Algorithmic Trading with Machine Learning and Deep Learning

Charlio Xu

September 16, 2017

1 Introduction

This project sets up an algorithmic trading platform which incorporates various kinds of machine learning and deep learning methods to learn and predict market movement directions in order to generate extra cumulative returns over the benchmark returns.

2 Domain Background

a. automatic trading of equities with predefined algorithms, order placement is determined by algorithms

b. prediction of market movement directions vs price prediction, then place long or short order

c. classification problem with labels of two categories

d. machine learning and deep learning algorithms can be used

3 Problem Statement

given a history of past prices of a stock or other equities, how can we predict its movement directions (upwards or downwards) in the future

4 Datasets and Inputs

a. historical daily closed prices of stocks from Yahoo, or high-frequency tick data

b. input will be time series price data of a stock: time-indexed daily closed prices (float)

5 Solution Statement

The problem is a classification problem, so I will use supervised learning of classification like logistic regression, SVM, or dicision tree, and deep learning methods like simple neural network or convolutional neural network to tackle the problem.

6 Benchmark Model

I will use classical simple trading strategies as benchmark models, which include simple moving average, momentum and mean reversion methods.

Moreover, I will also compare the machine learning and deep learning methods.

7 Evaluation Metrics

On the machine learning level, I will use accuracy for our classification problem.

On the algorithmic trading level, I will use the cumulative returns including transaction costs as the evluation metric and compare it with the realized return in a plot.

8 Project Design

For the core algorithms, I will use classification algorithms in scikit-learn as well as construct neural networks in TensorFlow or Keras.

For the learning process, I will first use vectorized backtesting in numpy to have a first look at the performance of an algorithm, and then use event-based backtesting with object-oriented programming paradign to enhance the learning process.

Data will be downloaded from Yahoo! Finance and preprocessed with Pandas. Since the data is time series, spliting it into training, validating and testing sets is easy by simply cutting the time line into three pieces.

Evaluation will be provided with an accuracy analysis, but more trading-oriented performance will be visualized in a plot to compare the cumulative returns between given by the algorithm and the realized historic return.