VWAP Prediction for Algorithmic Trading Using Machine Learning

Vikas Natesh, Jiangting Yue

December 20, 2018

Abstract

High frequency trading (HFT) is a type of algorithmic trading using high performance hardware to execute a large number of financial trades in a very short period of time such as seconds or even a nano seconds. Over the past decade, tools from machine learning have permeated this field and have augmented or replaced several traditional trading strategies dependent on time-series analysis such as ARMA. Predicting future prices in order to execute directional trades is difficult due to the noisy nature of stock price time series. Instead of predicting price directly, we aimed to predict the future direction and real value of the Volume Weighted Average Price (VWAP) using features from a limit order book. A limit-order book is simply a record of buy and sell orders made by market participants during trading hours. We applied several classical algorithms in machine learning such as the support vector machine, logistic regression, random forest, least absolute shrinkage and selection operator (LASSO regression), and long short-term model (LSTM) and analyzed the effectiveness of these models.

1 Introduction

When market participants want to buy or sell a certain amount of stock, there are two types of orders they can use, the market order and the limit order. A market order is a buy or sell order to be executed immediately at the current market price. A limit order is an order to buy a security at no more than a trader-specified price, or to sell a security at no less than a trader-specified price. Importantly, a market order guarantees execution but does not guarantee the best price for the participant. On the other hand, a limit order guarantees the trader's desired price but does not guarantee order execution. This also means that a limit order may not be executed immediately if the current price is not the desired bid/ask price. The limit-order book will contain a record of both executed and unexecuted limit orders maintained by the exchange. Over the course of the trading day, different limit orders will be sent by different participants to the exchange, creating a time series of order bids/asks and volumes. Figure 1 displays the bid (green) and ask (red) volumes for 5 different price levels of a particular stock at single point in time.

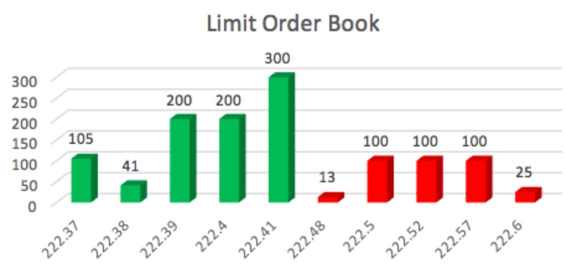

Figure 1: 5-price-level limit order at a single time point

Volume Weighted Average Price (VWAP) of a stock is the ratio of the total dollar value traded to total volume (number of shares) traded over a certain time interval. It can be considered as another way to average price information for a time interval.

$$VWAP(n) = \frac{\sum_{i=1}^{n} P_i * Q_i}{\sum_{i=1}^{n} Q_i} \tag{1}$$

Where $i = 1$ indicates the start of the interval and

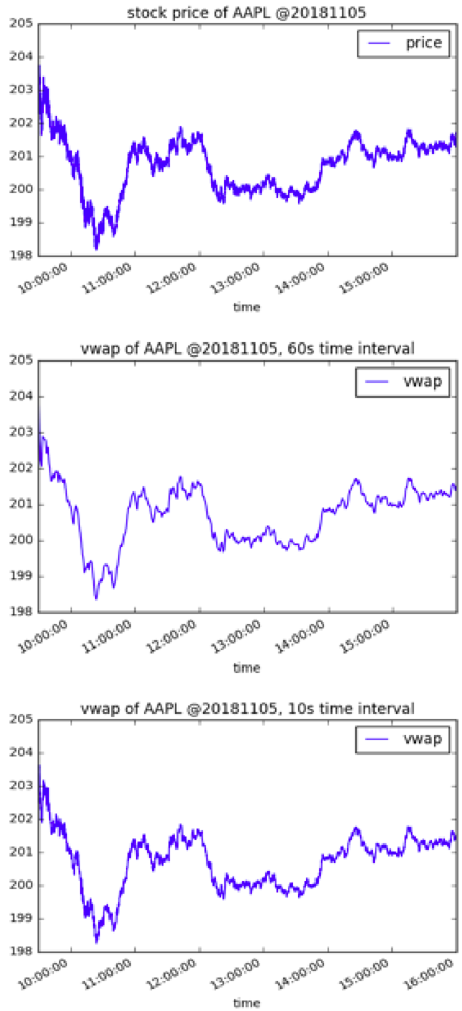

Figure 2: Time series plot of APPL price and VWAP

n is the interval size. The reason we want to predict VWAP instead of price is that the time series of stock price is quite noisy and could be very random in short time intervals. VWAP is a essentially a way to smooth the time series. Figure 2 displays a time series plot of AAPL (Apple Inc.) price and VWAP for 10sec and 60sec time intervals over the course of the trading day 2018-11-05. We can see that when the time interval increases, the VWAP series becomes smoother.

The standard deviation of AAPL stock price series at 2018-11-05 is 1.152. Meanwhile. the std. dev. of the AAPL VWAP series at 2018-11-05 is approx 0.84, as seen in the table:

We can see that the VWAP series is not as volatile as the stock price series, giving us the intuition that VWAP series may easier to predict than pure stock price series.

Time Interval	VWAP σ
10 sec	0.8414
30 sec	0.8424
60 sec	0.8437

2 Dataset and Features

We obtained raw order feeds for two stocks, Apple (AAPL) and Goldman Sachs (GS) using the Trading Physics service. This contains tick by tick trade event data from ArcaBook, BATS, Direct Edge, NASDAQ TotalView, and NYSE exchanges from 10-17-2018 to 12-04-2018 (33 trading days). Each row of data represents a trading event with milliseconds labeled and event types defined as follows: 1. limit order placed, 2. limit order cancelled partially, 3. limit order deleted, 4. limit order executed partially, 5. limit order executed fully. Using this raw data, we constructed the limit order book and computed the VWAP time series over three different intervals; 10s, 30s and 60s. Each data point (row) represents a snapshot of market status at that specific time point. Two consecutive data points have a fixed time difference (10s, 30s and 60s). Below is detailed explanation for each feature in our dataset.

['msec'] : milliseconds since midnight of the specific trading day

['bp1', 'bp2', 'bp3', 'bp4', 'bp5'] : 5 bid price levels. bp1 is the best bid price

['ap1', 'ap2', 'ap3', 'ap4', 'ap5']: 5 level ask price, ap1 is the best ask price

['bv1', 'bv2', 'bv3', 'bv4', 'bv5']: five volumes corresponding to 5 bid prices

['av1', 'av2', 'av3', 'av4', 'av5']: five volumes corresponding to 5 ask prices

['delta-bp1', 'delta-bp2', 'delta-bp3', 'delta-bp4', 'delta-bp5'] : change in bid price from the previous data point for 5 levels. For example, for delta-bp1:
$\Delta bp^1_t = bp^1_t - bp^1_{t-1}$

['delta-ap1', 'delta-ap2', 'delta-ap3', 'delta-ap4', 'delta-ap5']: change in ask price from the previous data point for 5 levels. For example, for delta-ap1:
$\Delta ap^1_t = ap^1_t - ap^1_{t-1}$

['delta-bv1', 'delta-bv2', 'delta-bv3', 'delta-bv4',

'delta-bv5']: change in bid volume from the previous data point for 5 levels. For example, for delta-bv1:
$\Delta bv_t^1 = bv_t^1 - bv_{t-1}^1$

['delta-av1', 'delta-av2', 'delta-av3', 'delta-av4', 'delta-av5']: change in ask volume from the previous data point for 5 levels. For example, for delta-av1:
$\Delta av_t^1 = av_t^1 - av_{t-1}^1$

['mean-volumn-diff']: difference between mean bid volume and mean ask volume over 5 levels i.e. $\frac{1}{5}\sum_{j=1}^{5} bv_t^j - \frac{1}{5}\sum_{j=1}^{5} av_t^j$

['spread']: difference between best bid price and best ask price i.e. $bp_t^1 - ap_t^1$

['vol-unb1', 'vol-unb2', 'vol-unb3', 'vol-unb4', 'vol-unb5']: This is the volume imbalance i.e. the difference between bid and ask volume divided by bid volume at each level. For example, for level 1:
vol-unb1 $= \frac{bv_t^1 - a_t^1}{bv_t^1}$

['mom-bp1', 'mom-ap1']: momentum of price at the best bid and ask level. For example, the momentum of bid price over 5 data points is
$mom_5^1 = \frac{bp_t^1 - bp_{t-5}^1}{bp_{t-5}^1}$

['vola-bp1', 'vola-ap1']: volatility (std dev) of bid and ask price at best bid and ask level.

Since our aim is to predict both direction and real value of the VWAP at the next time step (every 10s, 30s, or 60s), we have a binary output variable 'vwap-d' taking on values -1 or 1, and a real output variable 'vwap-v' representing the time series shifted in the future by 10, 30, or 60 sec.

3 VWAP Moving Direction Prediction

Given the rich set of features above, we performed feature selection via principal component analysis and random forest to narrow down the feature set to that which had the most predictive power. We used the explained variance ratio to choose 30 principal components. These optimal components achieved a 99% explained variance ratio. The random forrest method extracted 14 features. Figure 3 displays the accuracy of AAPL VWAP direction prediction using logistic regression on different feature sets.

Features	Training	CV	Test
PCA(30)	0.64685	0.64306	0.64827
RF(14)	0.64731	0.65846	0.64742
Original Data	0.63987	0.64194	0.63194

Figure 3: accuracy table

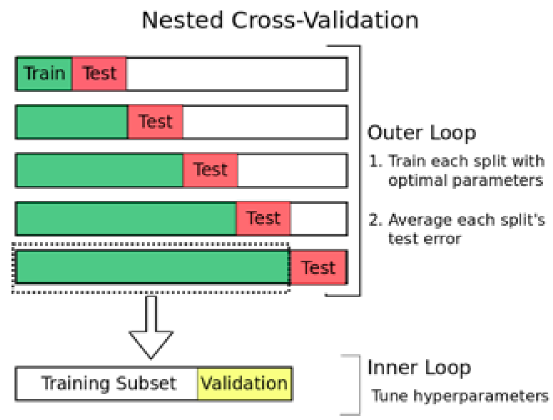

Figure 4: cross-validation

From this data, it is apparent that PCA and Random Forest feature sets have a very small predictive edge over using the full feature set. However, it is difficult to tell which feature selection method is better. Since PCA(30) had a better test performance, we choose to use the PCA(30) in the actual training step.

Since the VWAP series we want to predict is a time series, there exists a relation between data points at each time step. We perform cross validation to avoid the situation of using a future price to predict a past price. Figure 4 illustrates our method of cross validation. The outer loop continues to introduce a new test set each iteration while the inner loop performs grid search to optimize parameters of our models.

After cross validation, we applied 3 fundamental algorithms to predict VWAP direction: SVM, logistic regression, and random forest. Figure 5 displays the accuracy of the 3 models in predicting 10 sec and 60 sec VWAP.

The 60s VWAP prediction has poor predictive accuracy, only about 50% which is essentially a coin flip. One possibility is that the sample size (number of time steps) is too small. For example, Future 10 second vwap prediction

	Training	CV	Test
RF	0.6468	0.6436	0.6482
SVM(Linear)	0.6483	0.6481	0.6438
LR	0.6474	0.6489	0.6451

(RF: Random Forest, LR: Logistic regression)

Future 60 second vwap prediction

	Training	CV	Test
RF	0.5558	0.4962	0.502
SVM(Linear)	0.5077	0.5072	0.4907
LR	0.5158	0.501	0.4934

(RF: Random Forest, LR: Logistic regression)

Figure 5: model accuracy

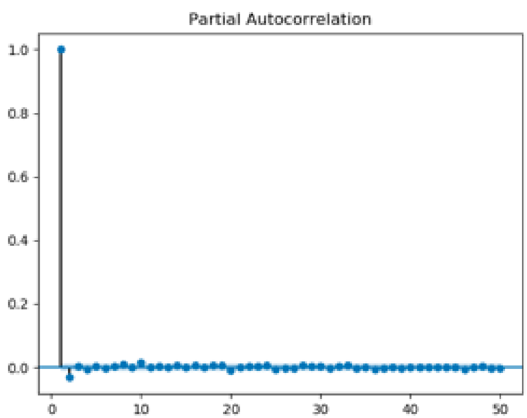

Figure 6: partial auto-correlation

in the 10 second case, the number of sample points is 6 times as large as the 60 sec case, allowing the 10 sec. case to realize a slightly better accuracy of 65% i.e. a 15% trading edge. In addition, 10 sec. data is more granular and contains more information on the VWAP time series. The 60 second case may perform better if we use more than 33 trading days. However, there are no significant differences in accuracy between the 3 algorithms. 65% accuracy on a 10 sec. VWAP seems to be our performance limit for these three algorithms.

Another reason why the 60 sec. case has poor performance is that the VWAP time series may itself be too noisy. Figure 6 displays the partial auto-correlation graph of the VWAP time series in our 60s time interval data. We can see that the only significant lag is lag 1 while the remaining 49 lags are statistically insignificant. The partial

group	predict Prob.	True Prob.
1	26.79%	19.31%
2	35.84%	28.99%
3	39.59%	34.92%
4	42.96%	43.00%
5	47.61%	45.21%
6	52.97%	52.00%
7	57.10%	57.93%
8	60.01%	63.22%
9	68.41%	70.62%
10	73.32%	76.18%

Figure 7: probability of VWAP decreasing at next time step

auto-correlation graph tells us there is fair amount of randomness in the 60s data, and may contribute to poor performance of the models for a 60s time interval.

In the random forest and logistic regression models, we can obtain the mathematical probability associated with the binary [-1,1] prediction at each time step. We found this information to be critical in that it provided a more accurate output variable to predict. After we trained a random forest model, we used it to predict the probability of [-1,1] at each time step. We divided the testing set into 10 groups based on the probability p of VWAP decreasing at a particular time step. Figure 7 shows a table of these groups, ordered in ascending order of p. We then performed frequency counting in each group to obtain the true probability of VWAP decreasing at each time step. Figure 8 indicates that the predicted and true probabilities are quite similar for each group with a mean-squared-error of 0.0122. Thus, predicting the probability of [-1,1] has much better accuracy than predicting direction [-1,1] alone. This is a reasonable outcome since the predictability of the stock market is not constant and one could use this probability estimation to characterize the varying predictability. From a trading perspective, we could take a trading action when we predict p to be in either group 1 or group 10, since these groups represent more certain events. Specifically, a prediction in group

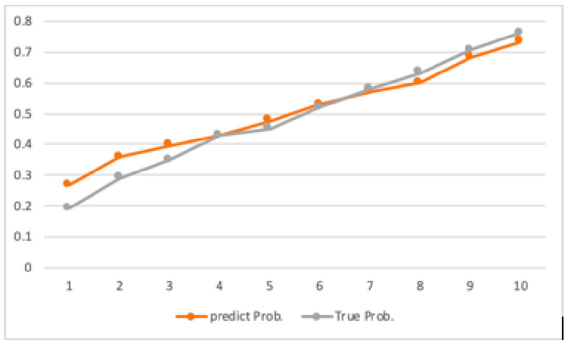

Figure 8: predicted vs true probability of VWAP decreasing

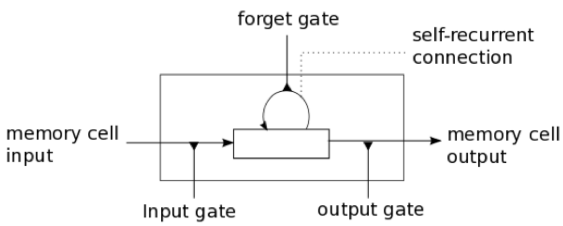

Figure 9: single LSTM unit

1 indicates that VWAP is not likely to decrease at the next time step while a prediction in group 10 indicates that VWAP is very likely to decrease.

4 VWAP Real Value Prediction

To predict the real value of VWAP at the next 10, 30, or 60sec time step, we used two regression algorithms: LASSO regression and the long short-term memory (LSTM) recurrent neural network. The LASSO algorithm is shown below:

$$\min_{w} \quad F(w,b) = \lambda \parallel w \parallel_1 + \sum_{i=1}^{m} w * x_i + b - y_i^2$$

After parameter tuning, we set the regularization parameter $\lambda = 1$. LASSO acts as a feature selector since coefficients w_i for some features will be set to 0 and have no effect on the prediction.

An LSTM network is a cyclic directed graph of nodes called memory units. Each unit contains an input gate, output gate, self-recurrent connection (cycle), and a forget gate. This structure allows an LSTM to remember or forget its previous state, depending on the current environment, making it ideal for time-series data. We built an LSTM

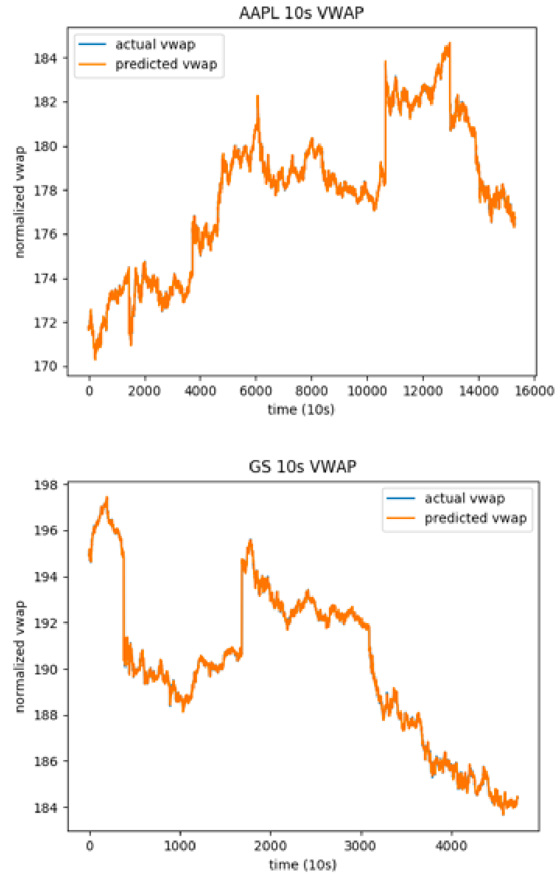

Figure 10: real vs predicted VWAP in LASSO

network with the following parameters: tanh activation, sigmoid recurrent activation, ADAM algorithm for learning rate update, 50 unit hidden layer, batch size 73 during stochastic gradient descent backpropagation. Training was performed for 40 epochs and validated using a training set comprised of the first 80% of recorded trading days and a test set composed of the remaining 20%. Figure 10 shows the true VWAP time series as well as the VWAP time series predicted by the LASSO model, for Apple and Goldman Sachs stocks. The model performed extremely well, with a negligible MSE and an R^2 goodness-of-fit score of 0.999. However, this also suggests overfitting of the data, which may be attributed to small sample size.

Figure 11 shows the training and test error of the LSTM model over 40 epochs. These errors converge to approx 1% after 40 epochs. Figure 12 shows the true and predicted normalized 10 sec VWAP time series of GS using the LSTM network. This model also performed very well, with a MSE

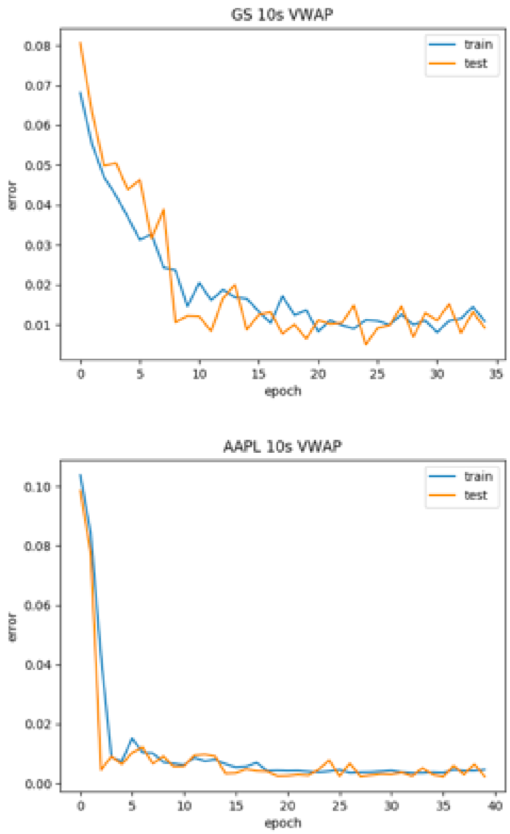

Figure 11: test vs train error over all epochs in LSTM

of 0.00031 and an R^2 goodness-of-fit score of 0.946. Although the fit is very strong, it is difficult to tell whether this is due to over-fitting.

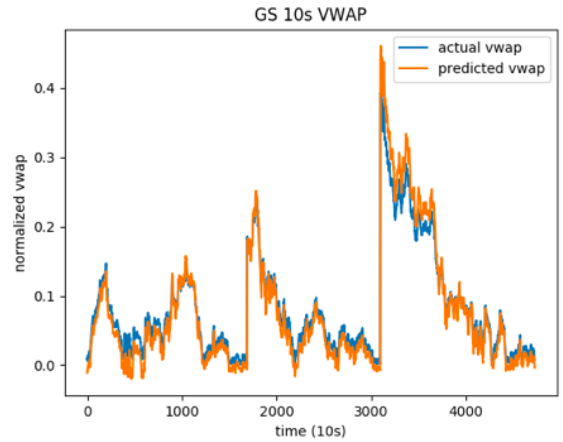

Figure 12: real vs predicted VWAP in LSTM

5 Discussion

The models described here can certainly be extended. For example, although only single classifiers were tested when trying to predict VWAP direction, we could've used an ensemble of classifiers to perform boosting. This may have yielded a better accuracy score. LSTM seems to be the most reliable predictive method. Stock time series are non-stationary so ARMA model assumptions do not hold. Average reduction in error rates obtained by LSTM is between 84% - 87% when compared to ARIMA, indicating the superiority of LSTM to ARIMA. Stationarity implicitly assumes that the same relationship holds between the N predictor points and the following N+1st point you're trying to predict. When using an autoregressive lag of N, we cannot learn dependencies over intervals longer than N. Since LSTM is a neural network, it is is capable of learning non-linear structure and non-stationarity becomes less of an issue for short term trades. However, LSTM has difficulty learning long-term patterns since gradients are only back-propagated a finite number of time steps. Nonetheless, LSTM can prove useful for the type of short-term prediction possible in the stock market. For example, one can execute a simple directional strategy based on predicting VWAP of a particular stock in the next 10/30/60 sec. If the current market price is above the predicted VWAP, it is a good intraday price to sell, as we expect the price to mean-revert towards the VWAP in the future. Likewise, if the price is below the predicted VWAP, it is a good intraday price to buy i.e. we can buy now and sell when the price increases (mean-reverts) towards the VWAP. Another strategy involves market making where a trading firm voluntarily agrees to buy/sell at a certain level. Essentially, a market making firm acts as a liquidity provider to the market. In this strategy, we treat the predicted VWAP as the "true" value of the Stock and then place both a bid and an ask limit order above and below this ?true? value. The hope is that these orders some counter-party accepts these orders, allowing the market making firm to make a profit from bid-ask spread.

6 References

- Long Short-Term Memory. Sepp Hochreiter, Jurgen Schmidhuber. Neural Computation 9(8):17351780, 1997

- Market Impact with Autocorrelated Order

Flow under Perfect Competition. Jonathan Donier.

- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. MIT Press, 2012.

- Modeling high-frequency limit order book dynamics with support vector machines. Alec N.Kercheval and Yuan Zhang. https://www.math.fsu.edu/ aluffi/archive/paper462.pdf

- Deep Learning for Limit Order Books. Justin A. Sirignano. https://arxiv.org/pdf/1601.01987.pdf. May 2016

- Forecasting Economic and Financial Time Series: ARIMA vs LSTM. Sima Namin and Akbar Namin. https://arxiv.org/pdf/1803.06386.pdf. March 2018

- Machine Learning for Market Microstructure and High Frequency Trading. Michael Kearns and Yuriy Nevmyvaka. High Frequency Trading - New Realities for Traders, Markets and Regulators. 2013

- Cartea, A., Jaimungal, S., and Penalva, J. (2015). Algorithmic and high frequency trading. Cambridge University Press, chapter 11.

- D. Palguna and I. Pollak. Mid-price prediction in a limit order book. IEEE Journal of Selected Topics in Signal Processing, 10(6):10831092, 2016.

- Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference. C. Lee Giles, Steve Lawrence, and, Ah Chung Tsoi. https://clgiles.ist.psu.edu/papers/MLJ-finance.pdf.

- http://www.tradingphysics.org/Feeds/