Project Milestone 2

Stock prediction by related data weights

Figuring out the weights of each related stock to make a prediction model

Name: Sun Wu Choi

Due: 2020/11/24

**Report**

**[1] Milestone 2 objective**

Due on 11/24: Provide a relationship map of Tesla and tweak the parameters to increase the relationship prediction rate. Think outside of the box to fundamentally solve the errors of this problem, especially think this relationship map like figuring out the connectivity of a brain. Optimal milestone is to have 30% or higher relationship prediction rate and figure out the relationship of companies that is related to Tesla, which can be LG Chemical. Figure out if Tesla and LG Chemical is a one-way relation, or both way relation. By achieving this relationship map between Tesla and LG Chemical, it can be recursively applied to any dataset and show a feedback map like mapping connectivity between neurons in a brain.

**[2] Implemented or experimented algorithm**

Hebbian learning was implemented in the conceptual level

Any two stock that rise, fall, stayed the same in the same day was awarded with 1 point and -1 point if one rise and one fall. Also, I made a relationship score between -1 to 1 by dividing this score with the length of days it compared two stocks. I used the data from Kaggle, stocks data of all the companies of USA from 1963.

Using this relationship generation, first interesting thing I was able to find was “Agilent Technologies, Inc. Common Stock” which I used as a control stock for testing, had the largest relationship of 0.38 with FBGX. Surprisingly FBGX was a 2X ETN for Russell 1000 growth index, which is an ETF. Agilent Technologies, Inc. Common Stock uses IT to provide some data and it was indexed as health care by the Russell growth index. Russell 1000 ETF had their portfolio as 15% IT and many more. Also, Russell 1000 had this company in their portfolio.

The analysis on this result was quite exciting and disappointing. It was able to find which and which stock rise and fell together, figuring out some sort of relations between them. But the data also had ETN and ETF which I must sort it out now. This will be a long job, so I am planning to do it before the final report.

**[3] Empirical analysis and potential area of improvement**

As I mentioned above, empirically I realized the data set must be cleaned up before I run the Hebbian learning again. I have to sort out the ETN and ETF. But the relationship shows some promising results. If I can only compare the relationships between actual company stocks and create a list of top 10 companies that are related, then I might be able to get something going on.

But this is only using a simple Hebbian learning, nothing too deep. I was wondering if there is a method to extract some hidden relationships using machine learning and I am still thinking about this part.

Another improvement is that relationships are bounded in time, which means if Tesla and LG Chemical makes a deal for 3 years, then after 3 years, the relationship becomes useless. So, to avoid that, I am going to only compare the relationship with 3 different time length, 1 year, 5-year, 10 year. I am excited to see how different the relationship comes out.

**[4] Endgame**

The endgame result will be a program that will tell you how much a certain stock is related to another stock. More like if you look up Tesla, then it will show you top 10 companies that is closely related with Tesla company. The decision to buy or sell is up to the user, this program only tells you the top relationships.

**Reflect**

**[1] Did I achieve the milestone**

Yes, I was overthinking this project but after getting the historical data of the stock market and actually playing around with it, I realized it was better to keep it simple yet work more on the theoretical reasoning of why this method works and what data it will derive from the dataset.

**[2] Challenge in the milestone**

the dataset is too broad and has some weird stock data, so I have to clean them up for the final report. This will take some time, but it will be worth it since it will make the algorithm to only compare with the actual companies, not the ETF or ETN.

**[3] Extra things done**

The data source I mentioned was using Rapid API, but I am only using Kaggle dataset for this project.

**Replan**

**[1] Upcoming plan**

To create a relationship table for each company to store the relationship data. The data will be sorted so that the highest relationship comes top and lowest on the bottom.

Also, I will think about an additional method that can find a hidden relationship using machine learning or DNN.

**[2] Immediate step**

Immediate step will be cleaning up the data so that it only contains stocks, no ETF or ETN.

Also, to tinker the compare time length to 1 year, 5 years, 10 years.

**[3] Milestone update**

Update will be that I will not only use Tesla to figure out the relationship, but I will use all the stocks and run the algorithm to create a relationship map like a neural network.