Forecasting Cryptocurrency Transaction Data

Jake Taylor
Department of Statistics
Stanford University
jakee417@stanford.edu

Samuel Wong
Department of Statistics
Stanford University
samwwong@stanford.edu

1 Introduction

Forecasting is used in many industrial settings when future decision making is complicated by the inherent uncertainty of a process of interest. Modern examples include forecasting future product demand for a web commerce retailer (1), efficient energy use as part of a smart grid (SG) system (2), or central processing unit (CPU) utilization across a network of virtual routers for a cloud based computer security company (3). Classical statistical analysis of time series data commonly starts with an Autoregressive-Moving-Average (ARMA) model (4) and builds upon this foundation with a variety of generalizations such as ARIMA, SARIMA, ARMAX, NARMAX, or VARMA (5). More recently, approaches involving Deep Neural Networks (DNNs) have been used to learn highly non-linear functions for large amounts of time series data while also incorporating a probabilistic component to measure the forecast's uncertainty (1). We would like to explore the application of these more recent approaches to analyze large amounts of cryptocurrency transaction data.

2 Dataset and Features

Our dataset comes from transaction data present in the blockchains of popular cryptocurrencies. Since these blockchains are all digital public ledgers, the data is readily available for us to use to perform our forecasting using Bayesian techniques. The data already has timestamps (hour, day, month, year), which we convert into time series features. The goal is to forecast the transaction volume as well as the total amount of value transacted every hour. We believe that accurate forecasting of these targets will create a strong tool for financial analysis in more traditional cryptocurrencies such as Bitcoin, Litecoin, and Dogecoin. Furthermore, with the growing popularity of dApps on the Ethereum blockchain, particularly in DeFi with the rise of DEX (UniSwap, SushiSwap, Curve, etc.), there still remains an open question of scalability in this space. We believe our predictions of transaction volume can help guide decisions for new entrants as to whether to stay within this current blockchain framework and risk higher Gas fees and an overall longer block confirmation time, or to build their product on a separate chain altogether.

3 Proposed Methodology

Consider an additive error decomposition of our time series data, $y(t) = x(t) + \epsilon(t), t \in \{0, \dots, T\}$. This allows us to model $y(t)$ by sequentially modelling $x(t)$ and then $\epsilon(t)$. We can first model $x(t) = f(y(t), \dots, y(t-\tau)), \tau < T$ where $f(\cdot)$ is taken to be some DNN fit on τ lagged covariates. After fitting $f(\cdot)$, we can estimate $\epsilon(t)$ using the residuals, $\epsilon(t) \approx y(t) - f(y(t), \dots, y(t-\tau))$. We can then learn some distribution $\epsilon(t) \sim \mathcal{D}(\vec{t}; \vec{\theta})$ that incorporates dependency between $\epsilon(t_i), \epsilon(t_j), i \neq j$. For example, $\mathcal{D}(\vec{t}; \vec{\theta})$ could be a Hidden Markov Model (HMM) consisting of $NB(r, p)$ or $\mathcal{N}(\mu, \sigma^2)$ components depending if we are looking at trade frequency (counts) or trade values (real valued). Finally, we can fit the parameters of this error distribution $\epsilon(t) \sim \mathcal{D}(\vec{t}; \vec{\theta})$ using some function $g(\cdot)$ which is fit with yet another DNN. We plan to use the Tensorflow (TF) and Tensorflow Probability (TFP) libraries to fit $f(\cdot)$ and $g(\cdot)$ respectively (6; 7) .

Stat271: Applied Bayesian Analysis, Spring 2021, Stanford University, CA. (template credit NIPS 2017.)

References

[1] D. Salinas, V. Flunkert, and J. Gasthaus, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," 2019.

[2] D. Kaur, S. N. Islam, M. A. Mahmud, and Z. Dong, "Energy forecasting in smart grid systems: A review of the state-of-the-art techniques," *CoRR*, vol. abs/2011.12598, 2020.

[3] S. R. Karingula, N. Ramanan, R. Tahsambi, M. Amjadi, D. Jung, R. Si, C. Thimmisetty, and C. N. C. J. au2, "Boosted embeddings for time series forecasting," 2021.

[4] P. Whittle, "Hypothesis testing in time series analysis," 1951.

[5] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications With R Examples*. Springer, 2006.

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[7] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," 2017.