Limit Order Book Dynamics

Our goal is to use the dynamics of the Limit Order Book (LOB) as an indicator for high-frequency stock price movement, thus enabling statistical arbitrage. Formally, we will the study limit order book imbalance process, $I(t)$, and the stock price process, $S(t)$, and attempt to establish a stochastic relationship $\dot{S} = f(S, I, t)$. We will then attempt to derive an optimal trading strategy based on the observed relationship.

Recap Next Steps

1. Complete in-sample backtesting of the 'naive' trading strategies.
2. Formulate stochastic control problem
3. Extra Reading: Bellman Equations, MDP, Partially Observable MDP

In-Sample Backtesting of Naive Trading Strategies

As a refresher:

We are a considering a CTMC for the joint distribution $(I(t), \Delta S(t))$ where $I(t) \in \{1, 2, \ldots, \#_{bins}\}$ is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and $\Delta S(t) = \text{sign}(S(t+\Delta t_S) - S(t)) \in \{-1, 0, 1\}$, considered individually for the best bid and best ask prices. The pair $(I(t), \Delta S(t))$ was then reduced into one dimension with a simple encoding.

From the resulting timeseries we estimated a generator matrix $\boldsymbol{G}$ and used it to obtain a one-step transition probability matrix $\boldsymbol{P} = e^{\boldsymbol{G}\Delta t_I}$. The entries of $\boldsymbol{P}$ contain the conditional probabilities $\mathbb{P}[\rho_{curr}, \Delta S_{curr} \mid \rho_{prev}, \Delta S_{prev}]$, from which we can solve for the probability of now seeing a given price change (ΔS_{curr}) conditional on the current imbalance, the previous imbalance, and the previous price change.

For example, one such conditional probability matrix $\boldsymbol{P_C}$ (using 3 imbalance bins) was:

	$\rho_n = 1$									$\rho_n = 2$									$\rho_n = 3$								
$\Delta S_n < 0 \rightarrow$	.67	.05	.04	.01	.03	.04	.00	.05	.05	.02	.50	.12	.01	.00	.02	.05	.01	.02	.00	.00	.52	.00	.01	.00	.00	.00	.00
$\Delta S_n = 0 \rightarrow$	.33	.95	.96	.99	.97	.96	.41	.93	.95	.96	.49	.87	.98	.99	.97	.91	.48	.96	.98	.95	.47	.95	.96	.93	.98	.88	.34
$\Delta S_n > 0 \rightarrow$	.00	.00	.00	.00	.00	.00	.58	.02	.00	.02	.01	.00	.01	.01	.01	.05	.51	.01	.02	.04	.01	.05	.03	.02	.02	.12	.66
	$\Delta S_{n-1} < 0$									$\Delta S_{n-1} > 0$									$\Delta S_{n-1} = 0$								

Immediately evident from $\boldsymbol{P_C}$ is that in most cases we are expecting no price change. In fact, the only cases in which the probability of a price change is > 0.5 show evidence of *momentum*; for example, the way to interpret the value in row 1, column 1 is: if $\rho_{prev} = \rho_{curr} = 1$ and previously we saw a downward price change, then we expect to again see a downward price change. In fact, the best way to summarize the matrix is:

$$\mathbb{P}[\Delta S_{curr} = \Delta S_{prev} \mid \rho_{prev} = \rho_{curr}] > 0.5$$

We backtested a number of naive trading strategies, outlined here, based on this key observation. In plain terms, the Naive trading strategies can be interpreted as follows:

Algorithm 1 Naive Trading Strategy

$cash = 0$
$asset = 0$
for $t = 2$: `length`$(timeseries)$ **do**
    if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{+}= data.BuyPrice(t)$
        $asset \mathrel{-}= 1$
    else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{-}= data.SellPrice(t)$
        $asset \mathrel{+}= 1$
    end if
end for
if $asset > 0$ **then**
    $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
    $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

Naive Trading Strategy Using the conditional probabilities obtained from $\boldsymbol{P_C}$, we will execute a buy (resp. sell) market order if the probability of an upward (resp. downward) price change is > 0.5.

Naive+ Trading Strategy Extending the naive trading strategy, if we anticipate no change then we'll additionally keep limited orders posted at the touch, front of the queue. We'll track MO arrival, assume we always get excuted, and immediately repost the limit orders.

Naive++ Trading Strategy We won't execute market orders or keep limit orders at the touch. Using the conditional probabilities obtained from $\boldsymbol{P_C}$, if we expect a downward (resp. upward) price change then we'll add a limit order to the sell (resp. buy) side, and hopefully pick up an agent who is executing a market order going against the price change momentum.

Naive- Trading Strategy We additionally considered a trading strategy, for benchmark purposes, which used only current imbalance to predict future price change. But actually this predicted $\mathbb{P}\left[\Delta S_{curr} = 0\right] > 0.5$ at all times, so we could not run a strategy off it.

Backtesting these trading strategies required a choice of parameters for Δt_S, the price change observation period, Δt_I, the imbalance averaging period, and $\#_{bins}$, the number of imbalance bins. Through a brute force calibration technique we found that $\#_{bins} = 4$ provided the highest expected number of successful trades for most tickers, so this was chosen as a constant. Similarly, we empirically saw that calibration always yielded $\Delta t_S = \Delta t_I$, so this was taken as a given. Then each backtest consisted of first calibrating the value Δt_I from one day of data by maximizing the intra-day Sharpe ratio, then using the calibrated parameters to backtest the entire year.

Algorithm 2 Naive+ Trading Strategy

$cash = 0$
$asset = 0$
$LO_{posted} =$ `False`
for $t = 2$: `length`$(timeseries)$ **do**
    if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{+}= data.BuyPrice(t)$
        $asset \mathrel{-}= 1$
        $LO_{posted} =$ `False`
    else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{-}= data.SellPrice(t)$
        $asset \mathrel{+}= 1$
        $LO_{posted} =$ `False`
    else if $\mathbb{P}\left[\Delta S_{curr} = 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $LO_{posted} =$ `True`
    end if
    if LO_{posted} **then**
        for $MO \in ArrivedMarketOrders(t, t+1)$ **do**
            if $MO == Sell$ **then**
                $cash \mathrel{-}= data.BuyPrice(t)$
                $asset \mathrel{+}= 1$
            else if $MO == Buy$ **then**
                $cash \mathrel{+}= data.SellPrice(t)$
                $asset \mathrel{-}= 1$
            end if
        end for
    end if
end for
if $asset > 0$ **then**
    $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
    $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

Algorithm 3 Naive++ Trading Strategy

$cash = 0$
$asset = 0$
$LOBuy_{posted} =$ `False`
$LOSell_{posted} =$ `False`
for $t = 2$: `length`$(timeseries)$ **do**
    if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $LOBuy_{posted} =$ `False`
        $LOSell_{posted} =$ `True`
    else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $LOBuy_{posted} =$ `True`
        $LOSell_{posted} =$ `False`
    else if $\mathbb{P}\left[\Delta S_{curr} = 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $LOBuy_{posted} =$ `False`
        $LOSell_{posted} =$ `False`
    end if
    for $MO \in ArrivedMarketOrders(t, t+1)$ **do**
        if $MO == Sell \ \wedge \ LOBuy_{posted}$ **then**
            $cash \mathrel{-}= data.BuyPrice(t)$
            $asset \mathrel{+}= 1$
        else if $MO == Buy \ \wedge \ LOSell_{posted}$ **then**
            $cash \mathrel{+}= data.SellPrice(t)$
            $asset \mathrel{-}= 1$
        end if
    end for
end for
if $asset > 0$ **then**
    $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
    $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

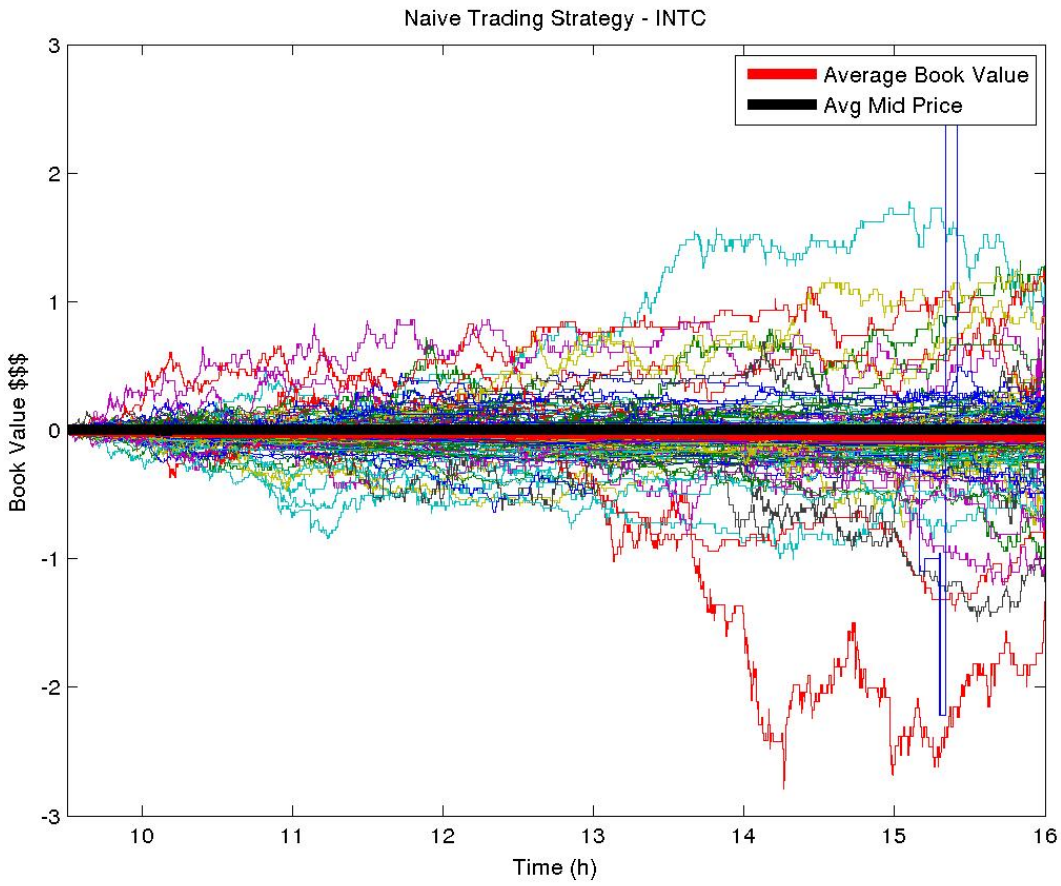

Figure 1: INTC: Bookvalue against time of trading day.

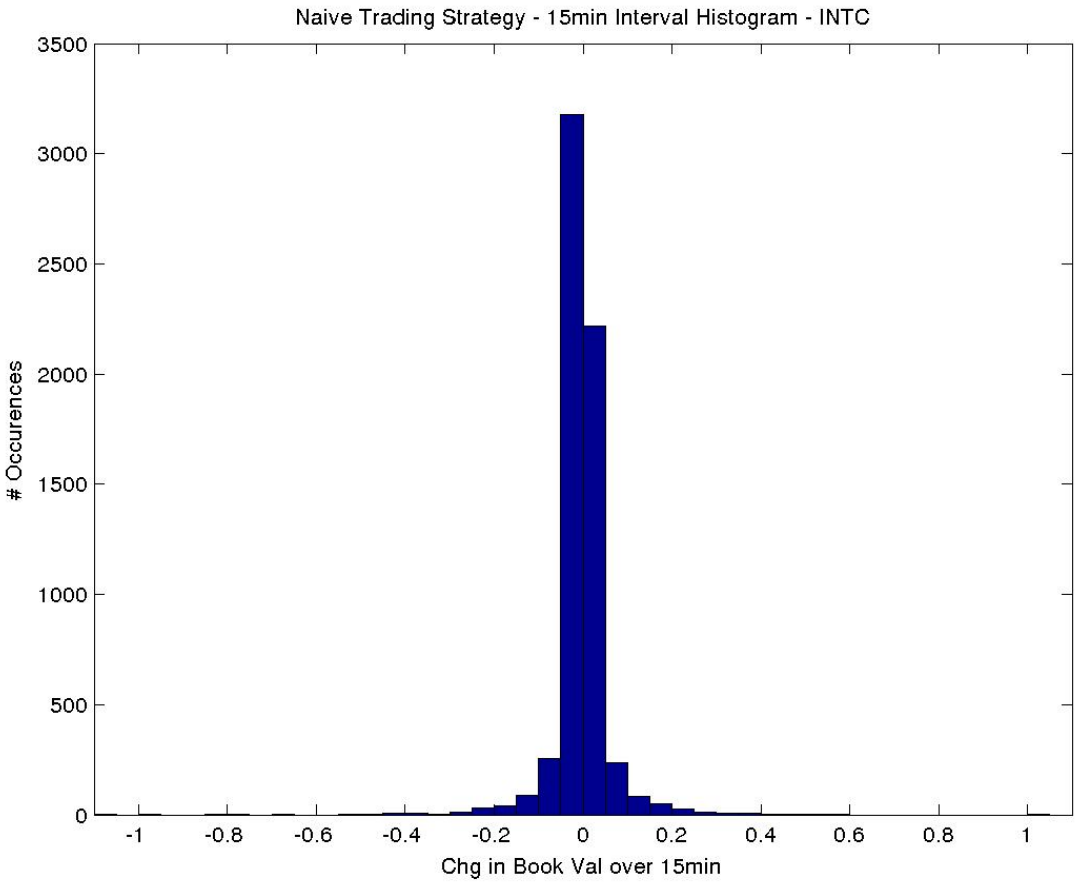

Figure 2: INTC: Histogram of 15min bookvalue changes.

Conclusions from Naive Trading Strategies

To properly compare the Naive trading strategies, it must be understood that the Naive+ strategy has the Naive built into it - thus it's actually the difference between the two that needs to be assessed to ascertain the effect of posting Limit Orders when no price change is predicted. As seen in Figure 3, the Naive trading strategy on average underperformed the average mid price, while the Naive+ (adding at-the-touch limit orders when no change was predicted) and Naive++ (adding limit orders to adversely selecting agents that traded against the price change momentum) strategies both on average generated revenue.

Question 1 Why is the Naive strategy producing, on average, normalized losses? Especially so when considering that we are in-sample backtesting. On calibration, we see that our intra-day sharpe ratio is around 0.01 or 0.02 when we choose our optimal parameters, so at the very least on the calibration date the strategy produces positive returns. The remainder of the calendar days are out-of-sample, as the parameters are (likely) not optimal. This suggests non-stationary data, and in particular not every day can be modelled by the same Markov chain. The problem may be exaggerated by the fact that we're calibrating on the first trading day of the calendar year, when we might expect reduced, or at least non-representative, trading activity. Further, we're currently obtaining the $\boldsymbol{P_C}$ probability matrix using only bid-side data, not sell-side or mid, and we're ignoring the bid-ask spread. Thus predicting a "price change" may be insufficient when considering a monetizable opportunity, as we won't be able to profit off a predicted increase followed by a predicted decrease unless the interim mid-price move is greater than the bid-ask spread (assuming constant spread). This suggests a potential straightforward modification to the strategy.

Question 2 Why do the Naive+ and ++ strategies outperform the Naive strategy? This is particularly interesting since the probabilities are being obtained from the same matrix. The obvious difference between the successful and nonsuccessful strategies is that the former (a) uses limit orders, and (b) executes when we predict a zero change, whereas the latter uses (a) market orders, and (b) executes when we do predict nonzero change.

(a) obviously leads to a different transaction price being used: if I buy with a LO I'm paying the bid price, whereas buying with a MO I pay the ask price. If I value the stock using the mid price, and the mid price doesn't move as a result of my transaction, then with LO I'm buying the asset for less than I'm valuing it at, and with MO I'm paying more than its value.

(b) seems to be the largest flaw in the Naive strategy, to which there are two factors. One, we are not predicting the magnitude of the price change, only whether it is zero or nonzero. Two, from the probabilities presented above, *we will only predict a price change if we've already seen a price change.* Thus we're effectively reacting too late.

Here's how this works adversely. Suppose a stock has bid/ask quotes of \$9.99/\$10.01, for a bid-ask spread of \$0.02 and a mid of \$10.

1. Imbalance = 1 (pressure for upward price move). $[NPV = 0]$
2. Bid/ask goes up to \$10.00/\$10.02. $[NPV = 0]$
3. Imbalance = 1. We predict another > 0 price change. $[NPV = 0]$
4. We buy 1 share (at \$10.02). $[NPV = -0.01]$

5. Bid/ask goes up to \$10.01/\$10.03. $[NPV = 0]$
6. Imbalance = -1 (pressure for a downward move). $[NPV = 0]$
7. Bid/ask goes down to \$10.00/\$10.02. $[NPV = -0.01]$
8. Imbalance = -1. We predict another < 0 price change.
9. We sell 1 share (at \$10.00). $[NPV = -0.02]$
10. Bid/ask goes down to \$9.99/\$10.01. $[NPV = -0.02]$

In this example the price goes up and back down by two cents to return to where it started, and in the process we lost \$0.02. Now imagine what happens if we price goes up by one cent, up by one cent, then down by ten cents, down by one cent. In this case we lose \$0.11. We're unable to predict that initial upward or downward price change, and only react to it.

Ideas to Explore and Next Steps

- Model the mid price instead of the bid or ask, hold the bid-ask spread as a constant (average observed), and predict price changes at least as great as the spread, instead of simply non-zero.
- Calculate imbalance using a weighted average of the best n bid (resp. ask) prices. This may reduce noise in the signal, have an effect on the size of the imbalance averaging window, and be a stronger predictor.
- Transition to exploring the stochastic control problem.

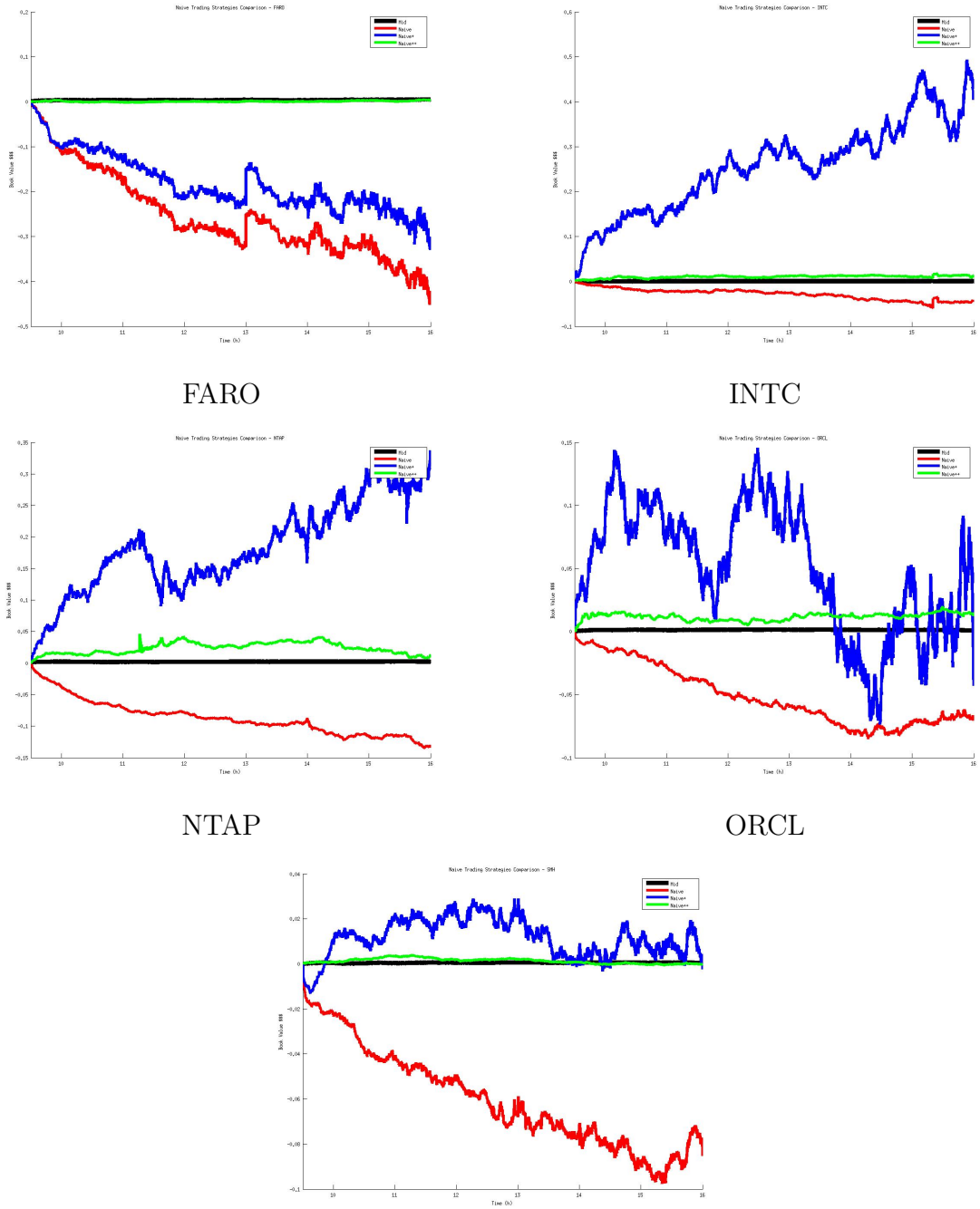

Figure 3: Comparison of Naive (red), Naive+ (blue), and Naive++ (green) trading strategies, with benchmark Midprice (black). Plotted are bookvalues against time of trading day, averaged across trading year.