Information Retrieval and Data Mining - Project 6

Shaun Dowling, Alessandro Ialongo, Andrey Levushkin, Matthieu Louis
University College London

Introduction

Trying to predict the direction of stock markets with a certain degree of confidence has always been the dream of many people. With the massive increase of available data - financial stocks tick data or social media data (facebook, twitter, etc.) for example - there has been a strong interest in developing mathematical models that leverage those in order to predict and profit from price movements.

A recent and popular model involves estimating the sentiment (i.e. the mood) of a relevant live stream of tweets as a predictor of market changes. One assumption the model makes is that stock prices are led by the behaviour of a large group of people and that this behaviour is correlated to the overall mood of that group of person.

Pipeline

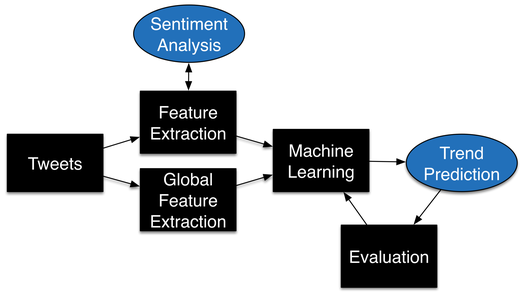

Figure 1: A diagram of our pipeline.

Feature Extraction

The key objective of this project is to use information contained within Twitter posts to predict the mood of the markets towards certain stocks. Tweets contain a great deal of information, links between users, links to entities (either explicitly view a tag or plain text) and network effect via re-tweets.

In their raw form, Tweets are not convenient for inference. As such we will focus a great deal of our time distilling the information contained within the Tweets while ensuring all important aspects are preserved. This approach gives a nice layer of abstraction between the Information Retrieval and the Machine Learning steps in the pipeline. We will be gathering these features using two independent MapReduce jobs.

Global Statistics

Firstly, we will run a global job to gather statistics that depend on the entire dataset (for example tf/idf). The statistics available and the manner in which they are gathered will be a significant area of experimentation.Some statistics that we plan to start with are:

- user average sentiment - to help us reduce bias from general opinion. This could be global or regarding a specific company.
- inverse document frequency - to be used with term frequency within tweets to potentially highlight particularly significant phrases
- cliques of users (strongly connected groups) - experiment with the effect of an entire connected group having a particular sentiment or sudden change of sentiment.

Extractor

Having gathered global statistics, we can split with another map the Tweets by time segment such that the reducer can then create a full feature vector for each time slice as we see fit. Architecturally, there will be an abstract Extractor component within the reducer that is given the full list of tweets to process. This component will be the central component of change in order to explore different options in feature extraction. The Extractor will also have access to whatever global statistics produced previously in order to compute more complex features

Key local features that initially be calculated will be things like

- entity mentions - a base feature which will be joined with a number of other to attribute features to the company whose stock we are interested in
- sentiment - tied to a specific entity or connected group of users
- word frequencies - again tied to a entity
- user ID - which users are mentioning a company
- number of re-tweets - potentially linked to other statistics so that they can be weighted accordingly

Machine Learning

The feature extraction process provides the raw material on which to construct models to explain the data and to formulate predictions about stock prices given new Tweets. In our project we will consider three main statistical models to uncover the patterns in the data. Each of these models will make use of a portion of the available Twitter data (between 70% and 90% of the data, ordered chronologically) in combination with stock prices to extract the optimal parameters (according to a loss function). The parameterised models will then be used to predict the more recent performance of the relevant stocks given the remaining portion of the Twitter data. These predictions will then be evaluated against the real-world performance. The models are:

Linear Regression

We will use regularised linear regression (ridge regression) with the mean squared error loss function to give us a simple baseline on our predictive performance. With this model, we hope to find a linear relation (i.e. linear coefficients) between the features previously extracted and stock prices. The regularisation will be parameterised by a lambda value which will determine the extent to which more complex (larger) coefficients will be penalised in our loss function.

Support Vector Machine

When predicting simple increase or decreases in stock prices (see Evaluation section), the problem becomes a classification rather than a regression one. Thus we can use a support vector machine (experimenting with possible kernels and respective parameters) to fit a hyperplane in the feature space between the time splits in the Twitter data that corresponded to increases in the relevant stock prices from those that corresponded to decreases.

Gaussian Process

For the regression task we will also attempt to describe the stock market as a Gaussian interaction of multiple samples defined by a suitable (kernel) covariance matrix. Defining this matrix will allows us to model periodicities, and the decay of correlation between adjacent samples. Out of the three, this model is the most elaborate as it allows for the greatest flexibility on the relation between the features and the stock prices. In fact the kernel covariance can encode very complex nonlinear relations between features and data-points. This requires also a higher degree of cross-validation to select the most effective kernel and its hyperparameters.

Performance Evaluation

ML algorithm performance

To demonstrate the effectiveness of our learning algorithms we proceed by splitting the data into testing and training sets. Than we predict the future stock price using the training set and compute the mean squared error of the prediction using the test set. An example of discrepancy between the predicted and actual price that we hope to minimise is shown below.

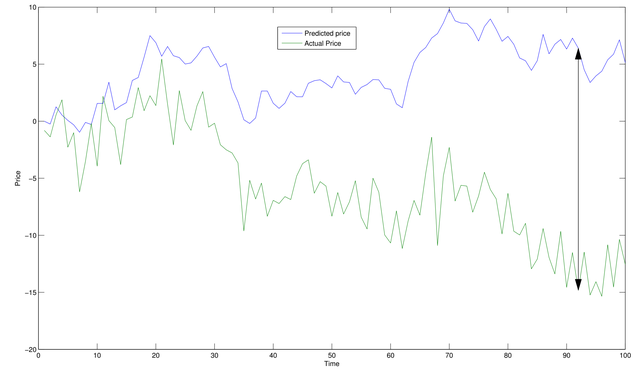

Figure 2: The arrow demonstrates the differences between the predicted price and the actual price in the test set that we hope to minimise.

By carrying out the above procedure with different sets of test and training data we will be able to establish the predictive power of our model as well as determine how far into the future we will be able to make effective predictions. It is expected that the performance of the prediction will deteriorate for predictions further into the future as shown in the figure above.

If the system will be used for training than the complex regression problem can be reduced to a simpler classification problem. Instead of predicting the exact price we will instead classify whether stock is going to go up or down after a specific amount of time. When evaluating this approach we intend to use the number of misclassification as a metric for determining the predictive power of our algorithm.

Back-testing

While mean square error in prediction is useful in evaluating algorithm effectiveness, low mean square error does not directly translate into trading performance.

The simple price model will need be further extended to incorporate stock liquidity and ensure that gains can be realised. Further optimisation is possible by incorporating trading fees as well as liquidity rebates to ensure that the system not only maximises prediction power but also profitability.