time of prediction, we perform simple averaging of the features of K nearest neighbours of the current state [17] and use the average features for prediction. To find the nearest neighbours, we define the proximity of states using the following distance function. For state $\mathbf{S} = \{S_1, S_2, S_3\}$ and $\mathbf{R} = \{R_1, R_2, R_3\}$ we define the distance between be the Euclidean distance between $\mathbf{S}$ and $\mathbf{R}$ if $S_1 - S_2$ and $R_1 - R_2$ have the same sign. Otherwise, we define the distance between $\mathbf{S}$ and $\mathbf{R}$ to be infinite:
$$d(\mathbf{S},\mathbf{R}) = \begin{cases} d_E(\mathbf{S},\mathbf{R}) & \text{if } \mathrm{sgn}(S_1 - S_2) = \mathrm{sgn}(R_1 - R_2), \\ \infty & \text{otherwise,} \end{cases}$$
where d_E is the Euclidean distance between the states and $\mathrm{sgn}(0) = 0$.

D. Prediction of the mid-price change

We propose to predict mid-price changes in two steps – the first one for predicting whether the mid-price will change a fixed amount of time (δ) into the future, and the second one to predict whether the mid-price change would be positive or negative. At time t, we compute the state $S(t)$. We then estimate the four conditional features of Eqs. (1,2,3,4) for $S(t)$ and its K nearest neighbours using the distance function defined in Eq. (III-C). The computation of these features is over the interval $[0,t]$, as described earlier. The features used for prediction are the simple average of the features of the K nearest neighbours of the current state.

In the first step of prediction, we solve a binary hypothesis testing problem where the null hypothesis is that the mid-price change is zero and the alternative hypothesis is that the mid-price change is non-zero. For this step, which we call the *change detection* step, we propose two rules. The first one selects the alternative hypothesis of a non-zero mid-price change if the estimated conditional mid-price change probability in Eq. (1) is above a threshold θ. Otherwise, the algorithm selects the null hypothesis of zero mid-price change. The second algorithm selects the alternative hypothesis if the estimated conditional time to first mid-price change in Eq. (4) is lesser than δ and selects the null hypothesis otherwise.

For the second step, called the *sign prediction* step, we propose two rules. The first rule is to predict an upward movement in mid-price if the feature defined in Eq. (2) is positive and predict a downward movement in mid-price otherwise. The second rule is to predict an upward movement in mid-price if the feature defined in Eq. (3) is positive and predict a downward movement otherwise.

We combine the two predictors in each step based on the error rates conditioned on the state. For each change detection rule, we measure change detection error rate and for each sign prediction rule, we measure sign prediction error rate, all conditioned on the state. A change detection error is the event where a change is predicted and no change is actually observed or the event where no change is predicted but change is actually observed. The change detection error rate is measured as the ratio of the number of change detection errors to the total number of steps when a prediction is made. Similarly, a sign prediction error is the event where an upward change is predicted and the mid-price actually moves downward or the event where a downward change is predicted and the mid-price actually moves upward. The sign prediction error rate is measured as the ratio of the number of sign prediction errors to the total number of times a change is predicted. The conditional error rates corresponding to the four rules, conditioned on each state are measured over the time steps $n = 0, \Delta t, 2\Delta t, \ldots, t-\delta$. If at time t the predictions of the two rules disagree at any of the two stages, then we pick the prediction of the one with the lower conditional error rate.

Therefore, the final prediction algorithm first predicts whether a change is going to occur or not based on the two change detection rules and their corresponding conditional error rates. If a change is predicted at this stage, the algorithm goes to the sign prediction stage and predicts whether the change would be positive or negative in a similar fashion. Otherwise, the algorithm predicts no change and stops. As mentioned earlier, the features and the error rates used for the final prediction are averaged over K nearest neighbours.

IV. Prediction with share imbalance and order flow imbalance

A. Prediction with share imbalance

The *share imbalance* $\mathrm{SI}(t)$ at time t is defined as

$$\mathrm{SI}(t) = \frac{\#_{\text{shares at the best bid}}(t)}{\#_{\text{shares at the best bid}}(t) + \#_{\text{shares at the best ask}}(t)} \quad (5)$$

We use a two-step prediction algorithm, where the change detection step predicts that the mid-price will change at time $t+\delta$ if

$$\mathrm{SI}(t) \neq 1/2.$$

If the change detection step predicts a change, then we predict the sign of mid-price change as follows:

$$\text{Sign prediction at } t = \begin{cases} \text{up} & \text{if } \mathrm{SI}(t) > 1/2 \\ \text{down} & \text{if } \mathrm{SI}(t) < 1/2 \end{cases} \quad (6)$$

B. Prediction with order flow imbalance

The *order flow imbalance* feature, $\mathrm{OFI}(t)$, defined in [7] incorporates temporal information into the characterization of supply-demand imbalance. At time $\mathcal{T}_i$, the bid price, ask price, number of shares at the best bid and number of shares at the best ask are denoted using $V_b(\mathcal{T}_i)$, $V_a(\mathcal{T}_i)$, $q_b(\mathcal{T}_i)$ and $q_a(\mathcal{T}_i)$, respectively. We let $\mathcal{T}_{i+1}$ be the first time instant after $\mathcal{T}_i$ when one of these quantities changes, signifying a change in the supply-demand imbalance. This is quantified using e_{i+1} defined in Eq. (7). We define the order flow imbalance at time t as the sum of e_i as measured at the time of the past E such events, where E is a user-defined parameter.

The change detection step predicts that the mid-price will change at time $t+\delta$ if

$$|\mathrm{OFI}(t)| > 0.$$

If the change detection step predicts a change, then we predict the sign of mid-price change as the sign of the flow imbalance.

V. Evaluating predictors using order execution algorithms

In order to test the effectiveness of our predictors, we incorporate them into an order execution problem whose goal is to execute X_0 shares within the time interval $[t_0, t_N]$. This can be done for both buying and selling. We describe it here for buying. We assume that trades are constrained to happen at time steps $t_0, t_1, \cdots, t_N$, which are equally spaced δ seconds apart. Our benchmark is a *uniform* execution strategy[2] which uses market orders of size $X_0/(N+1)$ at times $t = t_0, t_1, \cdots, t_N$.

We propose a modification to uniform execution that makes use of our predictors. Specifically, if at time t_i we believe that the price will go up soon, we buy more than suggested by the uniform strategy. If at time t_i we believe that the price will go down in the near future, we buy less than suggested by the uniform strategy. We simply follow the uniform strategy if no change is predicted. We use parameter $\pi \in [0,1]$ to indicate by how much we accelerate or decelerate buying in response to our predictions. The value $\pi = 1$ means a 100% increase in the number of shares in response to an upward change prediction and a 100% reduction in the number of shares in response to a downward change prediction. The value $\pi = 0$ corresponds to the uniform strategy. For the case of selling, the action that is opposite to the action while buying would be followed when mid-price change is predicted, and uniform execution would be followed when no change is predicted.

It is possible for this strategy to postpone the entire execution to the last instant (t_N). This could result in adverse execution costs as a result of our large order getting crossed at deeper levels of the book. In order to avoid large orders getting executed at the end, we perform uniform execution towards the end of the algorithm. In most cases, there are more than 100 shares at the best quote. Hence, we make sure that we perform uniform execution starting from the time instant when delaying execution by one more step would result in executing more than 100 shares at any future step. We show the exact procedure for buying using the pseudocode in Fig. 3. For the sake of clarity in the pseudocode, we do not show the steps when the order book is updated.

VI. Comparison with the Obizhaeva-Wang strategy

In this section, we describe an optimal order execution strategy that operates on very simple assumptions for order book dynamics. Following the conventions and notations in Obizhaeva and Wang's paper [19], the objective of the problem in its discrete-time version, is to buy X_0 shares using market orders over $N+1$ discrete time steps given by t_0, t_1, $\cdots$, t_N. For a risk-neutral trader, the objective is to minimize $\mathbb{E}\left[\sum_{n=0}^{N} \bar{P}_n x_n\right]$, subject to the constraint $\sum_{n=0}^{N} x_n = X_0$. In the preceding equations, x_n and $\bar{P}_n$ are the number of shares purchased at the n^{th} time step and the average execution cost at that step respectively. Since the objective is to buy using market orders, it is sufficient to consider only the ask side of the book. The assumptions of the model are as follows:

- The market participants are assumed to be comprised of one large trader and many small traders.

[2] In the literature, such strategies are sometimes called TWAP, for *time-weighted average price*. This is because the average price paid by this strategy is approximately equal to the average price of the security during $[t_0, t_N]$.

$$e_{i+1} = I_{\{V_b(\mathcal{T}_{i+1}) \geq V_b(\mathcal{T}_i)\}} q_b(\mathcal{T}_{i+1}) - I_{\{V_b(\mathcal{T}_{i+1}) \leq V_b(\mathcal{T}_i)\}} q_b(\mathcal{T}_i) - I_{\{V_a(\mathcal{T}_{i+1}) \leq V_a(\mathcal{T}_i)\}} q_a(\mathcal{T}_{i+1}) + I_{\{V_a(\mathcal{T}_{i+1}) \geq V_a(\mathcal{T}_i)\}} q_a(\mathcal{T}_i) \quad (7)$$

```
function EXECUTEBUYORDERIMBALANCE(X_0, N, δ, t_0, π)
    i ← 0
    while i ≤ N do
        t ← t_i
        SI(t) ← q_b(t) / (q_b(t) + q_a(t))
        if i ≠ N then
            if X_{t_i} / (N − i) < 100 then
                if SI(t) ≠ 1/2 then
                    % Predict that change will happen
                    if SI(t) > 1/2 then
                        % Predict upward movement
                        Execute min{(1+π)X_0 / (N+1), X_{t_i}}
                    else
                        % Predict downward movement
                        Execute min{(1−π)X_0 / (N+1), X_{t_i}}
                    end if
                else
                    Execute min{X_0 / (N+1), X_{t_i}}
                end if
            else
                Execute min{100, X_{t_i}}
            end if
        else
            Execute X_N
        end if
        i ← i + 1
    end while
end function
```

Fig. 3. Buying with price prediction based on the share imbalance, and ensuring that not more than 100 shares get executed at any time step. Pseudocode assumes that at each t_i, the order book is up to date.

- In the absence of the large trader, the ask side of the order book is assumed to have a steady state shape with $q \times dP$ shares over any price interval $[P, P + dP)$.
- In the absence of the large trader, the security has a fundamental price F_t, which follows a Brownian motion, and the steady state bid-ask spread is equal to s.
- The mid-price of the security, V_t, would follow the same dynamics, and the ask price would follow the dynamics of $V_t + s/2$ if it were not for the large trader who impacts their dynamics.
- The large trader submits market orders and any buy market order of size x is assumed to have a permanent price impact equal to λx.
- Additionally, it also has a temporary price impact that decays exponentially with time. For a trade of size x, placed at time $t_i < t$, the temporary impact at time t is equal to $x\kappa e^{-\rho(t-t_i)}$.

If the number of trades during the interval $[0, t)$ is denoted using $n(t)$, then the ask price at time t can be calculated by summing together the two price impacts as

$$A_t = F_t + s/2 + \lambda \sum_{i=0}^{n(t)} x_{t_i} + \sum_{i=0}^{n(t)} x_{t_i} \kappa e^{-\rho(t-t_i)}$$

If a trade of size x is executed at time t, the block order book assumption means that the ask price at t_+ obeys the following equation

$$\int_{A_t}^{A_{t_+}} q dP = x$$

From this equation, it can be inferred that the ask price immediately after the trade is $A_t + x/q$, and the average execution price is $A_t + x/(2q)$. Following [19], we fix $\lambda = 1/(2q)$, which means that if there are no more large trades, the ask price would converge to $A_t + x/(2q)$. Comparing $A_t + x/q$ with the equation $A_{t_+} = A_t + \lambda x + \kappa x$, we get that $\kappa = 1/q - \lambda$. For the block shaped order book, this translates to $\kappa = 1/(2q)$. The optimal number of shares at each time step can now be found using dynamic programming. Denoting the total number of shares that are unexecuted at time step t_n using X_{t_n} and assuming that any two time steps t_i and $t_{(i-1)}$ are spaced δ seconds apart, the solution is given by the recursive formula shown in Eq. (8) and stated as Proposition 1 in [19]. In Eq. (8), D_{t_n} is the deviation of the ask price from its steady state value, which is equal to the sum of all temporary impacts i.e, $D_{t_n} = \sum_{i=0}^{n(t)} x_{t_i} \kappa e^{-\rho(t-t_i)}$. The coefficients α_{n+1}, β_{n+1}, γ_{n+1} and ϵ_{n+1} are calculated as shown in Eq. (9). The boundary conditions for Eq. (9) are $\alpha_N = 1/(2q) - \lambda$, $\beta_N = 1$, and $\gamma_N = 0$.

A. Estimating model parameters

Under the block order book assumption, this model only has two parameters. The first one is the density q of limit orders in the ask side and the second one is the temporary price impact parameter ρ. As an estimate of the density parameter, we use the sample average of the number of shares, averaged over each price on the appropriate side of the order book and over time instants separated by 1 second.

The purpose of the resiliency parameter is to quantify the rate at which the order book reverts back to nominal levels after it encounters a shock. There have been many attempts in the literature to investigate and quantify resiliency. In [10], the authors take a nonparametric approach to characterizing resiliency. While [5] studies the nature of the order book event immediately after a shock, the authors in [10] study the evolution of the order book over many events around aggressive orders. The authors observe that bid-ask spreads are very low just before an aggressive order, consistent with [5]. After the aggressive order, the spreads revert back to levels that are close to the time average. However, estimating resiliency as described in the model is an open problem and beyond the

$$\begin{aligned}
x_n &= -\frac{1}{2}\epsilon_{n+1}[D_{t_n}(1-\beta_{n+1}e^{-\rho\delta}+2\kappa\gamma_{n+1}e^{-\rho\delta})-X_{t_n}(\lambda+2\alpha_{n+1}-\beta_{n+1}\kappa e^{-\rho\delta})] \quad (8)\\
\alpha_n &= \alpha_{n+1}-\frac{1}{4}\epsilon_{n+1}(\lambda+2\alpha_{n+1}-\beta_{n+1}\kappa e^{-\rho\delta})^2\\
\beta_n &= \beta_{n+1}e^{-\rho\delta}+\frac{1}{2}\delta_{n+1}(1-\beta_{n+1}e^{-\rho\delta}+2\kappa\gamma_{n+1}e^{-\rho\delta})(\lambda+2\alpha_{n+1}-\beta_{n+1}\kappa e^{-\rho\delta})\\
\gamma_n &= \gamma_{n+1}e^{-\rho\delta}-\frac{1}{4}\delta_{n+1}(1-\beta_{n+1}e^{-\rho\delta}+2\kappa\gamma_{n+1}e^{-\rho\delta})\\
\epsilon_{n+1} &= [1/(2q)+\alpha_{n+1}-\beta_{n+1}\kappa e^{-\rho\delta}+\gamma_{n+1}\kappa^2e^{-2\rho\delta}]^{-1} \quad (9)
\end{aligned}$$

scope of our paper. Based on the behaviour of the algorithm, we can identify two extreme cases between which all other cases would fall. We consider these cases for the purpose of our experiments in order to circumvent the estimation of ρ.

The first case is when $\rho \to \infty$. When ρ is very large, our experiments show that the strategy becomes indistinguishable from uniform execution. The second case is when $\rho \to 0$. In this case, the strategy behaves very differently from uniform execution. It executes half the shares at the first instant (t_0) and the other half at the very last instant (t_N).

VII. Performance and Comparison of Prediction Algorithms

We use NASDAQ's ITCH data from October 2013 to February 2014. This data set provides the complete order series from which limit order books can be constructed for all securities traded on NASDAQ. For order execution strategies that buy X_0 shares, we use the following quantity to measure the execution cost improvement of strategy A relative to strategy B:

$$R = \frac{\text{Cost of strategy B} - \text{Cost of strategy A}}{\text{Cost of strategy B}} \quad (10)$$

For the case of selling, we use

$$R = \frac{\text{Cost of strategy A} - \text{Cost of strategy B}}{\text{Cost of strategy B}} \quad (11)$$

Positive values of this quantity correspond to strategy A outperforming the strategy B.

A. *Order execution experiment setup*

One important issue in simulating execution strategies is that it is not possible to replicate market impact with historical data. This is because future data points, which are made up of public orders, cannot respond to simulated orders. In our experiments with market orders, we execute a market order at time t as if it will get executed when the order book is in its historical configuration at time t. But we do not modify the order book since it is extremely hard to simulate market impact. In other words, although simulated trades consume liquidity and large simulated trades can walk up the book, we simply simulate the process and do not modify the book. If an execution strategy prescribes a non-integer amount of shares at any stage, then we round it to the nearest integer.

The stock universe consists of $Q = 50$ liquid[3] stocks as measured over the period September 1 to September 30, 2013. We perform simulations on each trading day in the 5-month period of October 1, 2013 to February 28, 2014. Feature estimation starts at 9:30am, and we simulate the execution of X_0 shares between 2:30pm and 3:30pm on each day. In a real application scenario, a client would send an order to a broker and specify X_0. Therefore, the broker who runs the execution algorithm would not have any control over X_0. We note that we would get similar simulated execution costs per share for a wide range of X_0 since we do not modify the order book when our simulated orders are executed.

In order to find out whether our results are statistically significant, we compute the standard errors, $\hat{\eta}$, for the average relative improvement of our strategy's execution cost. Specifically, suppose that we simulate both the uniform strategy and our execution strategy with some fixed values of the parameters, on D days and Q stocks. Our strategy would produce the following execution cost improvements relative to uniform execution [calculated through Eqs. (10,11)]: $R_1, R_2, \cdots, R_M$ where $M \equiv DQ$ is the number of samples from D days with Q stocks on each day. We estimate the mean relative cost improvement and the standard deviation of relative cost improvement using the sample mean $\hat{\mu}$ and the sample standard deviation $\hat{\sigma}$, respectively. We assume that $R_1, R_2, \cdots, R_M$ are uncorrelated across stocks and have a common standard deviation. Then the standard error (i.e., the standard deviation of the sample mean) can be estimated as $\frac{\hat{\sigma}}{\sqrt{M}}$.

B. *Experiments with conditional features*

In our experiments we observe that the expected mid-price change at the time of the first mid-price change, conditional on the three-dimensional state, is the only sign prediction feature among those in Eqs. (1-4) that gives statistically significant improvements compared to the uniform execution strategy. We describe the results for the case when we always predict mid-price change and use the feature described in Eq. (3) for predicting the sign of mid-price change. We average this feature over $K = 20$ nearest neighbours and let $\pi = 1$. We show the results in Table I. The mean improvements compared to the uniform execution strategy are shown in basis points[4]

[3] Liquidity in our work is defined as the average daily volume of shares traded, averaged over September 1 to September 30, 2013. We pick 50 stocks with the highest average daily trading volume over this interval.

[4] One *basis point* is 0.01%, i.e., 0.0001, abbreviated "bp" in singular and "bps" in plural.

TABLE I
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 3600 SHARES USING THE FEATURE OF EQ. (3) WITH $\delta = 1$ SECOND

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	1.128	0.839
Sample standard deviation $(\hat{\sigma}) \times 10^4$	15.86	15.77
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.222	0.221
$\frac{\hat{\mu}}{\hat{\eta}}$	5.077	3.798

TABLE II
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 3600 SHARES USING SI WITH $\delta = 1$ SECOND

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	1.117	0.8425
Sample standard deviation $(\hat{\sigma}) \times 10^4$	8.247	8.298
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.1155	0.1162
$\frac{\hat{\mu}}{\hat{\eta}}$	9.671	7.254

TABLE III
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 3600 SHARES USING OFI WITH $\delta = 1$ SECOND AND $E = 10$ CHANGE EVENTS

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	0.2727	0.2342
Sample standard deviation $(\hat{\sigma}) \times 10^4$	0.1952	0.1856
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.0273	0.0259
$\frac{\hat{\mu}}{\hat{\eta}}$	9.978	9.0114

TABLE IV
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 10800 SHARES WITH $\delta = 1$ SECOND USING BOTH SI AND OFI, AND $E = 10$

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	0.8177	0.6781
Sample standard deviation $(\hat{\sigma}) \times 10^4$	5.725	5.621
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.0802	0.0787
$\frac{\hat{\mu}}{\hat{\eta}}$	10.20	8.615

in the first row. For both buying and selling, the improvement is around 1bp. The t-statistics presented in the last row of the table suggest that these mean improvements are statistically significant. We next compare these results with the results obtained by using predictors based on the share imbalance and order flow imbalance as explained in Section IV-A.

C. Experiments with share imbalance and order flow imbalance

Table II shows the mean and standard error of the improvements over the execution strategy for buying and selling 3600 shares using the share imbalance feature. The mean improvements are similar to those in Table I: they are significantly positive and around 1 basis point.

Table III shows the mean improvements for buying and selling with the order flow imbalance where the OFI is obtained by summing over $E = 10$ changes in the first level of the order book. This strategy also shows positive and significant improvements as compared to the uniform execution, but the mean improvements are only around 0.2-0.3bps.

D. Combining share imbalance and order flow imbalance

We now investigate whether combining the SI and OFI leads to improved performance. The basic idea is to assign a higher confidence to the prediction if the two features agree. Specifically, if the SI predicts that the price will go up and the OFI predicts that it will not, we classify it as a "weak up" prediction. If the SI predicts that the price will go down and the OFI predicts that it will not, we classify it as a "weak down" prediction. If both the SI and OFI predict up, we classify it as a "strong up" prediction; and if they both predict "down", we classify it as a "strong down" prediction. This is summarized in Eq. (12).

If we are buying 10800 shares in one hour with $\delta = 1$ second, then we would buy 3 shares per second if the price is always predicted to be flat. If the prediction is a strong up, weak up, strong down or weak down, we buy 6 shares, 4 shares, 2 shares or 0 shares respectively. Our strategy for selling is defined similarly. Table IV shows the results for this strategy. For comparison, Table V shows the results for trading 10800 shares in one hour using the SI only. When buying, this strategy buys 6, 3, and 0 shares when the prediction is up, flat, and down, respectively. While using the two features together still produces statistically significant improvements compared to the uniform execution strategy, the improvements do not outperform using the share imbalance alone.

In Tables VI and VII we show the performance of the execution algorithm with perfect knowledge of the direction of price movement δ seconds into the future. The tables show the improvements relative to the uniform execution

TABLE V
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 10800 SHARES WITH $\delta = 1$ SECOND, USING SHARE IMBALANCE ONLY

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	1.0977	0.8387
Sample standard deviation $(\hat{\sigma}) \times 10^4$	8.1827	8.2475
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.1146	0.1549
$\frac{\hat{\mu}}{\hat{\eta}}$	9.5801	7.2629

$$\text{Prediction at } t = \begin{cases} \text{strong up} & \text{if } \mathrm{SI}(t) > 1/2 \text{ and } \mathrm{OFI}(t) > 0 \\ \text{weak up} & \text{if } \mathrm{SI}(t) > 1/2 \text{ and } \mathrm{OFI}(t) \leq 0 \\ \text{strong down} & \text{if } \mathrm{SI}(t) < 1/2 \text{ and } \mathrm{OFI}(t) < 0 \\ \text{weak down} & \text{if } \mathrm{SI}(t) < 1/2 \text{ and } \mathrm{OFI}(t) \geq 0 \\ \text{flat} & \text{otherwise} \end{cases} \tag{12}$$

TABLE VIII
SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR EXECUTING 10800 SHARES WITH USING THE OBIZHAEVA-WANG STRATEGY, $\delta = 1$ SECOND, $\rho = 10^{-6}$

	Buying	Selling
Mean improvement $(\hat{\mu}) \times 10^4$	0.7284	-2.881
Sample standard deviation $(\hat{\sigma}) \times 10^4$	14.235	14.238
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.1993	0.1993
$\frac{\hat{\mu}}{\hat{\eta}}$	3.654	-14.45

strategy for $\delta = 1, 2, 10$ and 30 seconds. With larger δ, the mean improvement increases since the algorithm is able to forecast longer into the future. However, note that these perfect-foresight experiments outperform our predictors only when they are able to predict perfectly for tens of seconds into the future.

E. Experiments with the Obizhaeva-Wang strategy

As discussed in Section VI, we consider two instances of the Obizhaeva-Wang strategy, resulting from two extreme settings of the temporary impact decay parameter ρ. If $\rho \to \infty$, then the strategy is identical to uniform execution, and therefore the average improvements for both buying and selling are identically equal to zero. If $\rho \to 0$, then the strategy executes half the shares at t_0 and half at t_N. We observe in Table VIII that the mean improvement averaged over buying and selling is negative.

VIII. CONCLUSIONS

We have proposed several nonparametric methods to predict short-term mid-price changes in limit order books. One of the predictors relies on features conditioned on the state of the order book. These features are estimated during the trading day using mild assumptions about their behaviour. We also evaluate predictors based on the contemporaneous share imbalance and order flow imbalance. In order to evaluate our predictors, we incorporate them into an order execution strategy. As benchmarks, we use two execution strategies found in the literature whose trading schedules are independent of the observations of the order book. Through experiments on 50 liquid stocks over NASDAQ's order book data from October 2013 to February 2014, we find that our predictors result in statistically significant improvements over the benchmarks. Two of our strategies produce a statistically significant trading cost improvement of about 1bp compared to the uniform execution strategy. This can be economically very significant for asset managers with large portfolio turnovers and for brokers with considerable trading volumes.

REFERENCES

[1] A. Alfonsi, A. Fruth, and A. Schied. Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, 10(2):143–157, June 2010.

[2] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3(2):5–40, 2001.

[3] E. Bayraktar and M. Ludkovski. Liquidation in limit order books with controlled intensity. January 2012. Working paper, available at arXiv: http://arxiv.org/abs/1105.0247.

[4] D. Bertsimas and A. W. Lo. Optimal control of execution costs. *Journal of Financial Markets*, 1:1–50, 1998.

[5] B. Biais, P. Hillion, and C. Spatt. An empirical analysis of the limit order book and the order flow in the Paris Bourse. *Journal of Finance*, 50(5):1655–1689, December 1995.

[6] R. Cont. Statistical modeling of high frequency financial data: Facts, models and challenges. *IEEE Signal Processing Magazine*, 28(5):16–25, September 2011.

[7] R. Cont, A. Kukanov, and S. Stoikov. The price impact of order book events. *Journal of Financial Econometrics*, 2013.

[8] R. Cont and A. Larrard. Price dynamics in a markovian limit order market. February 2012. Working paper, available at SSRN: http://ssrn.com/abstract=1735338.

[9] R. Cont, S. Stoikov, and R. Talreja. A stochastic model for order book dynamics. *Operations Research*, 58(3):549–563, May–June 2010.

[10] H. Degryse, F. Jong, M. Ravenswaaij, and G. Wuyts. Aggressive orders and the resiliency of a limit order market. *Review of Finance*, 9(2):201–242, 2005.

[11] D. Easley, M. M. L. de Prado, and M. O'Hara. The microstructure of the 'flash crash': Flow toxicity, liquidity crashes and the probability of informed trading. *The Journal of Portfolio Management*, 37(2):118–128, Winter 2011.

[12] R. F. Engle. The econometrics of ultra-high frequency data. *Econometrica*, 68(1):1–22, January 2000.

[13] R. C. Grinold and R. N. Kahn. *Active Portfolio Management: A Quantitative Approach for Providing Superior Returns and Controlling Risk*. McGraw-Hill, 2nd edition edition, 2000.

[14] F. Guilbaud and H. Pham. Optimal high-frequency trading with limit and market orders. *Quantitative Finance*, pages 1–16, August 2012.

[15] A. A. Kirilenko, A. S. Kyle, M. Samadi, and T. Tuzun. The flash crash: The impact of high frequency trading on an electronic market. Working paper, available at SSRN: http://ssrn.com/abstract=1686004, May 26, 2011.

[16] J. Large. Measuring the resiliency of an electronic limit order book. *Journal of Financial Markets*, 10(1):1–25, February 2007.

[17] D. M. Mount and S. Arya. Approximate K nearest neighbors. http://www.cs.umd.edu/~mount/ANN/.

[18] Y. Nevmyvaka, Y. Feng, and M. Kearns. Reinforcement learning for optimized trade execution. 2006.

[19] A. Obizhaeva and J. Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16:1–32, 2013.

[20] D. Palguna. and I. Pollak. Non-parametric prediction of the mid-price dynamics in a limit order book. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 896–899, Aug 2012.

[21] D. Palguna. and I. Pollak. Non-parametric prediction of the mid-price dynamics in a limit order book. In *2013 IEEE Information Theory Workshop (ITW)*, pages 1–5, Sept 2013.

[22] D. Palguna. and I. Pollak. Non-parametric prediction of the mid-price dynamics in a limit order book. In *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2013.

TABLE VI

SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR BUYING 3600 SHARES WITH DIFFERENT δ VALUES IN SECONDS AND WITH PERFECT PREDICTION

	Buy $(\delta = 1)$	Buy $(\delta = 2)$	Buy $(\delta = 10)$	Buy $(\delta = 30)$
Mean improvement $(\hat{\mu}) \times 10^4$	0.1092	0.2051	0.7830	1.8219
Sample standard deviation $(\hat{\sigma}) \times 10^4$	0.2114	0.3729	1.145	2.484
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.00296	0.00522	0.0160	0.0347
$\frac{\hat{\mu}}{\hat{\eta}}$	36.908	39.272	48.835	52.35

TABLE VII

SAMPLE MEAN, SAMPLE STANDARD DEVIATION, AND STANDARD ERRORS OF RELATIVE EXECUTION COST IMPROVEMENTS FOR SELLING 3600 SHARES WITH DIFFERENT δ VALUES IN SECONDS AND WITH PERFECT PREDICTION

	Sell $(\delta = 1)$	Sell $(\delta = 2)$	Sell $(\delta = 10)$	Sell $(\delta = 30)$
Mean improvement $(\hat{\mu}) \times 10^4$	0.1017	0.1915	0.7643	1.799
Sample standard deviation $(\hat{\sigma}) \times 10^4$	0.2991	0.3683	1.582	2.433
Standard error $(\hat{\eta} = \frac{\hat{\sigma}}{\sqrt{M}}) \times 10^4$	0.00299	0.00516	0.0113	0.0307
$\frac{\hat{\mu}}{\hat{\eta}}$	33.985	37.144	48.330	52.81

[23] M. Potters and J.-P. Bouchaud. More statistical properties of order books and price impact. *Physica A*, 324(1–2):133–140, June 2003.

[24] S. Predoiu, G. Shaikhet, and S. Shreve. Optimal execution in a general one-sided limit order book. *SIAM Journal on Financial Mathematics*, 2:183–212, March 2011.

[25] E. Smith, J. D. Farmer, L. Gillemot, and S. Krishnamurthy. Statistical theory of the continuous double auction. *Quantitative Finance*, 3(6):481–514, 2003.

[26] S. Stoikov and R. Waeber. Optimal asset liquidation using limit order book information. September 2012. Working paper, available at SSRN: http://ssrn.com/abstract=2113827.