CS230

Price Formation in Stock Markets Using Fundamental Analysis

Paul Boehringer*
Department of Statistics
Stanford University
paulb360@stanford.edu

November 17th, 2020

Abstract

This paper describes a model to make predictions of asset prices. It focuses on stocks in the New York Stock exchange, though the methods presented in this paper apply directly to any stock traded world wide. The final model has two parallel networks. One LSTM that considers the price history of a given asset as an input in a daily time domain along with potentially correlated assets such as gold, forex pairs, and oil. A second parallel LSTM with input of the asset's financial status at less discrete quarterly intervals is also trained. The concatenated outputs of these two LSTMs are used to make predictions on the characteristics of an assset's price features for the next day. The model achieves superior results compared to using a lagged linear regression with the timeseries data.

1 Introduction

Financial markets are an increasingly competitive environment where everyone's goal can be summarized by the same cliche phrase; *buy low, sell high*. According to the world bank the total value of the world markets sits at over $100 trillion dollars at the time of this paper. This project aims to build an algorithm which can continuously outperform in a market filled with competition trying to do the same.

Markets are mostly made by a series of information of what happened in the past. There are various kinds of traders, some make decisions on nothing but the shape of an asset's price history; this is called *technical analysis*. Others will focus on the financial state of a company looking at trends in sales, liabilities, and how cash flows through the company more holistically; this is called *fundamental analysis*. One of the other largest methods of trading is macro-economic strategies that focus on investing in industries which analysts believe will outperform others and trying to capture growth of that industry to out preform the market average. Within all these groups there are teams that will stick to each of their respective investment paradigms with a giant spectrum on the degree of technicality.

LSTMs offer the ability to consider prior information in a high dimension and learn complex relationships in the features to make predictions about the future. An LSTM will be used in this paper to combine data traditionally used in the three domains of investing mentioned above. Price history of assets will be incorporated in an effort to account for the effect that technical analysis will have on its future price. Quarterly financial statements and insider trading info is included to account for the effect fundamental analysis would have on the asset. Finally metrics which are typically used for macro economic analysis will be used to attempt to account for different states of the economy through trading; this is called the market regime in the trading world. With these points in consideration it will be good to cover some significant literature on how deep learning is affecting the world of trading.

*any readers can contact me at paulboehringer0989@gmail.com with any questions

2 Related work

One of the first significant works that merged the field of finance and deep learning is when Cont et al., came out with work showing that an LSTM can use tick by tick order book data for extreme increases in SOTA accuracy in predicting the next direction of an equity's price movement [1]. This clear evidence of universal patterns in price formation was not received with wide acceptance as it goes directly against the efficient markets hypothesis that states these kinds of patterns will disappear as more traders catch on and take advantage of them. However the world of technical analysis welcomed this line of thinking with open arms as it justifies the approach of predicting stock prices using only the shape of a few graphs summarizing an equity in the recent past. Regardless of its controversy it shows the potential for LSTMs in finance.

Next, is the work of Bao et al. which achieves SOTA performance in predicting the next closing price of equities on the S&5P00 using an LSTM [2]. Notably this paper has a hint of macro economics included as its features contain various metrics derived from the current market regime in the form of various indicy values from markets world wide. Notably there have been others to use similar modeling techniques with their own data to again show that LSTMs have remarkable predictive power with regard to movement of markets [3].

One metric which is not often discussed in the news cycle but plays a clear role in price formation is the use of insider trading information *(note: this is data on legal transactions executives execute on their own company's stock. A controversial topic which has implications outside the scope of this paper).* Regardless of opinions there has been research that shows insider trading information is relevant to price formation [4], [5], [6]. Though the effect is recognized to change over time, insider's clear insight justifies its inclusion in the model.

Machine learning methods have been becoming more popular in financial and commodity markets across the globe, however many firms have taken a step back in adoption of ML methods due to strategies not scaling. Further, many say making decisions using complex models where insight into why the predictions they are making tend to work isn't viewed as enough of a benefit to ditch old-style technical analysis which traders feel they understand a causal relationship of price formation [7]. One of the project main goals is to include input from all paradigms of trading to lessen this valid criticism.

3 Dataset and Features

The raw data set consists of 3 main parts. Prices, financial statements, and insider trading history. Price data consists of daily OHLC and Volume data for every stock on the New York Stock Exchange from 1990 to 2018. This was retrieved with the R package *batchgetsymbols*. The quarterly fundamentals data which has 138 features for each company starting at the company's IPO and going until the company is delisted from the exchange or 2019, the end of data collection (which ever is first). This is about 7 million days of prices across all stocks and 350,000 quarters of financial statements. Finally there is the complete insider trading history for each company which consists over 3 million transactions scraped from form 4s registered with the SEC. These contain 18 fields per transaction. The final representation of this data in the input vector q *more on this later* is described in item 7 of the appendix. The rest of the data was scrapped from a Bloomberg terminal using code written by me, it is available in the project repository linked in the appendix.

3.1 Preprocessing

Now this data is turned into features and targets. First the easier formation of the two: the targets are created by taking the log return of the a company's open relative to it's prior close using the formula: $ln(p_t/p_{t-1})$. The log return of the high, low, and close columns are also calculated relative to the open on the same day. Note that the targets for the predictions of day $t = 1$ will be in the features of $t = 2$.

The following is done to create a single time step of the input vector $x_t \in 12$. First, as mentioned in the target description, the first 4 entries will be the actual vales of the targets from the prior day as they are the price changes for the past day at this next time step. Then the log return is calculated using the same method for the daily volume of the corresponding equity. The same is done for the volume weighted average price of oil, gold, silver, Forex pairs: USD/EUR, USD/JPY, USD/CAD, USD/NZD. Finally, note that x_t is input for the LSTM network with daily input.

There is also an input timeseries $q, with q_t \in \mathbb{R}^{54}$. This is for data that describes companies, but is only available on a quarterly basis. Of the 138 quarterly financial statement values scrapped, only fields with less than 5% of their data missing were kept. This led to only 34 of the 138 fields being kept from the three financial statements. Tables showing these 34 fields are included in the appendix as items 3, 4, & 5. Some metrics are kept with the as is value, one being how many days into the quarter the highest price occurs. Other metrics like cash on hand are represented by a log return of that value from prior quarter to that time setup. There was no experimentation with this sort of feature creation, domain knowledge of industry standards was used. The structure of a quarterly timeseries is shown below. Note that *deltas* are variables for which the change is relevant so log returns are computed, while *Values from Quarter* kept as the actual with no temporal comparison preformed.

<<<--- Continued for desired look back period	Values from 2 Quarters Back --> 1 Quarter Back					Values from 1 Quarters Back --> Current Quarter				
	Deltas			Values From Quarter		Deltas			Values From Quarter	
	Acctg Info (34 Cols)	Form 4 Info (7 cols)	Pricing Info (4 cols)	Index Info (4 cols)	Span Info (5 cols)	Acctg Info (34 Cols)	Form 4 Info (7 cols)	Pricing Info (4 cols)	Index Info (4 cols)	Span Info (5 cols)
<<<---	0 : 34	34 : 41	41 : 45	45 : 49	49 : 54	54 : 80	88 : 95	95 : 99	99 : 103	103 : 108

Table 1: Structure of Quarterly Data

4 Methods

Initially the model was not trained with the quarterly information. The model was a network trained on 1/3rd of the daily price and volume data for the stock market from 1980 till 2018. However this often would not converge. Training the same network architecture using data from similar companies based on their Standard Industrial Classifier codes [8] actually allowed for convergence and reasonable prediction in some cases. Finally, training a network on given individual company allowed for good model performance, however over fitting is an extreme concern if fitting for one company as a large enough network can force the training loss down to zero without any regularization.

The fact that the network would atleast converge for companies from the same industry gives confidence that there is some information relevant to price formation in the prior price history. However splitting sectors is a problem as it lowers the size of the dataset. In an attempt to make a more universal model and use all the data collected to train a network the features measured using information at a quarterly interval are trained on a new network using a parallel network structure shown in the figure below.

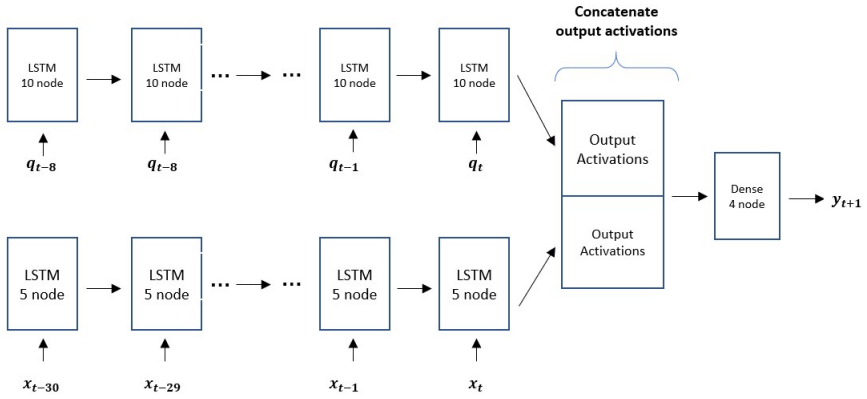

Figure 1: Structure of Final Model with Parallel Networks

Here the bottom portion of the network is what was described above; an LSTM using daily price history. However the output activations are not used directly to predict y_{t+1}. In this parallel network the activations

are output and concatenated with outputs from another LSTM which processes the fundamentals data with 10 nodes per layer. Notably the LSTM for the fundamentals includes 8 quarterly observations instead of the 30 days worth of prices.

5 Results

The network ended up making very good predictions. These are shown in the figure below. Before diving further into the predictions, some notes on model performance and various hyper parameters. The first significant note is that model size did not appear to have much of an effect on performance. That is, unless the number of nodes in the upper or lower LSTM changed by an order of magnitude the performance was very stable. 10 nodes for the quarterly LSTM and 5 for the daily were chosen simply because they are nice numbers. Regarding other hyper parameters such as learning rate, the optimizer, and batch size. None of these really had a significant effect. The main factor on whether the model would converge was the normalization. Initially the normalization was done on a per company basis. Then Keras was set to do the learning without any batch norm. This did not work, however not normalizing the data and then allowing batch norm to do this task allowed for convergence. This leads to the hypothesis that there is a more simple model somewhere between a linear regression and an LSTM that would be able to match the quality of predictions.

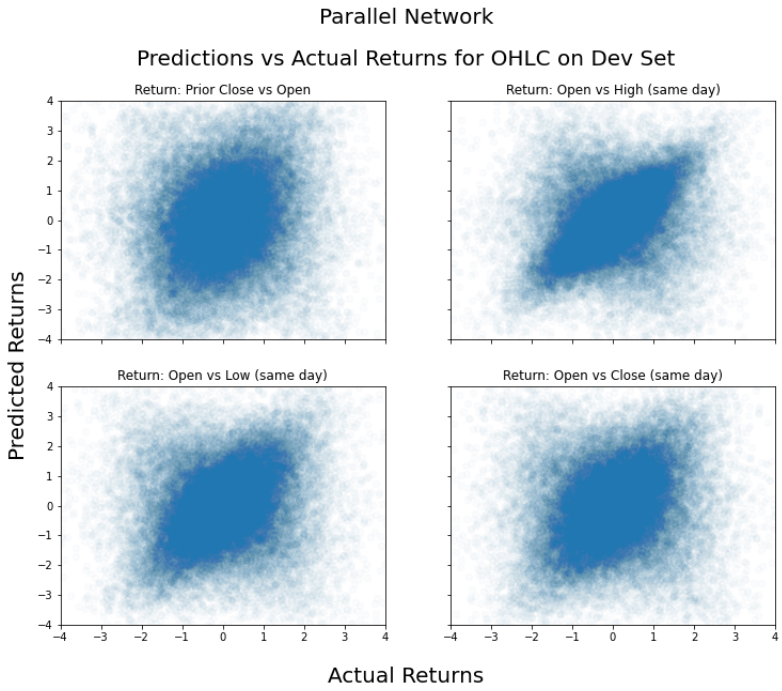

Figure 2: Predictions of Model Targets

While the predictions look quite good their performance is slightly misleading, atleast regarding the predictions on the return of a stock's low vs open for the day, and the same for the high vs open. This is because the lows and the highs tend to be around $\pm 1\%$ of the open. So any model can generally pick up on this trend and it will typically be within the vicinity. This is seen by the fact that the MSE for the low prediction was 0.31, but fitting a time series regression with the last 5 days of price data yielded a MSE of 0.412. While the neural network is a clearly significant increase in performance it is not huge. One feature that is notably different is that this model creates predictions further from 0 compared to a lagged linear regression whose results are much more flat. Note, the predictions of a linear regression model are shown in item 7 of the appendix. This is actually a great feature as if you can select predictions with large magnitude and they are more likely to atleast be in that direction.

6 Conclusion & Future Work

The model makes impressive predictions on price characteristics of equities for the next day. However, brainstorming on how to use this model in real time trading led to one realization that has significant implications; there is no timing prediction for the low or the high. Knowing which will come first is clearly important. This realization made it evidently clear that the model is not equip to make investment decisions on its own. Further, with the current dataset nothing can be done to add this sort of feature. This is because the price data is only Open, High, Low, Close, & Volume. There is nothing about when these features occur.

More work needs to be done on the timing of said predictions. The results achieved thus far imply there is potential to use this model in a decision making process. However in markets timing is key, any investment algorithm based off the current results would have to take a time blind approach which is not good. With a more comprehensive price timeseries datset timing predictions within the day could also be done. Further, others have done work showing that simple, but rigid rules such as trailing stop losses can drastically increase the performance [9]. Next steps would be to build a more comprehensive strategy back tester which can act on intra-day predictions (i.e. trading on an equity more than once in a day).

Overall the fact that the model has a different shape of predictions along with more "spread" in predictions than an lagged regression, one of the industry standards for these sorts of predictions shows that can be further pursued as part of the decision making process in an investment strategy.

7 Contributions

I would like to thank Avoy Datta for his guidance and recommendations on how to approach various aspects of this problem when I ran into issues. His advice on experimenting with attention boosted the of the model.

8 References

References

[1] Rama Cont Justin Sirignano. *Universal features of price formation in financial markets: perspectives from deep learning*, 2019.

[2] Jun Yue Wei Bao. *A deep learning framework for financial time series using stacked autoencoders and long-short term memory*, Jul 14, 2017.

[3] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira. Stock market's price movement prediction with lstm neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1419–1426, 2017.

[4] Guay W. R. Core, J. E. *Stock Market Anomalies: What Can We Learn from Repurchases and insider Trading?*, Apr. 3rd. 2004.

[5] F. Brochet. *Information content of insider trades before and after the Sarbanes Oxley Act*, Jan 17th, 2010.

[6] Huddart J. S. Petroni R. K. Ke, B. *What Insiders Know About Future Earnings and How They Use it: Evidence from Insider Trades*, Mar. 24th, 2002.

[7] Tyler Durden. *The Perils Of Trade With Machine Learning: One Pin Drop Can Make You Lose 20 Years Of Returns*, Oct 14. 2018.

[8] Wikipedia contributors. *Standard Industrial Classification*, Sept 27. 2020.

[9] Yufeng Han, Guofu Zhou, and Yingzi Zhu. *Taming Momentum Crashes: A Simple Stop-Loss Strategy.*

9 appendix

9.1 Appendix item 1: Github Repository

The code for the project can be found at the following link https://github.com/pb360/cs230_project.

9.2 Appendix item 2: Insider Trading Metrics

Name	Description
buy_count	number of purchases during quarter
buy_vol	volume of stock purchased during quarter
buy_dollars	dollars worth of stock purchased during quarter
sell_count	number of sells during quarter
sell_vol	volume of stock sold during quarter
sell_dollars	dollars worth of stock sold during quarter
delta_vol	total change in volume of shares owned
price_open	starting price of the quarter
price_low	lowest price through the whole quarter
price_high	highest price through the whole quarter
price_close	ending price of the quarter.
index_L_of_L	days into quarter the lowest of daily low prices occured
index_H_of_L	days into quarter the highest of daily low prices occured
index_L_of_H	days into quarter the lowest of daily high prices occured
index_H_of_H	days into quarter the highest of daily high prices occured
span_LL_to_HL	days between lowest of lows and highest of lows
span_LL_to_LH	days between lowest of lows and lowest of highs
span_LL_to_HH	days between lowest of lows and highest of highs
span_HL_to_LH	days between highest of lows and lowest of highs
span_HL_to_HH	days between highest of lows and highest of highs

9.3 Appendix item 3: Quarterly Balance Sheet Fields Kept

Balance Sheet		
Category	**Accounting Name**	**Bloomberg ID**
Assets	- Cash, Cash Equivalents, & STI	C&CE_AND_STI_DETAILED
	- Cash & Cash Equivalents	BS_CASH_NEAR_CASH_ITEM
	- ST Investments	BS_MKT_SEC_OTHER_ST_INVEST
	- Property Plant & Equipment Net	BS_NET_FIX_ASSET
	- Other LT Assets	BS_OTHER_ASSETS_DEF_CHRG_OTHER
	TOTAL ASSETS	BS_TOT_ASSET
Liabilities	- ST Debt	BS_ST_BORROW
	- LT Debt	BS_LT_BORROW
	TOTAL LIABILITIES	BS_TOT_LIAB2
Stockholder Equity	- Preferred Equity and Hybrid Capital	BS_PFD_EQTY_&_HYBRID_CPTL
	- Share Capital & APIC	BS_SH_CAP_AND_APIC
	- Minority/Non Controlling Interest	MINORITY_NONCONTROLLING_INTEREST
	TOTAL EQUITY	TOTAL_EQUITY
	- Total Liabilities and Equity	TOT_LIAB_AND_EQY
	- Shares Outstanding	BS_SH_OUT
	- Net Debt	NET_DEBT
	- Net Debt to Equity	NET_DEBT_TO_SHRHLDR_EQTY

9.4 Appendix item 4: Quarterly Cash Flow Statement Fields Kept

Cash Flow Statement		
Category	**Accounting Name**	**Bloomberg ID**
Cash From Operating Activities	- Net Income	CF_NET_INC
	- Depreciation & Amortization	CF_DEPR_AMORT
	- Chg in Non-Cash Work Cap	CF_CHNG_NON_CASH_WORK_CAP
	- Cash From Operating Activities	CF_CASH_FROM_OPER

9.5 Appendix item 5: Quarterly Income Statement Fields Kept

Income Statement		
Category	**Accounting Name**	**Bloomberg ID**
Revenue	- Revenue	SALES_REV_TURN
	- Other Operating Income	IS_OTHER_OPER_INC
	Operating Income (Loss) ~EBITDA~	IS_OPER_INC
	Pretax Income (Loss), Adjusted	PRETAX_INC
	- Income Tax Expense (Benefit)	IS_INC_TAX_EXP
	Net Income, GAAP	NET_INCOME
	- Preferred Dividens	IS_TOT_CASH_PFD_DVD
	Net Income Avail to Common, GAAP	EARN_FOR_COMMON
Per Share	Basic Weighted Avg Shares	IS_AVG_NUM_SH_FOR_EPS
	Basic EPS, GAAP	IS_EPS
Cash	Operating Margin	OPER_MARGIN
	Profit Margin	PROF_MARGIN
	Total Cash Common Dividends	IS_TOT_CASH_COM_DVD

9.6 Appendix item 6: Notes on creation of quarterly input vector

All of the values you see in appendix items 2 through 5 are concatenated together and form one vector q

9.7 Appendix item 7: Performance of Lagged Regression