Ruben Cubo
September 1, 2017

Learning the Stock Market in Depth

Background

The Stock Market has been one of the main drivers for creating and multiplying wealth historically, together with other investments such as real estate and entrepreneurship. This is reflected in historical market performances: investing in the S&P 500 has returned 7% on average after inflation, which is enough to double real value of investments in roughly 10 years according to the rule of 72.

However, there have been people that are capable of beating the market, i.e. indices such as the S&P 500, e.g. Warren Buffett. Technical and fundamental approaches have been used, although this work will be focused on the former. A myriad of indicators and methods exist, although in this work the focus will be on deep learning, and in particular, using recurrent neural networks (rNN).

Problem Statement

The main aim of this work is to estimate how a stock value will fluctuate with time by using rNNs. If one can do so and compare it to market evaluation by using e.g. S&P 500 or similar, one can give orders to buy or sell accordingly, optimizing returns which will hopefully beat the market, although portfolio optimization is well beyond the scope of this work.

The main issue when trying to predict market prices is high volatility: prices can vary by a large amount during a day and in between days for unknown or unrelated reasons. However, the author believes that these volatilities can be predicted in a model to some extent, although they can be very complex.

Datasets & Inputs

In order to do the analysis, stock market data is needed. For this work, Quandl's freely available stock data will be used, that will include historical data of open, high, low, close and trading volumes for each day. Both nominal and adjusted will be taken into consideration, with the latter being a more accurate view of the real value of the stock, since some factors such as dividends, splits and new offerings are taken into consideration.

Other data that can be interesting is different market indices. These indices can reflect a large part of the market (e.g. S&P 500), or be specialized in different areas. It is expected that a significant correlation will be observed between an individual stock value and the indices.

Solution Statement

The methods developed in this work should take one stock's historical data and give predictions of its value over some time horizon, e.g. 1 day, 7 days and 14 days. In order to test the accuracy of the algorithm, historical data at a later date than the training data will be used.

It should be noted that different fees and bonuses will not be considered, such as transaction fees, maintenance fees, taxes and dividends.

Benchmarking

Two benchmarks will be considered for this project:

- Market average: In particular the S&P 500. This is done in order to assess performance with respect to an index fund buy & hold strategy.
- A linear model with a Kalman filter, inspired by Martinelli & Rhoads work (see references)

Evaluation Metrics

Several evaluation metrics can be considered in these kinds of problems.

First, for the value estimation, the metric would be the Mean Square Error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

Where $\hat{Y}_i$ would be the estimated value of the stock price and Y_i would be the real value. N would be the total number of stocks and time instants considered.

Another measurement would be the simple return:

$$r_s = \frac{P(t+1) - P(t)}{P(t)}$$

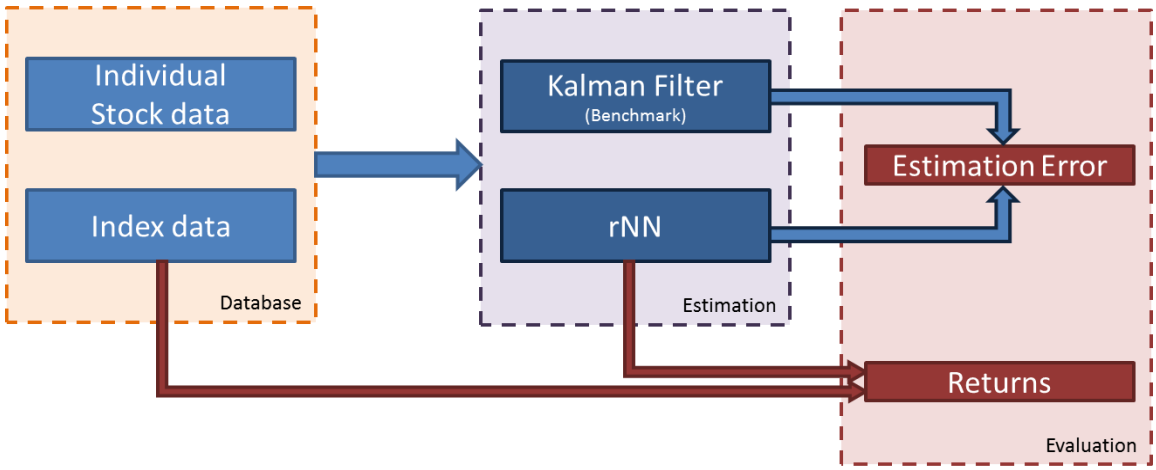
Where P(t) and P(t+1) would be the stock prices or moving averages at time t and t+1. Time here depends on the sampling: it can be one day, one week or even one year. One can evaluate the difference between this simple return and the market average, and see whether we can beat the market or not in a consistent way.

Project Design

The work is divided into three sections:

- Data acquisition and preprocessing: Stock data will be taken from Quandl open databases using the API they have for Python. It is unclear how many companies and indices will be considered. Data will be divided as usual, in training and testing sets. It should be noted that since the data are given in a time series, the only random component to be chosen is the split date.
- Estimation: Both the Kalman Filter and rNN estimation algorithms will be implemented.
- Evaluation: It will be divided in two parts:
    - Estimation error in the test data, by using MSE and comparing it to the benchmark.

- Returns evaluation: We want to beat the market, so hopefully our algorithms will outperform the usual market indices.



A flowchart of the project work is illustrated below for clarity

References

R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Journal of Basic Engineering, vol. 82, no. 35, 1960

R. Martinelli, and N. Rhoads, "Predicting Market Data Using The Kalman Filter", Stocks & Commodities, vol. 28, no. 1, pp. 44-47, 2010

R. Martinelli, and N. Rhoads, "Predicting Market Data Using The Kalman Filter, Pt. 2", Stocks & Commodities, vol. 28, no. 2, pp. 46-51, 2010

https://www.quandl.com

S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997