Financial Market Analyzer / Predictor

Contributors: Matthew Rastovac, David Chang, Sean Stockwell

Author: Matthew Rastovac

Introduction

The purpose of this project is to create a basically functional financial market performance estimator by utilizing past trends and economic theories. Further along, supervised machine learning may be used to create a more accurate predictor. Performance of the tool will be measured by the accuracy of the estimate, i.e. the disparity between the “predicted” market performance and the “actual” market performance.

General Approach

The project’s prediction algorithms will build in complexity. First, we will use basic characteristics of graphs (slope, common occurring patterns, etc.) to examine the current state of the market. The error on this phase is expected to be very high. Then, chart patterns (“technical analysis”) will be used to make more accurate estimates. These results should be much more accurate than the previous ones, and we can compare the different patterns used to find the “best” match for certain market states. The final phase, if reached, will likely utilize supervised machine learning to predict future performance by analyzing past performance during certain market states.

A “market state” is defined as a market’s performance potential at a certain point in time. For example, the peak of a bull run would have a different market state than the middle of a bear run. The nearby past data of that point typically determines the market state. The definition of “nearby” also depends on the type of analysis we are doing; some will require us to look years into the past, while others may only require a few days’ or weeks’ worth of performance to make the prediction.

We will be analyzing mostly index performances, since individual stocks, commodities, funds, etc. typically have high volatility. Since markets change slowly over time, we will be benchmarking our predictor over past market performances. The time frames to be used will mostly be within the range of 1995-2015, as there will be many market indicators to work with:



Phase 1

The first phase of the project will be to use basic graph properties to make the prediction. The following are all plausible methods:

1. Using the slope of a rise to predict a crash or the slope of a crash to predict a reversal
2. Using the frequency of peaks and troughs to estimate when the next one will be
3. Using the average increase each year to estimate average growth over the next

These are just a few of the ways we can use basic graph analysis to estimate market performance. Each of these methods has clear limitations, but this phase will server mostly to accustom us with using the pricing data.

Phase 2

The next phase will focus on using technical analysis to make more accurate predictions. A list of these technical analyses can be found under #3 of the Useful / Interesting Resources section. We will choose a few of these patterns that we think would fit well with the data and write algorithms to model them. There are a few python libraries that could also be used to implement some of these features.

The purpose behind this phase is to gain a clearer understanding of the markets work. Since human psychology is a large factor in market movements, basic graph indicators are essentially useless. Determining patterns in financial charts seems like a waste of time, but there are some reoccurrences that make it seem at least somewhat predictable. These patterns are not proven under any academic theory, but are sometimes accepted as natural due to their presence in empirical data. Also, they are truly the best estimate we have of the market short of machine learning.

Phase 3

This phase’s existence and production quality will depend on the timing left on the project and my confidence in being able to teach (or even implement) machine learning. We will be using supervised machine learning, meaning that we utilize an algorithm to analyze past data in order to predict future data. There are many algorithms we could implement and fine-tune as well.

Another thing to consider is how we allow the algorithm to grow. We could have it learn over a timeframe, test its performance, then rewrite it; alternatively, we could have it grow evolutionarily. This means that it forms the prediction by analyzing the timeframe, tests itself, and then attempts to rewrite itself. Both of these are valid forms to consider.

I myself have not used machine learning intensively, but it seems like a fun and useful introduction to the field. I am taking an online course on it now, so as we approach this phase I’ll be able to assess the problem more carefully.

Workflow

Our basic approach will be as follows:

1. Gather historical price data using fin2py package (seems to be easiest)
2. Understand and learn to use the data with NumPy and Pandas
3. Create helper functions to gather and analyze basic data, i.e. means, medians, slopes, timeframe data, etc. as well as calculate the accuracy of the algorithm
4. Implement analysis algorithms
5. Assess accuracy of algorithms by running functions and viewing through the web application

The web application I make will be a basic graph overlay that will allow you to pick and choose which algorithms you want to compare as well as if you want to include the actual performance data. This will hopefully help us find technical indicators and problems with the algorithm faster.

Supplemental Tools

The following Python libraries may be useful for the project:

1. NumPy – Scientific Computation - <http://www.numpy.org/>
2. Pandas – Scientific Computation - <http://pandas.pydata.org/>
3. yahoo-finance – Financial Data Source - <https://github.com/lukaszbanasiak/yahoo-finance>
4. TA-Lib – Technical Analysis - <http://ta-lib.org/> (general)

<https://github.com/mrjbq7/ta-lib> (python package)

1. Scikit-Learn – Machine Learning - <http://scikit-learn.org/stable/>
2. PyBrain – Machine Learning - [http://pybrain.org/](http://pybrain.org/pages/features)

Also, I will be working on a web demo that can be used to overlay the predicted and actual graphs to directly compare them. I will directly embed that in our code.

Useful / Interesting Resources

1. <http://www.forbes.com/sites/davidleinweber/2013/04/26/big-data-gets-bigger-now-google-trends-can-predict-the-market/#4acec991613a>
2. <http://www.investopedia.com/articles/07/mean_reversion_martingale.asp>
3. <http://www.investopedia.com/university/technical/techanalysis8.asp>
4. <https://github.com/twiecki/financial-analysis-python-tutorial>