User Categorization and Community Detection in Bitcoin Network

Dongyuan Mao and Yifei Zhang

***Abstract*— This project works on Bitcoin user categorization and community detection. A new user network generation method is presented to better contract user addresses. Users playing different roles are recognized with K-Means. Three different methods, K-Means, Node2Vector and Fiedler Vector Method are applied to analyze the network community structure. A major radiant community plus a minor community structure is detected.**

I. Introduction

Bitcoin is a worldwide cryptocurrency and digital payment system. It is the first decentralized digital currency, as the system works without a central repository or single administrator. It offers several compelling advantages to traditional country currency in that it eliminates the bank/clearinghouse to lower fees, it is usable across the globe, accounts cannot be frozen, and there are no arbitrary legal limits. Given the public nature of the Blockchain, in which every transaction is recorded, it provides an excellent source of data from which to infer and analyze flows of funds and connections between users.

This project works on Bitcoin user categorization and community detection by examining transaction network features and applying clustering algorithm. In Part III, we investigated into the original dataset and analyzed features of Bitcoin transaction network. An improved version of baseline user network generation is implemented to contract user addresses better in Part IV. K-Means is used for user categorization based on the results on user address contraction. Detailed analysis of the clusters derived by K-Means are given in Part V. Three different methods, K-Means, Node2Vector and Fiedler Vector Method are used for community structure detection and currency flow analysis. Methods and results are presented in Part VI. Part VII concludes the whole project.

II. Related Works

A. Ron D et al., Analysis of Bitcoin Network

This article first contract addresses to users. The reasoning is that all sending addresses in a single transaction belongs to the same owner. The main problem of this approach is that the addresses a user owned may not be in a single connected component.

Then the article dwells on whether most Bitcoins are stored or spent. It sums up all the Bitcoins that belong to addresses that only receive and never send. Then it removes all new addresses and old addresses to avoid biases. The conclusion is that most users store Bitcoins.

This article combines addresses into users based on co-occurrence in transactions. But it only uses co-occurrence on the same side (input/output) of a transaction. It ignores co-occurrence on the opposite sides of a transaction. In fact many users will generate new addresses frequently and transfer Bitcoins from their old addresses to the new ones, to hide their identity. Also many users will use new addresses to receive change of a payment.

B. Koshy P et al., Analysis of Anonymity in Bitcoin

In this article the authors tried to analyze traffic patterns on the Bitcoin network to see if it was possible to create mappings from Bitcoin addresses to IPs. They created CoinSeer which established an outbound connection to every listening peer whose IP address was advertised on the Bitcoin network, to collect their transactions and IP addresses.

They first came up an assumption that the creator of a transaction was the first relayer. Then, the author paired the first relayers IP address with Bitcoin addresses. Then, they think of a transaction owned by IP i which contains Bitcoin address b as a vote for the pairing between b and i.

C. Palla et al. with Community Detection

This paper presents a clique based method to explore overlapping community structure in large scale network. The method is based on the observation that, a typical community consists of several complete subgraphs that tend to share many of their nodes. Our work attempts to separate the Bitcoin network into stocking market, where users only trade with exchange center to get interest, and commodity market, where users trade commodity and service with Bitcoin. Thus Bitcoin investors can be ignored by the clique-based method making it very useful in analyzing the commodity market structure.

III. Dataset

We will use the dataset provided by ELTE Bitcoin Project. There are two sets of data provided by the website. We are considering using the first set as it contains timestamps which will be very useful in our project. The first set of data contains all blockchain up to 2013.12.28.

A. Data Files

The following data files are used in our project:

- txin.txt
  list of all transaction inputs (sums sent by the users), including txID, addrID and value
- txout.txt
  list of all transaction outputs (sums received by the users), including txID, addrID and value

TABLE I: Send Addr in TXN

Send Addr Number	Count	Percentage
1	20503156	68.87%
2	5678228	19.07%
3	1149961	3.86%
4	1506313	5.06%
5	327134	1.10%

TABLE II: Receive Addr in TXN.

Receive Addr Number	Count	Percentage
1	1933255	6.43%
2	26428808	87.95%
3	1031183	3.43%
4	164881	0.55%
5	102395	0.34%

B. Basic Metrics

The dataset contains 277443 blockchains up to 2013.12.28. Datafiles contain 29771506 txinID and 30048911 txoutID. 277475 transactions do not have input, which corresponds to the Bitcoin mining transaction. This number is in consistency with the number of blockchains.

A total number of 24618958 addrID is involved in all the transactions. Each transaction has a number of input addrID and output addrID. The relationship is as follows:

We can see from the tables that, most TXNs have 1 or 2 input addrID. However, only 6% of TXNs have 1 output addrID. Majority of TXNs have 2 output addrID.

This can be explained by the "change" account features during Bitcoin TXNs. When the output of a transaction is used as the input of another transaction, it must be spent in its entirety. Sometimes the coin value of the output is higher than what the user wishes to pay. In this case, the client generates a new Bitcoin address, and sends the difference back to this address. This is known as change. Therefore most TXNs have 2 output addrID. This will have a huge impact in our following analysis of the Bitcoin TXN network.

IV. User Network Generation

A. Goal

The anonymity of Bitcoin network lies in that, addresses, instead of users, are involved in each transactions. To perform user categorization and community detection, user-user network is needed. We want to generate a user-user network out of the current addr-addr network by contracting addrs which belong to the same user.

B. Baseline

Addresses appearing on the same transaction input side should belong to the same user. By contracting these addresses into a single user, we can get a preliminary user-user network. 12137803 users are generated out of 24618958 addresses by contracting addresses appearing on the same TXN input side. User-degree and user-TXN distribution are in Figure 1 and Figure 2.

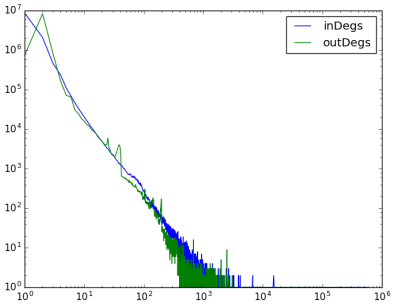

Fig. 1: User-Deg Distribution

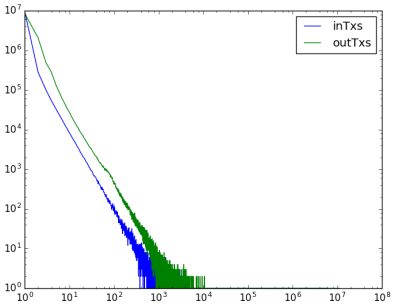

Fig. 2: User-TXN Distribution

We can see from the figures that user-degree distribution follows power-law. However, there is a peak at out degree 2(we will explain this later on). User-TXN also follows power-law distribution. Generally users are participating more in TXN-out side than in TXN in side. The reason for this is "change" account. Typically user on the TXN input side will also appear on the TXN output side.

A more detailed user distribution is in Table III. We can see from the table that more than 50% of users are actually "dead" users. They receive bitcoin from one user in one TXN and send the bitcoin to 2 users in another TXN. Among the 2 out degrees, one should be the change account for this user but didn't get recognized by the baseline algorithm. This gives the explanation why there is a peak at out degree equals 2 in the user-degree distribution graph.

This result gives us intuition on the composition of current user network. For user categorization, we are planning to merge or eliminate these dead users to get a better result.

Besides user distribution, we also investigated the composition of output side of TXNS. Results are in Table IV. We can see from the table that about 95.08% TXNs have less than 2 out userID. 54.66% have 1 change account and 1 receive account, which is the normal case. However, 34.47% of TXNs have 2 output userID. One of the output user in this case should be recognized as change account.

To sum up, the baseline version of user network follows the power-law distribution in user-degree and user-TXN participation. However, more than 50% of users recognized are "dead" users, which leads to the peak at outdeg equals

TABLE III: User Distribution wrt. Degree and TXN.

TXN in	TXN out	OutDeg	InDeg	UserCnt	Percentage
1	1	2	1	6453856	53.17%
0	1	0	1	1290888	10.64%
1	2	2	2	1264938	10.42%
1	2	3	2	443954	3.66%
1	1	1	1	414731	3.42%

TABLE IV: Receive Side Composition in TXN.

Change Account	Recv Account	Cnt	Percentage
0	1	1604830	5.34%
0	2	10359355	34.47%
1	0	183525	0.61%
1	1	16424005	54.66%

2 in Fig.1.

C. Improvement

The problem of baseline method is that it only utilized the input part of transactions. Many addresses of a single user may not be connected in the input contraction graph. As a result, the baseline method can not fully contract user addresses.

One proof of that can be derived from fig.3 that at the time 2013.12.28, there are 956185 wallets created on Blockchain.info – the largest cryptocurrency wallet service. According to our dataset, there are total 24617960 addresses. However, the baseline method have as much as 12137803 users, which is far more than reasonable. This means average number of addresses owned by one user is 2.0282, which is much lower than expected. According to Blockchain.info's wallet number, the average number of addresses is 25.746 – a little higher than expected because Blockchain.info is not the only Bitcoin wallet service in the world.

Another proof of that comes from fig.2. The Out-degree distribution has a bump at degree 2, which not only contradicts power law, but also overthrows the basic instinct that most transactions happen between two people. Instead it suggests that most transaction happens between one seller and two buyers. This wrong suggestion is caused by failure of recognition of change accounts. Most of the transactions should have one input user on the input side and one output user and a change account on the output side. The baseline method mistakenly count change account as a second user.

Because of the change mechanism of Bitcoin transaction, it is tenuous to say that all addresses in the output part of a transaction belong to a single user. Most likely, there is a change account in the output part. Based on this instinct, we propose a new contraction method to further contract

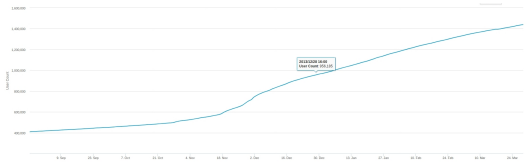

Fig. 3: The total number of Blockchain wallets created

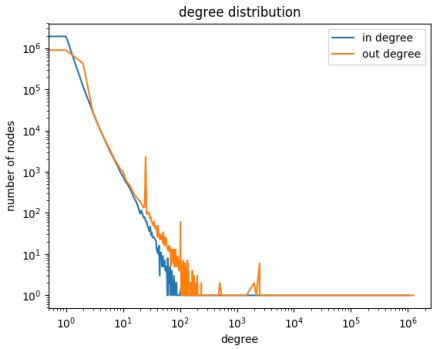

Fig. 4: Use-Deg Distribution

addresses on reasonable assumptions. The new algorithm's flowchart is shown as below: Using Union-Find to combine addresses.

1) Contract all addresses on the input part of a transaction.
2) Iterate through all transactions. If at least one change account is found in the output part of a transaction, contract all addresses on the output part except change accounts. Delete transactions that have already used for output contraction.
3) repeat step 2 until convergence.
4) Guess change accounts of the output part of the remaining transactions. Choose the most similar one in the output part according to the input user. Here, the similarity of two nodes is defined as the number of nodes that point to both of them plus the number of nodes that are pointed by both of them. If the highest similarity is 0, postpone the guessing.
5) goto step 2 until convergence
6) repeat step 3,4 until convergence, which means it can not guess the change accounts of the remaining transactions anymore.

The core essence of this new algorithm is that first, try to figure out the change account. Then, make a reasonable assumption that the rest of the output part belongs to a single user. This is because most of transactions are between two people, and the output par contains one receive address and one change address.

The result of our new algorithm gives 2167182 users, which is much smaller than the baseline method. The average number of addresses owned by one user is 11.36, which is much higher than the baseline, and a little lower than expected. Fig.4 shows the in/out degree distribution of the user graph produced by our new method. We can see that the bump in out degree line disappeared compared to Fig.2, which makes sense because transactions have two output users should be fewer than one output user.

V. User Categorization

A. Goal

Intuitively, users participating in Bitcoin transactions can be classified into the following categories:

- Trading centers
  Bitcoin-cash exchange site. Should have high degrees and involve in huge amount of TXNs.
- Miners
  Mine Bitcoins and trade for cash. Bitcoin value out should be much larger than value in. Out degree should be relatively small as the small amount of trading centers.
- Investors
  Buy and sell Bitcoins at trading centers to make money. Should have relative small in degree and out degree. Could be involved in a moderate amount of TXNs.
- Accumulators
  Accumulates Bitcoins from trading centers or other users. TXN value in is much larger than TXN value out.
- Merchants
  Accept Bitcoins as currency. Examples like gambling website and website accepting bitcoin donations. Should have a large in degree and a relatively small out degree. Value in and value out could be mostly the same.
- Customers
  Use Bitcoins as currency at merchants.

Please note that all the features of these categories listed above are based on intuition.

B. K-Means

K-means clustering is a method popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

For Bitcoin user clustering, the following features are extracted as input of K-Means:

- In TXN: Number of TXNs user appears on in side.
- Out TXN: Number of TXNs user appears on out side.
- Out Deg: Out degree of this user in user-user network
- In Deg: In degree of this user in user-user network
- Val in: Amount of Bitcoin flow into this user's wallet
- Val out: Amount of Bitcoin flow out from this user's wallet
- Hub&Auth score: Calculated from the user-user network

*Please note that InTXN corresponds to OutDegree, and OutTXN corresponds to InDegree.

The values of these features are normalized as follows:

$$norm\_val = \frac{val - mean}{stddev}$$

The improved version of user network generated in Part IV is used here. Clustering error for different value of k is shown in Fig. 5. Based on the result of Fig. 5 and experiments, we choose $k = 5$ as the number of clusters.

The average features for each cluster is given in Table V. We can see from the table that, different clusters have very different features. The intuitive explanations are as follows:

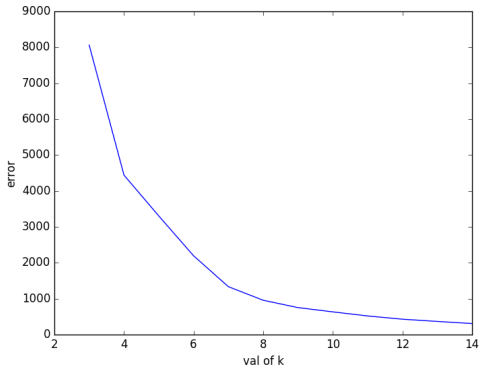

Fig. 5: K-Means Result

- Cluster 0:
  Users in this cluster are more likely to appear on output side on a transaction than input side. They also have much larger out degree than in degree. Meanwhile, value in is larger than value out in this cluster. We can infer that this cluster is the users of accumulators.
- Cluster 1:
  Obviously, the only user in this cluster is the trading center. This user is involved in huge amount of transactions. In and out degree is also very high. Hub and auth scores are also significantly bigger than users in other clusters.
- Cluster 2:
  Users in this cluster have roughly the same in and out degree. They participated in a bunch of transactions. This could be the cluster of investors.
- Cluster 3:
  Only 4 users are in this cluster. They have relative high transaction number and in-out degrees. Transaction value is also very high compare to other cluster. This could be the cluster of important merchants.
- Cluster 4:
  It is difficult to get some intuition from the features listed here. In and out degree are roughly the same, but TXN out is much larger than TXN in. Hub score for this cluster is low compared to Cluster 1, 2 and 3.

A summary of cluster-cluster edge distribution is given in Table VI. Fig. 6 gives a graph illustration of the relationship between different clusters. We can see from the table that:

- Cluster 0:
  Cluster 0 have very few out edges and self edges. Cluster 0 also have very few in edges from Cluster 2,3,4. However, 13.49% of edges are from Cluster 1 to Cluster 0. This proves our thoughts that users in Cluster 0 are accumulators. They get Bitcoin from the trading center and keep the money in their pocket for ever.
- Cluster 1:
  Cluster 1 is definitely the center of the whole network. 65.69% edges are related to this user. This proves the thought that this user should be the trading center.
- Cluster 2:

TABLE V: Average Features of Different Clusters.

Cluster	TXin	TXout	OutDeg	InDeg	ValIn
0	0.201	1.310	0.240	1.061	2.22e9
1	23709364	28748112	1243331	1049741	9.36e16
2	5.376	7.949	1.507	1.231	5.92e10
3	2557.25	2894	158.75	54.25	1.35e15
4	0.485	1.187	0.851	1.003	8.14e9

Cluster	ValOut	Hub	Auth	Cnt
0	1.66e9	4.56e-10	4.43e-5	470882
1	9.31e16	0.054	0.998	1
2	5.91e10	0.0009	3.27e-5	1049737
3	1.35e15	0.0009	9.04e-5	4
4	7.60e9	8.23e-9	4.15e-7	646558

TABLE VI: Cluster-Cluster Edge Distribution.

Cluster	0	1	2	3	4
0	0.03%	0.00%	1.39%	0.00%	1.82%
1	13.49%	0.00%	22.13%	0.00%	0.00%
2	0.40%	30.07%	6.80%	0.01%	8.07%
3	0.00%	0.00%	0.02%	0.00%	0.00%
4	0.40%	0.00%	6.68%	0.00%	8.69%

Cluster 2 have intimate relationship with Cluster 1. 52.20% edges are between them. This is in consistent with our thoughts that users in this Cluster are investors who trade with the trading center to get profit.

- Cluster 4:
  Surprisingly, Cluster 4 have no relationship with the trading center. It seems that Cluster 2 and Cluster 4 have a loose connection between each other.

In conclusion, Cluster 0 are accumulators who get Bitcoin from trading center and keep the money. Cluster 1 is the trading center. Cluster 2 are investors who trade with the trading center. Cluster 3 are merchants. Cluster 4 is much trickier to discern than other clusters. The ingredients of this cluster will be analyzed in Part VI.

VI. Community Detection and Flow Analysis

A. K-Means

1) Flow Analysis: TXN-TXN distribution and value flow distribution among different clusters are given in Table VII and Table VIII.

The contents of them are roughly in consistent with Table VI. Meanwhile, there are some differences:

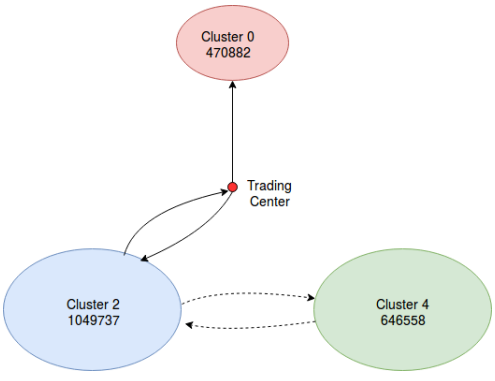

Fig. 6: Edge Relationship between Clusters

TABLE VII: Cluster-Cluster TXN Distribution.

Cluster	0	1	2	3	4
0	0.01%	0.00%	0.43%	0.00%	0.50%
1	5.71%	0.00%	32.70%	0.01%	0.00%
2	0.15%	52.68%	2.90%	0.00%	1.95%
3	0.00%	0.08%	0.01%	0.00%	0.00%
4	0.10%	0.00%	1.45%	0.00%	1.32%

TABLE VIII: Cluster-Cluster Value Flow.

Cluster	0	1	2	3	4
0	0.00%	0.00%	1.06%	0.00%	0.85%
1	2.51%	0.00%	31.92%	0.66%	0.00%
2	0.03%	34.48%	7.43%	0.42%	7.40%
3	0.00%	0.49%	0.58%	0.14%	0.01%
4	0.04%	0.00%	7.88%	0.01%	4.09%

- 13.49% edges from Cluster 1 to Cluster 0 only occupies 5.71% TXNs and 2.51% value flow. This implies that accumulator are more inactive compared to other users. The value exchanged between trading center and accumulators are also low compared to other TXNs.
- The edges and value exchanges between them occupies about 7% of total traffic. However, the TXN amount between them occupies only 2%. Therefore, the TXNs between Cluster 2 and Cluster 4 are of high value.

In conclusion, we can use edge distribution between clusters to represent TXN distribution and value flow.

2) Radiant Community Detection: We notice that majority of nodes in the network have 1 in degree and 1 out degree. Delving deep into the node degree features, we separate out the following radiant structure community with the trading center sitting in the core position:

- Cluster 0:
  388372 out of 470882 users have 1 in degree and 0 out degree. These users only connects to the trading center.
- Cluster 2:
  546449 out of 1049737 users have 1 in degree and 1 out degree. Besides that, 30433 users have 0 in degree and 1 out degree. These users only connect to the trading center.
- Cluster 4:
  344507 out of 646558 users have 1 in degree and 0 out degree. These users are connected to nodes in Cluster 2 and Cluster 4.

Fig. 7 gives the outline of this radiant community. These results certifies the conclusions we made in Part V.B. About 45.3% of users in the network are only connected to the trading center. They are either accumulators, investors or miners.

3) Community Detection after Removing Singletons: Since about half users are only connected to the trading center, these nodes are removed from the Bitcoin network. The community structure after removing these singletons is shown in Fig. 8.

The whole network forms a huge weakly connected component(WCC). The strong connected component(SCC) of the network contains part of Cluster 2 and Cluster 4. The size of

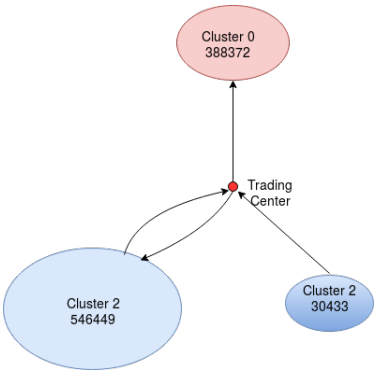

Fig. 7: Radiant Community Structure

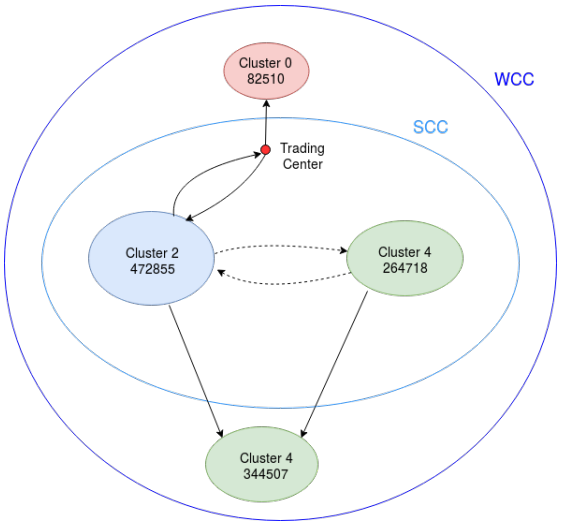

Fig. 8: Community Structure after Removing Singletons

the SCC is 744681, which is in consistent with the cluster size after separation of singletons.

After removing the trading center from the network, WCC size is 734826. The original huge SCC no longer exists. Size of SCC drops significantly to 57405. This indicates that, after removing the trading center, the graph falls apart into many small pieces. Currency flow in the new WCC is mostly "one way trip". It is rare for users to trade back and forth with each other.

B. Node2Vector

1) Algorithm: The key idea of node2vec comes from word2vec. Word2vec uses skip-grams to predict context words given target, and use negative-sampling to achieve computational efficiency. The essence of this algorithm is to project each word into a low dimension vector. Then it uses word co-occurrence to train this embedding matrix. Combining with negative-sampling, it trains binary logistic regressions for a true pair versus a couple of noise pairs. The loss function can be written as following:

$$J_t(\theta) = \log \sigma(u_o^T v_c) + \sum_{i=1}^{k} [\log \sigma(-u_{rand(j)}^T v_c)]$$

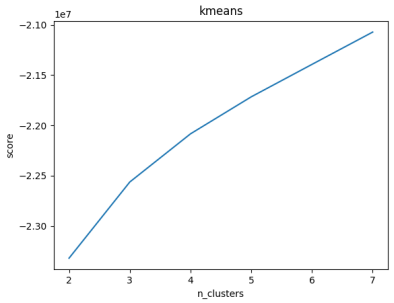

Fig. 9: K-Means Result of embedding matrix

TABLE IX: User property of clusters.

Cluster	0	1	2	3	4
nodes %	16.64%	16.03%	35.30%	16.04%	15.99%
inter user/node	0.14	0.15	0.00	0.17	1.53
in user/node	1.03	1.05	1.05	1.05	2.77
out user/node	1.46	1.48	0.00	1.49	3.76

Here we just replace words with nodes and word co-occurrence with node neighborhood. To detect communities, we need to group together closely linked nodes. Most of the edges should be inside groups between nodes from the same group and few of the edges should bridge across groups between nodes from different groups. Therefore, we choose BFS-like neighborhood to train the model. Because node2vec from snap consumes too much memory, we implemented our own light-weighted version. The detailed implementation of our algorithm shows as follow:

1) Choose dimension = 10, k_sample = 10, learning_rate = 0.01.
2) Each node has two vector: src_vec as u and dst_vec as v
3) Apply SGD to minimize loss function:
   - Compute gradient of one true pair and ten noise pairs. They are selected randomly by generating random permutation of nodes.
   - Take care of overflow of exponential term.
   - Modify src_vec first, then dst_vec.
   - Iterate until converge.

There are total 2129849 nodes, and this algorithm takes about 12 hours to complete. After getting the embedding matrix, we run kmeans on these 20 features(10 src and 10 dst). We tried different cluster numbers to find the optimal one. Fig. 6 shows total loss changing with cluster number. We can see that there is no obvious turning point, meaning that bitcoin society has loose community structure. As the increasing rate is linear after 5 clusters and to be consistent with former method, we decide to select 5 as final cluster number.

2) Results: Table IX to XIV shows the result of communities detected by node2vec. Table IX to XI summarized key features of each cluster. Table XII to XIV captured relations among these clusters.

Table IX shows user properties of all clusters. It captures user edges inside each group as "internal user per node",

TABLE X: Transaction & value property of clusters.

Cluster	0	1	2	3	4
inter trans/node	0.10	0.12	0.00	0.15	6.18
in trans/node	2.73	2.42	0.85	2.74	11.31
out trans/node	3.96	3.84	0.00	4.41	8.83
inter val/node	1.77E9	3.33E9	0.00	2.14E9	2.17E10
in val/node	1.45E10	1.71E10	5.58E8	1.54E10	3.65E10
out val/node	1.48E10	1.73E10	0.00	1.57E10	3.69E10

TABLE XI: Value & Transaction & user relations of clusters.

Cluster	0	1	2	3	4
inter val/trans	1.69E10	2.87E10	N/A	1.46E10	3.52E9
in val/trans	5.31E9	7.06E9	6.55E8	5.61E9	3.22E9
out val/trans	3.73E9	4.51E9	N/A	3.56E9	4.18E9
inter val/user	1.3E10	2.25E10	N/A	1.27E10	1.42E10
in val/user	1.4E10	1.63E10	5.32E8	1.47E10	1.32E10
out val/user	1.01E10	1.17E10	N/A	1.05E10	9.82E9

incoming user edges as "in user per node" and outgoing user edges as "out user per node". Here, user edges comes from directed user graph. An edge from user A to user B means that there exists a transaction where A send B bitcoins. There is at most one edge from A to B. Internal means a user inside the group send bitcoins to a user inside the group. Incoming means a user from outside the group sends bitcons to a user inside the group. Outgoing is the opposite.

From Table IX, we can see that Cluster 2 has more than twice number of nodes than other clusters. Other clusters have similar number of nodes. Cluster 0,1,3 have very similar user properties, where each user has more outgoing connections than incoming connections, and has much fewer internal connections. Cluster 4 has largely similar structure as these three clusters, but users in Cluster 4 are much more active. They have 10 times more internal connections and twice more incoming and outgoing connections. This suggests that members of Cluster 4 are more closely connected to each other than other clusters. Cluster 2 is rather interesting. It is the largest cluster, and have zero internal and outgoing connections. This means that every user inside this group only receive bitcoins from others, but never send bitcoins to others. These users are classified as hoarders, who invest into bitcoins for long term. They buy a certain amount of bitcoins and hope the moving average of bitcoin price raises

TABLE XII: Cluster-Cluster User Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	1.39%	1.33%	3.01%	1.33%	9.15%
S 1	1.35%	1.45%	2.85%	1.42%	8.89%
S 2	0.00%	0.00%	0.00%	0.00%	0.00%
S 3	1.42%	1.44%	2.77%	1.65%	8.97%
S 4	7.72%	7.45%	13.97%	7.52%	14.93%

TABLE XIII: Cluster-Cluster TXN Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	0.39%	0.37%	0.51%	0.38%	13.57%
S 1	0.37%	0.42%	0.53%	0.40%	12.55%
S 2	0.00%	0.00%	0.00%	0.00%	0.00%
S 3	0.40%	0.41%	0.49%	0.53%	14.62%
S 4	9.47%	7.96%	5.23%	9.14%	22.26%

TABLE XIV: Cluster-Cluster Value Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	1.61%	1.63%	0.10%	1.52%	10.16%
S 1	1.76%	2.92%	0.11%	1.91%	11.39%
S 2	0.00%	0.00%	0.00%	0.00%	0.00%
S 3	1.57%	1.78%	0.10%	1.88%	10.31%
S 4	9.86%	11.56%	0.76%	10.08%	18.99%

continuously in the future. On the contrary, people in Cluster 4 are classified as speculators, who buy and sell bitcoins regularly to earn from fluctuation of bitcoin price. They invest into bitcoins for short term. People in other clusters seems to be regular users who treat bitcoin as a new form of currencies. They use bitcoins to complete secret transactions when they want to protect their privacy and don't want anyone else know that they participated in this transaction. In conclusion, Cluster 2 is a community of hoarders, because incoming value per node is much larger than outgoing value per node; Cluster 0,1,3 are communities of regular users; and Cluster 4 is a community of speculators, because users in this cluster are much more active than other clusters.

Table X shows transaction and value properties of all clusters. It captures transaction/value edges inside each group as "internal trans/value per node", incoming transaction/value edges as "in trans/value per node" and outgoing transaction/value edges as "out trans/value per node". Here, transaction/value edges comes from directed TXN graph. A transaction from user A to user B forms a directed transaction/value edge from A to B. The weight of transaction edge is 1 and value is the input value. If there are multiple users on the output side of the transaction. Say there are n users, then there will be an edge from the input user to each of the output users. The weight of each transaction edge is 1/n and value is the corresponding output value. There can be more than one transaction/user edges from user A to user B.

From Table X, we can see that all clusters have similar transaction/value properties as user property. First, let's look at transaction properties. For Cluster 0,1,3, difference between internal and incoming/outgoing become larger. For Cluster 3, outgoing transactions become twice as many as incoming transactions. For Cluster 4, incoming transactions become more than outgoing transactions. Most of the transactions happened between Cluster 4 and other clusters. Also, most of the transactions happened externally instead of internally. Now, let's look at value properties. Other than the internal connection of Cluster 1,3,4 increases with respect to user properties, value properties are very similar to user properties.

Table XI shows value, transaction and user relations of all clusters. It captures ratios among value, transaction and user properties. From value per transaction ratio, we can see that although external value, transaction and user edges are much more than internal, internal value per transaction is much more than external for Cluster 0,1,2,3, and roughly equal for Cluster 4. Because people tend to make large transactions with people they trust, this means that people in Cluster 0,1,2,3 trust people in the same cluster more, while

people in Cluster 4 trust everyone as equal. This suggestion is reasonable due to different behavior of regular users and speculators. Regular users only want to protect their privacy and nothing else. Thus, to avoid fraud, they mostly make large transaction with familiar people. While speculators want to earn money and don't care about privacy. Thus, they don't care who is on the other side of the transaction as long as the transaction is proceeded successfully. From value per user ratio, we can see that all clusters have similar ratio for internal, incoming and outgoing transactions. This ratio seems to be constant between 9E9 to 1.5E10, where only Cluster 1's internal and incoming value per user are a bit higher than usual. This means that the expected total transaction value between two users are roughly the same, regardless of which groups do these two users come from.

Table XII to XIV shows cluster-cluster distribution of user, transaction and value properties. It captures relationship between different clusters. In these tables, rows represent source clusters and columns represent destination clusters. From these tables, we can see that most connections happened between Cluster 4 and other clusters. This means that bitcoin communities have a radiant structure, where Cluster 4 is at the center of bitcoin transactions which is heavily connected internally and connected to all the other communities externally, and other clusters are on the peripheral which are only connected with the center cluster externally and loosely connected internally.

C. Fiedler Vector

1) Algorithm: Fiedler vector comes from spectral clustering algorithm. This method aims to minimize conductance– connectivity between groups relative to the density of each group.

$$\phi(A,B) = \frac{cut(A,B)}{min(vol(A),vol(B))}$$

Fiedler vector is the second smallest eigenvector of Laplacian matrix. Laplacian matrix is degree matrix minus adjacency matrix. To ensure eigenvalues are non-negative real numbers and eigenvectors are real and orthogonal, we need to convert user transaction graph into an undirected graph. This implies that we ignore the difference between buyers and sellers. Another reason for doing this is to reduce computational complexity: computing eigenvalue of a symmetric matrix is much faster than a asymmetric matrix. After this simplification, we can still use the second smallest eigenvalue of Laplacian matrix to approximate minimum conductance, and get the corresponding eigenvector as an optimal cluster assignment of a relaxed problem. Then, we sort the components of Fiedler vector to group together components with similar value.

Because there are more than 2 million nodes, it is impossible to store the adjacency matrix. But good thing is there are only 3 million edges, which means it is a sparse matrix. Therefore we use scipy.sparse.linalg.eigsh to solve the smallest two eigenvectors of a sparse matrix (k=2). We set the initial vector to normalized ones to reduce the iteration for the first eigenvector, as we already know it is the smallest

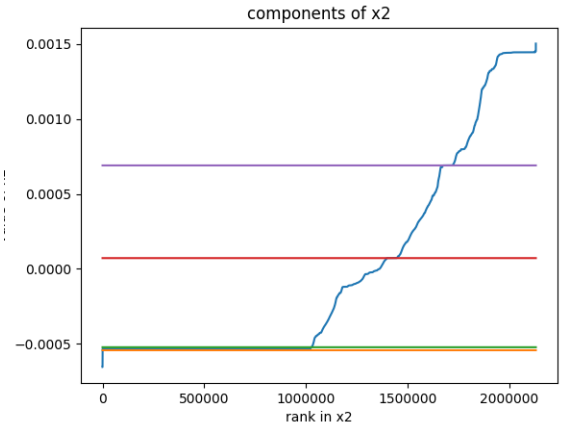

Fig. 10: Sorted components of Fiedler vector.

TABLE XV: User property of clusters.

Cluster	0	1	2	3	4
nodes %	0.02%	47.91%	19.92%	12.99%	19.16%
inter user/node	0.24	0.03	1.84	0.72	0.76
in user/node	1.73	0.95	1.98	0.87	0.31
out user/node	1.19	0.60	2.81	1.19	0.09

eigenvector. And set the number of Lanczos vectors to 5, as it it is recommended that ncv ¿ 2*k. sparse.eigsh uses Arnoldi iteration to reduce $O(n^3)$ complexity for dense matrix to $O(n^2)$ for sparse matrix. This algorithm takes about 12 hours to complete, and Fig. 9 shows the result. Because Fiedler vector attempts to minimize the cut of the partition, the nodes on the two sides of an edge tend to have similar values. This implies that nodes in a heavily connected community tend to have similar values. Therefore, the flat region of the sorted curve should be in the same group. As can be seen from Fig. 10, there are approximately 4 plateaus. Each plateau becomes a cluster. On the left-down corner of Fig. 10, there are a few nodes that have a much smaller value than the nearest plateau. To capture this detail and have the same number of clusters as former methods, we decide to single out these nodes to have a cluster of their own.

2) Results: Table XV to XX shows the result of communities detected by Fiedler Vector. Table XV to XVII summarized key features of each cluster. Table XVIII to XX captured relations among these clusters.

From Table XV, we can see that Cluster 1 has nearly half of total user nodes, and Cluster 0 has few user nodes. Cluster 0,1,4 have more incoming user edges than outgoing user edges, while Cluster 2,3 have more outgoing edges.

TABLE XVI: Transaction & value property of clusters.

Cluster	0	1	2	3	4
inter trans/node	0.21	0.03	7.95	0.56	0.33
in trans/node	3.86	1.65	7.42	2.58	0.55
out trans/node	3.94	1.58	5.93	5.27	0.43
inter val/node	2.73E7	2.00E8	3.77E10	9.83E9	2.7E9
in val/node	2.98E9	5.20E9	2.15E10	1.46E10	1.21E9
out val/node	2.86E9	5.25E9	2.16E10	1.46E10	9.7E8

TABLE XVII: Value & Transaction & user relations of clusters.

Cluster	0	1	2	3	4
inter val/trans	1.27E8	7.06E9	4.74E9	1.74E10	8.23E9
in val/trans	7.71E8	3.16E9	2.89E9	5.67E9	2.22E9
out val/trans	7.26E8	3.32E9	3.64E9	2.77E9	2.26E9
inter val/user	1.16E8	6.58E9	2.05E10	1.37E10	3.57E9
in val/user	1.72E9	5.5E9	1.09E10	1.67E10	3.91E9
out val/user	2.41E9	8.68E9	7.7E9	1.23E10	1.11E10

TABLE XVIII: Cluster-Cluster User Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	0.00%	0.00%	0.01%	0.00%	0.00%
S 1	0.00%	0.89%	17.67%	0.00%	0.00%
S 2	0.02%	27.66%	22.34%	6.11%	0.31%
S 3	0.00%	0.00%	6.11%	5.71%	3.31%
S 4	0.00%	0.00%	0.22%	0.81%	8.83%

Cluster 0,1 have loose connection internally, while Cluster 2,3,4 have strong connection internally. Cluster 1 has nearly no internal connection and Cluster 4 have nearly no outgoing connections. User properties are very similar between Cluster 2 and 3, except that people in Cluster 2 are much more active than Cluster 3. They are selling bitcoins in general. Cluster 0 and 1 have very similar user properties, except that people in Cluster 0 are much more active than Cluster 1. They are buying bitcoins in general. Cluster 4 is very different. People in Cluster 4 mostly connected with themselves, and only have incoming connections. They are also buying bitcoins, but are much more active internally than externally, while people in Cluster 0,1 are much more active externally. In conclusion, Cluster 4 is a community of hoarders, because incoming value per node is much larger than outgoing value per node; Cluster 0,1,3 are communities of regular users; and Cluster 2 is a community of speculators, because users in this cluster are much more active than other clusters.

From Table XVI, we can see that most clusters have transaction/value properties roughly similar as user property. First, let's look at transaction properties. Cluster 0,1,3 have very similar transaction properties as user properties, except that the difference between internal connection and external connection become larger. Cluster 2 is interesting. it's transaction property is the opposite of user property, which means while people in Cluster 2 have more outgoing user connections, they make much more transactions through internal or incoming user connections. Cluster 4's transaction property is also different from user property. Although people in Cluster 4 have more internal user connections, they make more transactions via incoming user connection. Now, let's look at value properties. Cluster 0,1,3,4 have value properties similar as user property. But the difference between internal connection and external connection becomes much larger for Cluster 1, while it becomes smaller for Cluster 2,3 and remain the same for Cluster 4. Just as transaction property, value property of Cluster 2 is also opposite to user property, which means although people in Cluster 2 have more external user connection than internal user connection, most value is exchanged via internal connection.

TABLE XIX: Cluster-Cluster TXN Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	0.00%	0.00%	0.02%	0.00%	0.00%
S 1	0.00%	0.31%	17.06%	0.00%	0.00%
S 2	0.01%	17.75%	35.66%	7.33%	1.52%
S 3	0.00%	0.00%	14.60%	1.65%	0.84%
S 4	0.00%	0.00%	1.63%	0.22%	1.41%

TABLE XX: Cluster-Cluster Value Distribution.

Cluster	D 0	D 1	D 2	D 3	D 4
S 0	0.00%	0.00%	0.00%	0.00%	0.00%
S 1	0.00%	0.52%	13.74%	0.00%	0.00%
S 2	0.00%	13.62%	41.02%	9.61%	0.29%
S 3	0.00%	0.00%	9.39%	6.98%	0.98%
S 4	0.00%	0.00%	0.24%	0.77%	2.82%

Table XVII shows value, transaction and user relations of all clusters. First, let's look at value per transaction ratio. For Cluster 1,3,4, we can see that although external transaction edges are much more than internal, internal value per transaction is much more than external, which means people in these clusters trust people in the same cluster more. For Cluster 2, cluster members have both more transactions and more value per transaction internally than externally, which means that they not only trust people in the same cluster more, but also are more willing to trade with these people. For Cluster 1, the situation is the opposite of other clusters. It seems that people in Cluster 1 trust external people more than internal people. Maybe people in Cluster 1 have competitive relationships, which make them not trust each other. Now, let's take a look at value per user ratio. We can see that cluster 0,2 have similar value per user ratio as value per transaction ratio. Cluster 1,3 have similar ratio for internal, incoming and outgoing transactions, which means people in these groups expect to trade similar amount of value regardless of which group do the other user come from. Cluster 4 is interesting. It's outgoing value per user is much higher than the internal or incoming, which means they only sell bitcoins to a certain few external people.

Table XVIII to XX shows relationship of user, transaction and value properties between different clusters. From these tables, we can see that most connections happened inside Cluster 2. But there are still some considerable amount of connections happened between Cluster 2 and Cluster 1,3, and inside Cluster 3,4. This means that bitcoin communities have two parts. One major part with Cluster 2 at the center which is heavily connected internally and connected to all the other communities externally, and Cluster 1,3 on the peripheral which are only connected with the center cluster externally and loosely connected internally. And a small part of Cluster 4, which is heavily connected internally, but has few external connections.

VII. Conclusions

In this project, we implemented a new method for user network generation and tried three different methods for user categorization and community detection for Bitcoin network. The conclusions are as follows:

- User degree and user-TXN in Bitcoin network follows power-law distribution.
- After contraction of the output part, the degree distribution follows power law without a bump at degree 2, and the average number of addresses owned by a single user is much closer to the expected value.
- K-Means separates the users into 5 different clusters with different features. Trading center, accumulators, investors, merchants can be recognized.
- A radiant community structure is detected by analyzing network structure. Half of users only trade with the trading center.
- Clusters generated by K-Means is in consistent with the WCC and SCC of the network. Trading center sits in the core position of the network. Network fells apart quickly after removing the trading center.
- A radiant community structure is detected by node2vec. Communities of hoarders, regular users and speculators are recognized. Hoarders and Regular users trust people in the same community more, while speculators trust everyone as equal.
- A major radiant plus a minor community structure is detected by Fiedler vector. Buyers, sellers, inactive and active groups are recognized. A group with competitive relationship is found, where group members don't trust each other.

ACKNOWLEDGMENT

We would like to thank Professor Jure Leskovec for teaching us the foundational material, Srijan Kumar for providing helpful advice as well as our TA for giving us feedback and tips for improving our model/project.

REFERENCES

[1] Ron D et al., Quantitative Analysis of the Full Bitcoin Transaction Graph
[2] Koshy P et al., An Analysis of Anonymity in Bitcoin Using P2P Network Traffic
[3] Clauset et al., Finding community structure in very large networks
[4] Palla et al., Uncovering the overlapping community structure of complex networks in nature and society
[5] ELTE Bitcoin Project website and resources: http://www.vo.elte.hu/bitcoin/downloads.htm