ALGORITHMIC TRADING USING MACHINE LEARNING TECHNIQUES: FINAL REPORT

Chenxu Shao*, Zheming Zheng†

Department of Management Science and Engineering

December 12, 2013

ABSTRACT

In this report, we present an automatic stock trading process, which relies on a hierarchy of a feature selecting method, multiple machine-learning algorithms as well as an online learning mechanism. Backward search was used in feature selection, while the local linear regression (LLR), $l-1$ regularized v-support vector machine (v-SVM), and the multiple additive regression tree (MART), were chosen as the underlying algorithms. Our trading model is simplified from real life trading. One strength of our approach is that the model, regardless of its many simplified assumptions, is more sophisticated than many of the reported model which uses simple buy-and-hold strategies. In addition, applying the online learning mechanism greatly improves the prediction accuracy. The learning results are impressively robust, rendering our process promising candidates for real life algorithmic trading.

1 Introduction

Nowadays investors and trading firms have been more aware of risks than any time in the past due to the non-stationary and chaotic stock markets under the impact of the 2008 financial crisis. Therefore, many firms now rely heavily on algorithmic trading in the stock markets, especially for high-frequency trading, because of the large amount of information and the importance of instantaneous decisions.

It has always been a challenging task to predict stock prices or even their trends. Our project aims at predicting the short-term pricing trend of selected stocks and simulate the trading results with a simple strategy, to provide a key reference for improving algorithmic trading and better trading strategies.

1.1 Trading model

In order to model the real life trading, we constructed a simplified trading model which is almost identical to the real life trading, with several simplifications: 1. only close price is considered (when selling/buying the stock, one would have to pay the bid/ask price, but they are relatively close to the close price) and trading actions happen each day before closing which is not realistic in real life. 2. The only transaction cost is the 0.3% close transaction fee [1] and income tax is not accounted. Despite these simplifications, the trading model is relatively realistic in that the actual transaction cost is considered compared to the models used in the literature [2;3;4].

1.2 Concepts and terminologies

We assume the trader has two kinds of assets: cash and stock shares. The present value (PV) of the trader's portfolio, is defined as the future amount of money that has been discounted to reflect its current value, as if it existed today. In this specific project, we assume the risk-free interest rate is zero[3], so the present value is calculated as:

$$\text{PV} = \text{Total Cash} + \text{Total Share} \cdot \text{Stock Price} \tag{1}$$

The rate of return (ROR), is defined as the ratio of money gained or lost (whether realized or unrealized) on an investment relative to the amount of money invested. So, the total return of the portfolio (i.e., the growth rate of the money) is calculated as:

$$\text{Total ROR} = \frac{\text{PV}_{\text{last day}} - \text{PV}_{\text{first day}}}{\text{PV}_{\text{first day}}} \tag{2}$$

The daily return is just the daily growth rate of the money, and can be calculated from the following formula:

$$\text{Total ROR} = (1 + \text{Daily ROR})^{\#\text{ of trading days}} - 1 \tag{3}$$

2 Methodology

2.1 Stock selection

The dataset was downloaded from Bloomberg™terminal with more than 20 indicators/features. The stock is almost randomly selected from *S&P 500* index components, but we do include two basic criteria: stock price and volume. Higher stock price indicates that the stock may be a principal component of the index whereas high volume shows that the stock is traded actively.

In this project, we selected and tested on 25 stocks from the *S&P 500* index components, they are listed as follows:

Table 1: List of selected *S&P 500* index components (tickers only)

AAPL	AMZN	AZO	BAC	BLK
C	CMG	CSCO	DAL	EMC
EXC	GE	GOOG	INTC	ISRG
MA	MS	MSFT	NVDA	PCLN
PFE	SCHW	T	WFC	WPO

*Email: chenxu@stanford.edu

†Email: zheming@stanford.edu

[3] Assuming the annual interest rate is 0.2%, then the daily interest rate is roughly 7.86×10^{-6} which is negligible

2.2 Trading strategy

The trading strategy is rather a simple one: knowing or believing that the stock price will grow tomorrow, one would long (buy) exactly one share of the stock, and one would sell all the holding shares and short (sell) an extra share only based on the knowledge or belief that the stock price will decrease tomorrow.

Using this strategy in the simplified model, we managed to achieve a total return of 740.2% (for 731 days) for AAPL (APPLE INC.) corresponding to a daily return of 0.2294%, provided that the future stock prices are known.

The predetermined present value as a function of time, as well as a typical stock price time series, are shown in Fig. 1:

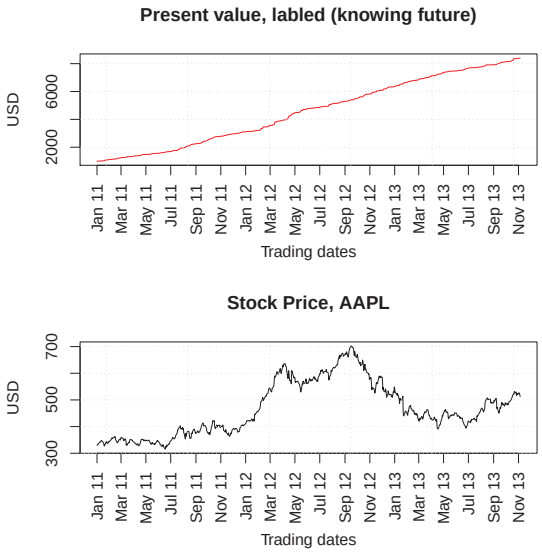

Figure 1: *Top*: PV v.s. time, *Bottom*: Stock price v.s. time

2.3 Feature construction

The features mainly contain indicators acquired from the Bloomberg™ terminal. In addition, there are several indicators constructed with the data, such as Sharpe ratio, Treynor ratio, *etc.*

A complete list of features are listed below:

Table 2: List of features used in the model

10-day Volatility	Quick ratio	P/E ratio	Net change
EBITDA	α	β	Risk premium
Earning/share	α for β	Log ROR	Benchmark ROR
High price ROR	Low Price ROR	Volume ROR	Williams.R%
PVT	Moving average / Price	Sharpe ratio	Treynor ratio

Due to the limitation of space, we will not introduce all the indicators/features, but note the latter eleven features are calculated using the acquired data.

The label is constructed with the closing price of the stock. If the closing price of the next day is greater than that of the present day, the label is set to 1, otherwise it is set to be -1 (for v-SVM) or 0 (for regressions).

2.4 Model selection

In the beginning, we considered four kinds of models: v-support vector machine (v-SVM), locally weighted linear regression (LWLR), logistic regression, and multiple additive regression tree (MART). MART was chosen because it uses data very efficiently and good results can be obtained with relatively small data sets. To determine which model(s) to use and if these models have more bias or variances so that we can tune the model(s), a learning analysis was performed. The learning curves of the four models are illustrated as follows:

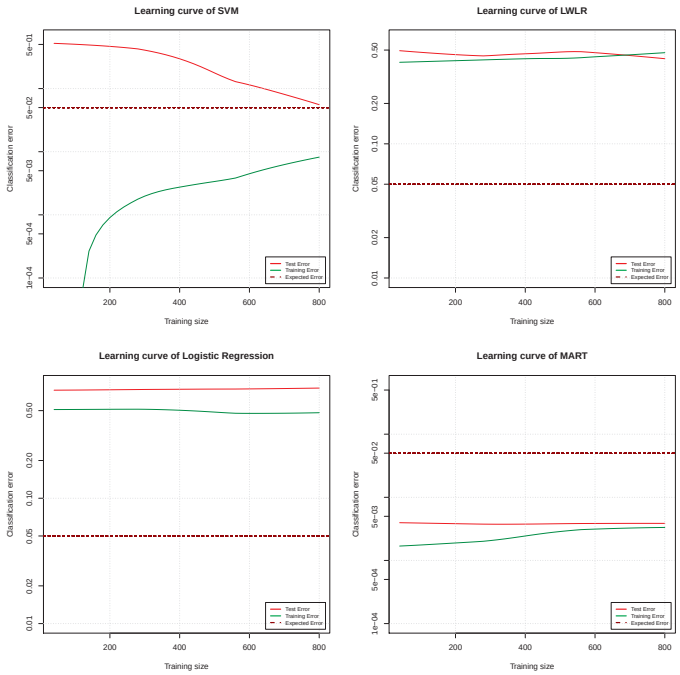

Figure 2: Learning curves of the four models. The underlying stock here is AAPL.

From the learning curves, it can be easily seen that v-SVM is less biased and clearly has a large variance, while LWLR, logistic regression and MART are biased. Since it is hard to construct more features but easy to reduce the size of features and change the size of training data, and considering the overall accuracy reflected from the learning curves, v-SVM and MART were selected as the underlying models. In addition, inspired by the idea of LLWR, a local linear regression (LLR) was also implemented, based on the belief that locally the stock price grows linear with time.

2.5 Feature selection

The learning analysis rendered v-SVM having a relatively high variance. To tackle this issue, the feature selection was performed. As the amount of features (20) is considerably small, instead of using heavy machinery such as mutual information, a simple backward search was performed.

For each of the stock, we leave one of the 20 features out and calculate the classification error. Then we have 20 classification errors corresponding to the absence of each of the 20 features. Next, We determine the 25% quantile of these errors and remove features without which the errors drop significantly, within the top 25%. Usually, 3 - 4 features are removed using this method.

2.6 Online learning vs offline learning

To implement the learning algorithms, we first compared the online learning mechanism versus the offline learning mechanism. In practice, the data of the stock from April 1, 2010 to December 31, 2010 were used as training data for offline learning and for the first prediction of online learning. Their accuracies and recalls as functions of initial training sizes are plotted as follows:

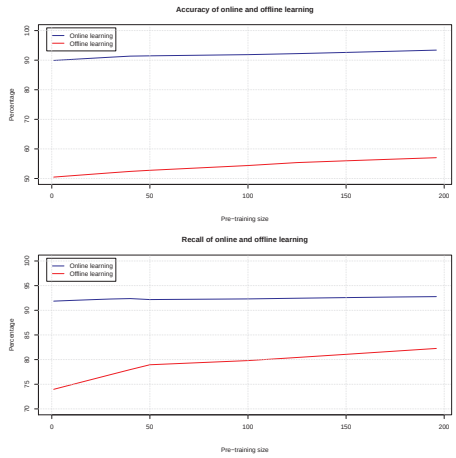

Figure 3: Accuracy (*Top*) and Recall (*Bottom*) curves of the two learning mechanisms

It can be seen from the above figure that the online learning mechanism has both higher accuracy and recall than the offline learning, which is reasonable: As the stock price is stochastic almost surely, the models need to be readjusted actively so that it catches the new trend/feature.

It is also worth noting that for the online learning, when doing the prediction, we use the stock data at the opening time which is the beginning of a day and the prediction was performed for the closing time of a day. At many occasions, there may be correlations between the opening price and the closing price (which can be one of the reasons why online learning is much more accurate than offline learning), but generally this strategy is still of practical use as one can always trade between the opening time and closing time.

Based on this result, online learning was then performed with the subsequent data from January 1, 2011 to November 9, 2013. On each step, the prediction was made for the current trading date and the trading actions were performed under the prediction, then the models were adjusted and retrained for the prediction for the next trading day.

3 Results and discussion

3.1 Local linear regression

The local linear regression was trained with a subset of features with *PVT* and *EBITDA* removed due to the large absolute values, which may cause divergence. For each iteration, the model was trained with the data from the past few days (three in this case), and then the prediction is performed (referred as "local learning" in the later text).

The precision, recall and the accuracy for the LLR model are listed in Table.3. It can be seen that although the accuracy of LLR is not high, which ranges from $63\% - 70\%$, but it is quite robust, with a variance of 2.774×10^{-4}. In addition, the precision and recall remain the same level as the accuracy. Moreover, although its accuracy is not impressively high, the model maintains positive total returns. All these features render this model very stable and practical.

Table 3: Learning results of LLR model on 25 stocks

Ticker	Accuracy	Precision	Recall	Total Return	Total Return (label)
AAPL	63.47%	64.75%	63.20%	176.25%	740.24%
AMZN	68.95%	70.03%	67.57%	127.08%	375.11%
AZO	67.44%	69.53%	68.81%	42.85%	265.74%
BAC	67.85%	67.70%	66.76%	6.30%	19.94%
BLK	67.17%	66.92%	70.51%	81.98%	286.16%
C	65.25%	65.17%	64.09%	27.39%	76.95%
CMG	68.67%	71.32%	70.05%	157.08%	472.86%
CSCO	65.25%	64.54%	64.90%	5.27%	21.73%
DAL	67.03%	68.12%	66.84%	8.72%	27.46%
EMC	67.99%	67.14%	66.76%	12.46%	34.20%
EXC	67.99%	66.30%	68.18%	11.40%	30.35%
GE	68.54%	69.23%	69.60%	8.77%	20.25%
GOOG	65.53%	67.40%	64.91%	208.45%	764.40%
INTC	67.31%	68.22%	66.94%	5.33%	27.85%
ISRG	67.03%	67.80%	65.40%	226.36%	657.79%
MA	69.63%	72.45%	71.36%	207.54%	477.73%
MS	65.80%	65.29%	65.65%	18.01%	40.69%
MSFT	68.67%	68.84%	67.13%	4.56%	31.79%
NVDA	65.39%	62.43%	65.89%	8.41%	32.33%
PCLN	67.44%	69.25%	69.25%	353.07%	1007.30%
PFE	67.31%	68.29%	65.12%	1.52%	17.78%
SCHW	71.00%	70.73%	71.51%	8.44%	25.57%
T	69.77%	70.57%	73.32%	1.57%	22.89%
WFC	67.44%	66.49%	69.06%	10.93%	35.34%
WPO	67.03%	68.59%	70.18%	175.56%	512.04%

The present value curses of AAPL using this model and predetermined label are plotted in Fig. 4. The present value grows with time, but is quite slow compared to the predetermined present value. In addition, it has many kinks, due to the many failures of predicting the trend of the future price. The total return of the trading strategy with this model is 176.2% for AAPL, corresponding to a daily return of 0.1391%. which is much lower than the predetermined returns. In general, the total return with LLR model is always much lower than the predetermined returns as listed in Table 3.

Despite the many issues, the LLR model managed to reach a positive returns for all 25 stocks for 731 days. And due to its simplicity (easy to implement) and speed (fast to train and predict), it can be a very powerful tool in high frequency trading.

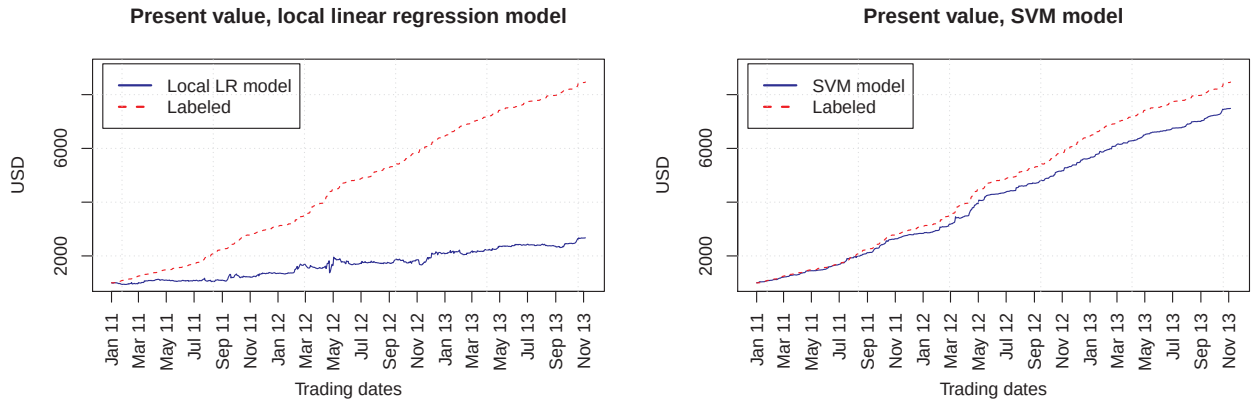

Figure 4: *Left*: PV of LLR, compared to the labelled PV, *Right*:PV of SVM, compared to the labelled PV. The underlying stock here is AAPL.

3.2 ν-support vector machine

The $l-1$ regulated ν-SVM ($C=1, \nu \approx 0.2$, with Laplace Radial Basis Function (RBF) kernel) was trained with a subset of features using feature selection (e.g., in the case of AAPL, the four features: *benchmark ROR, volatility, EBITDA* and α *for* β were removed from the feature set). The precision, recall and the accuracy for the ν-SVM model are listed as follows:

Table 4: Learning results of SVM model on 25 stocks

Ticker	Accuracy	Precision	Recall	Total Return	Tot. Ret. (label)
AAPL	91.66%	92.66%	90.93%	623.86%	740.24%
AMZN	90.56%	90.13%	91.35%	367.42%	375.11%
AZO	92.34%	91.92%	93.81%	216.68%	265.74%
BAC	95.49%	94.57%	96.40%	17.04%	19.94%
BLK	92.75%	93.96%	91.69%	243.68%	286.16%
C	90.83%	90.41%	91.16%	67.60%	76.95%
CMG	89.88%	90.61%	90.61%	346.93%	472.86%
CSCO	91.52%	92.07%	90.53%	20.17%	21.73%
DAL	94.66%	94.67%	94.92%	26.30%	27.46%
EMC	92.61%	92.63%	92.11%	33.15%	34.20%
EXC	91.93%	91.98%	91.19%	29.49%	30.35%
GE	94.80%	94.69%	95.20%	18.37%	20.25%
GOOG	92.20%	92.15%	92.88%	698.23%	764.40%
INTC	93.16%	93.05%	93.55%	26.89%	27.85%
ISRG	93.30%	93.44%	93.19%	606.14%	657.79%
MA	91.93%	92.06%	93.22%	452.84%	477.73%
MS	93.30%	94.07%	92.24%	37.62%	40.69%
MSFT	93.16%	92.62%	93.65%	29.77%	31.79%
NVDA	92.61%	91.64%	92.71%	30.92%	32.33%
PCLN	91.11%	91.93%	91.21%	965.17%	1007.30%
PFE	88.78%	87.80%	90.19%	14.72%	17.78%
SCHW	93.71%	94.43%	92.88%	24.87%	25.57%
T	92.34%	92.31%	93.26%	20.01%	22.89%
WFC	92.75%	94.27%	90.88%	32.42%	35.34%
WPO	91.66%	92.71%	91.52%	489.62%	512.04%

Clearly, the ν-SVM model has a very high accuracy, and good precision and recall as well. Compared to LLR model, the accuracy is significantly higher. As can be observed from the present value curve in Fig.4, the present value grows with time in a linear fashion, which is quite reasonable assuming the price change of the stock is nearly constant, thus with our trading strategy the amount one earns everyday is also nearly constant provided the prediction is correct. Although not obvious, it can be seen that the performance of the model is not very good near a sudden change of price trend (known as "change point", which is either a local optimum, or jump).

The ν-SVM model proved to be a very stable and powerful tool to predict stock price trend and together with the trading strategy, it managed to reach relatively high returns. It is worth noting that for different stocks the parameter ν and selected features are slightly different, and tweaking the parameters and performing ν-SVM algorithm takes longer time than simply doing the linear regression. Hence, although ν-SVM is more accurate, it is not as efficient as LLR.

3.3 Multiple additive regression tree

The precision, recall and the accuracy for the MART model are listed in Table.5. It is observed that the accuracies and recalls fluctuate a lot, whereas the precisions remain a very high level. Although the averaged accuracy is $\sim 81\%$, its variance is 0.02122, which is almost 10 times larger than that of LLR and ν-SVM, indicating that the performance of this model varies greatly.

Moreover, the total return of this model can even be significantly negative when the accuracy is actually high ($> 65\%$, see INTC and ISRG). This is because MART has a very poor recall, so that most of the time the trader is losing money by short selling the stock shares, which can cause significant loss when the trader is holding many shares in hand.

The unstable performance and the negative returns of MART reveal its disadvantages: first, its outputs of regression can lie outside of the range $[0,1]$ which would be classified as incorrect prediction; second, its extrapolation properties tend to be poor, which may cause the performance to be unstable; last, it is very sensitive to outliers, which further brings down the prediction accuracy.

In addition to this, it takes many iterations for MART to converge, the total number of iterations required for convergence varies from 200 - 1200. Performing hundreds of iterations costs a relatively large amount of time which is another disadvantage of this method. Therefore, it is not only less stable compared to ν-SVM and LLR, but less efficient than the two models.

Table 5: Learning results of MART model on 25 stocks

Ticker	Accuracy	Precision	Recall	Total Return	Tot. Ret. (label)
AAPL	98.50%	99.73%	97.33%	676.08%	740.24%
AMZN	92.34%	99.68%	85.14%	299.86%	375.11%
AZO	98.63%	99.22%	98.20%	246.61%	265.74%
BAC	53.08%	100.00%	4.99%	−42.55%	19.94%
BLK	69.90%	99.35%	41.29%	147.81%	286.16%
C	72.50%	100.00%	44.48%	38.02%	76.95%
CMG	99.73%	99.75%	99.75%	465.11%	472.86%
CSCO	83.31%	100.00%	66.02%	8.54%	21.73%
DAL	64.02%	100.00%	29.68%	−28.61%	27.46%
EMC	94.80%	100.00%	89.30%	30.05%	34.20%
EXC	70.45%	100.00%	38.64%	22.34%	30.35%
GE	85.23%	100.00%	71.20%	10.88%	20.25%
GOOG	97.54%	100.00%	95.25%	676.08%	764.40%
INTC	74.97%	100.00%	50.81%	−2.62%	27.85%
ISRG	65.39%	100.00%	31.06%	−534.42%	657.79%
MA	96.31%	99.73%	93.47%	407.28%	477.73%
MS	60.19%	100.00%	19.39%	17.16%	40.69%
MSFT	62.52%	100.00%	24.31%	−21.99%	31.79%
NVDA	81.26%	100.00%	60.06%	22.24%	32.33%
PCLN	90.56%	100.00%	82.17%	664.64%	1007.30%
PFE	84.95%	100.00%	70.03%	7.82%	17.78%
SCHW	68.95%	100.00%	37.81%	20.87%	25.57%
T	99.32%	100.00%	98.70%	20.18%	22.89%
WFC	60.47%	100.00%	20.17%	−27.29%	35.34%
WPO	98.77%	100.00%	97.69%	465.47%	512.04%

The performances of the three methods are summarized as follows:

Table 6: Performance of different models

Model	Avg. accuracy	Avg. learning time†	Have negative returns
LLR	67.40%	5-10s	No
v-SVM	92.36%	20-50s	No
MART	80.95%	1-5 min	Yes

† Note this is only a rough estimation based on observations.

As a result, v-SVM has the best overall performance. The time it used to learn and predict makes it best for daily trading. LLR turns out to be very stable and efficient, which can be used to assist high frequency trading. MART, which has a good average performance, is highly unstable and relatively slow in learning and predicting, and thus not suitable for algorithmic trading.

3.4 Further analysis on online learning

Notice that from the learning curve we can see that the accuracy (= 1 − test error) of linear regression ranges from 40% to 50% , whereas the actual averaged accuracy of LLR is 67.40%, which is higher. This is due to the advantage of online learning and local learning. Since the belief that the stock price is locally linear (but not globally), local learning should improve the quality of the prediction. Moreover, the online learning technique enables the model to evolve with time, so that it was adjusted and better for the next prediction.

In addition, for v-SVM, its accuracy varies from 50% - 95% by the learning curve, whereas the actually averaged accuracy of online learning turned out to be 92.36% , which is a number in between. This is because although the online learning technique enables the model to adjust itself with time, its variance is certainly larger than the offline learning as it always have no less training data than the offline learning, which may reduce the accuracy.

4 Conclusion and future work

In conclusion, we have developed a process flow of automatic trading using machine learning techniques. Feature construction, feature selection, and model selection were studied in detail. The online learning and offline learning mechanisms were also compared and discussed. The averaged prediction accuracy for the local linear regression model turned out to be 67.40% whereas that of the v-support vector machine turned out to be 92.36%, thus showing our process a good candidate for algorithmic trading.

Noting that v-support vector machine usually fails to predict correctly when the data point is close to a sudden change which is almost random seen from the data itself, and that the information of a sudden change in stock price trend may be included in the text information such as daily news. It is believed that a further step would be to investigate the usage of text-mining techniques to assist the machine learning process proposed in this project. Furthermore, it is noted that improving the learning speed of v-support vector machine as well as the efficiency of the code overall is needed. Finally, the real-time tests of the system as well as bringing more complexities (such as considering more realistic transaction costs) is necessary.

5 Acknowledgement

We would like to thank Professor Andrew Ng for teaching this great course of machine learning. Indeed, we enjoyed the class very much. The materials and knowledge we learned from this class is very practical and useful, and we have already put what we learned into practice. We also thank Alex Yuqing Dai, for assisting us in downloading the stock data from the Bloomberg™ terminal and for the many discussions he had with us.

References

[1] *https://usequities.nyx.com/markets/nyse-equities/trading-fees*

[2] Dempster, M.A.H.; Leemans, V. *Expert Syst. Appl.*, **2005**, 30, 543-552

[3] Fung, G.; Yu, J.; Lam, W. *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 481-493

[4] Debbini, D.; Estin, P.; Goutagny, M. *Modeling the Stock Market Using Twitter Sentiment Analysis*