Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin

Alex Greaves, Benjamin Au

December 8, 2015

Abstract

Bitcoin is the world's leading cryptocurrency, allowing users to make transactions securely and anonymously over the Internet. In recent years, The Bitcoin the ecosystem has gained the attention of consumers, businesses, investors and speculators alike. While there has been significant research done to analyze the network topology of the Bitcoin network, limited research has been performed to analyze the network's influence on overall Bitcoin price. In this paper, we investigate the predictive power of blockchain network-based features on the future price of Bitcoin. As a result of blockchain-network-based feature engineering and machine learning optimization, we obtain up-down Bitcoin price movement classification accuracy of roughly 55%.

1 Introduction

Bitcoin is a cryptographic protocol and distributed network that enables users to, store, and transfer digital currency. The scheme was first implemented by Satoshi Nakamoto in January 2009. Based on the transparency and security of the Bitcoin protocol, the Bitcoin ecosystem has gained significant traction from consumers, businesses, speculators and investors.

In recent years, Bitcoin has established itself as the leading decentralized, cryptographic currency. In the process, the value of individual Bitcoin currency has skyrocketed in value, from cents to over 1000 USD at its peak, leading many to take to Bitcoin as a means of speculation. The popularity of Bitcoin, the digital nature of the currency itself, as well as the availability of high-dimensional network and pricing data make machine learning prediction of the Bitcoin price particularly interesting.

2 Related Work

Our work builds on prior research to leverage blockchain network features, as a basis to conduct supervised machine learning prediction on the price of Bitcoin.

Ron et. al [2] used the Union-Find algorithm to group accounts belonging to the same individual or entity. Analysis demonstrated that many attempts were made to obfuscate an entities' transaction histories, finding many long chains and oddly shaped clusters in the graph that could not have been produced organically. In addition, their research showed that while the net flow of the graph far exceeds the number of Bitcoins in circulation, most Bitcoins are in fact not in circulation and have not been moved or used since their mining. This finding corroborates the hypothesis that significant money laundering for anonymization purposes is in effect.

Shah et al. claimed to produce a successful Bitcoin price prediction strategy based on Bayesian regression with a latent source model [3]. This model, originally developed by Shah to predict trending topics in Twitter, obtained significant success in predicting Bitcoin price, claiming a 50-day 89% ROI with a Sharpe ratio of 4.10, using 10-second historical price and Bitcoin limit order book features.

Madan et al. used bitcoin blockchain network features, as well as seconds-level historical bitcoin price in historical time deltas of 30, 60 and 120 minutes to develop features for supervised learning. Leveraging random forests, SVM and binomial logistic regression classifiers, price deltas 10 minutes in the future were predicted, obtaining results of approximately 55% accuracy.

3 Methods

3.1 Data Collection

We use the Bitcoin transaction data available on the CS224W website, which contains every Bitcoin transaction made prior to April 7, 2013. All transactions are available on a public ledger, and this data set is a large text file containing a line for every transaction. Each line includes the transaction id, sender, recipient, value (in BTC), and a timestamp. Transactions involving multiple senders and multiple receivers are represented by multiple lines with the same transaction id.

This file comes with a matching file containing one line for each group of addresses that appear together in some transaction, hence belonging to the same user or entity. We used this file to transform our data set into a simplified one indexed by ”user” entity rather than address using the Union Find algorithm, based on the intuition that for any given transaction, only one entity could be the sender of that transaction, even if multiple sending accounts were used.

The original data set contains nearly 37 million transactions between roughly 6 million addresses. Once reduced, we obtained the same number of transactions between just over 3 million unique users. We represent the data in a directed graph, where each node is a user and each edge is a transaction (from sender to receiver). Bitcoin mining is represented in the data as a transaction from one user to itself, and is hence manifested in the graph as a self-loop. This representation of the data allows us to explore various properties of the graphs and also use them as features in price prediction. In addition, to perform price prediction we needed a complete historical listing of prices for Bitcoin. We acquired from `api.bitcoincharts.com` the entire histories of several Bitcoin exchanges, where each line in a given history contains the timestamp and the exchange rate in USD. From these transactions we computed the average price of Bitcoin across all transactions as the official Bitcoin price at 15 second intervals dating back to just after the first Bitcoin was mined.

3.2 Feature Extraction

We compiled several network-based features to develop our supervised machine learning algorithms. The feature families we selected using the following approach: Features were developed for time-frames of one-hour, one-day, one-week and one-month prior to prediction time. The goal is to predict the price of Bitcoin in USD one hour in advance. All features are computed using only information available an hour prior to the target prediction time, and for node features we used both in- and out-degree. The following features were extracted:

- current Bitcoin price
- net flow per hour
- number of transactions per hour
- mean transaction value
- median transaction value
- average node in- and out-degree
- median node degree
- alpha constant of power law
- total number of Bitcoin mined
- number of new addresses
- mean initial deposit amount among new addresses
- number of transactions performed by new addresses

Many of these features are somewhat correlated with the price of Bitcoin, though of those most are only loosely related. In Figure 1, for instance, we see the total number of mined Bitcoin plotted against the current price.

In addition, we identified three addresses that had by the far the greatest influence on the network, with over 10% of all Bitcoin ever sent passing through at least one of these addresses. Creatively, we labeled these as A, B, and C, and of these the largest transactor (by over a factor of 2) was A. We believe this address to be associated with Mt. Gox, which was the world's largest Bitcoin exchange prior to their 2014 collapse. From each of these nodes, for the hour prior to prediction, we collected the following features:

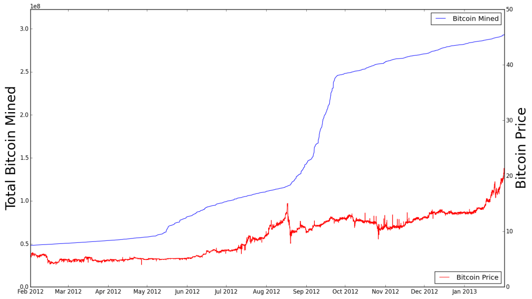


Figure 1: Here we see some correlation between the total number of Bitcoin Mined and Bitcoin Price over time

- total Bitcoin passing through
- net Bitcoin flow (received minus sent)
- number of transactions
- closeness centrality

In doing so, we hoped not only to extract important information for our classifiers and regressors, but also to gain a general idea of how the largest players in the Bitcoin market affect its valuation. We can gain some idea of the activity of the differing addresses by seeing their net balance, the total Bitcoin coming in minus that going out summed over a year long period. This information is summarized in Figure 2.

We collected data every hour from February 1, 2012 to February 1, 2013 to form our training set, and from February 1, 2013 to April 1, 2013 to form our test set. It was necessary to use data temporally after the training set data for testing in order to correctly simulate the real world scenario of predicting price based on past data.

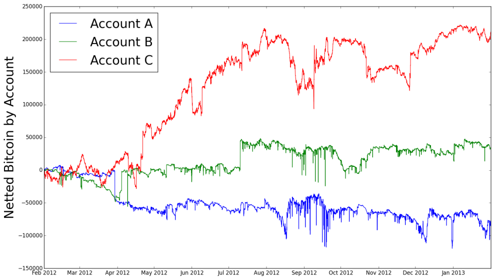


Figure 2: While account B maintains roughly a net zero account, account A loses Bitcoin over time and C gains Bitcoin

3.3 Feature Selection

Choosing which features to use is an important aspect of any regression or classification optimization. To prune features, we calculated the mutual information between each feature and the output, pruning all insignificant features from the model. Figure 3 gives the equation for I, the mutual information, where A is our feature data and B our output data.

$$I(A;B) = \sum_{b \in B} \sum_{a \in A} p(a,b) * log\left(\frac{p(a,b)}{p(a)p(b)}\right)$$

Figure 3: Equation for mutual information

As a general rule, the greater the mutual information, the more informative the feature. Many of the features we computed proved uninformative, and had mutual information scores barely above random chance. The features that proved the most informative and whose inclusion improved our results were:

- Current Bitcoin price
- Net Bitcoin flow for A, B, and C
- Closeness centrality for A, B, and C
- Mined bitcoin in the last hour
- Number of transactions among new addresses in the last hour
- Mean node degree in the last hour
- Net flow in the last hour

Interestingly, none of the features that looked farther back than the last hour were at all informative. Unsurprisingly, by far the most informative feature for price prediction was current price.

3.4 Network Algorithms

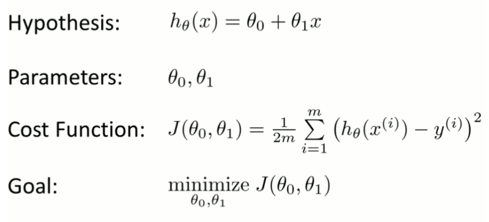
Union Find The Union Find algorithm is used to determine contiguous subsets within a network. This algorithm is applied as a preprocessing technique to relate multiple accounts related to a single owner entity. The algorithm is broken down recursively calling the Union and Find functions.

1. Union function: merges two disjoint subsets into the union of those subsets.

2. Find function: checks to see which other entities are part of the same subset as any of those already merged.

3.5 Learning Algorithms

Linear Regression Linear Regression (LR) is a predictive model that formulates a line of best fit between a scalar dependent variables and multiple explanatory variables. Linear fit occurs by minimizing the mean squared error between the predicted and actual output. Below we detail the hypothesis, parametrization (which can be extended in vectorized form), cost function, and goal of linear regression.



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Figure 4: Linear regression formulation

Logistic Regression Logistic Regression is a predictive regression model in which the dependent variable is categorical. In the simple case (as in our problem formulation) where there are only two catogories, Logistic Regression uses Maximum Likelihood Estimation to formulate the probabilities in which Logistic Regression will take on a particular class, with an iterative algorithm such as Newton's method used to obtain the fitted model. Logistic Regression is typically seen as a robust baseline method for classification. Below is the objective and MLE formulation for Logistic Regression, as well as the formulation of the logistic function.

$$\max_{\theta} \sum_{i=1}^{n} \log p(y_i | x_i, \theta).$$

$$p(y = 1|x) = \sigma(\theta^\top x) = \frac{1}{1 + \exp(-\theta^\top x)}.$$

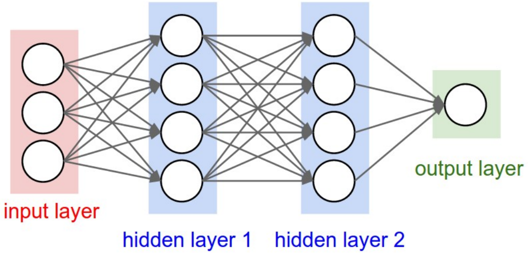
Figure 5: Logistic regression MLE and logit function

Support Vector Machine Support Vector Machine (SVM) is a discriminative classifier that generates a separating hyperplane. Error tolerance budget is included to make separating hyperplane robust in case of inseparable class data. Linear decision boundaries are augmented to more complex boundary shape through kernel implementation (e.g. polynomial, Gaussian and radial kernel). SVM obtains decision boundary by creating a margin which maximizes the functional and geometric margins between classes. SVMs have received attention for the impressive performance in classification. SVM can also be augmented for regression problems as Support Vector Regression (SVR) optimizing response variable distance from the decision boundary. Below we provide formulation of the optimization and constraint model for SVM.

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$

Figure 6: Support vector machine objective and constraints

Neural Network Neural Networks are a family of learning methods inspired by biological neural networks by modeling a system of interconnected neurons, which are tuned based on iterative learning. Feedforward neural networks connect a multi-dimensional input into one or more hidden layers of neurons before predicting an output. Dropout is used to prevent overfitting of the model. Hidden layers are modeled by affine transformation and final layers are modeled by softmax. In addition, a non-linearity (such as the tanh function) is often used after each layer. Below is a visualization of a Feed-forward neural network with two hidden layers, similar to the one we implemented.



4 Results

We leveraged several regression models to predict the price of Bitcoin one hour into the future, using our improved feature set. To set a baseline prediction, we chose a naive approach: we took the average percent price increase per hour (∼1%) and applied it to the current price to predict the price in an hour. We evaluated our performance by using mean squared error (MSE). Our results are summarized as follows:

Regression Model	MSE
Baseline	2.02
Linear Regression	1.94
SVM Regression	1.98

Table 1: Regression results

In addition to the above regression models, we also attempted several tree-based algorithms as well as K-Nearest Neighbors, all of which performed worse than baseline. With Linear and SVM based regression we were able to predict price change better than our baseline, as evidenced in Figure 8, where we plot the cumulative mean squared error as a function of time. By the end of the test period, the baseline prediction has accrued noticeably greater total error than our predictions.

In addition to a regression-based approach, we formulated the problem as a classification task, wherein we predict whether the price increased or decreased the following hour. We trained several different classifiers with the same set of features. In this case, our baseline model was simply choosing the most common output class, which was a price increase. We evaluated our performance by using classification accuracy. Our results are summarized as follows:

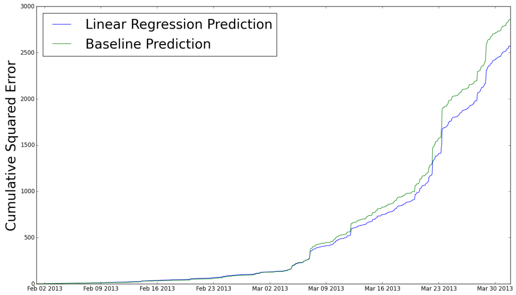


Figure 7: We see that our prediction and the baseline perform similarly at first but that our prediction does better as time goes on

Classification Model	Accuracy
Baseline	53.4%
Logistic Regression	54.3%
SVM	53.7%
Neural Network	55.1%

Table 2: Classification results

Our best classifier was the 2-hidden-layer neural network, though even with this we were still only able to predict Bitcoin increase or decrease slightly better than baseline. Nonetheless, we can still gain some insight as to the driving factors driving Bitcoin price. In Figure 9, for each data point we plot our two most informative features and mark each point with an 'x' or an 'o' for decrease or increase, respectively.

Our two most informative features were the net flow through account A per hour and the number of transactions made by new addresses in a given hour. We can tell from the graph that the features are correlated, as high number of transactions often implies near-zero net flow through account A. This is in line with our suspicion that account A is

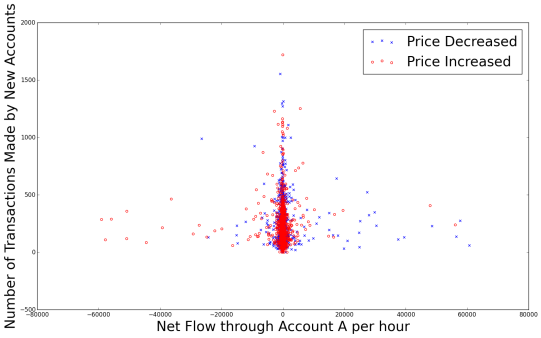


Figure 8

associate with Mt. Gox, an exchange which would give Bitcoin to new addresses. In addition, we can see that while the vast majority of data points have near-zero net flow through A, where this feature is not very informative, the extremes of this feature are quite informative. We can see that when this feature is large and positive the price tends to decrease and when it is large and negative the price tends to increase.

5 Discussion

In analyzing the Bitcoin blockchain, we extracted several network-based features, such as influential agents, flow features, and centrality measures. We then ran a battery of machine learning algorithms to train on these features. While our models were able to beat baseline performance, we ultimately found that only limited amounts of predictive information are embedded in these network features themselves. In reflection, this seems reasonable, given that Bitcoin price is technically dictated by exchanges whose behaviors lie largely outside the realm of the actual Bitcoin blockchain. While our original hypothesis was that there would be significant information embedded in the actual blockchain that might proxy Bitcoin exchange behavior, we found only limited success in this analysis.

Clearly, further work such as developing more non-network features is required to obtain a commercially viable Bitcoin price predictor. As mentioned in the related works, features based on the second by second Bitcoin exchanges are likely the most informative in predicting future price. Future research would be to include features that would more directly capture the information from these Bitcoin exchanges to supplement our network features. For example, analyzing user adoption, user behavior, and financial flow features within the Bitcoin exchanges themselves would likely provide complementary information in improving Bitcoin price predictive accuracy.

Nevertheless, we did uncover some interesting behavior relating the exchanges performed by account A (which we think is Mt. Gox) and Bitcoin price. In most of the data points, account A received roughly the same amount of Bitcoin as it sent. However, when account A sent many more Bitcoin than it received, Bitcoin price tended to increase, and when it received more than it sent, the price tended to decrease. This makes sense under our assumption, because if Mt. Gox is sending out a lot of Bitcoin, people are buying more Bitcoin than they are selling and thus there is a higher demand for it. This in turn would increase Bitcoin price. The opposite is true when more people are selling than buying.

Lastly, we gained some insights into the behavior of users that buy and sell Bitcoin. When the demand is neither particularly high nor low (net flow through A is near-zero), new users are willing to immediately use their Bitcoin, as seen in Figure 9. However, when demand is increasing or decreasing, we believe users are more likely to hoard their Bitcoin and use it less.

References

[1] Michael Fleder, Michael Kester, Sudeep Pillai *Bitcoin Transaction Graph Analysis*, MIT, 2014.

[2] Dorit Ron, Adi Shamir *Quantitative Analysis of the Full Bitcoin Transaction Graph*, The Weizmann Institute of Science, 2012.

[3] Dorit Ron, Adi Shamir *Quantitative Analysis of the Full Bitcoin Transaction Graph*, The Weizmann Institute of Science, 2012.

[4] Isaac Madan, Shaurya Saluja, Aojia Zhao *Automated Bitcoin Trading via Machine Learning Algorithms*, The Weizmann Institute of Science, 2012.