High Frequency Trading by Transient Profit Peak Classification using Machine Learning on Real-Time Data Streams

Vishnu Srivastava V2SRIVAS@UWATERLOO.CA
University of Waterloo, ON, Canada N2L 3G1

Abstract

Predicting the movement of the stock prices has been a major area of implementation of Machine Learning algorithms but due to large data processing and limited computing capability, most of this has been limited to long term predictions. Caching feature provides an alternative but it's not sufficient enough to account for large volatility in the market.This paper presents the transient peak classification algorithm combined with feature caching to take advantage of the volatility in market by classifying a data point as a peak or non-peak point based on the future profit estimation of a transaction.

1. Description of the Scientific Problem

A good portion of High Frequency Trading system uses a Financial Model driven approach i.e. the decision that the system makes whether to buy a stock or sell it in order to generate profit is decided by the rules defined by the model designer(Henrikson, 2011). These may execute a trade in order of microseconds thus their dependency on the reliability of rules and the latency between these systems and the Exchange server is high.

On the other hand the system which uses a Data driven approach are usually slow because they have to deal with the processing of large datasets(Pedro Domingos, 2003) and are not capable enough to withstand large market volatility. Thus these models were used for long term market prediction(Shen et al., 2012).

Project Proposal for CS886: Applied Machine Learning. University of Waterloo, Fall 2014.

2. Description of the Available Data

The data is available for 136 stocks of NSE India in two forms which can be classified as historic data and live streaming data.Both of these data have been utilized in formation of different features.

2.1. Historic Data

This data comprises of the change in price of a stock in an entire day . Each row of this data represents the statistics of a stock i.e. opening price, closing price, total volume of stock traded, highest value, lowest value and previous closing price. This data have been obtained from 3rd January, 2011 to present date from the National Stock Exchange of India's website. The following method is used to obtain this data: `www.nseindia.com/content/historical/EQUITIES/[year]/[month]/cm[day][month][year]bhav.csv.zip`

The variable within "[ ]" can be replaced with appropriate value to get the data. For a particular day this data isn't available until after the market closes.

2.2. Live Streaming Data

More than 1.6 million records per day are being obtained through live data streams. These are obtained from the stock broker's website. Athough no API have been provided by the broker, auto-login can be achieved by using CURL library for C++ or by HTMLunit library for JAVA. **This data consists of per second records of a particular stock**.The website URL: `https://newtrade.sharekhan.com/rmmweb/`

3. Preprocessing Environment

To deal with millions of records and hundreds of transactions the entire database(MySql) is shifted to RAM using DataRam's RAMDISK software. Special Exception handling routines have been implemented to clean the data before storing and rejecting corrupted data.

4. Plan for Analysis

A total of 16 features have been chosen for the peak classification. While most of the features generation involves arithematic operation on raw data, some of the features have been obtained by using Machine Learning algorithms. For this dataset, the **Random Forest** classification algorithm for the final classification has been implemented

4.1. Polynomial Regression on Historic Data

The historic data was used to predict the closing price of a stock. K-Fold Cross Validation was used to reduce overfitting.

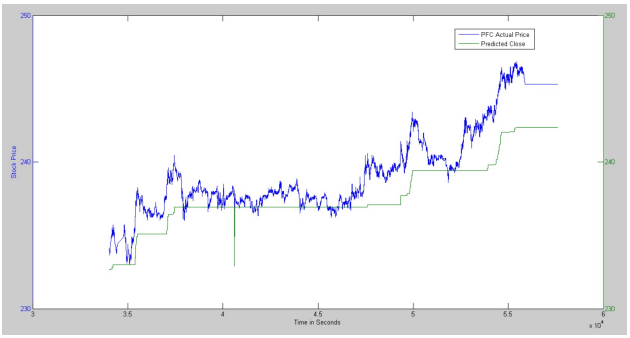

Figure 1. Polynomial regression to predict closing price of the PFC stock on 25 thSeptember, 2014

4.2. Multinomial Logistic Regression for Buy/Sell Prediction

$\ln(\frac{\pi_{buy}}{\pi_{sell}})$ will be calculated using features such as Sell quantity, Buy quantity, high price, low price and time. The result obtained from this calculation is used as a feature for transient peak detection.

4.3. Transient Profit Peak Detection Algorithm

A profit peak is defined as the point at which the price movement changes(buy to sell and vice versa) such that the difference between the current peak and the next peak is equal to or greater than minimum profit for a particular quantity. The peaks may not be identifiable on the first iteration thus certain filters need to be applied for appropriate peak selction. At first, peak prominance and peak width values are used to classify the peak points. Then the data is converted into frequency time graph, followed by the use of **low pass filter** to eliminate high frequency fluctuations. The output is again analysed for peak point classification by peak prominance and peak width measurements. This process will be repeated until no further peak

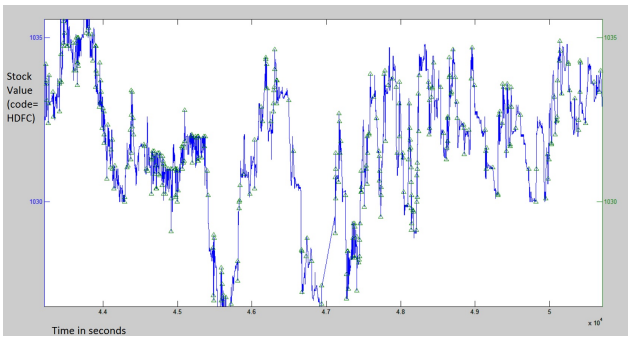

Figure 2. Detecting Peak points of a stock named HDFC, to be used as an output vector on 25thSeptember, 2014

points can be found. The result will be stored in an an output vector and will be used for classification.

4.4. Random Forrest Classification

The Random forest has the following advantages(Hastie et al., 2009):
i)Its accuracy is as good as Adaboost and sometimes better
ii) It's relatively robust to outliers and noise
iii) It's faster than bagging or boosting
iv) It gives useful internal estimates of error, strength, correlation and variable importance
v) It's simple and easily parallelized
vi)It works well on large datasets such as stock markets(Lauretto et al., 2013)
The Random forest algorithm is run in parallel using MATLAB's built-in function. The optimal number of features per decision trees and the number of decision tree to be used is decided by the continous error minimization by experiment using simulation.

4.5. Feature Caching

During the Live Data Stream the features vector will be stored in the memory. After obtaining more feature data, a new classifier will be generated at appropriate interval and a new code will be compiled.

4.6. Simulation

Simulations of Real-Time data streaming will be used by seperating training sample and test samples for the evaluation of the performance and the speed of response of the system. Caching intervals are set based on the trade-off between speed of computation and the reliability of the model.

References

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

Fredrik Henrikson. Characteristics of high-frequency trading. *Royal Institute of Technology, Sweden, Working Thesis*, 2011.

Marcelo S Lauretto, Barbara BC Silva, and Pablo M Andrade. Evaluation of a supervised learning approach for stock market operations. *arXiv preprint arXiv:1301.4944*, 2013.

Geoff Hulten Pedro Domingos. A general framework for mining massive data streams. In *Journal of Computational and Graphical Statistics, 12 (2003)*, 2003.

Vatsal H Shah. Machine learning techniques for stock prediction. *Foundations of Machine Learning—Spring*, 2007.

Shunrong Shen, Haomiao Jiang, and Tongda Zhang. Stock market forecasting using machine learning algorithms, 2012.