Project Report for CIS 419/519
Stock Recommendations using Machine Learning

Abstract

In this report, we present a simplified stock recommendation system. This system is based on patterns found from several features including the price and volume history of a list of equities. It utilizes various machine learning techniques to try to predict which stocks should be bought or sold to maximize return on investment going forward. We chose to employ neural networks, support vector machines with a gaussian kernel, and naive bayes as our main machine learning techniques. We trained each model on past data through a backtesting module and then presented each model with new data in order to make predictions. In addition, we utilized our module to test several different combinations of features and compared the accuracy, precision, and recall of various models.

1. Introduction

Many financial institutions and investors are wary of trading in the stock market since the financial collapse in 2008. Almost all financial institutions have turned to machine learning or other algorithmic techniques in order to reduce the risk in trading. These techniques are used both in high-frequency trading and longer-term trades based on a buy and hold strategy. Our system is targeted as a longer-term trade recommendation system, avoiding the real-time constraints of high-frequency algorithms. We attempt to solve a small subset of this larger financial landscape, making good short-term predictions for buying or selling stocks that the average investor might be able to utilize.

The principle of our trading strategy is based on the idea that price movements in a stock form technical patterns that represent the sentiment of a particular equity over time. These patterns can help predict the future price movement of a stock, which if known, can improve the overall return for an investor. In order to simplify our recommendation model we have limited our selection to the stocks in the Dow 30 and only look at closing price for the stocks in our list as the basis of our predictions. Also, in order to avoid the complication of taxes and trading costs we decided to subtract a flat 0.5% from each trade made.

1.1. Model assumptions

In our implementation, we make several assumptions to simplify the constraints. We believe that while these assumptions necessarily make the algorithms we develop less applicable to the real world, they are still not too far off from reality, and could potentially be used by individuals in the real world at some success, but at their own risk. We will first assume that the trader has both cash and shares of stocks. We can calculate the value of the trader's portfolio at any time. We can also calculate the rate of return of any investment the trader has made from the time it started to present time.

An additional assumption will be that the closing price reflects the sentiment of that day's price action enough to be considered the price for the entire day. This has the added benefit of avoiding intraday fluctuations in the stock market. We will also ignore all transaction fees and taxes associated with trading on the stock market and simply use a flat fee of 0.5%.

1.2. Definitions

$$Port.Value(V) = Cash + (numShares * stockValue)$$

$$ROI = \frac{(V_{now} - V_{beginning})}{V_{beginning}}$$

2. Methodology

2.1. Source of data

The dataset used was downloaded from Wharton Research Data Services (WRDS) with price and volume as the primary features and comprised the 30 stocks from the Dow.

It was necessary to normalize our data across different

University of Pennsylvania, CIS 419/519 Course Project.

stocks so volume and price are comparable to each other. That way, a stock that costs $100 does not dominate a stock that costs only $10. Volume was normalized by subtracting Average Daily Volume over the slice of time within our pattern's time frame.

2.2. Trading strategy

The strategy is to buy or sell a stock from the collection of stocks we follow within the Dow 30 and based on patterns or trends our models find, execute a trade at a given period of time or until our predicted price target is hit. Alternatively, our prediction is wrong and we exit our position with a stop loss set to 2%. This is to minimize our potential loss when we are wrong and maximize our profit when we're correct.

3. Progress so far

3.1. Backtesting

We have written a module to look at different slices of time and different feature sets in our historical trades data. We have also implemented pattern recognition in our backtesting code to confirm that there are trends on a daily basis that we could exploit.

Section 3.5 shows some example patterns that our system found with the current pattern in cyan. The other lines are past patterns that our system found that are potential matches and the dots to the right represent outcomes for each past pattern. The rightmost dots are the actual outcome and average of the past patterns.

3.2. Machine learning framework

At this point, we've also set up another framework that allows us to easily evaluate different classifiers, different amounts of data, and different features easily. This will help us decide which classifiers, features, and data to use to train each respective machine learning model for best results.

We've also reviewed numerous published papers to understand what existing work has been done, both successful and unsuccessful, which will ultimately aid in narrowing the scope on our final evaluation and help us determine best train and test practices.

3.3. Classifier Evaluation

So far we've evaluated a boosted decision tree and SVMs with polynomial, gaussian, and cosine similarity kernels using our framework. We've evaluated small subsets of data and different combinations of features for our preliminary trials in order to get some idea of performance without investing too much time. So far the gaussian kernel is performing the best of our set of tested classifiers, but requires additional tuning to be a good predictor in general. The boosted decision tree and other SVM kernels performed about the same as random guessing.

3.4. Future Work

Next, we plan to evaluate additional classifiers, including a neural network and naive bayes, with similar baseline variables for data and features.

As a stretch goal, we are also still trying to find additional sources of data to incorporate additional economic features into our dataset.

3.5. Figures

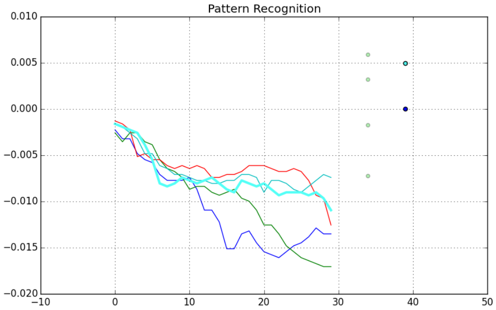

Figure 1. Examples of downward trend reversals

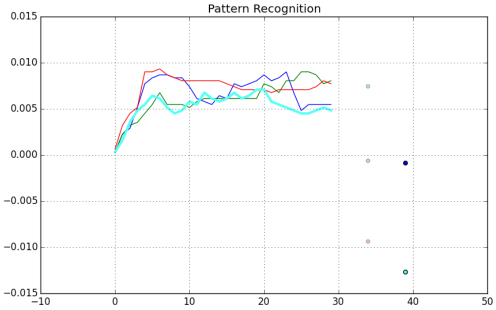

Figure 2. Examples of upward trend reversal

Acknowledgments

We would like to thank Eric Eaton and the TA's for their help with gathering data and getting started.

References

Chase Lochmiller, Yuan Chen. Predicting short term stock returns. Technical report, Stanford University, 2013.

Chenxu Shao, Zheming Zheng. Algorithmic trading using machine learning techniques. Technical report, Department of Management Science and Engineering, 2013.

Lin, Hui. Feature investigation for stock market prediction. Technical report, Stanford University, 2013.

Shunrong Shen, Haomiao Jiang, Tongda Zhang. Stock market forecasting using machine learning algorithms. Technical report, Department of Electrical Engineering, Stanford University, 2012.

Tianxin Dai, Arpan Shah, Hongxia Zhong. Automated stock trading using machine learning algorithms. Technical report, Department of Management Science and Engineering, 2012.