ISYE 6740 – Summer 2024

Clustering of Ethereum Wallet Transactions by Adjacency Symmetrization

Noah Casey

Background

The world of blockchain is an ever-evolving landscape, with a growing list of blockchain protocols, cryptocurrencies, and digital assets that are increasing in complexity. The uses of blockchain technology are beginning to provide significant societal benefits. There are discussions about utilizing blockchain for election integrity, cryptocurrencies being adopted in countries with unstable currencies, and blockchain technology offering a sense of control in an increasingly interconnected world. However, as with all technological advancements, there are drawbacks: scams, fraud, and untraceable criminal transactions tarnish the technology's reputation.

Ethereum is a popular blockchain protocol that prides itself on speed and environmental efficiency. The native cryptocurrency token of Ethereum is ETH, although other tokens can also be traded on the blockchain. Ethereum incorporates many features such as proof of stake, smart contracts, and web3. For the sake of relevance, we will focus on two primary features of blockchain: wallet addresses and transactions.

Each blockchain has a native token that serves as the base currency for transactions. This can be compared to a national currency that can be exchanged for other currencies at an airport. In this analogy, the native token, ETH, is equivalent to the national currency, while other cryptocurrency tokens on the blockchain represent other currencies that can be traded according to the exchange rate. These tokens can be exchanged between wallets or with a liquidity pool.

Wallets on the Ethereum blockchain are text-based addresses used to send and receive cryptocurrency. Each address is unique, ensuring that cryptocurrency transfers occur only between the intended parties. While a single person may own multiple wallet addresses, transferring cryptocurrency between them requires transactions. Sometimes, transactions involve a liquidity pool, also known as an automated market maker (AMM). AMMs facilitate seamless transactions and provide some price stability by allowing wallets to deposit a token and receive a counter-token in return, similar to a bank deposit.

Methods

The current project is to cluster wallets and transactions to glean insight into different categories of transactions that occur on the blockchain as well as differentiating wallets. It is the current thought that this may reveal transaction networks between organizations or a group of individuals that primarily use cryptocurrency. This may give way to techniques in finding criminal networks that use cryptocurrency. We may also be able to identify wallets of market makers or institutions that heavily use cryptocurrency. A successful model will provide an appropriate number of clusters with reasonable separability that may be investigated further to discover relationships between the wallets and transactions in each cluster.

We extracted transaction data from Ethereum block 16,000,000 to 16,100,000 using Chainstack's [5] public API. We sorted the wallets by total amount of ETH transacted and used the highest 1,000 wallets as our nodes for effeciency. A more in-depth study should obtain appropriate compute power and storage to handle 3TiB Numpy array handling. Also note that our method may neglect clusters that may exist in the full dataset.

We will use multiple approaches for our cluster analysis using a variety of initial transformations on the similarity matrix. We consider the weights to be the sum of the transactions from $i \to j$ in ETH. The graph will be weakly directed, so only wallets that receive or send ETH will be considered. For each transformation type, we will run the model using spectral clustering by constructing the graph Laplacian matrix and using eigendecomposition. Because the eigenvectors are unlikely to be perfectly sepearable, we will use the k largest eigenvectors, $v^1, v^2, \dots, v^k$. We will then utilize $z^1 = (v_1^1, v_1^2, \dots, v_1^k), z^2 = (v_2^1, v_2^2, \dots, v_2^k), \dots, z^m = (v_m^1, v_m^2, \dots, v_m^k)$ as the new data points for points $1, 2, \dots, m$ and evaluate using the rbf kernel in skikit-learn's SpectralClustering module. We will use the Davies-Bouldin Index as our evaluation criteria to determine the best number of clusters. This index is the average ratio of the distances within the cluster to the distances between clusters. Thus, clusters that are further apart and less-dispersed will result in a lower, better score. For each transformation type, we will retain the clusters determined by the algorithm that leads to the lowest Davies-Bouldin Index.

To assess each symmetrization, we will construct a simple dataset where each node (wallet) contains four features: total outgoing amount, total incoming amount, total outgoing transactions, and total incoming transactions. We will qualitatively examine the distribution of the groups on these features to determine whether or not the method is effective.

The Davies-Bouldin Index, is defined by

$$DB = \frac{1}{k} \sum_{i=1}^{k} max(\frac{\Delta X_i + \Delta X_j}{\delta(X_i, X_j)}).$$

Where ΔX_k is the within cluster sum of squares and $\delta(X_i, X_j)$ is the intercluster distance betweek clusters X_i and X_j.

We will use transformations to unipartiate weighted networks, which will maintain directionality of the network [3]. We will also use the naive graph transformation, which lacks the ability to retain directional information, as a control to test the effectiveness of more complicated models.

Naive Graph Transform

The most simple approach in a directed graph transform is to naively assume that if a transaction occurs between two parties, then there is a relationship. Thus, we will simply combine the weights from $i \to j$ and $j \to i$ to create a symmetric, weighted simliarity matrix. Given directed adjacency matrix A, we define

$$A_u = A + A^T$$

as our new similarity matrix.

Bibliometric Symmetrization

A natural requirement for clustering is that a symmetrization approach should create edges between similar nodes even though in the original network they do not exist. This ensures that if a two wallets transact with one wallet, they are related. We consider $B = AA^T$ and $C = A^T A$ where the former contains information about common outgoing edges and the latter contains information about common incoming edges. Thus, we will use

$$A_u = B + C$$

as our new adjacency matrix.

Degree-Discounted Symmetrization

We begin with the weighted, directed adjacency matrix. This transformation, known as degree-discounted symmetrization, takes the power-law degree distribution into account. This distribution claims that there are a few nodes in the network with a very high degree compared to the rest of the network. Generally, this transformation looks to give a higher level of similarity to nodes with low cardinality. That is, if only five wallets sent cryptocurrency to wallet x, then these have higher similarity than five-hundred wallets sending cryptocurrency to wallet y. The transformation considers the in and out transactions. Let

$$B = D_{out}^{-0.5} A D_{in}^{-0.5} A^T D_{out}^{-0.5}$$
$$C = D_{in}^{-0.5} A^T D_{out}^{-0.5} A D_{in}^{-0.5}.$$

Where in refers to transactions into a wallet and out refers to transactions coming out of a wallet. Thus, we have the similarity matrix

$$A_u = B + C.$$

Symmetrization Based on Random Walks

We define an adjacency matrix by

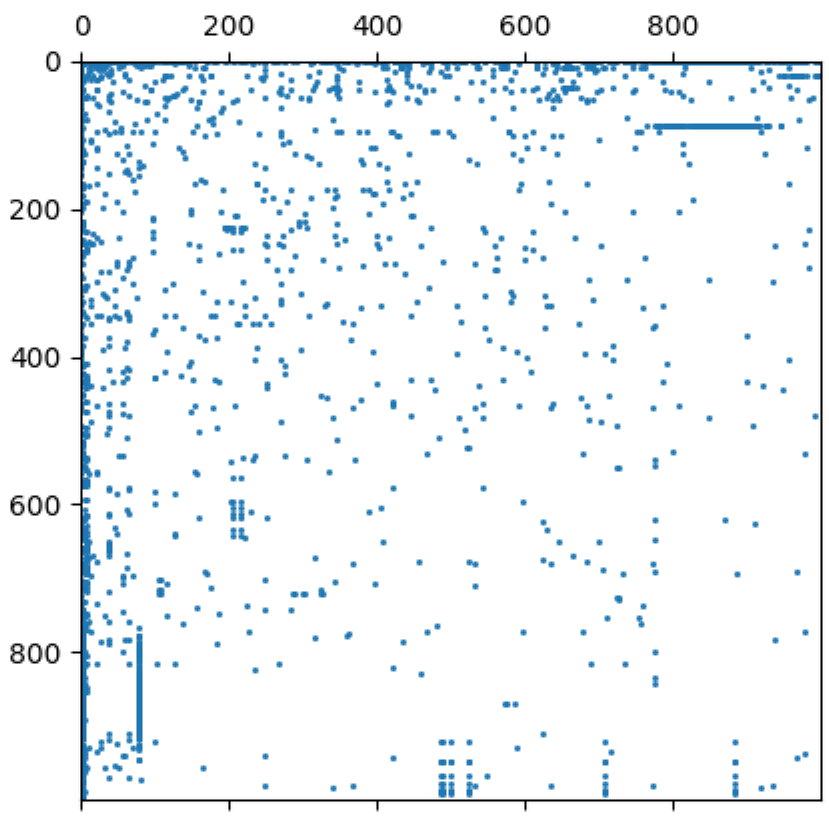
$$A_u = \frac{\Pi P + P^T \Pi}{2}.$$

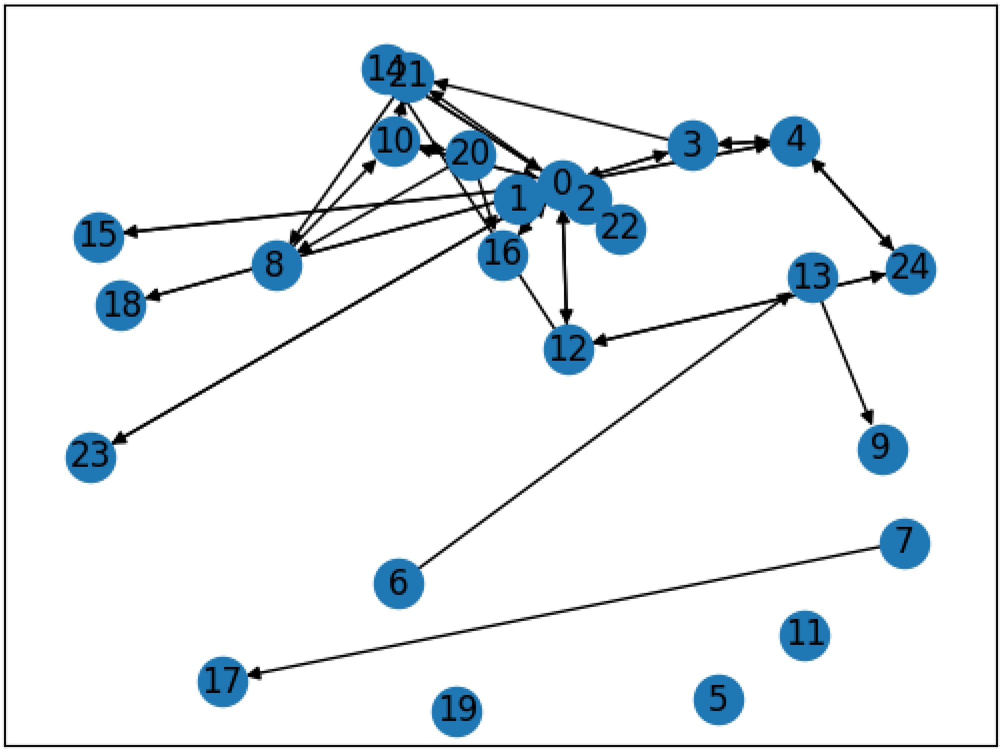
Where P is the transition matrix of the random walk and Π is the diagonal matrix with the probabilities of staying at each node in the stationary state. This transformation makes it easier to extract clusters that satisfy the criterion of low normalized cuts. That is, a group is well-connected with itself, but sparsely connected with the rest of the graph [4].

Results

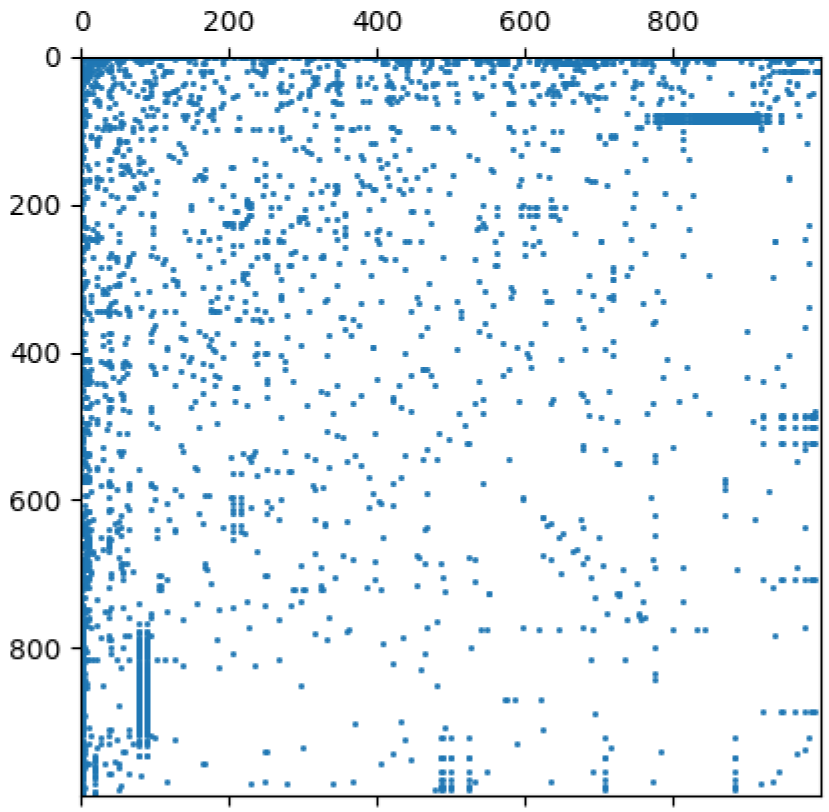
Analysis of the cluster groups proves that in this study, the Naive Graph Transform performs moderately well on our features, seemingly separating the clusters between most transactions and most ETH transacted with. The Bibliosymmetric Transform and Degree-Discounted transform appeared to cluster each feature rather well. It is worth noting that these two techniques are nearly identical mathematically minus some stabilization terms, so it is unsurprising that they yield similar results. However, the Random Walk Symmetrization technique appeared to perform rather poorly, which was expected when examining the network graph of the nodes, which are sparsely connected.

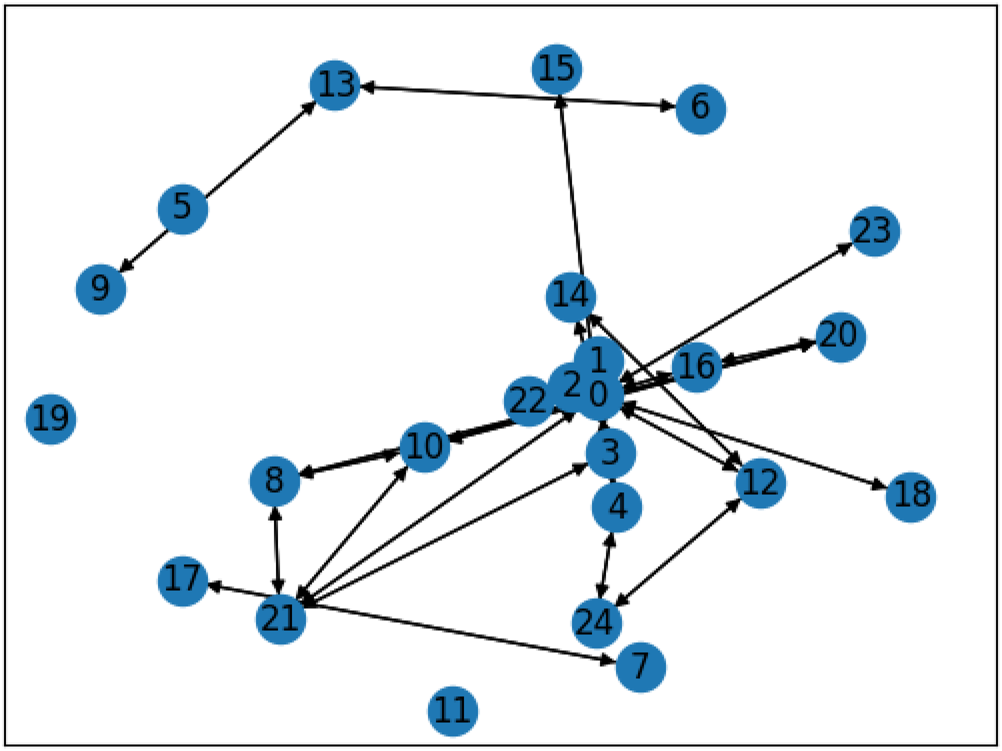
The original adjacency matrix and graph are of the following form. Note that all of the graphs below are limited to 25 nodes for readability.



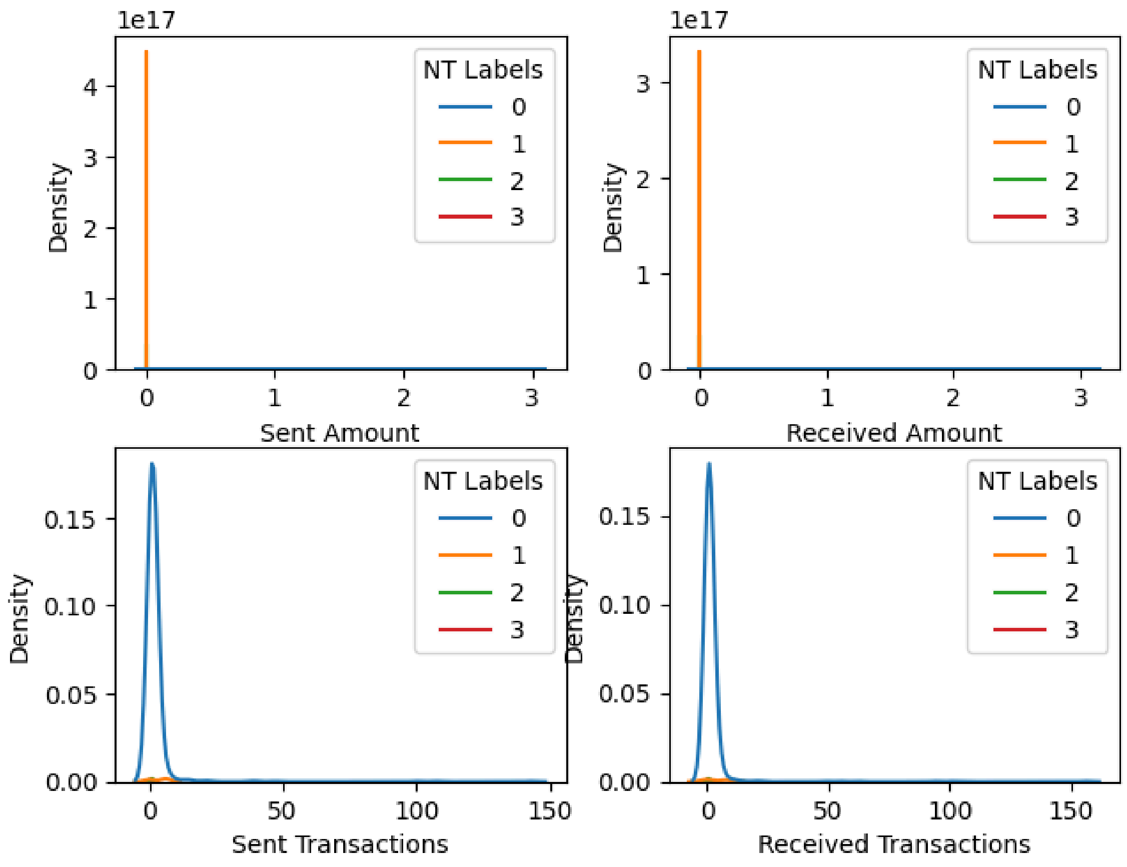


We will first look at the Naive Transform, which is considered the control as it is the simplest transform.



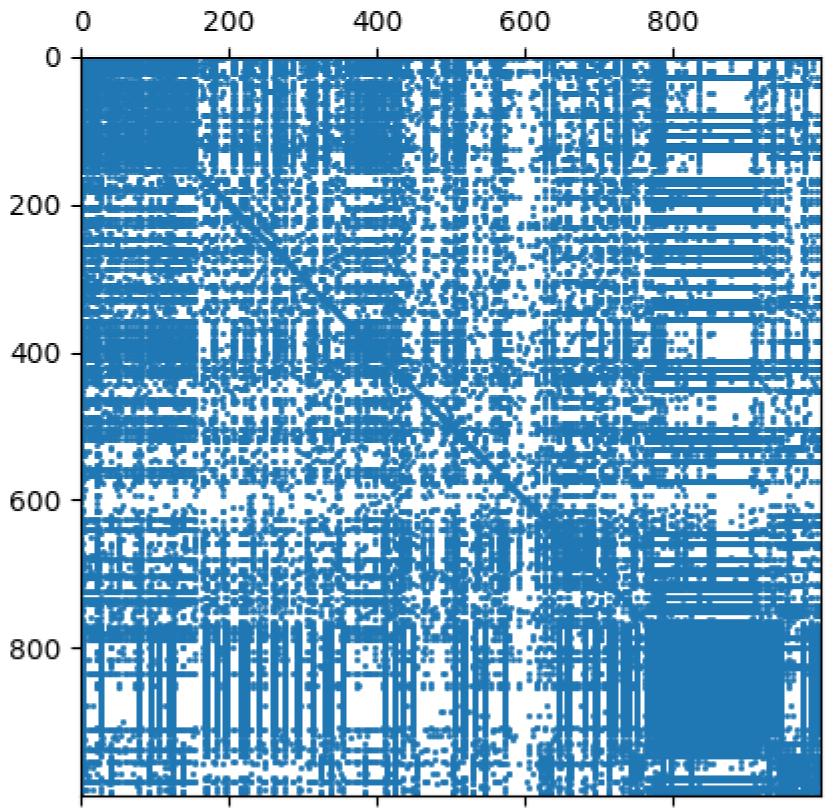
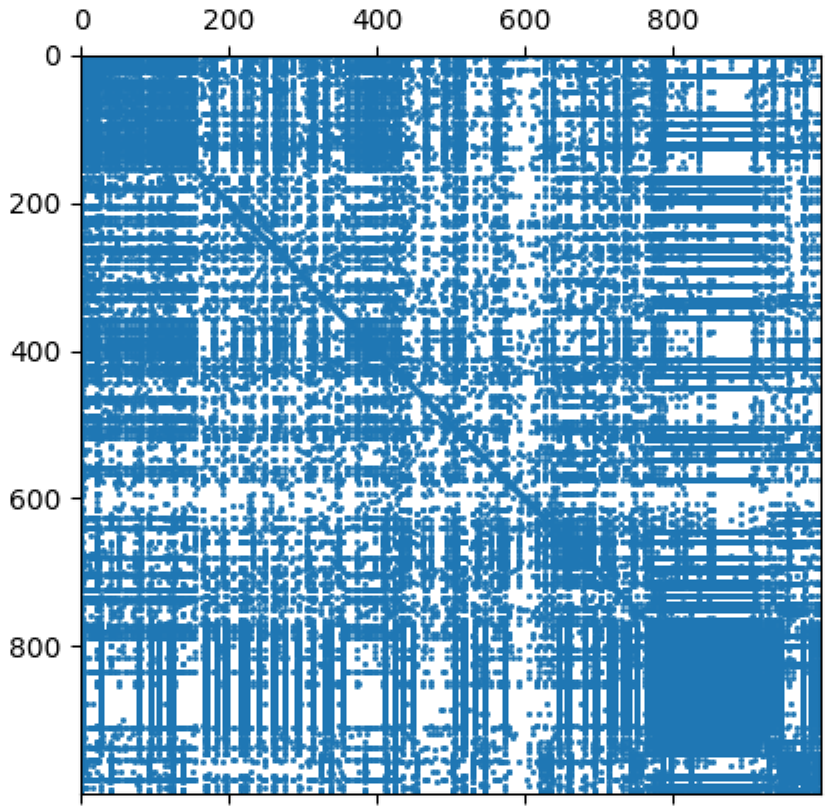


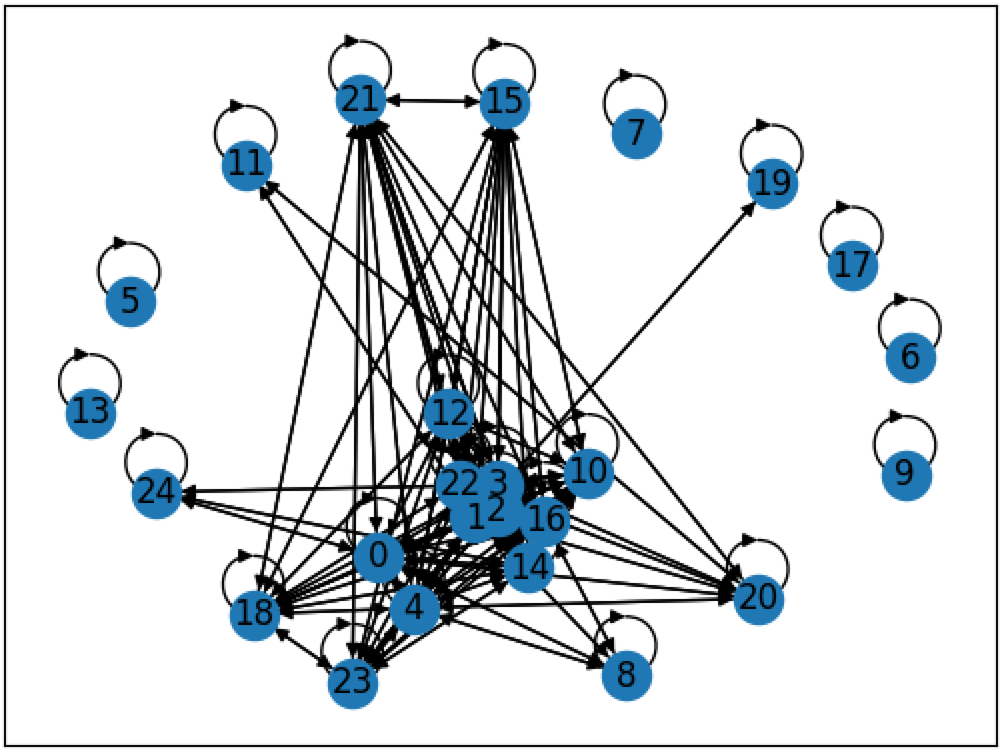
We see that the graph becomes bidirectional, knowing that it retains some concept of weights between nodes. The results for each of our features are displayed below.

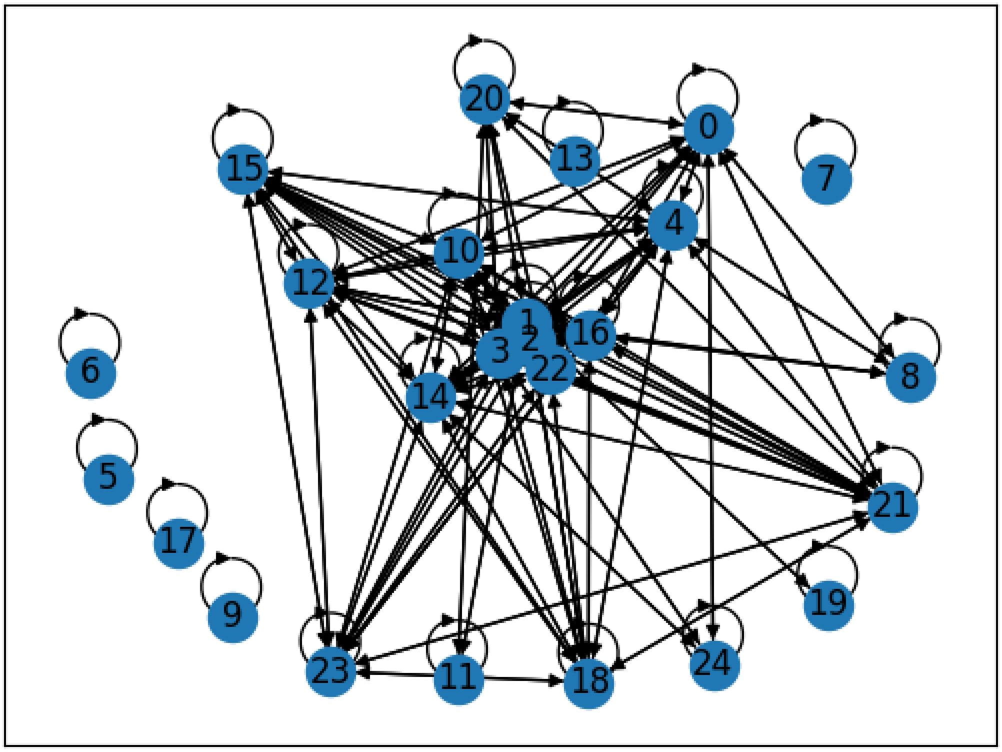


We see that two different groups are clustered on 0 transactions and 0 amount transacted with. This implies that there is a reasonable grouping happening in this method.

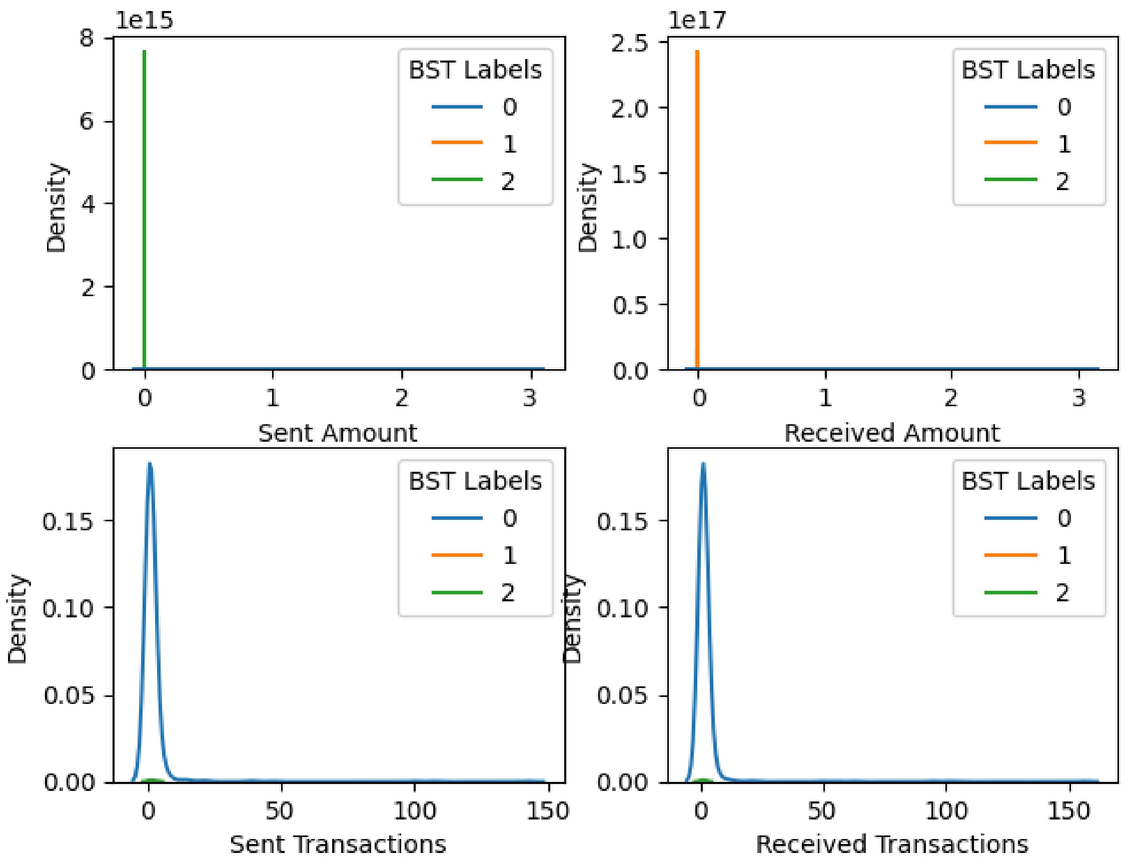
We will next consider both the Bibliosymmetric transform and the Degree-Discounted Transform respectively and take note that the results are qualitatively identical, meaning that these two techniques are essentially the same when it comes to clustering by our features.

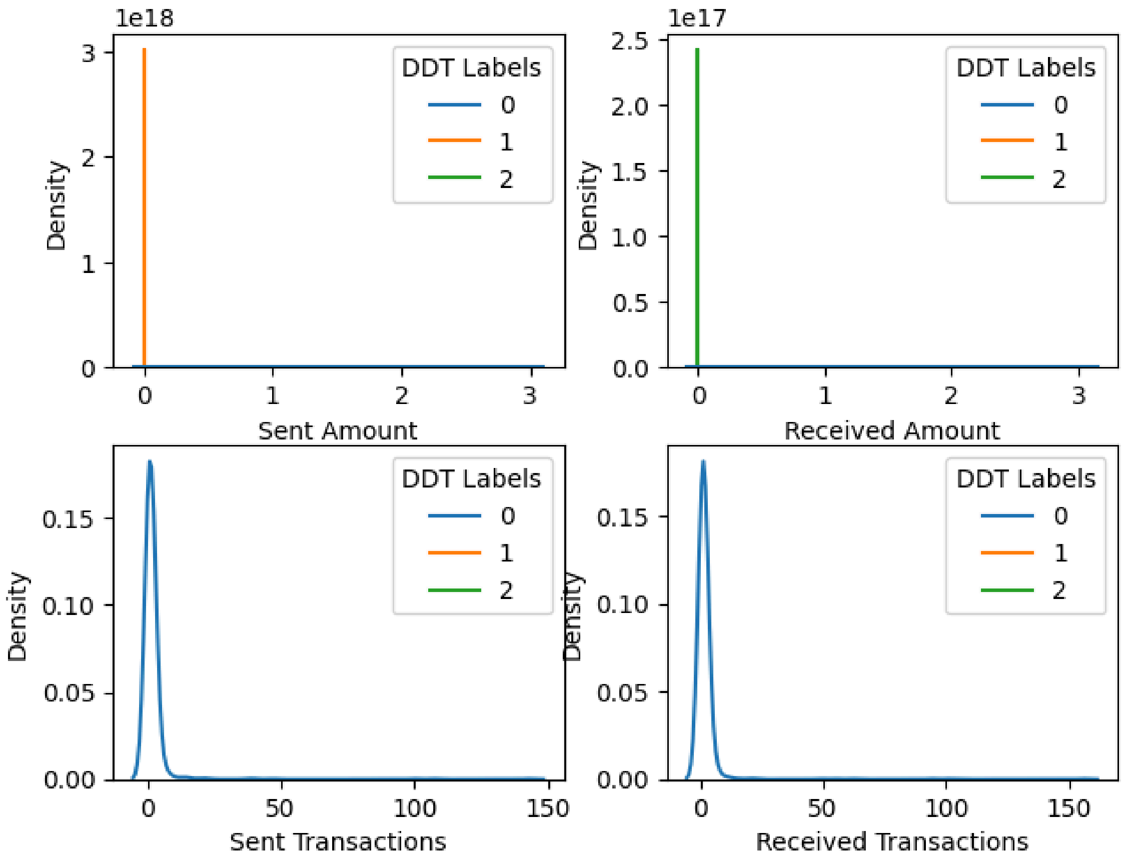

0
200
400
600
800
0
200
400
600
800

0
200
400
600
800
0
200
400
600
800





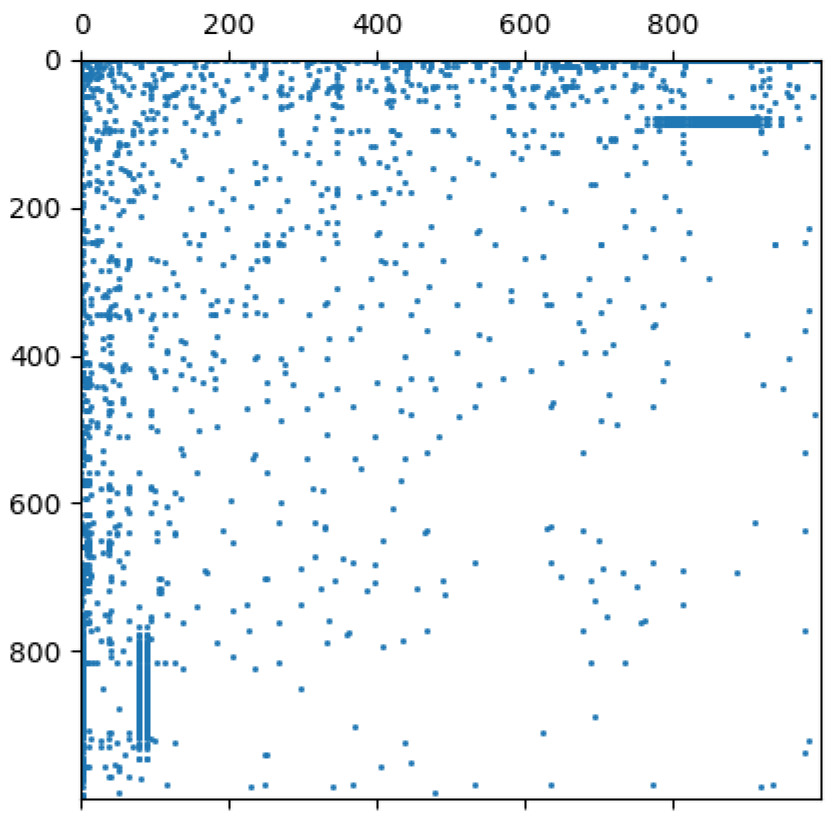
Take note of the strong interconnected relationships that are created in these transforms as well as the self loops that are in place. This likely assists with the strong separability in these two methods as seen below.

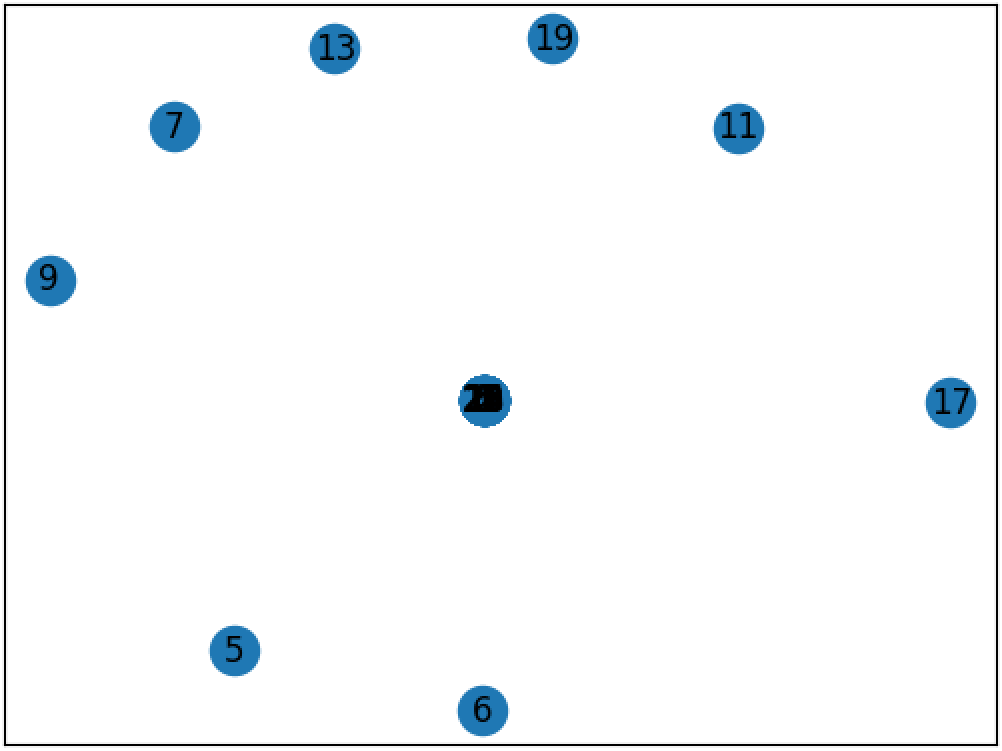

1e15
Density
Sent Amount
BST Labels
0
1
2
1e17
Density
Received Amount
Density
Sent Transactions
Density
Received Transactions


1e18
Density
Sent Amount
DDT Labels
0
1
2
1e17
Density
Received Amount
Density
Sent Transactions
Density
Received Transactions

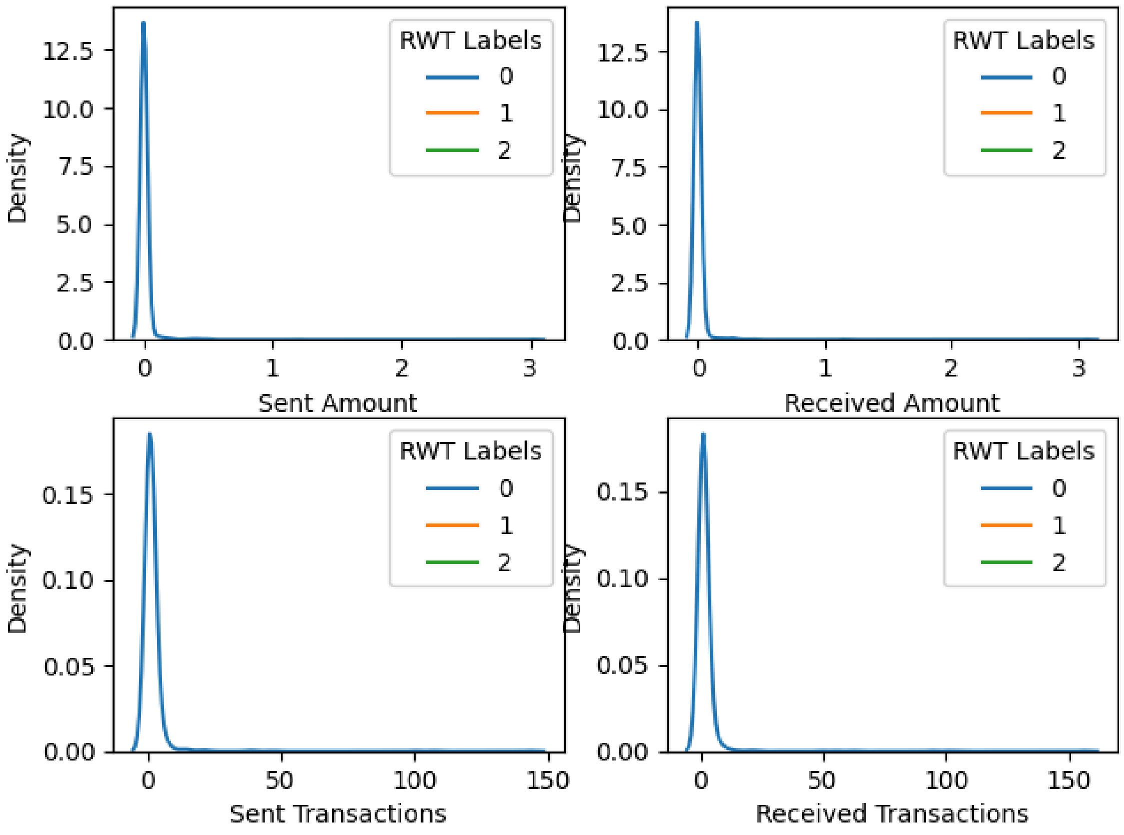
We see that each method has a different cluster around 0 sent amount and 0 received amount as well as one around 0 sent and received transactions. Thus, the clusters are likely better dispersed in this method as opposed to the Naive transform. In fact, this is further supported by the Davies-Bouldin Index, which was close to 1 on each of these methods and 5 on the Naive Graph transform.

On the other hand, we see a less performant result out of the Random Walk Transformation. The adjacency matrix becomes more sparse and the network graph follows, which likely leads to many different real clusters being interpreted as the same cluster.





As predicted, the results show a single cluster around 0 on all of our features, and while the other clusters may separate better, we can infer by the network graph that this is unlikely.



Additional Research

As mentioned previously, additional research should be more expansive in the number of wallets used, so wallets that transact with many other wallets with small amounts are considered. Furthermore, one may desire to use more recent blocks as the Ethereum blockchain is ever-changing. An extension of this study would be to determine wallet address types and analyze the clusters by how well they group types of wallet addresses, which may prove to be a better indication of a clustering technique's effectiveness. Sources of such information are difficult to find. Hence why we analyzed clusters with the features given.

Further research may also want to explore usage of the Random Walk Transformation, which did not perform well in this study. A different form of the Laplacian may be constructed as described in [3], which may yield better results as this method is more mathematically sound. There are also many other techniques in this survey that may be worth exploring for real-world network clustering, such as the transforms for bipartiate networks or transforms that don't retain directionality.

[1] Vlahavas George, Karasavvas Kostas, Vakali Athena (2024). Unsupervised clustering of bitcoin transactions. Financial Innovation. https://doi.org/10.1186/s40854-023-00525-y.
[2] Harry Sevi, Matthieu Jonckheere, Argyris Kalogeratos (2022). Generalized Spectral Clustering for Directed and Undirected Graphs. https://doi.org/10.48550/arXiv.2203.03221.

[3] Fragkiskos Malliarosa, Michalis Vazirgiannisa (2013). Clustering and Community Detection in Directed Networks: A Survey. arXiv:1308.0971v1.

[4] Venu Satuluri, Srinivasan Parthasarathy (2011). Symmetrizations for Clustering Directed Graphs. EDBT/ICDT '11: Proceedings of the 14th International Conference on Extending Database Technology.

[5] https://docs.chainstack.com/reference/base-trace-block.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html.

[8] Project code repository at https://github.com/noahcasey21/ethereum_spectral_clustering.