Draft version December 16, 2017
Typeset using LaTeX default style in AASTeX61

CHARACTERIZING THE ETHEREUM ADDRESS SPACE

James Payette,[1] Samuel Schwager,[2] and Joseph Murphy[3]

[1] *Department of Computer Science, Stanford University, Stanford, CA 94305, USA*
[2] *Department of Mathematical and Computational Science, Stanford University*
[3] *Department of Physics, Stanford University*

ABSTRACT

A decisive clustering of an inherently anonymous blockchain ecosystem would allow traits of specific users and, more broadly, overarching user groups to be inferred from publicly available blockchain data. Due to its built-in programming language, the Ethereum blockchain acts as a base layer upon which arbitrary smart contracts and decentralized applications can be developed. As such, we postulate that the Ethereum blockchain's formidable functionality and extensibility provide an exceptionally rich set of data compared to other popular blockchain ecosystems and that this data can be used to achieve an informed clustering of the Ethereum address space. Utilizing the k-means clustering algorithm in conjunction with Calinski-Harabasz scoring, we propose a segmentation of the Ethereum address space into four distinct behavior groups, which we herein discuss and evaluate both quantitatively and qualitatively.

Keywords: unsupervised learning, clustering, Ethereum, blockchain, cryptocurrency, smart contracts, decentralized applications

1. INTRODUCTION

Since the advent of Bitcoin, awareness and excitement around cryptocurrencies and the underlying blockchain technology that enables them have increased exponentially. Fundamentally, cryptocurrencies provide anonymity in that users operate via an address or set of addresses devoid of any personal information. However, also fundamental to the technology is the fact that blockchain data is completely publicly available and could therefore theoretically be used to successfully characterize or even identify users, resulting in considerable security implications (see Monaco (2015)) and other potential consequences and benefits. As such, we sought to gather a comprehensive dataset of Ethereum addresses and their associated metadata upon which we could apply cluster analysis to then divvy said addresses into behavior groups sharing similar attributes.

2. RELATED WORK

Several attempts have been made to identify addresses based on transactions from the Bitcoin blockchain e.g. Meiklejohn et al. (2013), Neudecker & Hartenstein (2017), Poikonen (2014); however, to our knowledge, this is the first such project focused on the Ethereum address space. Similar to our approach, other projects utilize several clustering methods, but k-means seems to be the primary algorithm employed due to its versatility and scalability with large data sets.

The main quantitative obstacle we had to navigate was making an educated estimate of the optimal number of clusters to use for learning, as this would inform our qualitative analysis by partitioning the address space into a discrete set of behavior groups. The issue of determining the optimal number of clusters, however, is not always a well-defined problem. Kodinariya & Makwana (2013) review six different evaluation techniques that range from quantitative to heuristic, including silhouette scoring–a measure of inter- and intra-cluster variance–and the so-called "elbow method", which attempts to estimate where the returns of adding additional clusters begin to diminish. Tibshirani et al. (2001) attempts to formalize the elbow method in a quantitative framework via the gap statistic. Our analysis has drawn from these works as they have guided our choice of clustering algorithms and evaluation metrics. Perhaps the most important upshot in examining related works was determining that the clustering and qualitative analysis of the address space is a problem open to experimentation and interpretation.

3. DATA SET AND FEATURES

Despite the public nature of the Ethereum blockchain data, gathering a significant dataset proved to be one of the most formidable challenges we faced throughout the course of our project. We relied on the Etherscan.io API for all of our data gathering efforts, and as such were subject to the constraint of a maximum of 5 requests per second. We carefully constructed Python scripts that made hundreds of thousands of requests to the API, handling all possible edge cases and failure scenarios. Ultimately, we were able to gather a dataset consisting of 250,000 addressess along with their respective Ethereum balances and full transaction histories. Please find our data collection scripts included in our code submission.

Using this data, we created a design matrix with each row corresponding to one of the 250,000 addresses and the columns corresponding to 34 different features, some of which are included in table 1. Due to the size of our data set, we had to make various algorithmic decisions in order to approach our analysis pragmatically.

4. METHODS

We experimented primarily with three clustering algorithms: k-means clustering, hierarchical or agglomerative clustering, and Birch clustering. In order to test the efficacy of our clusterings we used unsupervised evaluation metrics including Calinkski Harabaz scoring as well as heuristic evaluation metrics such as the so-called "elbow method" as described in Bholowalia & Kumar (2014), for example. Ultimately, we found that k-means, in many respects the simplest of the three clustering algorithms, combined with Calinkski Harabaz scoring to be the best method of analysis.

Given a client-specified number of clusters, K, the k-means algorithm divides the data into K clusters, generally unequal in size, with the objective of minimizing the inertia, or the sum of the squared distance between each cluster element and its cluster centroid. Our analysis utilized the k-means implementation from Pedregosa et al. (2011) [i.e. Scikit-Learn], which had the advantage of being very computationally efficient–a necessary condition given the size of our data set. While k-means was effective, the algorithm is *very* sensitive to data outliers. We discuss how this issue was addressed in section 5.

Given that the problem is unsupervised, evaluation metrics tend to report how "well" the data has been clustered. That is, unsupervised evaluation metrics measure intra- and inter-cluster variance to determine how effectively the

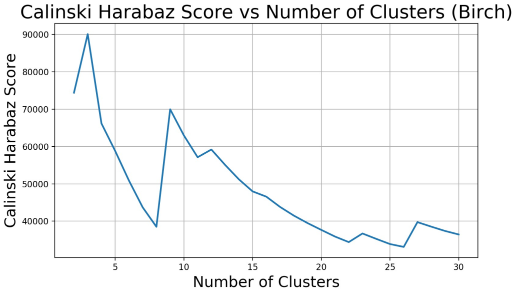

Figure 1. Calinkski Harabaz score versus number of clusters using the Birch clustering algorithm. Figure shown for comparison and corroboration of k-means result (Figure 2). Note, Birch clustering was not pursued because of its poor scalability.

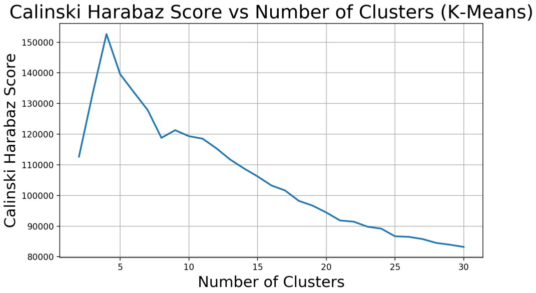

Figure 2. Calinski Harabaz score versus number of clusters using k-means clustering. Note the maximum here corresponds with the "elbow" of the Birch clustering plot, giving us confidence that an optimal number of clusters is about 4.

groupings upon which the clustering algorithm has converged segment the data. The Calinkski Harabaz score is a ratio of the inter-cluster dispersion mean and the intra-cluster dispersion mean. As such, a higher score represents a much more distinctive clustering of the data. The Calinski Harabaz score was employed over other unsupervised metrics such as a Silhouette score (where a Silhouette score of 1 indicates that a data point is perfectly classified by its cluster centroid and 0 indicates the data point has no preference between any of the clusters) because calculating the Calinski score is much more computationally efficient. For example, attempting to run Silhouette analysis on our entire data set of 250,000 addresses caused our machines to kill the task due to a lack of memory. In this sense, the Calinski Harabaz score is an informative yet practical option for determining cluster effectiveness.

5. EXPERIMENTS, RESULTS, AND DISCUSSION

We began our experimentation with a dataset of the top 10,000 Ethereum addresses, upon which we ran the k-means clustering and hierarchical or agglomerative clustering algorithms.

We then moved our attention the extensive dataset of 250,000 addresses that we gathered via the Etherscan.io API. We experienced the most success with the k-means and Birch clustering algorithms on this larger dataset, and specifically had a breakthrough with the k-means algorithm after performing a small amount of preprocessing to remove a small group of outliers that we realized had been distorting our cluster centroids. By calculating the sample mean and variance vectors for our 250,000-address dataset, we only allowed points satisfying the following criterion to remain post-preprocessing:

$$\|\bar{X} - X\| < 0.75\|S\| \tag{1}$$

Note that above, $\bar{X}$ represents the sample mean of a cluster k, X represents an arbitrary point in k, and S is a vector such that the ith component of S is the standard deviation of the ith feature over all points in k. Additionally, note that the choice of 0.75 was not arbitrary, but rather the optimal variance threshold we discovered via experimentation (Figure 3).

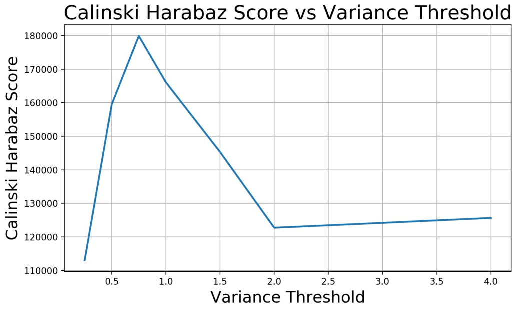

Figure 3. Calinski Harabaz Score versus variance threshold hyperparameter. Maximizing the score over our data set corresponded to choosing a cutoff variance threshold of 0.75.

Table 1. Features of interest and their intra-cluster sample means for each of the four clusters. .

Feature	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Total Ether	113.8	11.4	10.2	51.3
Total Number of Transactions	31.3	30.0	11.5	30.9
Number of Transactions (Incoming)	26.6	5.7	5.9	16.6
Number of Transactions (Outgoing)	4.7	24.3	5.6	14.3
Transactions per Month (Incoming)	5.4	1.4	1.4	0.89
Transactions per Month (Outgoing)	0.92	6.4	1.4	0.79
Ether Movement per Month (Incoming)	96.7	48.0	42.3	42.5
Ether Movement per Month (Outgoing)	74.4	50.3	41.0	44.7
USD Movement per Month (Incoming)	23,986.9	12,837.0	11,041.9	-
USD Movement per Month (Outgoing)	18,928.3	13,716.7	10,902.5	1679.6
Average Ether Transaction (Incoming)	43.9	76.4	55.1	153.8
Average Ether Transaction (Outgoing)	83.5	33.7	42.0	145.2
Average USD Transaction (Incoming)	7143.1	16,470.4	11,247.9	2030.5
Average USD Transaction (Outgoing)	16,670.2	7512.3	8665.3	3124.0

Note—For each of the sample means of each feature for each cluster, the associated sample variance is a considerable number of orders of magnitude smaller, so we do not report the sample variances here. Also, note that we do not report the USD Movement per Month for Cluster 4, as the result seemed suspect. Thus, we threw out the value, given that movement of total Ether contains similar information for our analytical purposes.

The table above provides a comparison between the four clusters that we created using k-means. First, we note that cluster 1 is the “wealthiest” among the four clusters. Interestingly, the addresses in cluster 1 appear to generally be on the receiving end of the transactions in which they are involved; however, we notice that the average magnitude of an incoming transaction for addresses in cluster 1 is significantly larger than that of an outgoing transaction.

Next, we notice that clusters 2 and 3 appear similar at first glance, but differ significantly with respect to outgoing transactions and general transaction magnitude. Cluster 2 appears to have considerably more outgoing transaction activity than cluster 3, and as a result cluster 2 moves significantly more value per month than does cluster 3. We also note that the average incoming transaction magnitude for cluster 2 is much larger than that of cluster 3; however, even though cluster 2 appears much more active with respect to outgoing transaction activity, the average magnitude of an outgoing transaction for an address in cluster 3 is over 1000 USD greater than that of an address in cluster 2.

Finally, we see that addresses in cluster 4 generally possess an intermediate amount of Ether and have nearly an equal amount of incoming and outgoing transactions. Additionally, we see that an average outgoing transaction for an address in this cluster is over 1000 USD greater in value than a corresponding incoming transaction. Finally, it is important to note that on a per month basis, cluster 4 seems to be the least active given that it has the lowest quantity of incoming and outgoing transactions per month among all four of the clusters.

6. CONCLUSIONS AND FUTURE WORK

Using an informed estimate of the proper number of clusters with which to group the Ethereum address space, we are able to succesfully report quantitative characteristics of each group and make several qualitative inferences about the clusters’ behavior traits. A major deliverable of the project is our data set of 250,000 unique addresses and corresponding feature vectors, which we manually scraped from the Ethereum blockchain. While we explored several clustering options (e.g. agglomerative and Birch), we use k-means as our primary clustering algorithm due to its scalability in light of our very large data set. In terms of evaluation metrics, we use Calinkski Harabaz scoring to measure the the ratio of the inter- and intra-cluster variance of our groupings. We also use variance reduction to remove outliers that inhibit our k-means analysis and maximize our Calinski Harabaz score.

Looking forward, in order to assess Ethereum's development over time, we find the relative amount of "gas" spent on smart contract execution versus normal transactions will be crucial in determining whether Ethereum is being utilized as a distributed supercomputer as intended by its creators or a vehicle for financial speculation.

Finally, now that we have determined an appropriate clustering of the Ethereum address space, we believe that generative models could be of interest in creating automated addresses replicating the behavior of users from the four groups we have identified. Such models could influence transaction activity, Ethereum and ERC20 coin prices, and the development of the Ethereum ecosystem as a whole.

7. CONTRIBUTIONS

7.1. *James (Jack) Payette: jpayette@stanford.edu*

Jack was instrumental in our data acquisition and analysis processes. With the help of Sam, Jack was able to employ use Etherscan.io to successfully scrape our data from the Ethereum blockchain. Jack also helped integrate various learning algorithms, implemented for us in Scikit-Learn, to run on our data set. Jack laid much of code-based foundation on which our project stands.

7.2. *Samuel (Sam) Schwager: sams95@stanford*

Sam originally posed the project idea of analayzing the Ethereum address space using publically available blockchain data. Sam's background knowledge of how cryptocurrencies operate was crucial for determining the motivation and scope of the project, as well as informing qualitative analysis of the clusters. Sam also helped to implement data collection scripts and explored alternative clustering algorithm options from Scikit-Learn's library. Sam was also an integral part of producing the final report.

7.3. *Joseph (Joey) Murphy: murphyjm@stanford.edu*

Joey used his knowledge of Matplotlib to create visualizations to help interpret our results and was an integral part of producing the final poster. Joey also contributed heavily to determining effective evaluation metrics with which to measure the success of our unsupervised clustering and used Scikit-Learn libraries to implement Silhouette and Calinski analysis. Joey made significant contributions to the final report.

REFERENCES

Bholowalia, P., & Kumar, A. 2014, International Journal of Computer Applications, 105

Kodinariya, T. M., & Makwana, P. R. 2013, International Journal, 1, 90

Meiklejohn, S., Pomarole, M., Jordan, G., et al. 2013, in Proceedings of the 2013 conference on Internet measurement conference, ACM, 127

Monaco, J. V. 2015, in SPIE Defense+ Security, International Society for Optics and Photonics, 945704

Neudecker, T., & Hartenstein, H. 2017, in International Conference on Financial Cryptography and Data Security, Springer, 155

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Poikonen, S. 2014

Tibshirani, R., Walther, G., & Hastie, T. 2001, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63, 411