Portfolio Recommendation System

Stanford University

CS 229 Project Report 2015

Berk Eserol

Introduction

Machine learning is one of the most important bricks that converges machine to human and beyond. Considering the last twenty years improvement in the area, it is easy to foresee that machine learning will continue being valuable for daily life. The improvement of machine learning can be attributed into four main reasons. Technical achievements in the area (1), growth of machine capacities and capabilities (2), advance connectivity and spreads of service technologies (3) and massive increase of the data amount (4) (Horvitz 2006).

Using the power of machine learning to analyze the historical data, future predictions and projections can be performed on many different subjects. Even though, it is affected by various different external events, in this application project, stock market prices are tried to be predicted using only their historical data and a portfolio recommendation result is generated via the output of the regression and scoring. The aim is to recommend a portfolio with high accuracy profit return. The system is designed to produce result on any market structure that can be represented with similar data. For the project, historical NASDAQ stock prices are used.

Related Work

Stock market can be associated with various different events and data. Earning announcements are one of the key events that affects the stock prices. Using company earnings data with Gaussian kernelled SVM can lead predictions with 64% accuracy (Pouransari H., Chalabi H. 2014). Other financial assets such as currency and underground resources can be another source of information and increase predictions up to 77.6% for some markets (Shen S., Jiang H., Zhang T. 2012). Additional features can enhance the prediction for 3, 5 or 10 days results up to 70% accuracy (Di X. 2014). Algorithm selection is another factor on the accuracy. 50% accuracy is achievable with neural networks (Lin H. 2013). Increasing the prediction range, for example 44 days with SVM can produce more accurate results up to 79% (Dai Y., Zhang Y. 2013). Other than regression approaches, without predicting the price but classifying the stock as positive/negative with SVM and creating an equally distributed portfolio can return nearly 3% higher than the market average (Arık S., Eryılmaz B. and Goldberg A. 2013)

Dataset and Features

All available historical price data for most of the companies that have stocks tradable through NASDAQ is collected[1]. The data contains daily values from the beginning of its trade date to a recent date (ideally one day before the calculation). Each day data is in the following format:

MSFT	Year	Month	Day	Open	High	Low	Close	Volume
	2015	11	12	53.4800	53.9800	53.1900	53.3200	34485500

[1] Yahoo Finance service and "herval/yahoo-finance" open source project (https://github.com/herval/yahoo-finance) is used for data retrieval.

Currently the system has 3010 company histories from their first day on NASDAQ to 11/12/2015. The data can be updated with most recent values to increase accuracy before the calculation.

The systems expects a set of inputs and use them to fill the feature vectors or as parameters. The expected inputs are the following

- **Set of NASDAQ stock names:**
  Instead of running on all 3010 companies, the system considers only the given set of stocks and only uses them in the recommended portfolio.
- **Budget:**
  Budget is the maximum amount of total stock price in the recommendation. The sum of the recommended stock price does not exceed the given budget amount.
- **Keep Time Interval (KTI):**
  The maximum intended time to keep the purchased stock. The keep time interval affects the features as a parameter. It can be minimum one day and maximum one year in the type of day count.
- **History Interval (HI):**
  History interval is used to trim the historical data to be considered. The data with the date outside of this interval is not considered.

Using the data bordered with inputs and using the parameters, the feature vector is filled such that

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
1	Interval Start Price	Interval Maximum Price	Interval Minimum Price	Interval Volume Average	Interval Start Day	Interval Start Month	Interval Start Year	Interval Close Price

The interval is referring to the two times of the keep time interval. The data set is created by an interval length sliding window for every data day.

Methods

As a preprocessing according to the given input parameters, for every historical day of the given companies, a new data point is created in the form of the feature vector. The interval is considered as two times of keep time interval in order to make a better prediction. As an example, if the given interval is five days, then the data points are generated according to ten days intervals. The last day is considered as a new point (half of the interval) and the gain is calculated accordingly.

As a learning method, locally weighted linear regression (LWR) is used minimizing

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} w^{(i)}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

Where the weights are calculated as $w^{(i)} = \exp(-\frac{|x - x^{(i)}|}{2\tau^2})$

τ is the bandwidth parameter.

The bandwidth parameter is used to prevent overfitting and underfitting. The θ is calculated using the normal equations

$$\theta = (X^T W X)^{-1} X^T W \vec{y}$$

Train is performed during the query time for the new point and predicted interval close price is determined. The difference between last close price and predicted interval close price is the initial score of the stock. If the initial score of a stock is negative then it is removed from the recommended bundle (RB). The positive initial score stocks are rescored between [0, 1] based on their initial scores such that total second scores of the stocks in the recommended bundle is 1. The budged is distributed proportional to the second scores of the recommended bundle stocks.

Results

The calculation of the training error is performed on the same data changing the latest date of the system into an earlier date and comparing the result with actual results. Using the "leave one out cross validation", error values are calculated for meaningful subsets of the feature set and best result is achieved with using the all predefined features. The bandwidth parameter is used as another variable for minimizing the training error (best at 0.8).

The titles of the graphs represent the given input to the system in the orders of **KTI/Budget/HI.** Each graph has four dates (execution of the system) of recommendation results indicating the day of the market buy order requested. The stocks are assumed sold KTI days later and the charts show the virtual profit in terms of dollars after KTI days of the given date. The following system training errors (profit errors) are calculated according to the virtual profit of each stock that appears in the recommended bundle (RB). The result considered as error if it appears in the recommended bundle and the virtual profit is not positive after KTI days such that

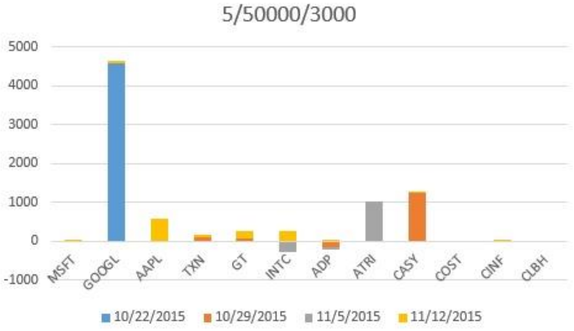
$$\hat{\varepsilon}(p) = \frac{1}{m}\sum_{i=1}^{m} 1\{p\left(s^{(i)}\right) < 0\} \qquad s \in RB$$

$\hat{\varepsilon}(p)$	10/22/2015	10/29/2015	11/05/2015	11/12/2015
5/50000/3000	0.00	0.40	0.80	0.00
8/50000/4800	0.00	0.00	0.60	0.00
10/50000/6000	0.00	0.00	0.33	0.00
13/50000/7800	0.00	0.00	0.00	0.00

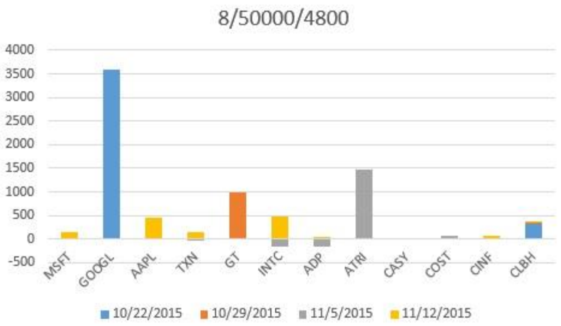
Using these 16 executions of the system with the predefined trending set of stocks, the average profit error is 0.1331 (profit accuracy is 86.69%). The total virtual profit combining the result of four execution of the system per given set of inputs are the following.

Total Virtual Profit	5/50000/3000	8/50000/4800	10/50000/6000	13/50000/7800
	7639.448	7441.044	3654.637	6835.232

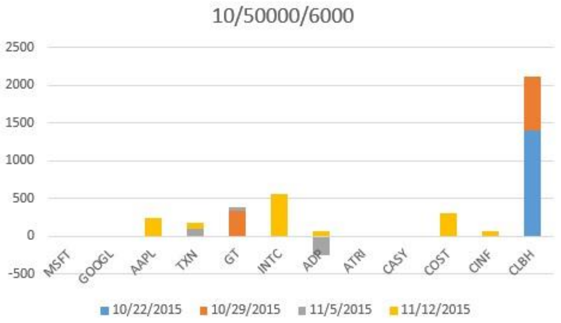
A set of 12 trending stocks {MSFT, GOOGL, AAPL, TXN, GT, INTC, ADP, ATRI, CASY, COST, CINF, CLBH} is selected as an example user input. Four executions of the system with different buy market order date results are represented in each charts. Positive results are considered as correct and negative results are considered as errors. Detailed virtual profit results after KTI days later are shown separately.



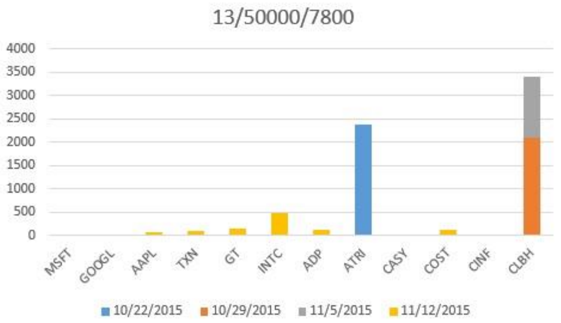
5/50000/3000	10/22/2015	10/29/2015	11/05/2015	11/12/2015
MSFT	0	0	0	10.53998
GOOGL	4587.117	0	0	54.55957
AAPL	0	0	0	575.2796
TXN	0	102.9202	0	55.04013
GT	0	68.99985	0	184.0801
INTC	0	0	-289.6	260.2996
ADP	0	-140.76	-89.6802	2.209974
ATRI	1.390014	-16.0803	1026.02	0
CASY	0	1234.801	0	12.85
COST	0	0	-15.1202	0
CINF	0	0	-18.6001	33.18004
CLBH	0	0	0	0



8/50000/4800	10/22/2015	10/29/2015	11/05/2015	11/12/2015
MSFT	0	0	0	149.73
GOOGL	3587.04	0	0	0
AAPL	0	0	0	445.5594
TXN	0	0	-20.4601	141.6
GT	0	994.4969	0	0
INTC	0	0	-164.56	468.4398
ADP	0	0	-167.861	36.25985
ATRI	0	16.22	1460.599	0
CASY	0	0	0	0
COST	0	0	60.25988	0
CINF	0	0	0	80.52
CLBH	321.3	25.6	0	6.3



10/50000/6000	10/22/2015	10/29/2015	11/05/2015	11/12/2015
MSFT	0	0	0	0
GOOGL	0	0	0	0
AAPL	0	0	0	242.439652
TXN	0	0	100.649817	75.680086
GT	0	341.779257	39.600072	0
INTC	0	0	0	556.19919
ADP	0	0	-259.01175	71.440152
ATRI	0	0	0	0
CASY	0	0	0	0
COST	0	0	0	310.780082
CINF	0	0	0	59.360159
CLBH	1398.92	716.8	0	0



13/50000/7800	10/22/2015	10/29/2015	11/05/2015	11/12/2015
MSFT	0	0	0	0
GOOGL	0	0	0	0
AAPL	0	0	0	76.71973
TXN	0	0	0	91.5201
GT	0	0	0	148.1999
INTC	0	0	0	478.71
ADP	0	0	0	129.7206
ATRI	2394.962	0	0	0
CASY	0	0	0	0
COST	0	0	0	111.9799
CINF	0	0	0	0
CLBH	0	2108.62	1294.8	0

Conclusion and Future Work

The Portfolio Recommendation System shows that adding a portfolio layer on top of the stock regression results is increasing the success rate (profit accuracy) up to 86.69%, when success is calculated by the profitability of the recommendations. Moreover, it helps to reduce the risk by distributing the budget over a set of stocks and tries to minimize the reflection of the regression errors to the profit.

The project can be enriched with additional features and alternative algorithms. Another suitable algorithm for the problem would be SVR instead of LWR. Some other hybrid solutions can also be applied such as determining the positive stocks with SVM and rate them with a regression algorithms.

References

Horvitz, E. 2006. Machine learning, reasoning, and intelligence in daily life: Directions and challenges. Technical Report TR-2006-185, Microsoft Research.

Pouransari H., Chalabi H. 2014. Event-based stock market prediction, Stanford University

Di X. 2014. Stock Trend Prediction with Technical Indicators using SVM, Stanford University

Lin H. 2013. Feature Investigation for Stock market Prediction, Department of Aeronautics and Astronautics, Stanford University

Dai Y., Zhang Y. 2013. Machine Learning in Stock Price Trend Forecasting, Stanford University

Arık S., Eryılmaz B., Goldberg A. 2013. Supervised classication-based stock prediction and portfolio optimization

Shen S., Jiang H., Zhang T. 2012. Stock Market Forecasting Using Machine Learning Algorithms