CMPSC 383
Multi-Agent and Robotic Systems
Spring 2017

Name: Colton J. McCurdy

Pledge:

1 Overview of Progress

Since the start of this final project, I have made some major transformations to what I plan to do. Originally, I wanted to simulate a stock market with numerous nodes who would communicate with eachother and buy and sell artificial stock. Due to various constraints, this would not be feasible in the limited amount of time that we have to complete this assignment and time that I am able to dedicate to this project. My next idea and one that I started implementing late last week was a stock market bot that would follow the price of a single stock and determine when to buy and sell shares of that stock using a well-known algorithm using simple moving averages. The details of the algorithm and a more complex algorithm which uses exponential moving averages will be discusses in later sections of this document.

Due to issues connecting to and obtaining stock information from the popular Yahoo Finance website, creating a bot for determining when to buy and sell stock would be difficult without this information. While I did try other websites such as the Nasdaq, I did not have luck obtaining the necessary data. After revisiting the problem many days later, I was finally able to find a website that would provide me with the information that I would need to make this project possible.

After thinking more deeply about creating a stock market bot, I felt like I would then need to choose a single company to evaluate the effectiveness of the bot. I did not like this and for this reason, I decided to move to an environment where I was more familiar, bitcoins. While I have had experience creating a bot that would simulate the buying and selling of bitcoins, I had a minor role in the project and would like to optimize the approach using a language that is more fit. Additionally, I would like to publically release under my GitHub account to display my knowledge of the language.

At this point in time, I have a framework for the tool. I have functions that connect to the CoinDesk Bitcoin Price Index API (`http://www.coindesk.com/api/`), to get the current and historical bitcoin prices for an arbitrary date in time. Being able to get the historical data for a given date range is important for the moving averages that will be used to determine when the bot should buy and sell bitcoins. While that is my current progress, I have a written process of how I will go about completing the project, (e.g., how the algorithms will be impletemented, how the functions will be structured, etc.).

2 Tools

As mentioned earlier, while I have had experience creating a bot that would analyze the price index of bitcoins, it was not created in a way that I would have liked it to be. Also, I was not able to contribute as much as I would have liked due to my lack of experience. This project provides me with an opportunity to create this tool using the programming language of my choice, namely the Go programming language.

2.1 Google's Go Programming Language

I have been particularly interested in the Go programming language — often just referred to as Go — recently due to its increasing popularity in industry. Go's syntax is often compared to a combination of Python and C, it often requires minimal code versus a more verbose language such as Java. Not only have I found this enjoyable, Go is often recognized for its ability to comlete tasks concurrently. Go was designed with concurrency at the top of the list of objectives of language features. In this project I will leverage the power of Go's hybrid, lightweight threads, called goroutines.

2.2 CoinDesk

As mentioned earlier, one of the issues that I faced when trying to obtain stock information was finding an API to connect to to obtain a given stock price index. While searching for an API to do this for the stock market, I came across CoinDesk which did exactly what I wanted except was focused on the bitcoin price index. After visiting CoinDesk's API website and seeing the clear documentation and useful examples, I was sure that this was what I wanted to work with. Not only does CoinDesk have routes for obtaining the current bitcoin price index, they have routes to obtain historical data for any given date range, which is exactly what I would need to perform the moving average calculations. For these reasons, I changed the focus of my project from the stock market to the bitcoin market.

3 Algorithms

In this bitcoin bot, the most popular algorithm for determining when it is appropriate to buy and sell bitcoins will be implemented. Both of the following algorithms use historical data to determine whether the price index is on a path to increase or decrease, indicating that bitcoins should be bought or sold, respectively.

3.1 Simple Moving Averages (SMA)

The simple moving average is often the baseline technique when determining when to buy and sell stocks. This technique has an infinite number of configurations. The most common configuration considers the price index from 50 days ago to that of 200 days ago. What this does is help smooth

out votality The values of 50 and 200 are chosen because the curve being evaluated is smooth. Shorter date ranges while more close to the actual price index are much more volatile.

To calculate the simple moving average, you can use the following equation, where n is the number of days being evaluated. When considering two configurations of the moving average, when they cross is when the bot would buy and sell.

$$SMA_n = \frac{i_1 + i_2 + \cdots + i_n}{n} \tag{1}$$