SOEN498 - Project Report

Anne-Laure Ehresmann and Maximilien Malderle

April 2019

1 Abstract

Bitcoin is the most popular cryptocurrency at this time, and has accrued a wealth of data on its users and their transactions. However, the sheer size and complexity of the interactions make it troublesome to identify clear patterns and shifts in user behaviour. We seek to classify this data to isolate impactful transactions on the network to help identify shifts, to map out user-interaction, notably in an attempt to characterise the flow of transactions and the transformation of spending patterns. The goal is to produce a list of interesting addresses, which can be used by the cryptocurrency community to provide more high level information on the specific platform, promoting more transparent overviews, encouraging a more trustworthy evolution.

2 Introduction

As of 2019, the number of daily bitcoin transactions hovers just below three hundred thousand. Bitcoin still holds the primary spot as a viable cryptocurrency, with both its price and popularity far above all others, despite thousands of competing cryptocurrencies emerging in the last decade. It is often referred to as the "cryptocurrency gold-standard", essentially underpinning the value and confidence in the cryptocurrency community. One of the interesting features (and necessity for its function) of Bitcoin is its transparency: Coins are not digitally stored on personal computers, every single transaction must be recorded by a ledger, which keeps track of the entire bitcoin network. There already exists a number of analytical works done taking advantage of this public ledger, notably with attempts to predict market-price[5], de-anonymising users[6], or optimizing decision-making for inter-cryptocurrency trading[7]. Our aim is to use multiple methods on different time slices of the Bitcoin network to determine if clusters of user interactions can be found, while inspecting the quality of these methods on such large data sets. Some interesting questions guiding our project are: Are there clusters of users constantly interacting with each other? What is the size of the transactions they handle? How is money dispersed and accumulated by different users? Is it possible to identify big shifts in transactional habits? All this may result in a list of interesting addresses. In the future others could use these addresses along side the already available public address-books to correlate it with real-world events, or to identify the predominant industries on the network. This could offer more information on specific block chain networks, adding to the credibility of a certain currency, or by flushing out the negative ones, ultimately improving confidence in cryptocurrencies.

3 Material and Methods

3.1 Material

For this project, we used two data sets: The *transaction data set*, containing information pertaining to each transaction, their sum, their receiving address and previous transaction, as well as general metadata; and the *user-graph*, describing the interactions between users (each with an assigned weight dependent on how "related" they are, for instance, number of transactions, total sum of bitcoins traded, etc.).

As extracting the data from the block chain is a initially time-consuming process, we instead relied on the transaction data set provided by [8], which contains all transactions up to approximately the 9th of February 2019. For the second data set, we intended to use tools provided by Reid and Harrigan and described in [1] to extract the user-graph directly from the block chain. Unfortunately, the tools were out of date, and did not properly handle the new method of storing the block chain that more recent bitcoin clients use. (Attempts to use older clients led to additional problems, and newer forks of the tools did not implement some features which we critically needed). We found that not only does there exist few tools to extract the user-graph, but additionally, the majority of these which are publicly available were built on top of Harrigan's tools, rendering them useless (unless an update was provided elsewhere, which we have not identified here). We ended up relying on an algorithm from Di Francesco Maesa et al. in [2], which we implemented [9] with the transaction data set as input. We note that the algorithm makes the following assumptions:

- All input addresses in the same transaction come from the same user.
- If an input address is used in two transactions, the union of both input address sets of each transaction belong to the same user.

This is a reasonable assumption to make, one also made by Reid and Harrigan in [1], as "multiple input addresses in a transaction are a strong indication that the entity that issued the transaction actually owns or shares the private keys corresponding to all the addresses" [11]. Others have used different or additional rules, such as in [11], which makes the additional assumption that all output addresses belong to the same user, or [12], which makes additional assumptions based on usage categories of the transaction. For edge weights, we merely counted the number of times two users interacted with each other, and labelled their edge accordingly.

The transaction input and output data sets from MIT are around 50 GB total. This is a quite unmanageable dataset, along with the useful transaction information being divided among 4 files. For this reason, for community detection, we instead relied on only the first 100 million transactions (about 3.7GB of data total), which proved to be more manageable for our programs. To be able to do more insightful analysis, useful transaction information needs to be synthesized in a lightweight easily manipulated data structure. To do this we created a DataFrame with the format (txID, blockId, InAddr, OutAddr, Date, Sum USD), condensing all the useful information found across the distributed transaction files. The ”txID” is used to identify the transaction, ”blockID” is used to connect the transaction to the Blockchain, ”InAddr” is the biggest input address, ”OutAddr” is a comma separated string of the output addresses, ”Date” represents the transaction date and the ”USD$_SUM$”$representstheUSdollarvalueofthetransaction.Allthetransactionswillbestor$

These files will contain a lot of uninteresting transactions, to further filter the information, we will cluster each year using K-means algorithms. Each cluster will represent a different class of transactions based on the US dollar value. This will reveal what percentage of the network transactions belong to a certain class of spending. This information can show the evolution of the network from year to year, but more importantly, can help identify which classes are the most impactful to value, leading to an interesting place to look for influential users.

3.2 Methods

We relied on the FluidC algorithm [10] for community detection. Informally, given a parameter k and a parameter i, FluidC randomly picks k users, assigns them a max *density*, and for each iteration until the ith iteration terminates, or until communities stop changing, FluidC has the density of each community "flow" to the neighbouring vertices, (with the density of each vertex v in a community c equal to $\frac{1}{v \in c}$. If two flows meet, they will both attempt to push at each other (with the one with higher density winning over) until the communities settle.

Condensing the 4 files into one transaction required the joining of the inputs and outputs files with the transaction overview file and the block information file. The input file is 27 GB, the output file is 23 GB, the transaction overview file is 7 GB and the block file is 500 MB. To achieve maximum efficiency on the join we converted the files to pyspark DataFrames. We leveraged sparks ability to store distributed data structures using HDFS to store the results in AWS S3 buckets. The output is stored as a data parquet, column-orientated storage and compression format, compatible with the Hadoop framework. This allows for quicker access times for future manipulation.

The K-mean algorithm was performed on each year with a $k = 5$ and a maximum iteration of 33. The k was selected to represent the assumption of there being 5 classes of economic actors. The ”Big Fish”, representing the institutional class, the ”Sharks”, representing the investment class, the ”Tunas”, representing the the industrial class, the ”Salmons”, representing the business class, the ”Minoes”, representing the individual class. This classification is designed to build an ”Ecosystem” representing the economic environment of the Bitcoin network. Like any ”Ecosystem” each member has a profound impact on its environment. This information can be analysed to better understand the makeup of the Bitcoin network at a given time, suggesting the health of the ”Ecosystem” when contrasted with the value of

Bitcoin. An interesting class to inspect for interesting addresses using the FluidC algorithm, is the "Big Fish".

4 Results

4.1 FluidC

Clearly, FluidC cannot be applied on disconnected graphs, so we detected connected components on our user graphs, and ran an instance of FluidC on each individually. On smaller (<10 000 transactions) datasets, we find that the graph is often disconnected, however with 10 million transactions the user graph is fully connected. We ran the algorithm for 10 communities, and obtained the following (with "interaction" referring to at least one transaction with a unique user of the same community): the highest number of other members of that community. For these, we identified a number of communities

Community	Num. Users	Interaction Mean	Interaction Std. Dev.	Top Num. of Interactions
0	51731	2.091	17.42	1026
1	48367	1.848	9.85	1881
2	66852	2.41	18.77	3684
3	65866	11.48	28.64	3246
4	246820	2.52	26.94	9141
5	148764	2.11	6.05	1512
6	64050	2.06	6.76	1087
7	14053	2.89	82.48	9777
8	70055	2.06	12.97	2978
9	295376	2.32	8.81	2213

Within these communities, we extracted the most "prolific" users, dependent on their number of interactions with other members of the community, and looked up their associated addresses on bitcoin address lookup search engines. We were able to identify some interesting users and communities from these results:

(Note that the user id is entirely dependent on our instance of the algorithm, it is necessary to extract the user's associated addresses using one of the files generated by our user-graph generator.)

For instance, user 24422, associated with 31 addresses, had 2 addresses appear on the "Bitcoin top 100 'Rich List' 20th March 2011"[16].

User 503532 is identified to be Allinvain, a user who was the victim of a theft in june 2011 of more than half a million dollars' worth of bitcoins[15]. Allinvain posted a list of his addresses on [17], a number of which we identified with user 5035232.

User 6161 is possibly Gavin Andresen according to a Reddit post[18], as it was associated with the address 15VjRaDX9zpbA8LVnbrCAFzrVzN7ixHNsC .

Community 2, notably with its most prolific users 1938400, 1957420, 1943150, 1927430, 700203, and 523175, all had addresses found to have payed or received bitcoin to/from addresses linked to `satoshidice.com`, a bitcoin gambling service.

Equally with Community 7, its top user had many of its addresses receiving bitcoins from a dice address.

User 165589, in Community 3, had his top address (1PJnjo4n2Rt5jWTUrCRr4inK2XmFPXqFC7) be identified in old bitcoin email chains [14] as the main address of slushpool.com, the oldest bitcoin mining-pool (also on the list of richest addresses in [16]), which leads us to conclude that community 3 is a community of miners of that pool. While not confirmed, the second most "prolific" user's most used address is a vanity address which hints at this: 1Mining2pfZRiFxGBQ6Jw8YNheUkHhyMfn .

User 142482 had one of its addresses (159FTr7Gjs2Qbj4Q5q29cvmchhqymQA7of) involved in more than 16000 transactions, be identified as a potential spammer[13].

4.2 Data

The Dataset transformation and reorganization yielded 9 *snappy.parquet* files

Year	Storage Size
2010	2.2 MB
2011	63.5 MB
2012	266.5 MB
2013	681.5 MB
2014	1010 MB
2015	1890 MB
2016	3180 MB
2017	4110 MB
2018	389.1 MB
Total	11592.8 MB

This transformation has condensed 50 GB of information into a total of roughly 11.5 GB of data, which brings together information from 4 sources. The files produced have an average size of 1.15 GB, which is a lot more manageable. The number of transactions is increasing every year (2018 contains only the records until February), with files of 4 GB already being very cumbersome to run analysis on, maybe further subdivision i.e. quarter years would be beneficial in the future.

4.3 K-Means

The K-Means Algorithm was very demanding on our systems. Files of the range 2015-2017 required 32-48 hours, on a MacbookPro 2014 Quadcore i7 16 GB RAM, an AWS m4.xlarge Quadcore 16 GB RAM Instance and an AWS c5n.xlarge Quadcore 16 GB RAM Instance. The c5n.xlarge was the most effective solution. The restriction for writing to a parquet or file is 5 executors, due to write blocking errors, so additional cores would not be beneficial, the computing power of the core is more impactful.

As for the algorithms performance there is a sample of the 33 max iterations of the 2016 dataset

Iterations/K	1	2	3	4	5
0	26.81%	47.30%	-28.41%	98.99%	80.08%
1	6.97%	21.07%	38.73%	95.07%	46.87%
2	21.04%	-1.26%	39.88%	86.70%	38.30%
3	30.16%	12.45%	34.59%	53.08%	34.39%
4	32.26%	24.03%	28.97%	27.55%	33.29%
5	31.64%	25.95%	26.16%	18.94%	37.80%
6	29.59%	24.48%	22.18%	11.17%	26.94%
7	26.72%	22.64%	21.00%	3.80%	19.37%
8	22.68%	18.93%	19.45%	2.38%	14.70%
9	20.05%	17.07%	17.22%	1.63%	8.55%
10	16.07%	13.21%	14.80%	0.00%	5.70%
11	13.10%	10.68%	13.00%	0.00%	5.56%
12	11.52%	8.82%	14.07%	0.00%	8.40%
13	10.41%	7.72%	15.32%	0.00%	13.10%
14	10.33%	6.80%	19.75%	0.00%	27.93%
15	11.51%	6.69%	23.33%	0.00%	19.02%
16	10.50%	7.50%	18.73%	0.00%	13.51%
17	9.41%	7.20%	11.01%	0.00%	3.78%
18	7.97%	6.07%	7.40%	0.00%	2.26%
19	6.92%	5.26%	5.00%	0.00%	1.67%
20	5.88%	4.32%	3.98%	0.00%	2.15%
21	4.48%	3.18%	2.78%	0.00%	1.52%
22	3.40%	2.39%	1.66%	0.00%	0.36%
23	2.71%	1.80%	1.53%	0.00%	0.29%
24	2.31%	1.57%	0.97%	0.00%	0.07%
25	1.90%	1.26%	0.81%	0.00%	0.00%
26	1.68%	1.14%	0.51%	0.00%	0.00%
27	1.54%	0.97%	0.70%	0.00%	0.07%
28	1.30%	0.84%	0.49%	0.00%	0.07%
29	1.11%	0.67%	0.53%	0.00%	0.00%
30	0.88%	0.55%	0.38%	0.00%	0.07%

Here we can observe the percentage change in the 5 centroids across the maximum 33 iterations. It is interesting to observe that K = 4 stops changing in iteration 10, the reason being, there existing an extreme value in the dataset, one large enough to dominate an entire K. This is a positive tradeoff because this address is an outlier deserving special investigation.

It is useful to observe the average percentage change in milestone iterations.

Iterations	Avg Percentage Change
5	28.10%
10	9.96%
15	12.11%
20	3.27%
25	0.79%
30	0.38%

In this test set the first 15 iterations are vital moving the centroid by minimum 10% on average. By iteration 25 the changes seem to converge to roughly 1%, 8 extra iterations smooths the final K's, but this represents 25% of the tasks, which in some cases represents a 12h timeframe.

Market information generated for 2011:

2011	Minnoes	Salmon	Tuna	Shark	Big Fish
K Value	$123.38	$23,010.53	$78,130.37	$322,925.69	$1,312,409.91
Amount in Cluster	1836659	4567	640	54	142
% Transactions	99.7067%	0.2479%	0.0347%	0.0029%	0.0077%
Spent in Cluster	$226,638,249.02	$105,123,467.85	$50,003,437.12	$17,437,987.51	$186,362,206.62
% Sum	38.7042%	17.9525%	8.5393%	2.9780%	31.8260%

Market information generated for 2017:

2017	Minnoes	Salmon	Tuna	Shark	Big Fish
K Value	$629.1020998 7	$204,513.17	$770,110.31	$3,129,964.14 7	$8,718,726.09
Amount in Cluster	32639271	107315	15174	1156	372
% Transactions	99.6215%	0.3275%	0.0463%	0.0035%	0.0011%
Spent in Cluster	$198,575.996 M	$223,035.665 M	$118,693.088 M	$36,543.968 M	$31,580.212 M
% Sum	32.6375%	36.6576%	19.5081%	6.0063%	5.1905%

Percentage change in the Market from 2011-2017:

Percentage Change	Minnoes	Salmon	Tuna	Shark	Big Fish
K Value	409.8899%	788.7808%	885.6734%	869.2521%	564.3295%
Amount in Cluster	1777.1002%	2349.7920%	2370.9375%	2140.7407%	261.9718%
% Transactions	-0.0852%	0.0796%	0.0116%	0.0006%	-0.0066%
Spent in Cluster	87618.0422%	212165.4372%	237369.8590%	209565.2892%	16945.6100%
% Sum	-6.0667%	18.7052%	10.9688%	3.0283%	-26.6356%

Percentage change in the Total Market from 2011-2017:

Change	2011	2017	% Change
Total Transactions	1842062	32763288	1778.6203%
Total Sum	$585,565,348.11	$608,428,931,495.50	103904.5315%
Price	$4.72	$13,850.40	293440.6780%

The most interesting finding is that the amount of transactions in the different K clusters has not changed more than 1% even with an 1770 times increase in transactions. It should also be taken into account that the distribution of the sum transactions shifted towards a more "Minnoe" and "Salmon" oriented environment, moving away from a 30% controlled "Big Fish" "Ecosystem", this along with the cheer amount of transactions should explain the 293 000 times increase in market price.

5 Discussion

One major flaw of FluidC is that it ignores edge weights. As such, members who constantly interact with each other are given the same weight as members who only participated in a single transaction together. As such, it is biased towards users which have interactions with a large number of unique users, as opposed to those who may have many transactions, but to the same users repeatedly. Additionally, we have personally not assigned any additional weight to the edges depending on the sum of bitcoins involved in a transaction. This means that, when pulling the most prolific users from each community, we underestimate the users which may have done a small number of very large transactions, as opposed to those who only perform small but numerous transactions.

Overall, the most troublesome part of our project was the size of the data set. Indeed, even the size of the results are difficult to manage: the most prolific users are commonly associated with hundreds of different addresses, many of which are involved in only a few (<5) transactions, making it difficult to verify whether these are truly a ”community” due to the lack of information on these small transactions.

It would be interesting to run this research on our synthesized datasets: Once identifying a user, we could look for him/her again and notice the shifts in the community he is associated with. It would equally be interesting to run FluidC multiple times for same or differing values of k, to see how confident it is in its communities. Given that our programs ran for massive amounts of time, we were unable to do this enough times to report any changes of value.

References

[1] Fergal Reid, Martin Harrigan, *An Analysis of Anonymity in the Bitcoin System*, arXiv:1107.4524

[2] D. Di Francesco Maesa, A. Marino, L. Ricci *Uncovering the Bitcoin blockchain: an analysis of the full users graph*, 2016 IEEE International Conference on Data Science and Advanced Analytics, DOI 10.1109/DSAA.2016.52

[3] Andrei Novikov, *PyClustering*, DOI:10.5281/zenodo.1491324

[4] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, https://bitcoin.org/bitcoin.pdf

[5] https://github.com/philipperemy/deep-learning-bitcoin

[6] *Breaking Bad: De-Anonymising Entity Types on the Bitcoin block chain Using Supervised Machine Learning*, DOI:10.24251/hicss.2018.443

[7] Bitcoinist, *Do A.I. and Cryptocurrency Work Well Together?*, `https://bitcoinist.com/ai-and-cryptocurrency-work-well-together/`

[8] Daniel Kondor, *Bitcoin Network Dataset*, `https://senseable2015-6.mit.edu/bitcoin/`

[9] Generate Bitcoin User Graph, `https://github.com/alehresmann/generate-bitcoin-user-graph`

[10] Ferran Parés, Dario Garcia-Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, Toyotaro Suzumura, *Fluid Communities: A Competitive, Scalable and Diverse Community Detection Algorithm*, arXiv:1703.09307

[11] Chen Zhao, Yong Guan. A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS.Gilbert Peterson; Sujeet Shenoi. 11th IFIP International Conference on Digital Forensics (DF), Jan2015, Orlando, FL, United States. IFIP Advances in Information and Communication Technology,AICT-462, pp.79-95, 2015, Advances in Digital Forensics XI. ¡10.1007/978-3-319-24123-4_5¿. ¡hal-01449078¿

[12] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, Stefan Savage,*A Fistful of Bitcoins: Characterizing Payments Among Men with No Names*

[13] Andrea Pinna, Roberto Tonelli, Matteo Orrú, Michele Marchesi *A Petri Nets Model for Blockchain Analysis*, arXiv:1709.07790

[14] “1PJnjo... belongs to Slush’s pool.” `https://bitcointalk.org/index.php?topic=127245.25;imode`

[15] Adrianne Jeffries, How to steal Bitcoin in three easy steps, December 19, 2013 01:10 pm, https://www.theverge.com/2013/12/19/5183356/how-to-steal-bitcoin-in-three-easy-steps

[16] Bitcoin Top 100 rich list 20th march 2011. https://bitcoinreport.blogspot.com/2011/03/bitcoin-top-100-rich-list-20th-march_20.html

[17] Allinvain, *I just got hacked - any help is welcome! (25,000 BTC stolen)*, June 13, 2011, 08:47:05 PM https://bitcointalk.org/index.php?topic=16457.msg214423#msg214423

[18] vongesell, *Finding Satoshi, Gavin's net worth ($200k to $300k)*, "Perhaps this was the wrong route so I went back to the google results for 15VjR... before eventually finding references to this as the Faucet address. Gavin was known to make the Faucet. So that makes it his." https://www.reddit.com/r/Bitcoin/comments/16tmw5/finding_satoshi_gavins_net_worth_200k_to_300k/