Unsupervised Clustering of Bitcoin Transaction Data

AMSC 663/664 Project

External Advisor: Dr. Armao

By: Stefan Poikonen

Bitcoin: A Brief History

- Bitcoin is a decentralized cryptocurrency used for digital transactions
- Based on a 2008 paper by Satoshi Nakamoto
- The Bitcoin Network was first implemented January 1st, 2009
- In early 2014 market capitalization of Bitcoin surpassed $8 billion
- Some merchants who accept Bitcoin: Amazon.com, Overstock.com, TigerDirect.com, OKCupid.com, Expedia.com
- Silk Road, the deep web market, heavily utilized Bitcoin

Bitcoin: How It Works

- User downloads a Bitcoin client, which generates a private key
- The associated public key (public address) is easily computed
- The private key acts as a form of digital signature
- A user “signs” a purchase by applying their private key to a transaction, which includes the recipient’s public address
- The resulting signed transaction is broadcast to the Bitcoin network
- Transaction blocks are verified by “miners”, who are rewarded in newly minted Bitcoin
- A public ledger of all past transactions (the “block chain”) is maintained

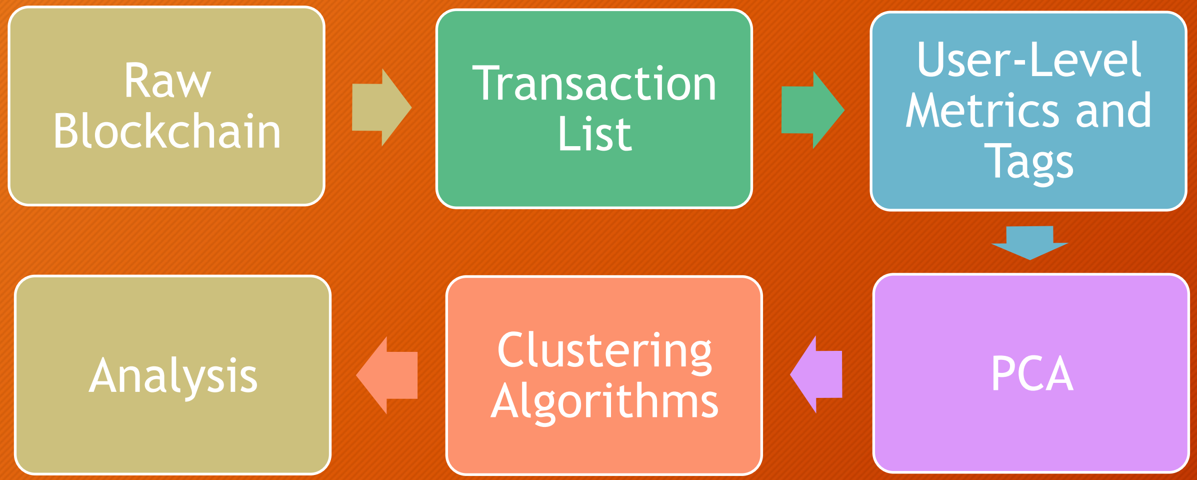
Previous Bitcoin Research

- Deanonymization
- Economics of Bitcoin
- Cryptocurrencies as tax havens
- Exchange rate variability
- Some global metrics: total trade volume, # of total transactions, distribution of transaction sizes, etc.
- Cryptographic Security of Bitcoin System

Project Goal

- Categorize Bitcoin transactions
- Without a training set, this is accomplished via unsupervised learning of transaction data
- Form clusters
- Evaluate the efficacy of the clusters
- List potential anomalous transactions

High Level Flow of Project

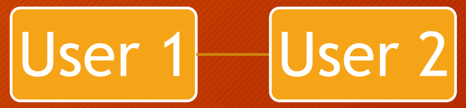


The Data

- The public ledger, or “block chain” is available to download
- Currently 22-23GBs
- Reid and Harrigan describe transformations to the raw block chain to transaction line tables
- Around 50 million transactions
- Each transaction line contains the following data elements:
  - Source ID
  - Destination ID
  - Timestamp
  - Amount

How To Maximize Information Extraction?

- We may note each transaction line only has four data elements
- Source ID and Destination ID may serve as indices
- We compile metrics (across all past transactions) for each user
- Examples of user-level metrics:
  - Average transaction amount
  - Timestamp of first transaction (i.e. when user joined Bitcoin network)
  - Largest transaction
  - Peak number of transactions in one month period
  - Value of BTC still held by user
  - Page Rank of user in the Bitcoin network



Tags: Additional Data

- Blockchain.info maintains a database of “tagged” public addresses
- Tags associate a public address with an entity, cause, website, etc.
- These tags have been categorized: political, charity, hacking, etc.
- We can compute the number of times a given user has been adjacent to certain categories, or other measures of a users closeness to a particular category of tag

1Q4G4ZJ1AN1aHkC9YnPQGWYEAxJrW62rJL	Wikileaks	http://wikileaks-donation.weebly.com/
1Dorian4RoXcnBv9hnQ4Y2C1an6NJ4UrjX	Dorian Nakamoto fundraiser	http://www.reddit.com/r/Bitcoin/comments/1ztjmg/andreas_im_fundra...
1DzBEBqzrNsRg8oeRbGWNUr4V2VSjdS7iQ	Wheelchair Fund	http://www.reddit.com/user/IamAlso_u_grahvity/submitted
1436j9Kw2veuQbY1FzPd4VFGZzejLEBjhb	FileZilla Donations	https://filezilla-project.org/

Augmented Data Line

- Each augmented transaction line may contain the following elements (and others):
  - Source ID, Destination ID, Timestamp, Amount
  - Source ID’s:
    - Timestamp of first transaction (i.e. when user joined Bitcoin network)
    - Average Transaction
    - Value of BTC still held by user
    - Page Rank of user in the Bitcoin network
  - Destination ID’s:
    - Timestamp of first transaction (i.e. when user joined Bitcoin network)
    - Average Transaction
    - Value of BTC still held by user
    - Page Rank of user in the Bitcoin network
  - Source ID:
    - # of times adjacent to “charity” tag
    - # of times adjacent to “computer parts” tag
  - Destination ID:
    - # of times adjacent to “charity” tag
    - # of times adjacent to “computer parts” tag

Clustering And Scale?

- Each column of data has different scale
- Measuring “distance” between augmented transaction lines in clustering is dependent on this scaling
- It would be possible to learn a metric with “good” scaling, if we had a training set.
- Without a training set we:
  - Normalize data: means → 0 and variances → 1
  - Log transformations to enhance normality of some data columns
  - Perform a Principal Component Analysis

Clustering Algorithms

K-Means

- Suppose we want to create k clusters.
- 1) Initialize *k* centroids. (Random selection from n data vectors is most common.)
- 2) Loop through the remaining *n-k* transactions, assigning to the nearest centroid.
- 3) Recompute centroids of updated clusters.
- Repeat steps 2) and 3) until transactions no longer switch between clusters

Fuzzy C-Means

- Similar to K-means, but assignment to clusters is expressed with level of certainty:

Centroid computation:

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^m * x_i}{\sum_{i=1}^{N} u_{ij}^m}.$$

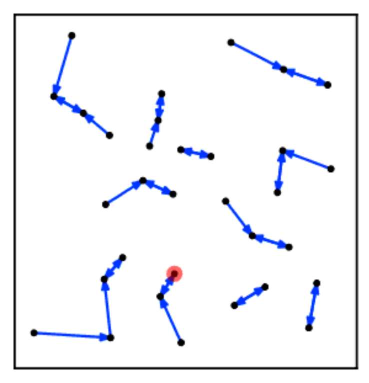
Updated membership value of x_i in cluster j:

$$u_{ij} = \left(\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}} \right)^{-1}$$

More Clustering Algorithms

- CURE Clustering Algorithm
  - Form of agglomerative hierarchical clustering
  1) Choose well-scattered set of points (different sampling methods proposed)
  2) Shrink towards means by multiplying by $0<\gamma<1$
     - Let these points be centroids of clusters
  3) Assign remaining points to nearest cluster centroid
  4) Merge two “most similar” clusters
  5) Recompute merged clusters centroids
  6) Goto 2) and repeat
  - $O(n^2 \log(n))$, but can use sampling, which essentially reduces n.

More Clustering Algorithms



- Nearest Neighbor Chain Algorithm
  - Initiate n-clusters, push clusters onto stack
  - Find nearest neighboring cluster.
  - If cluster already in stack, merge.
  - Else nearest neighbor goes to top of stack.
  - Nearest cluster may be defined by “single-linkage”, “full-linkage”, “Ward’s Method”, “centroid distance”, etc.
  - Computationally tenable for n = 50,000,000?

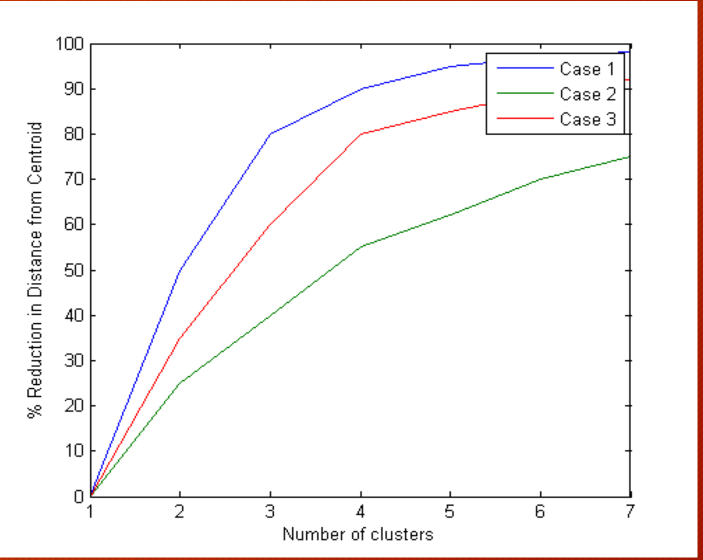
Validation

- As the number of clusters increase we expect:
  - Decrease in distance to cluster centroids
  - Greater compactness within clusters
  - Receiver Operator Curve/Area Under Curve
- Predictive Validation?
  - This would require an external dataset of categorized or synthetic transactions

$$(\Sigma_{C_j}(\frac{max(D(R_i, R_k))}{min(D(C_{jc}, C_{mc}))}))^{-1}$$

with $R_i, R_k \in C_j$ and C_{ac} represents the centroid of cluster a.

$$\frac{Avg(D(R_i, Centroid(Cluster(R_i)))^2)}{Avg(D(R_i, Centroid(AllData))^2)} \forall R_i \in R$$



Anomalous Transactions?

- Post-clustering find data points who distance from cluster centroids is greatest.
- These potentially represent anomalous transactions.

Implementation

- Code will be implemented primarily in C/C++
- Run a desktop with an Intel i5-3570K CPU
- 16GB of DDR3 RAM
- Python parsing of the block chain as convenient.
- If time permits, CUDA and/or OpenMP might be used for CPU and/or GPU parallelization respectively for key computationally intensive segments.

Time Line

- Now-November 15: Data transformation, parsing, user-metric computation, tag-metrics, etc.
- November 15-December 15: PCA and K-means clustering
- February 1-March 31: Fuzzy C-means clustering, CURE clustering, other clustering algorithms (time permitting)
- April 1 – April 25: Analysis of cluster quality, parallelization (time permitting)
- April 25 – May 15: Paper and presentation

- Milestones correspond with the completion of each bullet above.

Deliverables

- C++/Python code for transforming data to transaction line table
- C++ code for computing user-level metrics
- C++ code for computing tag-related metrics
- C++ code for normalizing data prior to PCA
- C++ code for computing K-means clusters
- C++ code for computing Fuzzy C-means clusters
- C++ code for other clustering (time permitting)
- Evaluation metrics from clustering with different numbers of clusters across different clustering algorithms
- First-Semester Progress Report
- Mid-Year Status Report
- Final Reports
- Weekly Reports

The End

- Questions?

Bibliography

Aiken, Michael. "Future Funds: The Latest on Bitcoin and Cryptocurrency." *Diplomatic Courier*. Diplomatic Courier, 4 Sept. 2014. Web. 02 Oct. 2014.
Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." *Consulted* 1.2012 (2008): 28.

Biryukov, Alex, Ivan Pustogarov, and R. Weinmann. "Trawling for tor hidden services: Detection, measurement, deanonymization." *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 2013.

Moore, Tyler, and Nicolas Christin. "Beware the middleman: Empirical analysis of Bitcoin-exchange risk." *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2013. 25-33.

Raskutti, Bhavani, and Christopher Leckie. "An Evaluation of Criteria for Measuring the Quality of Clusters." *Telstra Research Laboratories* (1999): Web. <http://ww2.cs.mu.oz.au/~caleckie/ijcai99.pdf>.

Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "CURE: An Efficient Clustering Algorithm for Large Databases." (1998): Web. <http://www.cs.sfu.ca/CourseCentral/459/han/papers/guha98.pdf>.

Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "CURE: an efficient clustering algorithm for large databases." *ACM SIGMOD Record*. Vol. 27. No. 2. ACM, 1998.

Ding, Chris, and Xiaofeng He. "K-means clustering via principal component analysis." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.