Bitcoin Price Prediction: Mixed Integer Quadratic Programming Versus Machine Learning Approaches

Marco Corazza(✉) and Giovanni Fasano

Ca' Foscari University of Venice, 30121 Venice, Italy
{corazza,fasano}@unive.it
https://www.unive.it/data/people/5592848,
https://www.unive.it/data/people/5592722

Abstract. Reliable Bitcoin price forecasts currently represent a challenging issue, due to the high volatility of this digital asset with respect to currencies in the Forex market. Since 2009 several models for Bitcoin price have been studied, based on neural networks, nonlinear optimization and regression approaches. More recently, Machine Learning paradigms have suggested novel ideas which provide successful guidelines. In particular, in this paper we start from considering the most recent performance of Bitcoin price, along with the history of its price, since they seem to partially invalidate well renowned regression models. This gives room to our Machine Learning and Mixed Integer Programming perspectives, since they seem to provide more reliable results. We remark that our outcomes are data–driven and do not need the fulfillment of standard assumptions required by regression–based approaches. Furthermore, considering the versatility of our approach, we allow the use of standard solvers for MIP optimization problems.

Keywords: Bitcoin · Regression problems · Support Vector Machines · Quadratic Mixed Integer Programming

1 Introduction

This paper details a novel viewpoint to study a price forecast problem, associated with *Bitcoin* [2,4,7], that represents the proposal with the largest market capitalization among the crypto assets. Bitcoin was initially created by an anonymous researcher (or possibly a team of people), under the nickname of Satoshi Nakamoto. The actual identity of such creator has never been revealed so far, and anonymity is expected to be likely maintained also in the future. No private/central bank is responsible for minting novel bitcoins[1], so that Bitcoin/USD rate is merely the result of negotiations among private stakeholders

[1] Observe that typically the symbol *Bitcoin* is used to identify the crypto asset, while *bitcoins* represent its coins.

M. Corazza et al. (Eds.): MAF 2022, *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pp. 162–167, 2022.
https://doi.org/10.1007/978-3-030-99638-3_27

through online exchanges, so that exchanges prevent from double-spending and arbitrary generation of new bitcoins. Direct peer–to–peer transactions involving bitcoins among private investors are also allowed, recurring to special secure protocols that impede abuses. In order to take record of finalized Bitcoin movements, a special distributed ledger, namely a *blockchain*, both does not allow reversibility for transactions and guarantees public information on transactions amount, though preserving anonymity.

In the last decade a number of approaches were introduced in the literature for Bitcoin price prediction (see e.g. [1] and [6], along with therein references), involving investors, researchers, practitioners, as well as private and public institutions, so that some approaches may be undoubtedly considered more methodologically sound with respect to others. Among the main difficulties which negatively affect an accurate prediction of Bitcoin price we find its high volatility, that is yielded by several causes, including a relatively small experience of investors, an unstructured market, the extreme liquidity of bitcoins and the high leverages on Bitcoin transactions.

Unlike the cited references we report here a data–driven approach, based on both Multiobjective Optimization and a Mixed Integer Quadratic Programming formulation, in order to reliably foresee Bitcoin price and exploiting its historical performance.

2 Our Problem

As from [5], in Fig. 1 we summarize some relevant information associated with the price of Bitcoin, from past transactions between 2009 and the end of September 2021. The abscissa axis reports the scaled values of the *Stock–to–Flow (SF)* associated with Bitcoin, i.e.

$$SF = \frac{Stock\ of\ Bitcoin\ at\ a\ given\ date}{Flow\ of\ bitcoins\ in\ a\ given\ time\ window},$$

where *Flow of bitcoins in a given time window* refers to an interval of 463 days (according with the suggestion from the current literature – see also [5]). On the ordinate axis of Fig. 1 we report Bitcoin price. Moreover, we compute the sets $L_{East-South}$ and $L_{West-North}$, being respectively

- $L_{East-South}$: the weak Pareto front associated with both the maximization of the stock-to-flow SF and the minimization of Bitcoin price;
- $L_{West-North}$: the weak Pareto front associated with both the minimization of the stock-to-flow SF and the maximization of Bitcoin price.

Furthermore, we also indicate in Fig. 1 some circled points, corresponding to so called *support vectors*, obtained applying a standard Support Vector Machine (SVM) approach (see also [6] and [5]) to the linear separation problem between the points in the sets $L_{East-South}$ and $L_{West-North}$. The stripe delimited by the lines through these support vectors represents an area where *no extreme* Bitcoin transactions were experienced. In other words, loosely speaking this last area

contains all the pairs of Bitcoin vs. its *SF*, obtained during its history, neither contained in $L_{East-South}$ nor in $L_{West-North}$. Thus, this (reasonably thin) area is likely expected to contain also most of the future transactions.

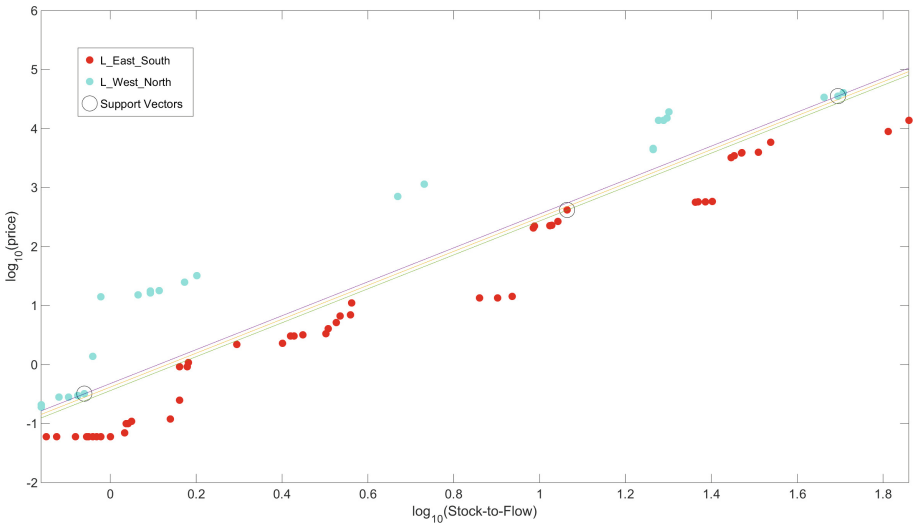

Fig. 1. The weak Pareto fronts $L_{East-South}$ and $L_{West-North}$, corresponding to *extreme* past transactions of Bitcoin price vs. its stock-to-flow ratio: cyan points correspond to relatively favorable (i.e. large priced) transactions, while red points correspond to relatively poor (i.e. poorly priced) transactions.

2.1 Our MIP Viewpoint vs. SVMs

First observe that if N represents the number of all the points in $\mathbb{R}^2$ corresponding to Bitcoin price vs. its *SF* (i.e. our dataset), we consider the next assignment for the labels $\{y_i\}$:

$$y_i = +1 \text{ if the point } P_i \text{ belongs to } L_{West-North},$$
$$y_i = -1 \text{ if the point } P_i \text{ belongs to } L_{East-South}.$$

Then, setting $N_0 = |L_{East-South} \cup L_{West-North}|$, in order to apply the SVM approach in Fig. 1, the solution of the next Convex Quadratic Programming problem is required (see also [1] and therein references for a complete justification)

$$\begin{aligned} \min_{\beta,\ \beta_0,\ \xi} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N_0}\xi_i \\ \text{s.t.} \quad & (\beta^T x_i + \beta_0)y_i \geq 1 - \xi_i, \quad i = 1, \ldots, N_0, \\ & \xi_i \geq 0, \qquad\qquad\qquad\quad i = 1, \ldots, N_0, \end{aligned} \tag{1}$$

In (1) the pair (β, β_0) represents the coefficients of the line (central line in Fig. 1) which best separates the sets $L_{East-South}$ and $L_{West-North}$, i.e. it corresponds to the largest possible stripe delimited by the support vectors in Fig. 1. The quantities x_i, $i = 1, \ldots, N_0$, in our application represent *scalars*, corresponding to a value (i.e. *SF*) in the abscissa axis. Nevertheless, in a more general framework they may represent n–real *vectors*, each representing a vector of values including the *SF*; this explains why in (1) we preferred to introduce the (more general) inner product between the n–real vector (of coefficients) β and the vector x_i, for any i. A similar consideration holds also for the formulations (2) and (3) below.

We highlight that some elements of the solution of this mathematical programming problem, namely β^* and β_0^*, provide the optimal estimates of the coefficients of the central line inside the area delimited by the weak Pareto fronts $L_{East-South}$ and $L_{West-North}$. Similarly, this also happens for the subsequent formulations (2) and (3). Note that these optimal estimates are determined accordingly to a data-driven SVM-based optimization approach which, loosely speaking, regresses the Bitcoin price on the SF. Note also that the above (optimal) central line constitutes the long term Bitcoin price forecaster; similarly, again, for the mathematical programming problems (2) and (3).

Moreover, it can be proved that the quantities $\{\xi_i\}$ will be all equal to zero if and only if the two sets $L_{East-South}$ and $L_{West-North}$ are linearly separable. We also observe that after assigning $i \geq 1$ labels $\{y_1, \ldots, y_i\} \subseteq \{+1, -1\}$ to the points $\{P_1, \ldots, P_i\}$, not contained in $L_{East-South} \cup L_{West-North}$, then in (1) we can replace N_0 by N_i, being $N_i \supset N_0$ and $N_i = N_0 + i$ (i.e. N_i points in the dataset have been labelled). Thus, it can easily be proved that adopting the procedure in [5], including one point of our dataset at a time, starting from N_0 up to N, we can iteratively refine the solution of (1) by solving

$$\begin{aligned} \min_{\beta,\, \beta_0,\, \xi} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{j=1}^{N_i} \xi_j \\ \text{s.t.} \quad & (\beta^T x_j + \beta_0) y_j \geq 1 - \xi_j, \quad j = 1, \ldots, N_i, \\ & \xi_j \geq 0, \qquad\qquad\qquad\quad j = 1, \ldots, N_i. \end{aligned} \tag{2}$$

Thus, after solving $N - N_0 + 1$ SVMs *(i.e.* $i = 1, \ldots, N - N_0 + 1$*)*, each corresponding to a binary separation problem, all the N points in the dataset will be classified as either *closer* to $L_{East-South}$ or *closer* to $L_{West-North}$. We strongly highlight that, as detailed in [5], there are applications (e.g. from *semi–supervised learning*) where the labels $\{y_1, \ldots, y_N\}$ are not all known when solving the first SVM in the sequence: this justifies the above iterative procedure.

Now, considering a completely different perspective we might alternatively replace the above iterative SVM–based procedure with a unique Mixed Integer Quadratic Programming (MIQP) reformulation. In this regard, let us preliminarily set

$$\begin{cases} A \equiv L_{West-North}, \\ \\ B \equiv L_{East-South}. \end{cases}$$

Then, in place of the above $N-N_0+1$ SVMs we can propose to solve the MIQP problem (note that $M \gg 1$ is not an unknown but it represents a *Big–M*, i.e. a large enough constant value)

$$\begin{aligned}
&\min_{\beta,\ \beta_0,\ \xi,\ \gamma} \quad \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \\
&\text{s.t.} \quad -(\beta^T x_i + \beta_0) + 1 - \xi_i \le (1-\gamma_i)M, \quad i \ : \ x_i \notin A \cup B, \\
&\qquad (\beta^T x_i + \beta_0) + 1 - \xi_i \le \gamma_i M, \qquad\quad i \ : \ x_i \notin A \cup B, \\
&\qquad -(\beta^T x_i + \beta_0) + 1 - \xi_i \le 0, \qquad\quad i \ : \ x_i \in A, \\
&\qquad (\beta^T x_i + \beta_0) + 1 - \xi_i \le 0, \qquad\quad i \ : \ x_i \in B, \\
&\qquad \xi_i \ge 0, \qquad\quad i = 1, \ldots, N, \\
&\qquad \gamma_i \in \{0,1\}, \qquad\quad i \ : \ x_i \notin A \cup B,
\end{aligned} \tag{3}$$

where:

- the objective function maintains the same structure with respect to the formulation (1) (i.e. a convex quadratic functional);
- $2(N - |A \cup B|)$ linear constraints are added with respect to (1);
- $2(N - |A \cup B|)$ binary unknowns (the unknowns $\{\gamma_i\}$) are added with respect to (1), so that if $(\beta^*, \beta_0^*, \xi^*, \gamma^*)$ is the final solution of (3) then:
  - $\gamma_i^* = 0$ means $x_i \in B$ (and in (1) we will equivalently have $y_i = -1$),
  - $\gamma_i^* = 1$ means $x_i \in A$ (and in (1) we will equivalently have $y_i = +1$).

Remark 1. We highlight that the formulation (3) is *equivalent* to solve the $N - N_0+1$ SVMs in (2). Indeed, under mild assumptions (3) provides the same results of the $N-N_0+1$ SVMs in (2). However, note that (3) unlike (2) contains integer unknowns, that typically increase the computational burden and require a more sophisticated solver.

Several additional properties can be proved for the formulation (3), with respect to considering a sequence of $N - N_0 + 1$ SVMs, including some interesting numerical results. The reader may refer to [5] and [3] for a more thorough description, along with additional suggestions and a complete analysis of the outcomes of the above methodologies on several practical applications. We also remark that C represents the unique parameter included in the formulation (3), and its assessment typically follows two guidelines: on one hand it is chosen large enough to penalize misclassification (i.e. when C is large we tend to reduce the number of nonzero unknowns $\{\xi_i\}$); on the other hand, a too large value for C may imply a relatively large time of computation. In our *Matlab* implementation we set $C =$ `inf` and no numerical odd was experienced.

References

1. Aggarwal, D., Chandrasekaran, S., Annamalai, B.: A complete empirical ensemble mode decomposition and support vector machine–based approach to predict Bitcoin prices. J. Behav. Exp. Finance **27**, 100335 (2020)

2. Blockchain information for Bitcoin (BTC). https://www.blockchain.com/charts/total--bitcoins
3. Caliciotti, A., Corazza, M., Fasano, G.: Regression models and machine learning approaches for long term bitcoin price forecast. Ann. Oper. Res. (2022, submitted)
4. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf
5. Pontiggia, A., Fasano, G.: Data Analytics and Machine Learning paradigm to gauge performances combining classification, ranking and sorting for system analysis. Working Paper 05/2021, Department of Management, University Ca' Foscari of Venice (2021)
6. Sreekanth Reddy, L., Sriramya, P.: A research on bitcoin price prediction using machine learning algorithms. Int. J. Sci. Technol. Res. **9**, 1600–1604 (2020)
7. Vigna, P., Casey, M.J.: The Age of Cryptocurrency: How Bitcoin and Digital Money Are Challenging the Global Economic Order, 1st edn. St. Martin's Press, New York (2015)