Stock Market Price Prediction Using Linear and Polynomial Regression Models

UNM Department of Computer Science

Lucas Nunno University of New Mexico - Computer Science Department

Introduction

The main question driving this project is: Is it possible to predict stock prices in the future and if so, how accurate are these predictions?
There were a number of objectives in this project.

1. Investigate linear and polynomial regression methods and see how these models can be used to predict stock price.
2. Explore how the the training set window size effects regression quantification metrics. In other words, does a model perform better when it has a more detailed picture of the past?

Motivation

Stock market price prediction is a problem that has the potential to be worth billions of dollars and is actively researched by the largest financial corporations in the world. It is a significant problem because it has no clear solution, although attempts can be made at approximation using many different machine learning techniques.

The project is a direct application of real-world machine learning applications including acquiring and analyzing a large data set and using a variety of techniques to train varying models and predict potential outcomes from these fitted models.

Materials & Methods

The following resources were used in order to conduct this research:

- Center for Research in Security Prices (CRSP) US Stock Databases
- Python scientific and data analysis libraries including: numpy, pandas, and scikit-learn

The process of converting the static price data into a vector of price predictions can be visualized as follows.

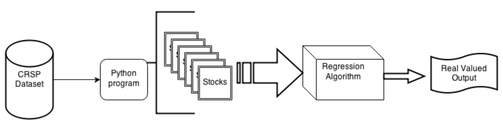


Figure 2: Data-flow of the program showing how stock data turns into prediction value vectors.

Methods (contd.)

A number of data transformations are required prior to and after feeding our data into our regression black boxes:

- Every training dataset must be normalized to a Gaussian normal distribution between -1 and 1 before the input matrix is fit to the chosen regression model. This is for each feature including: Date, Closing Price, Shares Outstanding, Volume, and Return on Investment.
- The data is then fed into the following regression algorithms: Linear, Support Vector Polynomial, Support Vector Radial Basis Function (RBF).
- These results are then compared using regression metrics algorithms to gauge performance.

Results

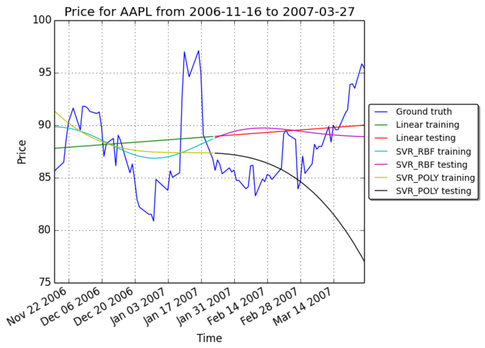
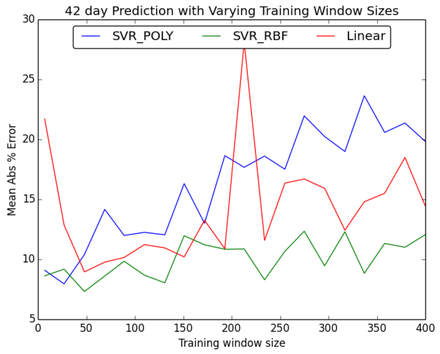


Figure 1: Comparison of several regression methods of a single stock on a fixed time-frame and their training and testing models visualized.

The preceding figure shows a comparison of regression methods. The training data is the fit the model performed on *known* data the testing data is the prediction that the model made on *unknown* (withheld) data.

The window sizes were varied in order to determine if that affected the prediction's accuracy. Each error percentage represents the average of 100 trials of random stock and initial date selection. SVR w/ RBF hovered around $5 - 10\%$ price prediction error for a 2 day window, and a more consistent 10% for 42 day prediction (pictured). More window variation results are in the paper.



Analysis

- Notice in Figure 1 how the polynomial method fits better than the rbf kernel for the first month and then diverges, this explains our results that the polynomial kernel had higher mean absolute error, but did better for shorter price projections.
- Linear regression performed better than expected, since it is usually the canonical example to overly-simplistic models for a complex adaptive system.
- If using these regression models in real life, one might wish to only predict a week or less in advance and use the tightest fitting regression model, since updated data is readily available.
- All the regression methods performed the best on a window size of ≈ 50. Training windows on the order of years tended to have exponentially worse performance.
- SVR with the RBF kernel was the overall best performer, with window size not greatly effecting it's comparative ranking.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] Center for Research in Security Prices The University of Chicago. Us stock databases. web, 2014.

Conclusions and Future Research

It is interesting how linear regression can perform better than polynomial methods at certain intervals due to the reduced chance of linear regression overfitting the training data. In some cases, we found that for long term projected market fluctuations linear regression performed well. This case was especially true when a polynomial method would overfit the training data and have increased performance at the beginning of the testing data, but at the cost of very inaccurate results in the later prediction dates. Conversely, linear regression was less accurate at the beginning of the prediction, but wouldn't perform as badly as a polynomial regression method that diverged.
Future research could be done in the following areas for this problem domain:

- Using additional linear and polynomial regression methods as black boxes.
- Continuous parameter adjustment and comparison to find optimal combinations for the problem domain.
- Integration of more features to apply decision trees, and other methods for industry and sector price regression.