Georgia Institute of Technology

4133 Advanced Optimization

Convex and Non-Convex Classification of S&P 500 Stocks

Matt Faulkner
Chris Fu
James Moriarty
Masud Parvez
Mario Wijaya

coached by
Dr. Guanghui (George) Lan

December 19, 2017

Contents

1 Introduction

This study focused on the Companies of the SP 500, and their metrics as collected on, April 24th, 2017. The objective was to classify whether particular companies were leading or trailing their 200 Day Moving Average based on the up-to-date financial metrics and information.
The Standard and Poors 500 Index, also known as the S&P 500, is a measure of the value of the market at a point in time for the outstanding shares of the best performing 500 American publicly traded companies. These stocks are viewed as leading indicators for the performance of large-cap U.S. equities, as selected by the S&P Index Committee. The index is possibly the most followed and thoroughly analyzed in the world, and accordingly, there is wealth of accurate, current, and accessible data pertaining to the finances of its component companies.
When choosing companies to invest in, investors want to determine whether the companys stock price will rise in the future or whether it will fall. If the price rises, the investor makes a profit, but if it falls the investor takes a loss. The determination of whether a stock will increase or decrease in price is a simple machine learning problem. More precisely it is a classification problem. This project seeks to classify S&P 500 securities as leading or trailing their 200 day moving average. Both convex and non-convex models are explored.

2 Data Collection

The data collection infrastructure was built collaboratively via a group git repository from source at https://finance.yahoo.com/ and https://www.wikipedia.org/. The collection process starts with a web scraper that parses Wikipedias daily official listing of S&P 500 companies. This scraper identifies the companies of the S&P 500 by their respective tickers, names, and industries. This list is refreshed daily and cached in the project repository. The codebase includes a python3 command line tool which calls Yahoos Financial API for information on specific companies. A daily script fetches financial attributes for every company in the cached S&P 500 list and stores the resulting data in a local csv file.
Before the data is stored, it is run through a filtering process that enumerates all values. This cleaning stage allows for flexible filtering, adjustments, and creation of new company attributes. The command line tool can report filtered or unfiltered data for any public company (including non S&P 500). Examples of attributes created include percent change from one year analyst consensus and percent change in after hours trading. The filtering process also facilitates feature scaling and transforming the labels into binary or continuous values.
Future improvements to the data collection process may include:

- Implementing a hosted database to overcome some limitations imposed by the current csv system
- Further automation of the collection process and more frequent samplings
- Collecting data on non-S&P 500 companies
- Integrating more data sources (in addition to Yahoo and Wikipedia)

3 Final Dataset

Our final data set *binary-04-24-2017.csv* represents a snapshot of the specified financial attributes for 428 S&P 500 companies **Appendix A**. There are 18 quantitative predictors and 10 sector qualitative data instances. The 10 sectors are Consumer Discretionary, Consumer Staples, Financials, Healthcare, Industrials, Information Technology, Materials, Real Estate, Telecommunication Services, Utilities, and Energy. The energy sector is the reference base model for the qualitative data. The data points are not time dependent as we only grabbed the information for a single day: April 24th, 2017.
Dependent Variable (Response):

- Percent Change from 200 Day moving Average - This compares the current stock price change and compared to the 200 previous days moving average: can be positive or negative

Independent Variables:
Quantitative:

1. AverageDailyVolume - The average value that represents the volume of stocks that have been traded for each instance in our data set over a year
2. BookValue - The market stock price of each instance
3. DaysRange - The price spread for a defined period, day
4. DaysVolume - The day volume of the previous day for a given stock
5. DividendYield - How much dividend yield a stock achieves for its stockholders
6. EBITDA - Earnings before interest, taxes, depreciation, and amortization
7. EPS - Earnings per Share
8. EPSYearGrowthEstimate - Estimate of the earnings per share the instance is expecting or estimating to achieve in the year to come
9. MarketCapitalization - The market cap is the value of the outstanding shares of a company
10. OneyrTargetChange - An analysts expectation of price for an instance in one year

11. PEGRatio - Price per earnings to growth ratio for determining the relative trade off between price of stock and EPS
12. PERatio - Price/Earnings ratio; ratio of companys stock price to the companys EPS
13. PercentChangeAfterHours - Percent change a company receives after banking and trading hours
14. PriceBook - This is the price of the stock at book value
15. PriceEPSEstimateCurrentYear - The estimated EPS for the current year as predicted by analysts
16. PriceEPSEstimateNextYear - The estimated EPS for the next year as predicted by analysts
17. PriceSales - A ratio value achieved by dividing the per share stock price and per share revenue
18. ShortRatio - A ratio of consumers attempting to short a company in the hopes a stock will decrease

Qualitative Data (by Sector)

1. Energy (Reference/Base) - Includes energy equipment & services and oil, gas & consumable fuels
2. Consumer Discretionary - Sector of goods and services
3. Consumer Staples - Includes beverages, food & staples retailing, food products, household products, personal products, and tobacco
4. Finance - The financial sector, included banks, capital markets, and insurance
5. Health Care - Includes biotechnology, health care equipment & supplies, health care providers & services, healthcare technology, life science tools & services, and pharmaceuticals
6. Industrials - The industrial realm, includes machinery, road rail, and airlines
7. Information Technology - Includes communication equipment, IT services, and software

8. Materials - Includes chemicals, construction materials, construction & packaging, metals & mining, and paper & forest products

9. Real Estate - Includes equity real estate investment trusts and real estate management & development

10. Telecommunication Services - Includes diversified and wireless telecommunication services

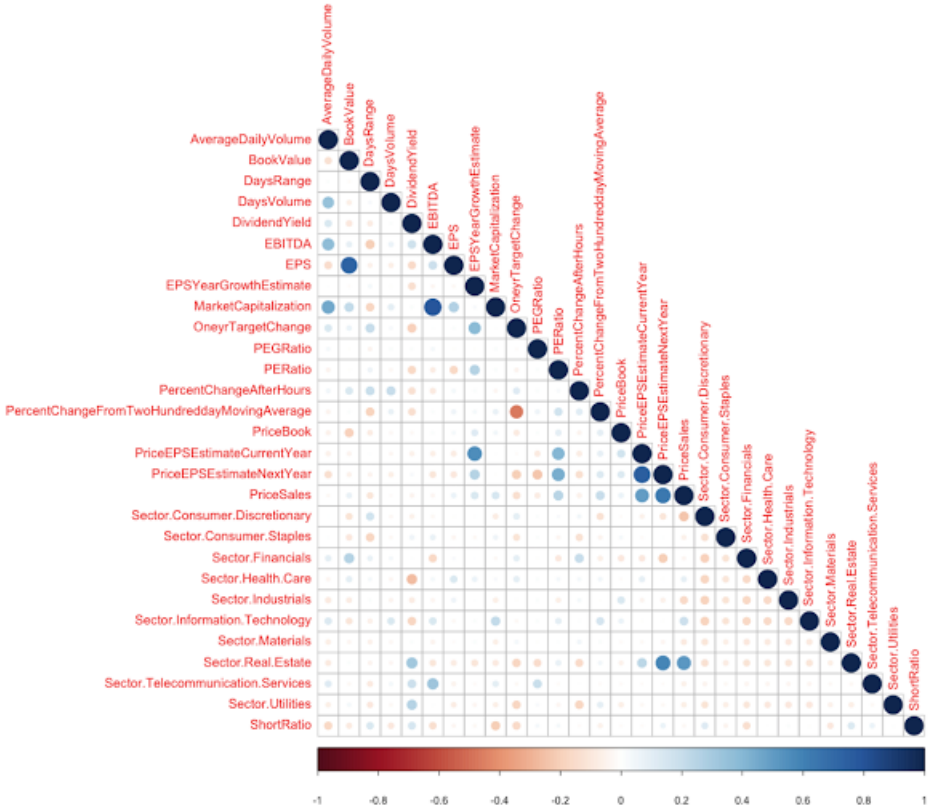

Figure 1: Covariance matrix of financial attributes contained in the dataset

The attribute used as the label for the classification models is the percent change from 200 day moving average. From this attribute a binary label was created: -1 for negative percent change, 1 for non-negative percent change. Other features needed to be scaled in order to be used by the classification models. To accomplish this scaling the following equation was used:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Figure 2: Scaling equation

4 Models

4.1 Soft-Margin SVM

Soft-Margin SVM is an extension to SVM in which the *hinge function* is utilized due to data not linearly separable. The *hinge function* is as follows:

$$\max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b))$$

This function is zero if the constraint in (1) is satisfied ($\vec{x}_i$ is on the correct side of the margin). If data is on the wrong side of the margin, the function's value is proportional to the distance on the margin. The model we wish to minimize is as follows:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b))\right] + \lambda||\vec{w}||^2,$$

λ determines the tradeoff between increasing margin-size and ensuring that the $\vec{x}_i$ is on the correct side of the margin.

4.2 Logistic Regression

Logistic Regression is a binary classification model that is formulated by solving the following optimization problem:

$$\max_{\theta}\sum_{i=1}^{N}\left\{-\log\left[1+\exp\left(-\theta^T u^{(i)}\right)\right] - \left[1 - v^{(i)}]\theta^T u^{(i)}\right\}$$

Figure 3: Logistic Regression Optimization Formulation

This classifier was chosen because it is a robust model that is formulated via a convex optimization problem, which can be solved using a variety of algorithms.

4.3 Stochastic Gradient Descent

Stochastic Gradient Descent, derived from the gradient descent optimization method aims to minimize an objective function written as a sum of differentiable functions. SGD is chosen because it converges faster and require less computation compared to other algorithms such as Gradient Descent and Mirror Descent that were discussed in class. The algorithm for SGD is as

follows:

$$\min \frac{1}{N} \sum_{i=1}^{N} \left(a_i^T x - b_i \right)^2 \quad x \in R^n$$

For $k = 1, ...,$
$G_k = 2(a_{i_k}^T x_{k-1} - b_{i_k}) a_{i_k},$
where i_k is an uniform random variable over $1, ..., N$
$x_k = x_{k-1} - \gamma_k G_k$
End

5 Cross Validation

In order to be unbiased with our prediction, we used the cross validation technique. We had 10 fold cross validation for each of our predictive model. For each fold, we shuffled our data and used that shuffled data in all of our convex and nonconvex algorithms. We averaged our prediction for each algorithm after 10 fold. We also used different split to choose our test and train data and we came up with 70-30 split as being the most reasonable with this dataset.

6 Results and Analysis

6.1 Convex Model

We tested 2 different types of convex classification models: soft margin support vector machines and logistic regression models. For each model, 3 different loss functions were tested: the L-2 norm, the L-1 norm, and the elastic net (a convex combination of the L-1 norm and L-2 norm). All of the models tested were optimized using stochastic gradient descent. An important parameter is the learning rate of the descent algorithm. The first test was to run the models using SKLearns default learning rate:

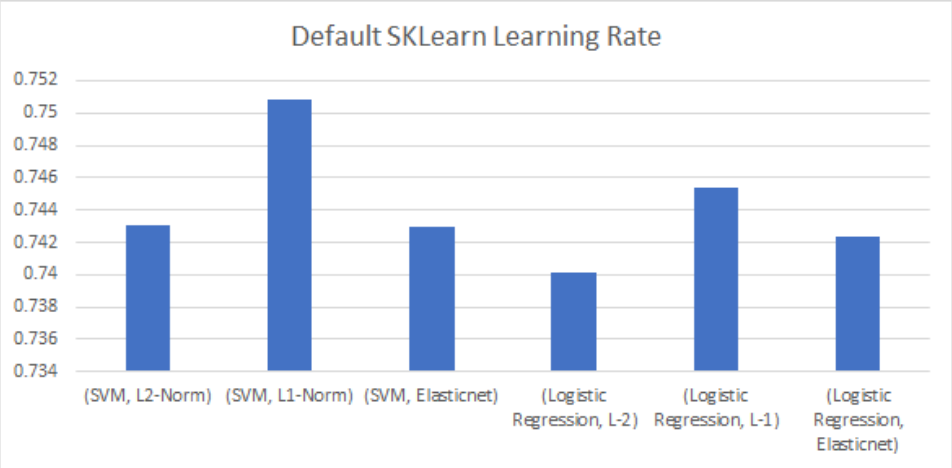

Figure 4: Percent of test data correctly classified by each model (default learning rate)

As can be seen in the above figure, the SVM model does better than the logistic regression, with the L-1 Norm being the best loss function for both models. With the goal of improving the accuracy of the models, the learning rate was modified to be based on the Lipschitz Constant. The next iteration of models tested used an inverse scaling method for the learning rate with an initial rate of $1/\sqrt{2L}$. As the algorithm went through more iterations, the learning rate decreased.

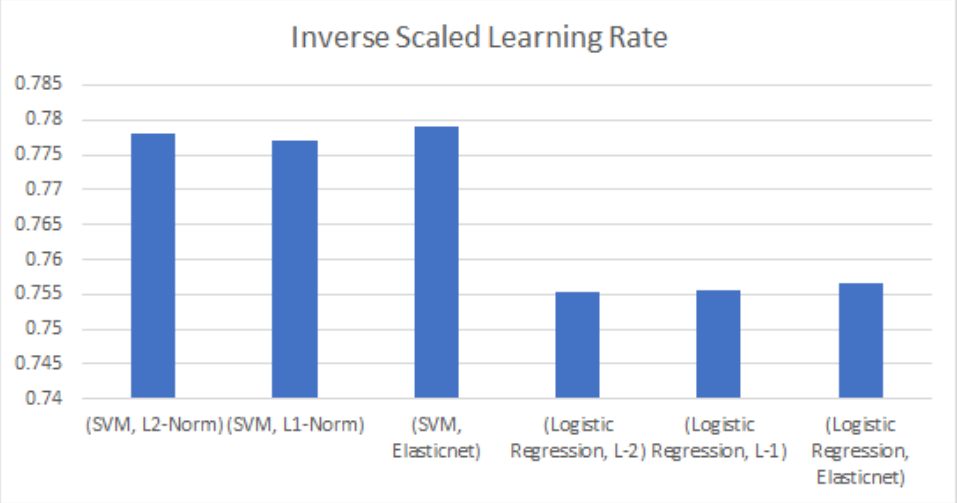

Figure 5: Percent of test data correctly classified by each model (inverse scaled learning rate)

Clearly changing the learning rate to a more optimal value improved the accuracy of the model, particularly with the SVM models. The elastic net loss function has a slight edge in this set of tests. The last set of tests was to hold the learning rate constant at $\gamma = \frac{1}{\sqrt{2L}}$

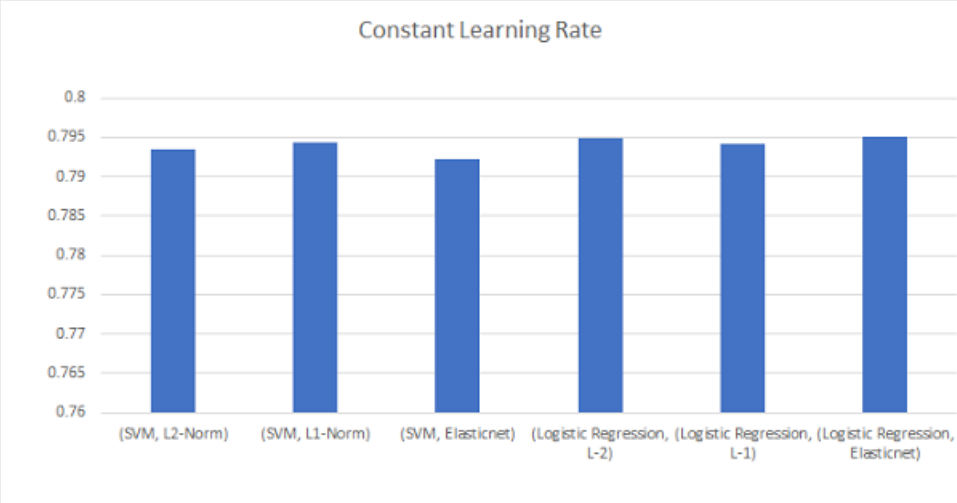

Figure 6: Percent of test data correctly classified by each model (Constant learning rate)

This set of models performed the best out of all of the convex models tested. It is noteworthy that the accuracy of the model was improved significantly

by optimizing the learning rate to $\gamma = \frac{1}{\sqrt{2L}}$. In this set of tests the accuracy is more than 4 percent higher than the original default learning rate. That the constant learning rate models performed the best is also consistent with theoretical analysis.

6.2 Non-Convex Model

In addition to using convex models, the team used non convex models to compare to the accuracy of the convex models. The non convex models used were Neural Networks and Naive Bayes. Neural networks model consist of input layer, hidden layer, and output layer. Input layer refers to the 28 attributes and output layer refers to our label % change of 200 moving average. For the neural networks model, 28 nodes were used for input layers. The accuracy for this model using 70% training set and 30% testing set as described in the Cross Validation section is 80%. Also, the team varied the number of hidden layers of neural networks to see how it would affect the accuracy. It turned out that the accuracy decreased as the number of hidden layers increased as shown on Figure 7.

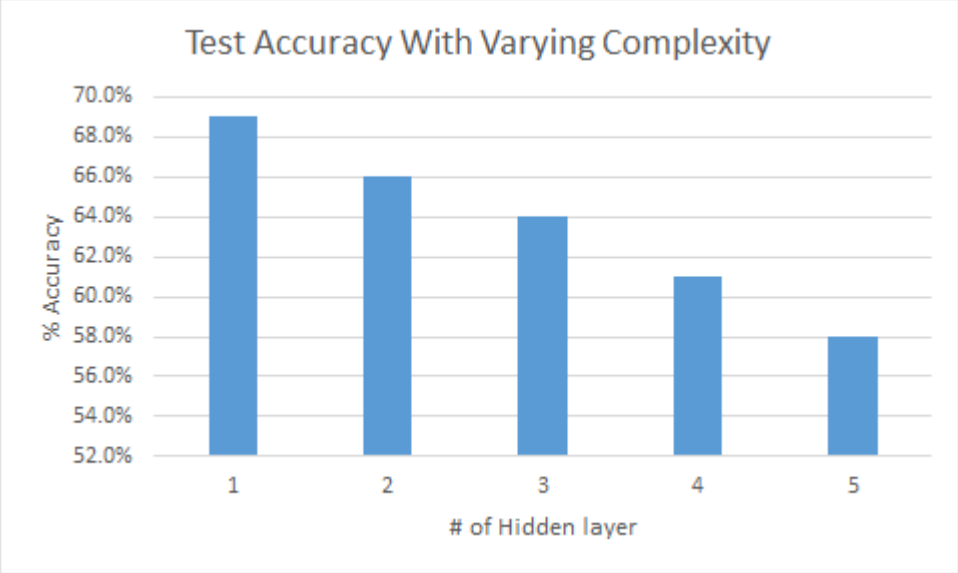

Figure 7: Percentage accuracy with varying number of hidden layers using Neural Networks algorithm

Next, we used Naive Bayes model to see how it differs compared to the convex model we built. The accuracy obtained is 65% with 70-30 split.

7 Conclusion

Through the model used, 80% classification accuracy with stochastic gradient descent support vector machine is obtained. Meanwhile, our model surpassed default SKLearn accuracy by 5% using calculated Lipschitz-based gamma. From multiple models, the benefits of convex models (SVM) and non convex models such as Neural Networks and Naive Bayes were displayed.