Identifying Volatility Regimes in Bitcoin Prices

Svetozar Valtchev, Jemisa Sadiku, Jianhong Wu

Laboratory for Industrial and Applied Mathematics, York University, Toronto, ON, Canada M3J 1P3

October 31, 2019

Abstract

In this paper we apply various clustering techniques to volatility and market sentiment measures of historical Bitcoin prices in order to identify hidden structural patterns by separating the data into different classes known as regimes. Regimes are a common indicator for traders and asset managers to look at when deciding what actions they want to take. For instance, hedge funds might avoid trading in highly volatile times where-as option traders might prefer. To separate the data, we perform `K-means`, `Ward`, `Birch` and `complete-linkage Agglomerative` clustering algorithms. The resulting clusters are then used as state vectors in a Markov Chain model that predicts the market's state in the short term horizon. Finally, we provide an improvement to our one time-step-ahead forecast using historical results and the conditional probability framework.

keywords — **bitcoin, BTC, cryptocurrency, clustering, volatility regimes, alternative assets, financial forecasting**

1 Introduction

Bitcoin (BTC) is a virtual currency, part of a larger class of digital currencies known as cryptocurrencies, where payments are pseudo-anonymous and independent of third parties such as banks and governments [1, 2, 3, 4].

Introduced in 2009, BTC is a decentralized cryptocurrency utilizing the blockchain technology to record transactions in a public ledger. Unlike fiat currencies and precious metals, Bitcoin is not a physical good, but instead just a record on a database maintained by a network of users known as "miners", public to the world [5, 3, 6]. A tremendous interest has been given to BTC price volatility which, according to [7], is more volatile than gold. A study by [8, 9] demonstrated that financial assets display common windows with large price changes that tend to cluster together. Furthermore, we encounter drastic changes in behaviour at the market extremes such as the 2008 global financial crisis where volatility, mean returns and correlation patterns in stock returns changed abruptly. Identifying these regime changes is critical for the analysis of equities, fixed income securities and a great number of macroeconomic variables [10]. Given a firm's trading strategies, a fund manager can utilize the current volatility conditions to adjust which ones to give preference to, or when to step away from the market.

Clustering analysis plays a critical role in identifying regime changes, and there have been a number of clustering techniques developed and successfully applied for regime changes in time series data. For example, in [11], the TreeGNG algorithm was used for clustering analysis of a high dimensional data set, full of noise. In this paper we examine how common clustering techniques perform when used to partition the volatility regimes of historical Bitcoin prices. We will then analyze how these regimes correlate to one another, and use the correlation to better predict future regimes. Allocation can be made proportional to these transitional probabilities, to optimize any objective the firm may desire. Strategy returns and risk levels can then be trivially back-tested, but are outside the scope of this paper.

2 Background

2.1 Volatility Regimes

Trading strategies utilizing heavy shorting positions perform best in bear markets. In contrast, a high frequency strategy will tend to perform best in a highly liquid and volatile market. It may even be the case that a strategy outperforms it's returns in conditions unlike those it was initially intended for. We call these market conditions, volatility regimes, as volatility is the central factor in identifying these time periods. For instance, the financial crash of 2008, can be weakly coined as a highly volatile, heavily downward trending market incident.

We refer to this as a weak definition, as there isn't a necessary condition on what is deemed high volatility, and heavily down trending. Furthermore we are conditioned to identify simple linear patterns in our world, extending our assumptions indefinitely, while missing out on much of what only a computer might see, and falling victim to countless biases. For example one might think a low volatile, heavily bullish market might be a market state, but this may not even be possible depending on the volatility measurement chosen, as we will see later on. These intuitive groupings we make, are not a systematic approach to differentiating regions in our state space, and also lack concrete threshold levels as well as a statistical backing.

2.2 Clustering

Clustering is a mathematical technique that separates a data set into clusters, where the data points in the same cluster share similarity and data points belonging to different clusters are dissimilar [12, 13, 14].

Clustering helps find structures in a data set and is one of main pattern recognition tasks, and widely used by to inform strategy decision making by investors, financial creditors, stock holders, etc [13].

In what follows, we will explore the K-Means [15], Ward's Method [16, 17], Complete-Linkage [18, 19, 20, 21] and Birch [22] algorithms for the identification of Bitcoin volatility regimes. Our analysis is general enough however, to be extended to other cryptocurrencies and asset groups.

3 Data Cleansing and Preparation

3.1 Resampling

Our training data is taken from `www.BTCe.com`, a Bitcoin exchange which publishes all trading data that passes through their exchange since their inception in 2011. It's important to note that real world trading data has no set sampling rate for transactions. Instead, in one second there may be a large number of orders, while the next may have none. In this paper we limit our data sample to BTC data starting from 2015 since the asset was relatively liquid after that. Our goal is to have an hour-by-hour prognosis of the market, hence a minute-by-minute sampling rate is a reasonable choice. Due to the nature of the financial markets, when re-sampling such unstructured data in a given 60 second window, there is the possibility of having more than one measurement, or no measurements at all. In case of more than one measurement, we take the mean of all data points within that time interval in order to avoid approximating the market price as the initial or final price alone. If we have no data over a 60 second window, we proceed by filling the data point as the last entry, known as a forward-fill. This again is intuitive for a price measurement, as the best approximation of an asset's current value is it's last traded price.

3.2 Volatility and Direction Measurements

For the sake of simplicity, we will limit our analysis to only two variables. The two common factors for identifying market regimes are volatility and market direction. Volatility index may be an intuitive choice, but since BTC is relatively new, no large financial institution has created a market standard. Services like `btcvol.info` provide 30 and 60 day volatility measurements, but these are too long of time periods for our analysis. We proceed by building our own measurement similar to `btcvol.info` [23] by using the standard deviation of our data in each preceding 60 minute window shown in Figure 1.

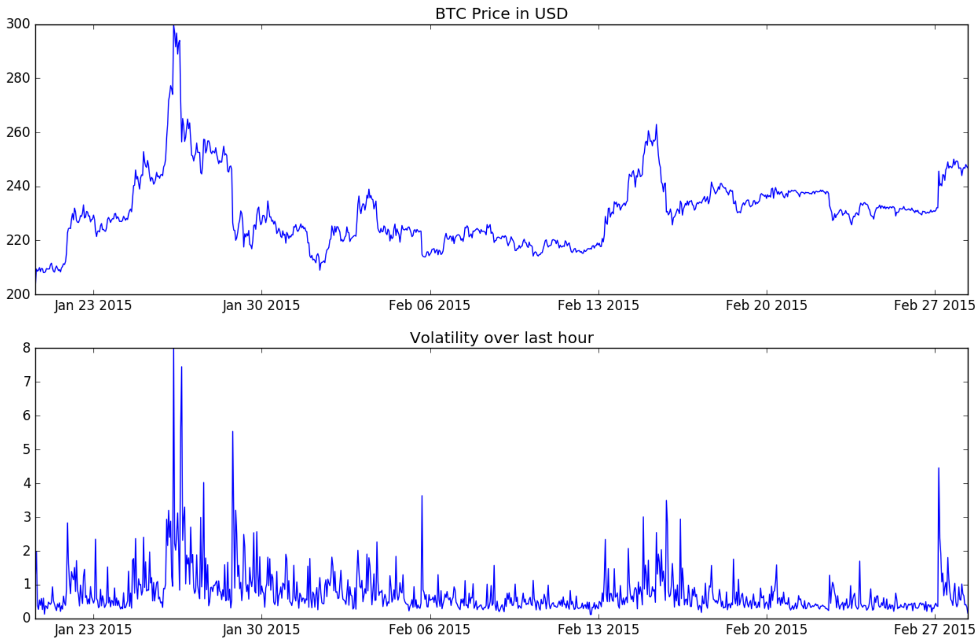

Figure 1: Example of BTC price and 60-minute volatility measure from mid January to late February in 2015

A better measuring tool used in financial instruments is the implied volatility which is calculated by reverse engineering options prices in the market. However, in this study this is not currently viable with the immaturity of the Bitcoin derivatives market. To demonstrate the market direction we calculate short term measurement of the change in price from one minute to the next by the following equation

$$\Delta_t = P_t - P_{t-1}$$

where P_t is the price of a Bitcoin in USD at time t. This can be taught of as a speed indicator computed every minute for BTC, but it's important to realize this as a highly noisy measurement due to the random fluctuations in minute to minute trading. To account for this, we apply a common moving average technique of length 60 minutes, and compute a new smoother measurement

$$S_t = \frac{\sum_{i=0}^{59} \Delta_{t-i}}{n}$$

for our price direction as can be seen in figure 2. The choice of averaging out the past 60 samples is again due to the fact that we want a measurement related to the market direction hourly, so any more data would be correlated with the previous measure, while any less would be failing to utilize all the information we have.

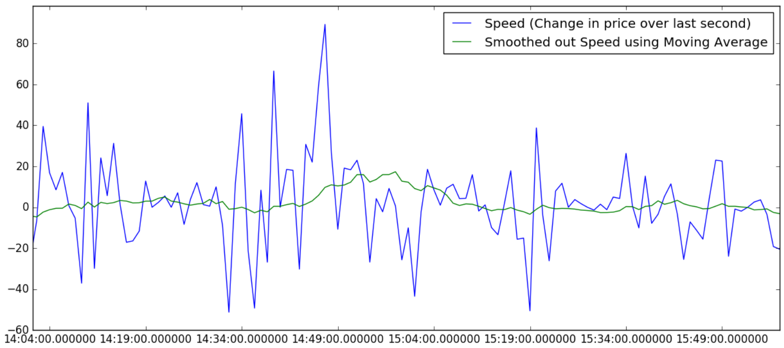

Figure 2: Example of data smoothing using a 60-minute moving average on BTC prices over a two hour period

It is also possible to use other measurements for determining the trend in the short term market conditions such as the spread between a long and short moving average filter. In such a scenario, a short term smoothing jumping over the long term would indicate that the market is currently experiencing a climb higher than the recent historical average, and possibly making a run up. Trend filtering for long and short momentum indicators is explored by [24].

3.3 Regularization

The last thing we want to is standardize our different measurements. We do this using a simple z-score transformation define by

$$z_i = \frac{X_x - \bar{X}}{\sigma_X}$$

with $\bar{X}$ and σ_X being the sample mean and standard deviation respectively, for each measure. Since we may utilize some algorithms which rely on Euclidean distance measurements, it's important that we perform this step in order to assure our measures are of the same scale. Furthermore, since some of the algorithms employ an optimization step, it is ideal to have our data standardized or normalized to assure faster convergence [25]. We choose standardization over normalization in order to keep open boundaries in our domain.

3.4 Final Slicing/Sampling

Since our data is now sampled on a minute to minute basis, with measures relying on the past 60 minutes at each step, we re-sample our data one more time with an hourly range. This is to assure we are not feeding our algorithms highly correlated data points, since the volatility over the last hour from one minute to the next is nearly identical. We choose to formulate our final data set with one data point per hour, so as to have 8760 points per year's worth of data.

4 Results

4.1 Volatility Clusters

As a base case, we consider a fund who's strategies are not market dependent. A simple approach may be to divide our market in two (or more) groups based on linear boundaries across each variable. For two arbitrary measures X and Y we can segment our data such that

$$f(X,Y) = \begin{cases} \text{State 1,} & \text{if X} \geq \bar{X}, \text{Y} \geq \bar{Y} \\ \text{State 2,} & \text{if X} \geq \bar{X}, \text{Y} \leq \bar{Y} \\ \text{State 3,} & \text{if X} \leq \bar{X}, \text{Y} \geq \bar{Y} \\ \text{State 4,} & \text{if X} \leq \bar{X}, \text{Y} \leq \bar{Y} \end{cases}$$

Graphically this will split our data into convex regions with linear boundaries as seen in Figure 3. However, the regions do not divide the data in a useful fashion and there are few problems with this approach including outliers having a strong weight on our boundaries and partitioning through a cluster of points which are similar in nature.

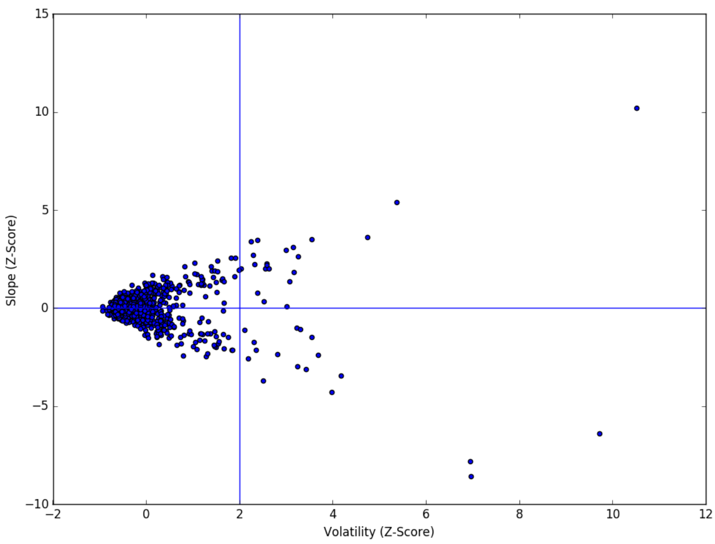

Figure 3: Basic partitioning of 2-D data producing four distinct market regions.

Instead, we will take a more systematic approach to cut our data. Before we begin applying our algorithms to our clean data, we first analyze it visually to identify any obvious clusters and/or any underlying structure. We show 40% of our data in Figure 4 in order to see our data distribution without cluttering the plot.

The data does display a common shape known as a Markowitz Bullet [26] in finance, which shares similar measurements. There are no obvious disjoint clusters but that is not too troubling as there are

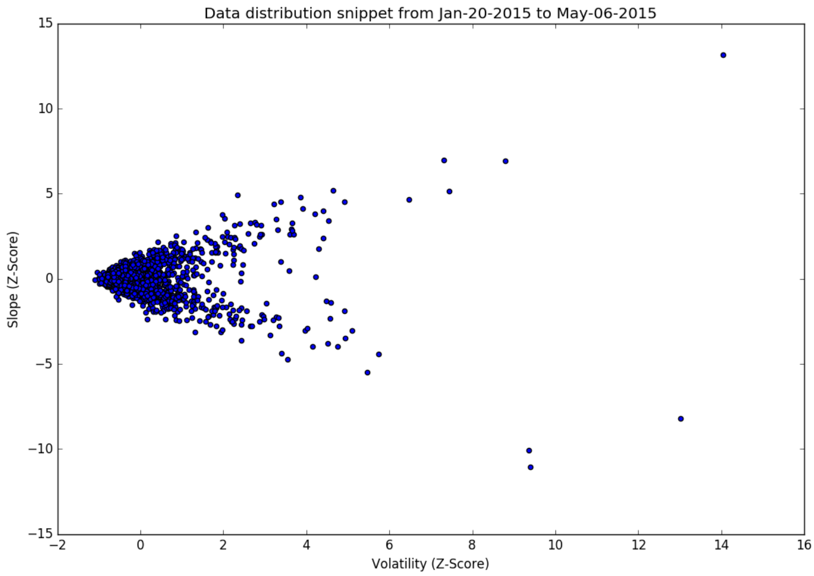

Figure 4: Distribution of cleansed data

specific regions and symmetries present that should partition the data reasonably.

4.1.1 Convergence Times

Before diving into a full clustering analysis, we test our algorithms on smaller batches of data to see how they perform. We are dealing with over 15,000 data points and so some algorithms may not scale well to this size. The following chart and plot show how long (in milliseconds) each method took to execute for different sample sizes:

Algorithm	100	500	1000	2500
K-Means	25	26	37	80
Agglomerative	2	75	1003	10616
Ward	8	75	1010	10098
Birch	28	36	71	154

Table 1: Execution times in milliseconds of each algorithm for varying amount of data points.

Our Agglomerative and Ward algorithms display exponential executions times, and did not converge within reasonable time on an year's worth of data. The aforementioned results were obtained on a standard CPU with optimized numerical analysis packages, but performance times may vary. However, the importance to note here, is the order of the growth of the processing time for each algorithms as data size increases.

4.1.2 Number of Clusters

Determining the number of clusters is the next logical question one would ask, since all our algorithms require this parameter. In the case where a firm has some group of regimes already in operation in their

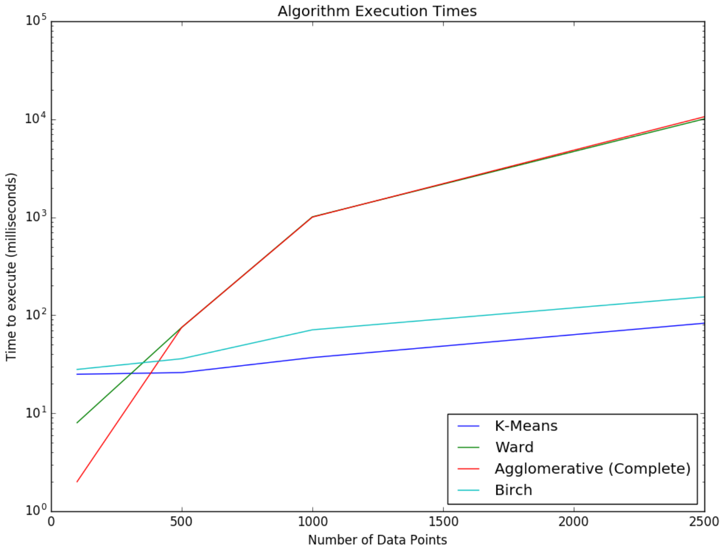

Figure 5: Algorithm convergence times for different sample sizes (in milliseconds)

diagnosis, it is intuitive to try to mimic those clusters already in place, and choose that as your value.

Another factor we want to keep in mind is that there is heavy symmetry in our data and a flat low volatile regime is a must in our results for obvious reasons, so an odd number of clusters is a good logical requirement to have (1 for neutral state and equal number of bull states as bear states).

A common heuristic at this point is to use the Elbow Method [27], which looks at the scree plot, and tries to identify where the elbow joint lies on the graph. This however is limited by your scale of the plot, and a complete solution can rely on many different factors and parameters. Other choices for our clusters is described in greater detail in [28] and more specifically for the k-means algorithm in [29]. We decided to stop adding clusters at the point where any new cluster reduced our total inertia by less than 20%. Since this was at 4 clusters, we decided to continue our analysis with 5 to assure the symmetry criteria mentioned above.

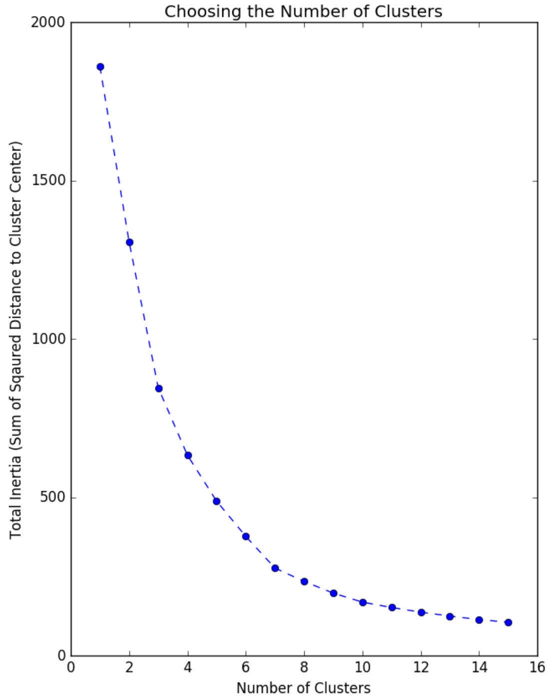

Figure 6: Example of a scree plot for the k-means algorithm measuring the accuracy of the cluster, in this case using a total inertia measurement.

The results of K-Means and Birch clustering can be seen in Figure 7. Notice that our decision boundaries are linear but not simply vertical or horizontal cuts, giving rise to some interesting groupings. For example, in the bottom left plot, a point in region 1 (green) can have a higher volatility than a point in region 3 (purple), yet be classified as a lower volatility regime group.

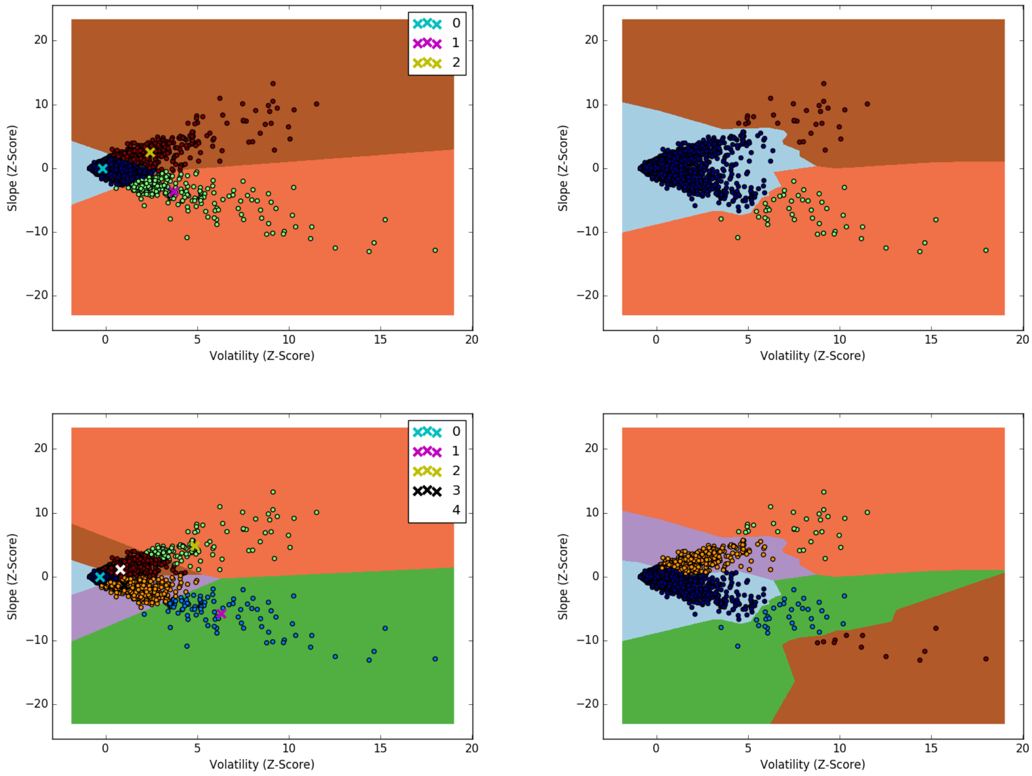

Figure 7: Clustering using 3 and 5 clusters, using K-Means on the left and Birch on the right

The Birch algorithm gives us non-linear boundaries altogether, and the clusters also display a shift from the symmetric structure we first hypothesized. The statistical inference picked up on slight dissimilarities between up-trending and down-trending market conditions given the same volatility. This is the power of clustering, distinguishing underlying patterns systematically.

4.2 Markov Chains

4.2.1 State Matrix and Probabilities

Applying the new models, we proceed to scan historical data and compute the frequencies of how each point's cluster changed in the next time period (1 hour). In other words, we count how many times points began in cluster C_i and ended up in cluster C_j an hour later.

This can be summarized in a nice Markov state matrix defined as

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n} \end{pmatrix}$$

where n is the number of clusters we have, and $s_{i,j}$ is the number of points our model would consider being in cluster C_i at some time period, and in cluster C_j in the subsequent time period. We can build this model easily by looking back at our data, and can trivially convert this matrix into a probability matrix of going from cluster C_i to cluster C_j in one time period (1 hour), using a simple binomial view. More precisely

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

where

$$p_{i,j} = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} s_{i,j} = 0 \\ \frac{s_{i,j}}{\sum_{j=1}^{n} s_{i,j}} & \text{otherwise} \end{cases}$$

Here, each element $p_{i,j}$ is the probability of going from cluster C_i to cluster C_j in a one hour time frame, as measured by the historical frequency of such an event occurring.

We can further introduce a confidence for each of our probabilities, as given by the Wald method

$$\hat{p} \pm z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

where $z_{\frac{\alpha}{2}}$ is set to 1.96 for a 95% confidence measure as determined from the standard normal distribution, and n being the row sum in matrix S for a particular probability. Note here that we can have division by zero, and so our accuracy measure of such a sample is undefined as it should be with a sample size of 0.

4.2.2 Transitions

If we define our current state as the elementary basis vector of length n:

$$X_0 = e_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

with one element in state s_i, we can use our probability matrix P to compute the probability of where we will lie in N time periods by $X_N = X_0^T P^N$. For example given the probability matrix from our K-Means three clusters model, and a current market condition of state three, we can predict in two hours that we will lie in cluster C_i with probability

$$X_N = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.9611 & 0.0132 & 0.0255 \\ 0.5638 & 0.1737 & 0.2624 \\ 0.6528 & 0.1301 & 0.2169 \end{pmatrix}^2$$
$$= \begin{pmatrix} 0.8425 & 0.0595 & 0.0979 \end{pmatrix}$$

and similar results can be formulated for the confidence intervals.

This kind of a model is useful as it not only gives the volatility regimes we currently lie in, but can also forecast the future. This inform our decision on allocating our trading strategies accordingly for the current time period, and inform the diagnosis of how they will likely change in the near term horizon, which may be useful for strategies that rely on holding a position for a longer period.

5 Conclusions

Using a group of common clustering algorithms, we identified market volatility regimes using common market measures. Patterns arose that gave way to predictive models not otherwise attainable by common means of the human eye. More importantly, these cluster divide the market into well defined regions which we can easily check and optimize for given a series of trading algorithms simple back-testing procedures. We aim to perform such back-testing in the future for a more in depth analysis of the strength of our results and predictors. We also aim to add volume and news sentiment measures to hopefully increase our predictive power.

It was shown that both the `K-Means` and `Birch` algorithms were reasonable choices for the division of the data, and results for which is better may be different from fund to fund. Choices of 3 or 5 were both reasonable choices for our total market regimes, though further work can be done here.

The Markov chain model was applied to each cluster in order to approximate our market in the time period ahead. The steady state probabilities of the chain were preserved in this fashion, (used to predict the number of expected time periods of stay in each market state over a long period), as well as any other properties of the Markov chain framework.

Lastly, an improvement to the model was explored as to increase the accuracy of our predictions. Other similarity measures, again, can be explored further.

6 Acknowledgments

This research was supported by Natural Sciences and Engineering Research Council of Canada and the Canada Research Chair Program.

References

[1] Björn Segendorf. What is bitcoin. *Sveriges riksbank economic review*, 2:71–87, 2014.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[3] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

[4] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2(6-10):71, 2016.

[5] Asma Salman and Muthanna G Abdul Razzaq. Bitcoin and the world of digital currencies. *Financial Management from an Emerging Market Perspective*, pages 271–281, 2018.

[6] Luisanna Cocco and Michele Marchesi. Modeling and simulation of the economics of mining in the bitcoin market. *PloS one*, 11(10):e0164603, 2016.

[7] Marc Gronwald. The economics of bitcoins–market characteristics and price jumps. 2014.

[8] Jack Fan and Isaac Kleshchelski. Trend following and volatility regimes.

[9] Rama Cont. Volatility clustering in financial markets: empirical facts and agent-based models. In *Long memory in economics*, pages 289–309. Springer, 2007.

[10] Andrew Ang and Allan Timmermann. Regime changes and financial markets. *Annu. Rev. Financ. Econ.*, 4(1):313–337, 2012.

[11] Kevin AJ Doherty, Rod G Adams, Neil Davey, and Wanida Pensuwon. Hierarchical topological clustering learns stock market sectors. In *2005 ICSC Congress on Computational Intelligence Methods and Applications*, pages 6–pp. IEEE, 2005.

[12] SR Nanda, Biswajit Mahanty, and MK Tiwari. Clustering indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12):8793–8798, 2010.

[13] Mansoor Momeni, Maryam Mohseni, and Mansour Soofi. Clustering stock market companies via k-means algorithm. *Kuwait Chapter of Arabian Journal of Business and Management Review*, 33(2578):1–10, 2015.

[14] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications.* "ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadephia", 2007.

[15] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[16] Fionn Murtagh and Pierre Legendre. Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm. *arXiv preprint arXiv:1111.6285*, 2011.

[17] J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[18] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal. British Computer Society*, 20:364–366, 1977.

[19] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

[20] Ian Davidson and SS Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 59–70. Springer, 2005.

[21] K Sasirekha and P Baby. Agglomerative hierarchical clustering algorithm-a. *International Journal of Scientific and Research Publications*, 83:83, 2013.

[22] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96*, pages 103–114, 1996.

[23] elidourado. The bitcoin volatility index, 2016.

[24] B. Bruder, T.L. Dao, J.C. Richard, and T. Roncalli. Trend filtering methods for momentum strategies. 2011.

[25] Ismail Mohamad and Dauda Usman. Standardization and its effects on k-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 6:3299–3303, 09 2013.

[26] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[27] David J. Ketchen and Christopher L. Shook. The application of cluster analysis in strategic management research: An analysis and critique. *Strategic Management Journal*, 17(6):441–458, 1996.

[28] Y. Jung, H. Park, D.Z. Du, and B. Drake. A decision criterion for the optimal number of clusters in hierarchical clustering. 2002.

[29] D.T. Pham, S.S. Dimov, and C.D. Nguyen. Selection of k in k-means clustering. *J. Mechanical Engineering Science*, 219:103–119, 2004.