

Relative and absolute equity performance prediction via supervised learning

Alex Alifimoff
aalifimoff@stanford.edu

Axel Sly
axelsly@stanford.edu

Introduction

Investment managers and traders utilize two different types of indicators to guide their trading decisions. “Technical” indicators are composed of information primarily visible on price-history charts: these include the recent price changes and volume for a particular security. “Fundamental” indicators, like Price/Equity Ratio and Quarterly Revenue, are based upon numbers primarily found on a company’s balance sheet. This paper seeks to evaluate the effectiveness of both fundamental and technical indicators in informing modern trading and investment decisions by using them as features in supervised learning to predict absolute and relative equity performance.

Our work attempts to build on a previous CS229 term project, “Automated Stock Trading Using Machine Learning Algorithms” by Dai, Shah, and Zhong by expanding return prediction to the investment-grade time horizon.

Absolute Return Prediction

The holy grail in modern financial markets is absolute return prediction. We began our project by attempting to use six months of monthly price and volume data¹ and two quarters of Price/Equity Ratio, Current Market Capitalization, and Price/Book Ratio² as our features to predict the change in equity price over the next ten months using a variety of supervised learning algorithms. In total, we used 18 features. With the exception of past return data, we preprocessed all of our features inputs by removing the mean and scaling to unit variance.

We selected 800 equities from the Russell 2000 index³ as our training set. We aimed to classify returns over the next year as either “positive” or “negative”. If p_0 is the price of an equity today, we decided to predict the value of $1 \{p_x - p_0 \geq 0\}$ for some time period x . We used Support Vector Machines, Logistic Regression, Decision Trees, and Gaussian Naive Bayes to try to classify the data. Our results on the twelve month timeframe are summarized in table 1.

Table 1. Results - 12 Month Absolute Prediction

Algorithm	Training Accuracy	Testing Accuracy	Precision	Recall
SVM 1(Gaussian)	65.6%	61.5%	82.0%	26.6%
SVM 2(Gaussian)	63.5%	62.3%	98.6%	00.3%
Logistic Regression	61.3%	58.1%	66.2%	44.3%
Decision Tree	100%	56.0%	63.2%	43.9%
Gaussian Naive Bayes	60.8%	58.9%	78.8%	25.0%

¹ Technical indicators

² Fundamental indicators selected because of their relevance to financial analysts

³ Unfortunately, there isn’t robust data available for technical indicators for small-market capitalization stocks, so we had to omit some of the original 2,000 Russell components.

SVM 1 corresponds to a Gaussian SVM with a class-weight inversely proportional to the frequency of examples in that class and a penalty parameter of 22.0. SVM 2 refers to a Gaussian SVM with the same penalty parameter.⁴ The underlying distribution for this dataset was 62% positive examples. As a result, the best performing algorithm is only marginally better than educated guessing.

We elected to use the same feature set to predict returns over just the next month, hypothesizing that our indicator set might contain more information about short term price fluctuations than long-term movements. Our results for the one month time horizon are summarized in table 2.

Table 2. Results - 1 Month Absolute Prediction

Algorithm	Training Accuracy	Testing Accuracy	Precision	Recall
SVM 1(Gaussian)	74.4%	73.4%	93.2%	21.5%
SVM 2(Gaussian)	72.6%	72.3%	99.6%	00.5%
Logistic Regression	72.7%	71.3%	88.2%	26.8%
Decision Tree	100%	64.3%	75.4%	35.2%
Gaussian Naive Bayes	72.7%	71.8%	92.8%	16.8%

SVM 1 refers to Gaussian-kernel with a penalty parameter of 20 and class weights inversely proportional to class frequency. SVM 2 refers to Gaussian-kernel with penalty parameter of 18.

While these results once again seem promising, the underlying distribution of returns was 73% positive examples. As a result, it became quite clear that attempting to predict the absolute return using this particular feature set was primarily an exercise in futility.

We decided the next most apt step in refining our algorithm was redefining our feature set: low training and testing accuracy convinced us that our features contained very little information about what we were ultimately trying to predict. We decided to attempt to vectorize the majority of a company's financial reports to determine if they contained any information that might be useful in predicting future performance. However, obtaining robust financial records for the small-market capitalization equities like those in the Russell 2000 proved to be next to impossible. To conduct analysis on full financial statements, we needed to change our training set to larger capitalization stocks.

We used all five hundred components of the S&P 500 as our training set. We used the same technical features as before (price and volume data for past six months) in addition to 13 key fundamental indicators⁵ from the past three quarters of the company's financial statements. In total, the technical feature set consisted of 12 features and the fundamental feature set consisted of 37 features.

⁴ These were the best performing SVMs, selected by 10-fold cross validation.

⁵ We used Cash on Hand, Stockholder's Equity, Retained Earnings, Beginning Net Income, Long Term Assets, Current Assets, Accounts Payable, Income, Invested Capital, Revenue, Earnings per Share, Earnings per Share (Operating), Current Liabilities, and Long Term Liabilities. We had three quarters of data for each indicator. Finally, we used the current market sector, as our final fundamental indicator.

We utilized these two feature sets to demonstrate the performance of absolute return prediction for the S&P 500. We again tried to predict returns one month into the future. Our results, compared to a “best guess” of choosing the dominant market trend⁶, are shown in table 3.

Table 3. Performance under various feature sets - Gaussian Kernel SVM⁷

Training Set	Accuracy Improvement vs. Best Guess
Fundamental & Technical Features	-4.2%
Technical Features	-5.0%
Fundamental Features	-4.5%

Our initial results on the Russell 2000 dataset were only confirmed by our analysis on the S&P 500. The S&P analysis was even worse! Fundamental and technical indicators carried very little information that could be used to predict future absolute returns.

We concluded that absolute return prediction is highly subject to outside influences that have little to do with the numbers on a company’s financial reports and past return data. Typically, macroeconomic influences that drive market confidence have much greater effects on equity prices than a company’s personal financial health.

Relative Return Prediction

We decided to try to refocus our algorithms by attempting to remove market risk from our prediction. Instead of classifying whether the return of a particular equity was positive or negative, we decided to examine the relationship between an equity’s return and the return of the market overall. In this phase of the project, we attempted to predict:

$$1 \left\{ \frac{(p_x - p_0)}{p_0} - r_{index} \geq 0 \right\}$$

where r_{index} is the return of the S&P 500 market index from time period 0 to time period x .

Once again, we looked at the S&P 500 components and used our 37-dimensional feature set of fundamental and technical indicators to predict relative returns. Our results are shown in table 4.

Table 4. Performance of Relative Return Prediction for different learning algorithms

Algorithm	Training Accuracy	Testing Accuracy	Precision	Recall
SVM (Gaussian)	73.2%	60.1%	47.2%	66.0%
Logistic Regression	67.0%	55.0%	51.6%	59.7%
Decision Tree	100.0%	58.0%	54.6%	56.6%
Gaussian Naive Bayes	54.8%	54.0%	19.8%	84.9%

SVM refers to a Gaussian-kernel Support Vector Classifier with a penalty parameter of 20. In this example, the underlying distribution of positive and negative examples was 50% and 50%, so any performance over 50% is a gain over random-guessing. Our best algorithm actually produces a relatively appreciable gain over random guessing.

⁶ For example, in a well-performing market, as in the time period we were examining, the best guess is that all equities will perform well.

⁷ Penalty parameter of 20 and no class weighting

We were interested in understanding which feature-types were the most useful in predicting future equity returns. We grouped our features into four sets: I Fundamental (the most recent quarter’s financial statement features), II Fundamental, III Fundamental (the oldest quarter’s financial statement features), and Technical and reconducted our analysis under each feature set. The results of this analysis are in table 5.

Table 5. Performance of Relative Return Prediction with different feature sets⁸

Feature Set	Accuracy Improvement vs. Best Guess
All Fundamental & Technical Features	10.1%
I, II Fundamental, All Technical	8.0%
I Fundamental, All Technical	5.0%
All Technical Features	-2.0%
All Fundamental Features	8.0%

Analysis of Results

We believe that our results provide some insights into the structure of equity markets. Our algorithms used features that were almost entirely company-specific in that they did not provide a picture of the overall health of the economy. An individual’s decision to invest in a capital market is highly driven by macroeconomic factors like the risk-free interest rate, unemployment, and current government policy towards the market. Since our model did not use these factors to make predictions, it isn’t entirely surprising that it was difficult for us to predict absolute equity performance.

However, our model is useful for evaluating the relative performance of equities. There is substantial information contained within our feature set about how one equity will perform relative to another. This insight is inline with modern financial theory, which suggests that equity performance is a function of a particular stock’s movement with the market, and that market risk is determined by other, primary macroeconomic, factors. Our model captures at least some element of idiosyncratic risk.

Our analysis also reveals which elements of the feature set are most useful in predicting equity returns. By far, the fundamental feature set was the most useful for evaluating company performance. Likewise, the most recent fundamental data contained far more predictive power than the older data. This corresponds with our intuition that recent company financial events are most relevant to a company’s performance.

Further Work

Our work demonstrates that supervised machine learning has the ability to aid asset managers and traders in projecting relative returns. However, our work is far from complete. We see several ways in which our work could be extended.

First, we think it would be interesting to build on our analysis by including macroeconomic features like unemployment, GDP, and housing starts. It is possible that these metrics would be useful in predicting absolute returns. Alternatively, these same features could be used to predict generalized market returns. A semi-accurate regression of future S&P returns would greatly increase the value of our research.

⁸ Conducted using Gaussian SVM with penalty parameter of 20.0.

Our analysis only looks at a selection of equities for one point in time. Extending the analysis to multiple time periods would demonstrate if the predictive trend is time dependent or could be more generally applied. This kind of analysis is the next step to truly determine if a firm could use a supervised machine learning model to generate a consistent profit.

Our initial analysis only looked at the S&P 500 and the Russell 2000. Market capitalization directly constrains the investment capability of financial actors. For example, a large bulge-bracket bank can't invest in micro-cap stock because the maximum investment amount is constrained by stock capitalization. This means that equities with different market capitalizations could have vastly different behaviors. Extending our analysis to different market capitalization ranges could potentially yield some quite interesting insights into market structure and behavior.

Furthermore, our work only considers the application of supervised learning to return prediction. However, large financial institutions are often concerned with other metrics, like volume, which become extremely relevant when the institution needs to move large amounts of equity around for clients. Leveraging the predictive power of machine learning to develop tools to predict other financial indicators, like volume, is a natural extension of our work.

However, we also believe that there are a number of interesting applications of unsupervised and reinforcement learning to financial markets. We believe that unsupervised learning could aid in understanding the components that drive return variance in modern financial markets. Reinforcement learning, on the other hand, presents several opportunities for developing a full fledged trading strategy in tandem with a predictive model like ours. Our references include a very exciting application of reinforcement learning to order execution.

References & Data Sources

Data was gathered using a combination of CompuStat (company financials), CRSP (past return and volume data), and Bloomberg (selected company financials). We implemented all of our models using the phenomenal scikit-learn library available for Python. Finally, we would like to thank Andrew Ng for his phenomenal course and the entire teaching staff for their help and wisdom throughout the quarter!

Hsu, C , "A Practical Guide to Support Vector Classification". Retrieved September, 2014 Available: <http://www.csie.ntu.edu.tw/%7Ecjlin/papers/guide/guide.pdf>

Dai, T , "Automated Stock Trading Using Machine Learning Algorithms". Retrieved September , 2014 Available: <http://cs229.stanford.edu/proj2012/ShahDaiZhong-AutomatedStockTradingUsingMachineLearningAlgorithms.pdf>

Nevmyvaka, Y , "Reinforcement Learning for Optimized Trade Execution ". Retrieved September , 2014 Available: <http://www.cis.upenn.edu/~mkearns/papers/rlexec.pdf>

Pedregosa, F , "Scikit-learn: Machine Learning in Python". Retrieved September , 2014 Available: <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>