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 $\dot{S} = f(S, I, t)$.

Recap Next Steps

1. Run cross-validation on the old CTMC imbalance model, also varying the averaging time.
2. Check for a unit root in the imbalance time series using the augmented Dickey-Fuller test, after transforming the data using the logit function.
3. Instead of running a HMM where the hidden state informs the observable imbalance, try having the hidden state affect the transition matrix between imbalance states.
4. Consider a CTMC where the state is actually the pair (I_{k-1}, I_k), with a $k^2 \times k^2$ transition matrix. Cross-validate and compare with regular CTMC.
5. Same as above but with HMM.

Cross-validation of CTMC

To cross-validate the CTMC calibration, the following steps were taken:

1. An imbalance averaging time (in ms) and number of imbalance bins were fixed. The infinitesimal generator matrix $\boldsymbol{G}$ was calculated on the resulting timeseries.
2. An embedded discrete Markov chain transition matrix $\boldsymbol{A}$ was obtained from $\boldsymbol{G}$. This effectively says: conditional on a transition from bin i, what are the transition probabilities to bin j?
3. The stationary distribution, and number (n) of steps required to converge to the stationary distribution, was calculated. That is: for $\epsilon > 0$, calculate n such that $||\boldsymbol{A}^{n+1} - \boldsymbol{A}^n|| < \epsilon$.

4. Find the average number of steps in the timeseries that are required to observe n transitions. This is the size of the timewindow against which to cross-validate.

5. Remove the cross-validation timewindow (call this the "removed series") from the full timeseries (call this the "remaining series"). Calculate two infinitesimal generator matrices $\boldsymbol{G}_{removed}$ and $\boldsymbol{G}_{remaining}$.

6. Calculate two error terms for the resulting matrices:

$$err = \sqrt{\frac{1}{\#trials} \times \sum_{trials} \left(\frac{1}{\#bins^2} \sum_{ij} (\boldsymbol{G}_{remaining}(ij) - \boldsymbol{G}_{removed}(ij))^2 \right) x}$$

$$\boldsymbol{Err} = \sqrt{\frac{1}{\#trials} \times \sum_{trials} (\mathbf{1} - \boldsymbol{G}_{removed} \div \boldsymbol{G}_{remaining})^2}$$

where, for $\boldsymbol{Err}$, division and squaring are entry-wise and not matrix-wise.

num bins averaging time	stationary n	Timewindow size	err	$\boldsymbol{Err}$
3 bins, 100ms	172	13629 steps (5.8% of series)	0.020345	50% - 464%
3 bins, 500ms	150	2982 steps (6.4% of series)	0.012794	37% - 331%
3 bins, 1000ms	120	1412 steps (6.0% of series)	0.010105	33% - 264%
3 bins, 2000ms	121	866 steps (7.4% of series)	0.006786	27% - 185%
3 bins, 3000ms	129	715 steps (9.2% of series)	0.005877	24% - 214%
3 bins, 5000ms	114	497 steps (10.5% of series)	0.004026	15% - 154%
3 bins, 10000ms	134	476 steps (20.3% of series)	0.001945	7% - 84%
3 bins, 20000ms	167	492 steps (42% of series)	0.001326	5% - 33%
5 bins, 100ms	53	2509 steps (1.1% of series)	0.140246	96% - 7419%
5 bins, 500ms	46	554 steps (1.18% of series)	0.034655	66% - 1035%
5 bins, 1000ms	40	289 steps (1.24% of series)	0.023179	63% - 911%
5 bins, 2000ms	37	168 steps (1.44% of series)	0.012056	52% - 1441%
5 bins, 3000ms	38	137 steps (1.76% of series)	0.009366	44% - Inf%
5 bins, 5000ms	32	93 steps (1.99% of series)	0.006778	41% - 529%
5 bins, 10000ms	37	83 steps (3.55% of series)	0.003355	28% - Inf%
5 bins, 20000ms	29	56 steps (4.79% of series)	0.002009	23% - Inf%
5 bins, 30000ms	29	53 steps (6.79% of series)	0.001533	21% - Inf%

HMM Calibration - Initial Guesses

A source of variability in the HMM calibration was the random matrices provided as initial guessed for the Baum-Welch algorithm. (That is, the transition matrix $\boldsymbol{T}_{guess}$ and emission

matrix $\boldsymbol{E}_{guess}$ were chosen to contain random numbers, and then subjected to the constraints of probability matrices, namely having rows sum to 1.)

Instead, $\boldsymbol{E}_{guess}$ was chosen to have all rows equal to the actual observed distribution of imbalance bins. So, for a 3-bin calibration, if the three bins were empirically observed to occur 15%, 60%, and 25% of the time, then each row of $\boldsymbol{E}_{guess}$ would be $\begin{bmatrix} 0.15 & 0.6 & 0.25 \end{bmatrix}$.

Choosing a good guess for $\boldsymbol{T}_{guess}$ is substantially more difficult, and literature on the topic is non-existent. Intuitively, because the states are hidden, no empirical observations should inform the transition values.

We tried the following choice of matrices: diagonal elements were chosen to be equal to a value between 0.9 and 1.0, and off-diagonal elements were all chosen equal such that rows summed to 1. Thus, an example matrix would be:

$$\boldsymbol{T}_{guess} = \begin{bmatrix} 0.98 & 0.01 & 0.01 \\ 0.01 & 0.98 & 0.01 \\ 0.01 & 0.01 & 0.98 \end{bmatrix}$$

We tested calibrating an HMM with 3 hidden states and 3 bins, with initial guesses for diagonal elements ranging from 0.9 to 0.999 in increments of 0.001. Most variation in calibration results was due to re-ordering of the hidden states. Up to reordering, then, the resulting calibrations were:

Diagonal 0.999-0.970	Diagonal 0.969 - 0.900
$\boldsymbol{T}_{est} = \begin{bmatrix} 0.6105 & 0.3579 & 0.0316 \\ 0.0972 & 0.7679 & 0.1349 \\ 0.0120 & 0.2307 & 0.7572 \end{bmatrix}$	$\boldsymbol{T}_{est} = \begin{bmatrix} 0.7565 & 0.2357 & 0.0078 \\ 0.1362 & 0.7843 & 0.0795 \\ 0.0211 & 0.3296 & 0.6493 \end{bmatrix}$
$\boldsymbol{E}_{est} = \begin{bmatrix} 0.0000 & 0.0779 & 0.9221 \\ 0.0181 & 0.9819 & 0.0000 \\ 0.9827 & 0.0147 & 0.0027 \end{bmatrix}$	$\boldsymbol{E}_{est} = \begin{bmatrix} 0.9803 & 0.0186 & 0.0011 \\ 0.0154 & 0.9626 & 0.0219 \\ 0.0000 & 0.0734 & 0.9266 \end{bmatrix}$

Table 1: 10s imbalance averaging period.

Diagonal 0.999-0.941	Diagonal 0.940 - 0.932	Diagonal 0.931 - 0.900
$\boldsymbol{T}_{est} = \begin{bmatrix} 0.9362 & 0.0595 & 0.0044 \\ 0.0412 & 0.9291 & 0.0297 \\ 0.0073 & 0.0965 & 0.8962 \end{bmatrix}$	(bogus results)	$\boldsymbol{T}_{est} = \begin{bmatrix} 0.9291 & 0.0412 & 0.0297 \\ 0.0593 & 0.9363 & 0.0044 \\ 0.0965 & 0.0073 & 0.8962 \end{bmatrix}$
$\boldsymbol{E}_{est} = \begin{bmatrix} 0.9943 & 0.0054 & 0.0003 \\ 0.0038 & 0.9934 & 0.0028 \\ 0.0012 & 0.0073 & 0.9915 \end{bmatrix}$		$\boldsymbol{E}_{est} = \begin{bmatrix} 0.0038 & 0.9934 & 0.0028 \\ 0.9943 & 0.0054 & 0.0003 \\ 0.0012 & 0.0073 & 0.9914 \end{bmatrix}$

Table 2: 1s imbalance averaging period.

2-dimensional CTMC

Next we considered a CTMC that tracks not only the imbalance bin, but jointly the imbalance bin and the price change over a subsequent interval. That is to say, the CTMC state was the pair $(I(t), S(t))$ where $I(t)$ is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and $S(t) = \text{sign}(S(t + \Delta t_S) - S(t))$. For 3 bins, this was encoded into one dimension $Z(t)$ as follows:

$Z(t)$	Bin $I(t)$	Price change $S(t)$
1	Bin 1	< 0
2	Bin 2	< 0
3	Bin 3	< 0
4	Bin 1	0
5	Bin 2	0
6	Bin 3	0
7	Bin 1	> 0
8	Bin 2	> 0
9	Bin 3	> 0

Here bid and ask prices were considered separately rather than considering the change in mid price. Calibrating a CTMC on the two resulting timeseries $Z_{bid}(t)$ and $Z_{ask}(t)$ yielded some interesting results:

imbalance Δt_I: 1000ms, price Δt_S: 500ms

$$\boldsymbol{G}_{Z_{bid}} = \begin{bmatrix} -0.9928 & 0.0217 & 0 & 0.2826 & 0.5870 & 0.0870 & 0 & 0.0145 & 0 \\ 0.0118 & -0.9647 & 0 & 0.1412 & 0.5882 & 0.2000 & 0 & 0.0118 & 0.0118 \\ 0 & 0.0909 & -1.0000 & 0 & 0.3636 & 0.5455 & 0 & 0 & 0 \\ 0.0146 & 0.0005 & 0 & -0.0792 & 0.0562 & 0.0034 & 0.0036 & 0.0006 & 0.0003 \\ 0.0016 & 0.0052 & 0.0003 & 0.0435 & -0.0897 & 0.0300 & 0 & 0.0080 & 0.0011 \\ 0.0003 & 0.0025 & 0.0022 & 0.0053 & 0.0919 & -0.1277 & 0 & 0.0017 & 0.0237 \\ 0 & 0.0345 & 0 & 0.4138 & 0.4138 & 0.1034 & -1.0000 & 0.0345 & 0 \\ 0.0179 & 0.0179 & 0 & 0.2232 & 0.5536 & 0.1250 & 0.0089 & -0.9732 & 0.0268 \\ 0.0094 & 0.0189 & 0 & 0.1132 & 0.5189 & 0.3113 & 0 & 0.0094 & -0.9811 \end{bmatrix}$$

$$\boldsymbol{G}_{Z_{ask}} = \begin{bmatrix} -0.9915 & 0.0169 & 0 & 0.2881 & 0.5678 & 0.1017 & 0 & 0.0169 & 0 \\ 0.0106 & -0.9681 & 0 & 0.1277 & 0.5638 & 0.2340 & 0 & 0.0213 & 0.0106 \\ 0 & 0.0588 & -1.0000 & 0 & 0.2941 & 0.5882 & 0 & 0 & 0.0588 \\ 0.0121 & 0.0005 & 0 & -0.0775 & 0.0580 & 0.0034 & 0.0027 & 0.0005 & 0.0003 \\ 0.0016 & 0.0058 & 0.0002 & 0.0448 & -0.0898 & 0.0297 & 0 & 0.0065 & 0.0011 \\ 0.0003 & 0.0025 & 0.0039 & 0.0059 & 0.0907 & -0.1311 & 0 & 0.0008 & 0.0270 \\ 0 & 0.0476 & 0 & 0.1905 & 0.5714 & 0.1429 & -1.0000 & 0.0476 & 0 \\ 0 & 0.0440 & 0 & 0.1319 & 0.6374 & 0.1429 & 0 & -0.9890 & 0.0330 \\ 0.0085 & 0.0254 & 0.0085 & 0.0847 & 0.5169 & 0.3220 & 0 & 0.0169 & -0.9831 \end{bmatrix}$$

Using these matrices, we can compute conditional probabilities. For example, we can ask: conditional on being in bin 1 (more bid volume than ask) and on the bid price changing, what is the probability that the change will be greater than 0? less than 0?

Again, converting the generator matrix to the embedded discrete time Markov chain matrix proves enlightening for these calculations:

$$\boldsymbol{A}_{Z_{bid}} = \begin{bmatrix} 0 & 0.0219 & 0 & 0.2847 & 0.5912 & 0.0876 & 0 & 0.0146 & 0 \\ 0.0122 & 0 & 0 & 0.1463 & 0.6098 & 0.2073 & 0 & 0.0122 & 0.0122 \\ 0 & 0.0909 & 0 & 0 & 0.3636 & 0.5455 & 0 & 0 & 0 \\ 0.1839 & 0.0065 & 0 & 0 & 0.7097 & 0.0435 & 0.0452 & 0.0081 & 0.0032 \\ 0.0174 & 0.0581 & 0.0029 & 0.4855 & 0 & 0.3343 & 0 & 0.0891 & 0.0126 \\ 0.0022 & 0.0197 & 0.0175 & 0.0416 & 0.7199 & 0 & 0 & 0.0131 & 0.1860 \\ 0 & 0.0345 & 0 & 0.4138 & 0.4138 & 0.1034 & 0 & 0.0345 & 0 \\ 0.0183 & 0.0183 & 0 & 0.2294 & 0.5688 & 0.1284 & 0.0092 & 0 & 0.0275 \\ 0.0096 & 0.0192 & 0 & 0.1154 & 0.5288 & 0.3173 & 0 & 0.0096 & 0 \end{bmatrix}$$

$$P(S(t+\delta t_S) > S(t) \mid S(t+\delta t_S) \neq S(t) \;\&\; I(t)=1) = \frac{P(S(t+\delta t_S) > S(t) \;\&\; I(t)=1)}{P(S(t+\delta t_S) \neq S(t) \;\&\; I(t)=1)}$$

$$xy$$