**Evolution of Bitcoin Miners**

## Overview

The aim of this project is to understand how bitcoin mining applications evolves over time. Our first task will be to understand what the most widely-used miners are. Then, we would like to understand what features in the code make the miners unique. Finally, we will test whether any particular types of changes to the algorithm occur in response to major bitcoin price fluctuations.

## Background

The official bitcoin wiki lists 14 miners:

<https://en.bitcoin.it/wiki/Mining_software>

Most of these are hosted on github. Project download information on GitHub can serve as a proxy for miner popularity. This would be the first step. Even if we aren’t able to do that, we can assume the miners listed on the Bitcoin wiki are the most prevalent ones (note that making this actual determination using raw data is another project). We can still track the commits throughout time and search for trends in the changes.

Once we do that, we can mine the commits for text describing the changes. Coupled with an analysis of the major code design characteristics of the algorithms, we will cluster the changes according to certain categories.

After this, we can regress changes in a particular category of code modification on price fluctuations and additional variables (affiliation with bitcoin pool, language miner is programmed in, OS, perhaps other variables).

## Project Steps

1. Obtain information about project download frequency on Github as a proxy for miner popularity. Since there are so many bitcoin miners, it is prudent to separate the ones actually used for mining from hobby projects. I am not sure whether this information is part of Github’s API. For the miners not hosted on Github, attempt to find analogous information on the miner’s site.
2. Use results from step 1 to narrow the list of miners we will focus on. If step 1 is unsuccessful, focus on the miners listed on the Bitcoin [wiki page](https://en.bitcoin.it/wiki/Mining_software). Track commits throughout time for the past year or two for the most popular bitcoin miners we are focusing on. This should be relatively straightforward given Github’s robust API.
3. Perform text-mining on the json files in order to cluster the category of changes to the miners. This is likely to be a challenging step because it will entail understanding how the miners are actually coded.
4. If step 3 is unsuccessful, we can simply track the commits per time period for given miners.
5. Obtain bitcoin price data. Analyze price fluctuations and isolate events where the price changed by more than 20% in a 2-3 day period. This will be considered extreme price fluctuation events.
6. Create a linear regression with number of changes in a particular category within a 2-week period as the response variable. The main explanatory variable is price fluctuation itself (binary variable – either a price fluctuation occurred or it didn’t). Other variables are whether the particular miner is affiliated with a certain bitcoin pool, the language the miner is programmed in and other variables TBD). If step 3 is unsuccessful, our response variable will simply be the number of commits as a function of price fluctuations. This will obviously be less insightful.

## 

## Timeline

|  |  |  |
| --- | --- | --- |
| **Date** | **Weekly deadlines** | **Ongoing Challenge Task Delegated to 1 Team Member** |
| 10/19 | Obtain Github project download frequency. Narrow list of miners to focus on. Extract information about commits throughout time for the most popular bitcoin miners. | Works on understanding the code in mining algorithms |
| 10/26 | Finish text-mining of commits |
| 11/9 | Cluster commit content as per result of text mining |
| 11/16 | Build regression analysis with price fluctuations as one of the explnatory variables. |
| 11/23 | Complete analysis |
| 11/30 | Prepare report and final presentation |