Detection and Visualization of Bitcoin Anomaly

1st Suibin Sun
Shanghai Jiao Tong Univ.
sun1998@sjtu.edu.cn

***Abstract*—As one of the most popular cryptocurrencies, Bitcoin is under many studies all over the world. While the number of Bitcoin users is rapidly increasing, the anomalies and attacks of Bitcoin transactions are also happening more and more. Members of Bitcoin network want to detect anomalies as soon as possible to prevent them from harming the network's community and ecosystem. Many machine learning techniques have been proposed to deal with this problem for example k-means clustering, which is a famous and classic method of clustering data points into different classes. On the other hand, as the continuous integrity of Bitcoin network, we need to detect anomalies in real time. Here we use BPR(Bayesian Probit Regression) algorithm to realize an online learning system. In the end, we build a visualization web app to observe real time statistic and detection of recent Bitcoin transactions.**

***Index Terms*—Bitcoin, Anomaly Detection, Online Learning, Visualization**

I. Introduction

Bitcoin is a cryptocurrency and worldwide payment system. It is the first decentralized digital currency, as the system works without a central bank or single administrator. The system works as a peer-to-peer network, a network in which transactions take place between users directly, without an intermediary. These transactions are verified by network nodes through the use of cryptography and recorded in a public distributed ledger called a blockchain. Bitcoin was invented by an unknown person or group of people under the name Satoshi Nakamoto and released as open-source software in 2009 [1].

With the increase of users and the expansion of the market, there are more and more abnormal transactions in Bitcoin. At present, there is no effective and intuitive detection method. Due to the lack of number and accuracy of ground-truth, unsupervised learning algorithms based on machine learning can be used to detect anomalous behavior. Here we use two relevant methods to detect anomalies in Bitcoin transaction network. In the transaction network, each transaction is a node and the Bitcoin flows between transactions are edges.

As Bitcoin transactions continue, the transaction network is dynamically updated. So we need an online learning strategy to update our model and detect real-time transactions. Here we use perception algorithm to perform a process of online learning.

In order to make the result as intuitive as possible for human operators to derive insights about the system, we build a visualization of features such as average in-degree, average out-degree of a node in transaction network.

II. Methods

Firstly in this section we will show how we collect the data we need. Then we are going to introduce the methods we use in the project. We will elaborate the unsupervised learning methods we used to distinguish whether a transaction is anomaly or not in part II-B. Then we describe the online learning method we use in such question in part II-C. The last method we claimed here is the web app we build for visualization in part II-D.

A. Data Collection

We parse the Bitcoin transaction data from the website *blockchain.info* [3]. We use almost two weeks to download part the blocks[1] and extract all transactions(about 200 billion transactions until May 10th, 2018) packaged in those blocks. This is such a big data problem that only part features of the transactions possess 1GB storage. The cost of processing data is too high and this is a waste of my work time. For calculation efficiency without loss of generality, I only choose data in 100000 blocks here which stands from block height 400000 to 499999. This segment of blockchain is created since Feb. 26th 2016 and ended in Dec. 19th 2017.

B. Unsupervised learning

1) Feature extraction: Now we get all transactions since the first block was created. The information we need is their in-degree, out-degree and the amount of each transaction. In-degree means he number of users who paid in this transaction while the out-degree stands for the number of users who gained Bitcoin in this transaction. And the amount of this transaction is the total amount in this transaction. We form these three data as three features of one data point.

An easy fact in Bitcoin network is that the miners would get award if he created this new block. This award is also a transaction recorded in each block. In order to avoid disturbing to the result of clustering, we remove such special transactions which has 0 in-degree and fixed amount currently 12.5 Bitcoin.

Another characteristic in this problem is that, values of all three features scale from 1 to a big number such as 100000. So we need a pretreatment to normalize these numbers. Here we take log of feature values as their available features.

[1] 222138 blocks(height 400000 to 522137) out of totally 522137 blocks until May 18th, 2018

2) k-means: It is a classic and useful method proposed by MacQueen J. [2] in 1967. Now we have to use k-means method to well cluster these transactions into normal ones and anomalous ones.

Suppose we have m points $(x_1, x_2, ..., x_m)$ where $x_i \in \mathbb{R}^3$ for each $i = 1, 2, ..., m$. We seek to partition these m points into k clusters $S = (S_1, S_2, ..., S_k)$ to solve

$$\min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

where μ_i is the centroid point of the cluster S_i for each $i = 1, 2, ..., k$.

Picking value of k during performing k-means clustering method is always a question. Here we use Calinski Harabaz score [4] to evaluate which k is better for us. It works to find a k to maximize

$$s(k) = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1} \tag{1}$$

where m is the number of samples, B_k, W_k are the covariance matrix of two different clusters and two data inner cluster respectively. The score is greater, this k is better for us.

3) Detection principle: Since this is an unsupervised learning problem, our clustering of these points still can't be used to distinguish which classes are normal and which classes are abnormal.

Now we make some assumption to derive the question into a labeled question. Assume the data set $(x_1, ..., x_m)$ drawn from Multivariate Gaussian Distribution

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)) \tag{2}$$

while the value of μ and Σ can be estimated by

$$\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3}$$

and

$$\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)(x_i - \mu)^T \tag{4}$$

Then we will judge a data point x_i as an anomaly if $p(x_i; \hat{\mu}, \hat{\Sigma}) < \epsilon$ for some chosen threshold ϵ. For example, if we assign $\epsilon = 0.05$, that means we decide that a transaction with a happening probability less than 5% is an anomaly.

C. Online Learning

Here we use BPR [5] to perform the procedure of online learning. Suppose the weights of features w meet Independent Gaussian distribution.

$$p(w) = N(w|\mu, \Sigma) \tag{5}$$

while Y is a one-dimension array which equals to the inner product of w and features x. Add it with a β^2 disturb:

$$p(y|w) = N(y|x^T w, \beta^2) \tag{6}$$

$$p(y|w) = N(y|x^T \mu, x^T \Sigma x + \beta^2) \tag{7}$$

Since we can observe the label y of a new data Y, we can use KL distance to estimate the distribution of y and then the posterior:

$$p(y|Y) = N(y|\tilde{m}, \tilde{v}^2) \tag{8}$$

$$\tilde{m} = x^T \mu + Y v(Y \frac{x^T \mu}{\sqrt{x^T \Sigma x + \beta^2}}) \tag{9}$$

$$\tilde{v}^2 = (x^T \Sigma x + \beta^2)(1 - w(Y \frac{x^T \mu}{\sqrt{x^T \Sigma x + \beta^2}})) \tag{10}$$

With the estimated distribution of y, we can calculate the posterior:

$$p(w|y) \propto p(y|w)p(w) \tag{11}$$

and

$$p(w_d|y) = N(w_d|\tilde{\mu}_d, \tilde{\sigma}_d) \tag{12}$$

$$\tilde{\mu}_d = \mu_d + Y x_{i,d} \frac{\sigma_d^2}{\sqrt{x^T \Sigma x + \beta^2}} v(Y \frac{x^T \mu}{\sqrt{x^T \Sigma x + \beta^2}}) \tag{13}$$

$$\tilde{\sigma}_d = \sigma_d [1 - x_{i,d} \frac{\sigma_d^2}{x^T \Sigma x + \beta^2} w(Y \frac{x^T \mu}{\sqrt{x^T \Sigma x + \beta^2}})] \tag{14}$$

So finally our update algorithm is:

(1) initialize $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, ..., \mu_D, \sigma_D^2$,

(2) input a new data y with label Y, *for d=1,...,D*: update μ_d and σ_d by formula 13 and 14.

D. Visualization

In order to show the result more intuitively, we build a simple web app that plot the realtime data of Bitcoin network including average transaction fees recorded in a block, the difficulty coefficient of mining, the number of transactions per day. We get the data also from *blockchain.info*. Some anomalies can be recognized from these charts.

III. Result and Evaluation

A. Unsupervised Learning

Since we choose different k ranges from 2 to 14, we get the score follows Figure 1. As a result, we choose 5 as our k in the following k-means methods.

After a long time running of k-means algorithm, we got a figure clustering points into 5, which is shown in Figure 2.

Now, we run the detection method to distinguish anomalies. The detected anomalies in Figure 3 seem to appear at the border of the plot, which indicated that the abnormal activities are usually extreme. Note that although the dimension of data points is 3, we use the first two dimension to plot the figures since 3D plotting is too time-wasted and less intuitional.

B. Online Learning

We take the model trained by blocks height 400000 to 499999 as a available model. Then we feed the following blocks 500000 to 522137 into the model and continuously update the parameters. The label of newly added points are also shown in Figure 3 with blue and red points.

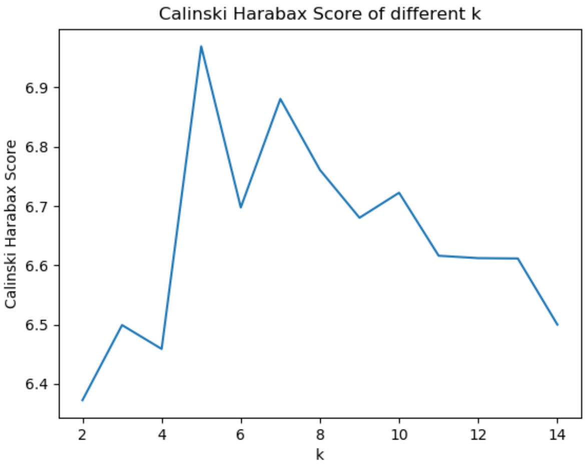


Fig. 1. Calinski Harabax Score over different k. When $k = 5$, the score is the greatest. So we choose $k = 5$ for following methods.

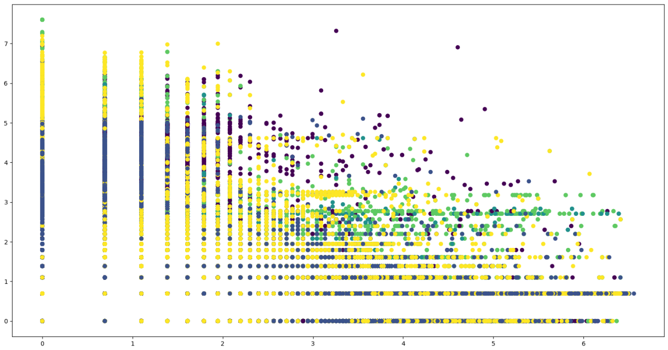


Fig. 2. 5 clusters shown in different colors

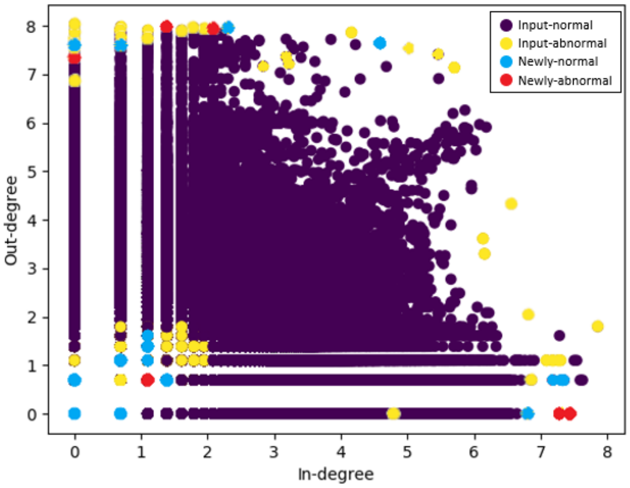
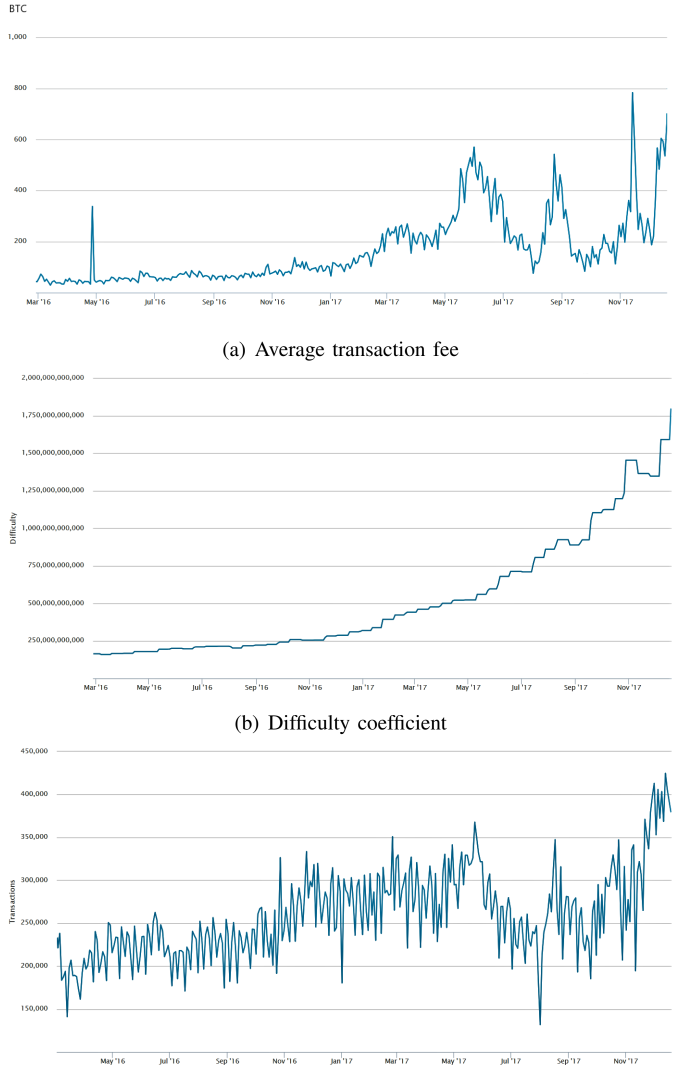


Fig. 3. 2D data points plotting.

C. Visualization

The charts described in part II-D are shown in the following Figures 4. Actually some abnormal points can be observed in average transaction fee chart.



(a) Average transaction fee

(b) Difficulty coefficient

(c) Number of transactions per day

Fig. 4. Visualization of Bitcoin network

Acknowledgment

Thanks to Teacher Fu. and TAs during this course and this project. Also thanks to the data sharing website *blockchain.info* which significantly decrease my work load.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[2] MacQueen J. Some methods for classification and analysis of multivariate observations[C]//Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967, 1(14): 281-297.

[3] Blockchain Block Explorer. (2017). Retrieved May 26, 2018, from: https://blockchain.info

[4] T. Calinski, J. Harabasz, ”A dendrite method for cluster analysis”, Comm. in Statistics, vol. 3, no. 1, pp. 1-27, 1974.

[5] W.R. Gilks, S. Richardson, David Spiegelhalter. (1995). *Markov Chain Monte Carlo in Practice* CRC Press