How does Market Microstructure affect the Performance of Deep Learning Models at High Frequency?

Nicholas Westray
nicholas.westray@nyu.edu

NYU Courant & AllianceBernstein - Multi Asset Solutions
(Joint work with P. Kolm (NYU) and J. Turiel (UCL))

Advances of ML Approaches for Financial Decision Making
and Time Series Analysis
28th March 2022

Outline

Introduction

- The focus today will be on Systematic Trading in US Equities at high frequency.
- Generating Alpha Signals (return predictors) is critical in this endeavour for both buy and sell sides
  - (Sell Side) - Improving performance in next generation trading algorithms.
  - (Buy Side) - Developing and deploying profitable HF strategies.
- The holy grail for many of these firms is to take as raw input a raw limit order book (or a collection of order books) and produce high frequency price/return forecasts

Signal Generation

- This type of alpha generation is enormously challenging for a number of reasons :
  - Enormous amounts (TB/PB) of data.
  - Specialist infrastructure required to store, process and analyze.
  - Data is noisy, non-stationary and fat tailed.
  - Field is extremely competitive, every single one of your competitors is trying to do the same thing with the same data.
- Current state of the art is to employ quants to extract and handcraft features using expert domain knowledge which then become high value IP.

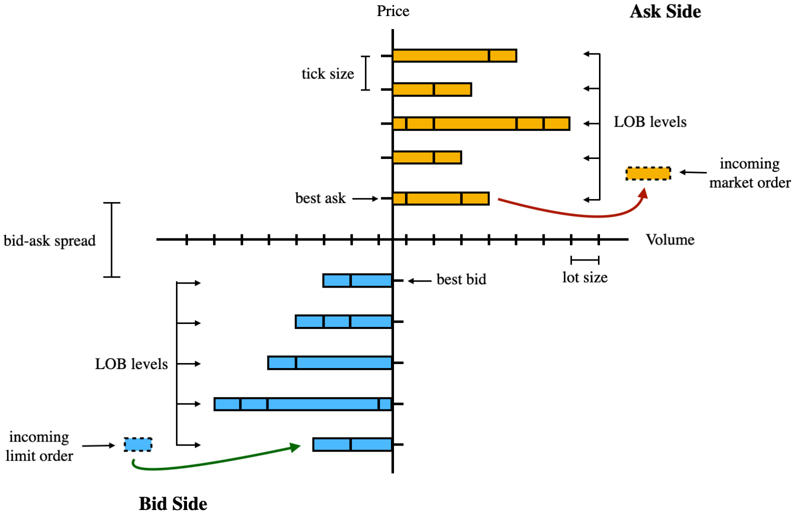
The Bitter Lesson

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation.
.... Researchers seek to leverage their human knowledge of the domain [to improve performance], but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to.

Rich Sutton - The Bitter Lesson, March 2019[1]

[1] http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Limit Order Book (LOB) Review



General Problem Statement

- Focus on the top 10 (non-zero) levels of the LOB and define the vector,

$$x_t := (a_t^1, v_t^{1,a}, b_t^1, v_t^{1,b}, \ldots, a_t^{10}, v_t^{10,a}, b_t^{10}, v_t^{10,b})^\top \in \mathbb{R}^{40}, \tag{1}$$

- For each stock and time t, fix a horizon h and consider the timeseries regression problem

$$r_{t,t+h} = g(x_t, x_{t-1}, \ldots, x_{t-W}) + \varepsilon_t \tag{2}$$

- $r_{t,t+h}$ are the forward returns (hereafter r_t).
- The function g is a Neural Network,
- W denotes the length of the lookback window (typically 100),

Literature Review

- Due to success of NNs in classification, existing literature reformulates the regression problem as one of classification.
- There are three main groups of authors who have focussed on this problem,
  - Tsantekedis et al. [4] - Focus on 5 Finnish stocks (FI-2010), investigate different architectures.
  - Sirignano et al [5] - Focus on S&P 500 with a single architecture, investigate questions of universality.
  - Zhang et al [3] - Focus on 5 LSE stocks, investigate sophisticated CNN-LSTM Inception networks and introduce new hardware to fit these models (IPUs).

Outstanding Practitioner Questions

- Should I transform the raw LOB before inputting into the NN?
- Are there recommendations for practitioners when choosing between architectures?
- Can I look at model predictive performance in terms of stock characteristics/microstructural properties?
- What kind of horizon do these alphas have?

Our Setup/Contribution

- Recast the problem as standard regression and include multiple horizons in the output.

$$r_t := (r_{t,1}, \ldots, r_{t,H})^\top \in \mathbb{R}^H, \quad \text{where } H \geq 1\,. \tag{3}$$

- We refer to the forecasts as an *alpha term structure* at time t. Our forecasting models take the form

$$r_t = g(x_t, x_{t-1}, \ldots, x_{t-W}) + \varepsilon_t \tag{4}$$

- We are going to :
  - Understand the effects of different RHS in the regression.
  - Compare multiple architectures across a large set of symbols
  - Explore the relationship between stock characteristics and model predictive power.

Data Description

- We use data for the time period January 1, 2019 through January 31, 2020 (LOBSTER, WRDS) for 115 stocks from Nasdaq.
- Number of updates is not constant across stocks, so we define returns in multiples of Δt or *number of average price changes* where

$$\Delta t := \frac{2.34 \cdot 10^7}{N}, \tag{5}$$

- The numerator is the number of milliseconds in a trading day and the denominator N denotes the average number of non-zero tick by tick mid-price returns.
- Insert a fixed latency buffer of 10ms for all intervals to mimic production setting.

Model Universe

- ARX - Autoregressive with exogenous features (linear model)

$$r_t = w_0 + \sum_{i=1}^{100} v_i^\top x_{t-i} + \varepsilon_t \,,$$

- MLP - Multilayer Perceptron (4 layers). Briola et al. [8]
- LSTM - Long Short Term Memory Network (128 hidden units)
- LSTM-MLP - LSTM (128 hidden units) $\rightarrow$ MLP (64 hidden units)
- LSTM (3) - Deep LSTM (150 hidden units). Model of Cont and Sirignano. [5]
- CNN-LSTM - State of the art model proposed by Zhang et al. [3]

Model Hyperparameters

Model	Input	Number of layers	Number of parameters$^{(*)}$	Learning rate	Batch size	Training epochs	Early stopping
ARX	OF[1]	1	2.0×10^3	10^{-4}	256	50	Yes
CNN-LSTM	OF	27	1.3×10^5	10^{-3}	256	50	Yes
LSTM	OF	2	1.0×10^5	10^{-5}	256	50	Yes
LSTM (3)	OF	4	4.6×10^5	10^{-5}	256	50	Yes
LSTM-MLP	OF	3	8.4×10^4	10^{-5}	256	50	Yes
MLP	OF	4	1.3×10^6	10^{-5}	256	50	Yes
ARX	LOB	1	4.0×10^3	10^{-4}	256	50	Yes
CNN-LSTM	LOB	27	1.4×10^5	10^{-3}	256	50	Yes
LSTM	LOB	2	1.1×10^5	10^{-5}	256	50	Yes
LSTM (3)	LOB	4	4.7×10^5	10^{-5}	256	50	Yes
LSTM-MLP	LOB	3	9.4×10^4	10^{-5}	256	50	Yes
MLP	LOB	4	2.3×10^6	10^{-5}	256	50	Yes

Table 1: Summary of the inputs and hyperparameters used. $^{(*)}$The number of parameters are approximated to the nearest order of magnitude and truncated for readability.

[1] OF denotes Order Flow, a well known differencing transform applied to LOBs

Model Fitting

- To mimic a real life production setting we perform (for each symbol) rolling fits over our time period.
- We choose a (1,4,1) configuration
  - The first week is for validation (early stopping).
  - The middle 4 weeks are for training.
  - The final week is for out of sample testing.
- We apply winsorization and Z-scoring to all independent and dependent variables used in regressions.
- We use the ADAM optimizer & Tensorflow (Keras).
- All computations leverage the GPUs on the NYU Greene[2] and Hudson[3] HPC environments

[2] NYU Greene
[3] NYU Hudson.

Results

- We use out of sample r-squared, (R^2_{OS}) as the evaluation metric, calculated daily for each :
  - Model
  - Stock
  - Horizon
  - Out of sample date
- First we look at dependence on horizon, so average out stock and dates.
- This gives us a curve across different horizons, for each model.

Order Flow Imbalance or Limit Order Book Inputs?

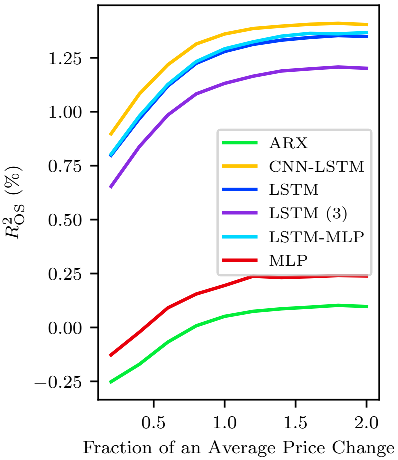


Figure 1: Left Panel OF. Right Panel LOB.

Discussion

- Recall the key questions we set out to address :
  - Understand the effects of different RHS in the regression.
  - Compare multiple architectures across a large set of symbols
- OF input is clearly better than LOB - stationarity of inputs is important.
- LSTM based models outperform non-LSTM models.
- Depth/CNN layers do not seem to outperform plain LSTM after converting to OF.
- Significant alpha at all horizons for the OF models. Small R^2 but high profitability due to shortness of horizon.

Stock Characteristics

- We have seen that a regular LSTM model with OF input is an excellent (non-complex) model.
- Use the following stock characteristics to study model performance :
  - Tick Size - Fraction of Time that spread = \$0.01 (Large tick stocks approximately 1)
  - Log Updates - Log Number of updates/day
  - Log Trades - Log Number of trades/day
  - Log Price Chg - Log Number of price changes/day.
- All numbers computed per stock by averaging across the time period.

Methodology

- We fix the model to be (LSTM, OF), average across horizons and out of sample data points.
- We are left with a single R^2_{OS} per stock (115 points).
- These are plotted against the stock characteristics in a scatter
- Results are almost identical for (CNN-LSTM, OF)

Cross Sectional Performance

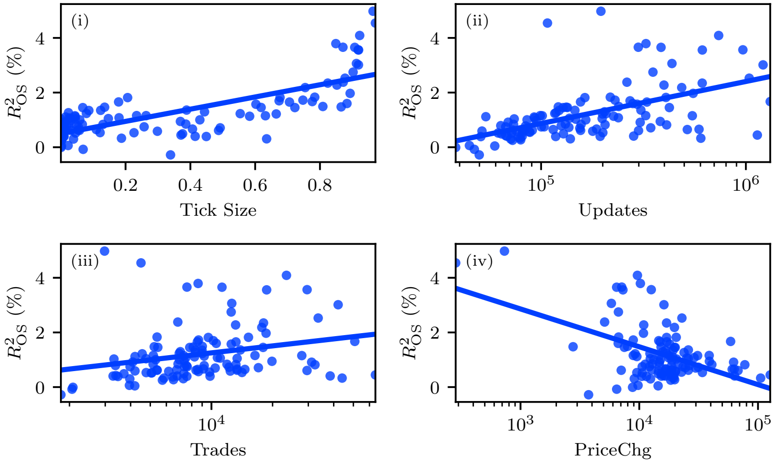


Figure 2: Cross Sectional Performance - (LSTM, OF) model.

Discussion

- There are clear dependencies on updates, tick size and trades
- Regression analysis shows that the best characteristic is in fact a combination, Log(Updates/PriceChg). This explains performance (R^2_{OS}) with an adj. R^2 of 75%.
- Why?
  - When Log(Updates/PriceChg) is large, you have a stable order book with lots of updates per price change. This is also a property of Large tick stocks (hence the correlation).
  - You have lots of data and complicated patterns forming between time series of imbalances and price changes.
  - The LSTM model is able to capture and model this.
- We have good O/S performance across the vast proportion of our universe

Predicting Performance

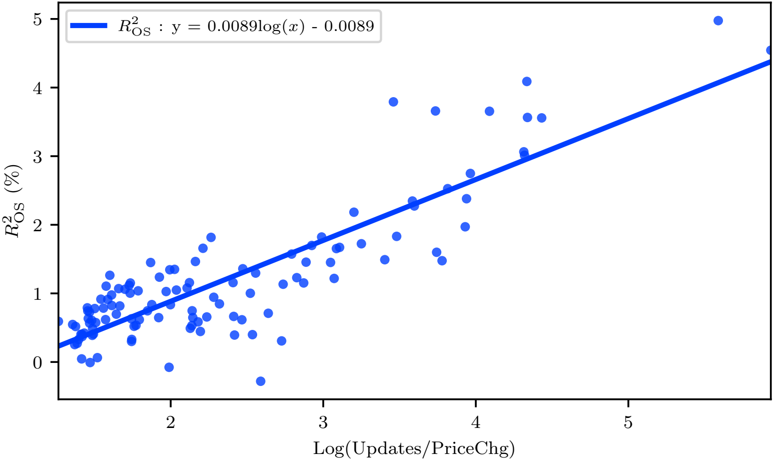


Figure 3: Cross Sectional Performance - Log(Updates/PriceChg).

Additional Results/Robustness Checks

- Many additional questions addressed in the paper - Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book
- How far ahead can we predict returns? (about 2-3 price changes).
- How sensitive are the results to the fixed window length $W = 100$? (interestingly not very)
- What if we use an OFI (Order Flow Imbalance) or Volume only LOB RHS? (removing prices helps but OF is the best)
- Motivation for results in terms of inductive biases - an interesting new concept from the ML literature.

Conclusions

- Built a framework to evaluate different regression inputs and deep learning architectures for return predictions.
- Shown that stationarity of the inputs is critical to getting good outcomes.
- Shown that the predictive power depends strongly on microstructural properties of the underlying stock, specifically on the ratio of number of updates and the price changes.
- Evidence that we need more effort on finding the best architecture for return prediction as in our experiments simple ones seem to perform as well as complicated ones.

References

- Hornik, K. "Approximation capabilities of multilayer feed forward neural networks".
- Hochreiter, S. and Schmidhuber, J. "Long short term memory"
- Zhang et al. "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books"
- Tsantekidis et al. "Using Deep Learning for price prediction by exploiting stationary limit order book features"
- Cont, R. and Sirignano, J. "Universal Features of Price Formation in Financial Markets: Perspectives From Deep Learning"
- Cont et al. "The Price Impact of Order Book Events"
- Xu et al. "Multi-Level Order-Flow Imbalance in a Limit Order Book"
- Briola et al. "Deep Learning modeling of Limit Order Book: a comparative perspective"