A Real-time System of Bitcoin Price Prediction

Zhao, Cai

Table of Contents

Abstract

This report is about a real-time system of Bitcoin price prediction using some big data technologies, including the implementation of data collecting, data preprocessing, modelling, trading strategy, real-time prediction and visualization of the predicted results. The main method of this system is from the paper published by MIT Professor Shah in Oct 2014 [9], which is bayesian regression.

1. Introduction and Task

Bitcoin is a cryptocurrency, a form of electronic cash. It is a decentralized digital currency without a central bank or single administrator that can be sent from user-to-user on the peer-to-peer bitcoin network without the need for intermediaries [1].

The task is to build a real-time system of Bitcoin price prediction.

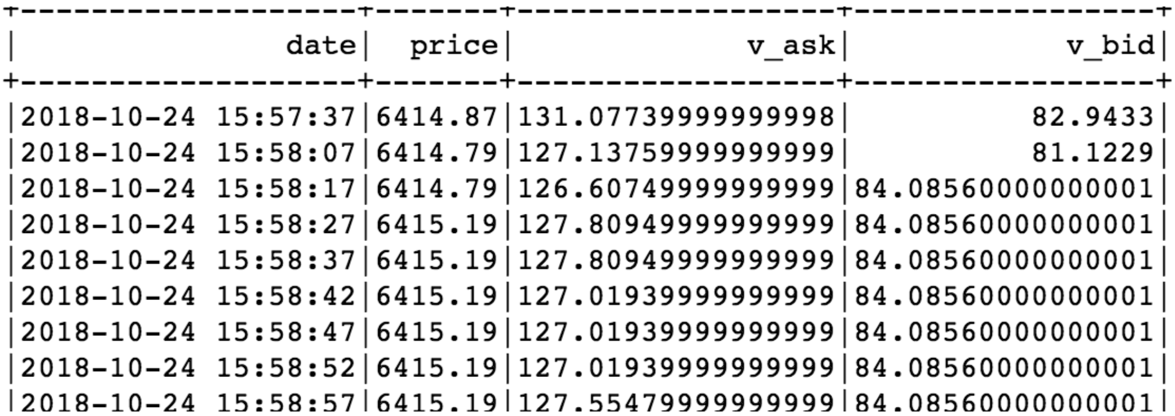
The reminder of this report is organized as follows. Section 2 describes the dataset. Section 3 describes the overview of system architecture. In section 4, approach is presented. In section 5, some details of implementation are performed. Original trading strategy and results are put in section 6. Modified strategy and results can be seen in section 7. Real-time prediction is shown in section 8. Visualization of results of prediction are demonstrated in section 9. Finally, benefits of using big data technologies are in section 10.

2. Dataset

Data related to Bitcoin price and order book is obtained from Okcoin.com every 10 second interval in real time by its APIs [2]. Founded in 2013, OKCoin is a world leading digital asset trading platform. OKCoin currently provides fiat trading with major digital assets, including Bitcoin, Bitcoin Cash, Ethereum, Ethereum Classic, and Litecoin [3].

I chose to run python program in Alibaba cloud to request Bitcoin history price and order book from Okcoin.com and save the data in MongoDB (a document database with the scalability and flexibility that you want with the querying and indexing that you need [4].) installed in the cloud platform. This real-time data collection mechanism allowed me to collect high-granularity Bitcoin price data and accumulate roughly 100,000 unique price points for use in our modeling step.

Figure 1 shows an example of dataset. The date field is the timestamp of requesting data from okcoin. The price field is the last price (close price) of Bitcoin at that time point. The v_bid field is total volume people are willing to buy in the top 60 orders and v_ask is the total volume people are willing to sell in the top 60 orders based on the current order book data.



```
+-------------------+-------+------------------+----------------+
|               date|  price|             v_ask|           v_bid|
+-------------------+-------+------------------+----------------+
|2018-10-24 15:57:37|6414.87|131.07739999999998|         82.9433|
|2018-10-24 15:58:07|6414.79|127.13759999999999|         81.1229|
|2018-10-24 15:58:17|6414.79|126.60749999999999|84.0856000000001|
|2018-10-24 15:58:27|6415.19|127.80949999999999|84.0856000000001|
|2018-10-24 15:58:37|6415.19|127.80949999999999|84.0856000000001|
|2018-10-24 15:58:42|6415.19|127.01939999999999|84.0856000000001|
|2018-10-24 15:58:47|6415.19|127.01939999999999|84.0856000000001|
|2018-10-24 15:58:52|6415.19|127.01939999999999|84.0856000000001|
|2018-10-24 15:58:57|6415.19|127.55479999999999|84.0856000000001|
```

Figure 1: an example of dataset

3. Overview of System Architecture

The overview of system architecture is shown in Figure 2. Data obtained from Okcoin.com every 10 second interval in real time by its APIs is saved in MongoDB installed in the Alibaba cloud platform. The raw data is preprocessed by using MLlib (Apache Spark's scalable machine learning library [7].) k-means algorithm, RDD (Resilient Distributed Datasets) and Spark SQL (Apache Spark's module for working with structured data [6]). In the process of training the model, MLlib, RDD and DataFrame are used. Finally, the results of prediction by the model are saved in in MongoDB installed in the Alibaba cloud platform as well; and the results can be visualized by a adminMongo (a web based user interface to handle all your MongoDB connections/databases needs [8].)

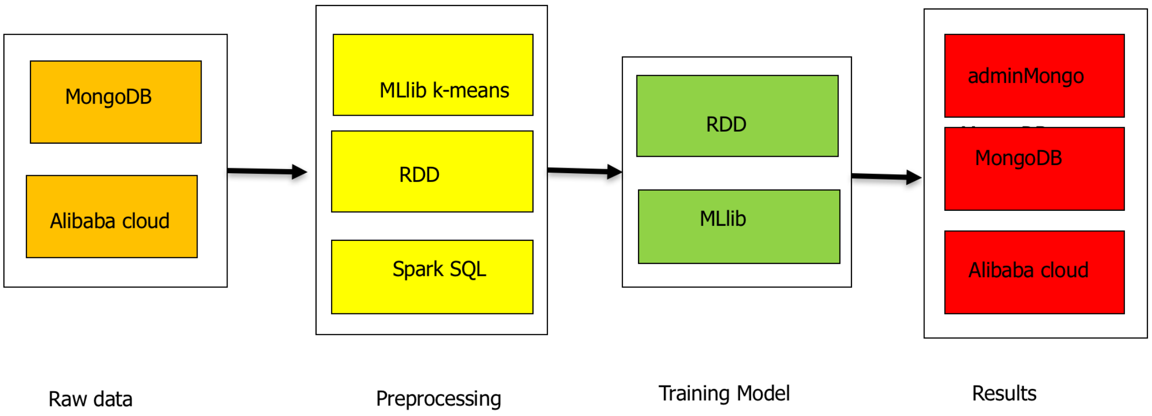


Figure 2: Overview of System Architecture

4. Approach

Bayesian regression is used for predicting price variation of Bitcoin. This idea is from the paper published by MIT Professor Shah in Oct 2014 [9]. Based on this price prediction method, they devise a simple strategy for trading Bitcoin. The strategy is able to nearly double the investment in less than 60 day period when run against real data trace. However, I have done some optimizations basing on this method from the paper when using some big data technologies to implement this method. I choose to use the sample entropy to choose the effective centres, while the in the paper, the choosing method is not published. The whole details of this method can be obtained from the original paper.

The main steps of the method from the paper are as follows:

1. use the historic time series to generate three subsets of time-series data of three different lengths: S_1 of time-length 30 minutes, S_2 of time-length 60 minutes, and S_3 of time-length 120 minutes.

2. at a given point of time, to predict the future change Δp, use the historical data of three length: previous 30 minutes, 60 minutes and 120 minutes - denoted x^1, x^2 and x^3

3.use x^j with historical samples S^j for Bayesian regression to predict average price change Δp^j for $1 \leq j \leq 3$.

4. calculate r = (vbid − vask)/(vbid + vask) where vbid is total volume people are willing to buy in the top 60 orders and vask is the total volume people are willing to sell in the top 60 orders based on the current order book data.

5.The final estimation Δp is produced as

$$\Delta p = w_0 + \sum_{j=1}^{3} w_j \Delta p^j + w_4 r$$

where **w** = (w_0, . . ., w_4) are learnt parameters.

For finding S_j,$1 \leq j \leq 3$ and learning **w**, the steps are as follows:

1. Utilize the first time period to find patterns S_j, $1 \leq j \leq 3$(previously divide the entire time duration into three, roughly equal sized periods)

2. The second period is used to learn parameters w and the last third period is used to evaluate the performance of the algorithm. The learning of w is done simply by finding the best linear fit over all choices given the selection of S_j , $1 \leq j \leq 3$. Now selection of S_j, $1 \leq j \leq 3$.

3.take all possible time series of appropriate length (effectively vectors of dimension 180, 360 and 720 respectively for S_1, S_2 and S_3)

Each of these form x_i and their corresponding label y_i is computed by looking at the average price change in the 10 second time interval following the end of time duration of x_i.

4. To facilitate computation on single machine with 128G RAM with 32 cores, they clustered patterns in 100 clusters using k−means algorithm. From these, they chose 20 most effective clusters and took representative patterns from these clusters.

5. The one missing detail is computing ‘distance’ between pattern x and x_i , this is squared l2-norm. use $\exp(c\cdot s(x, x_i))$ in place of exp(−||x− x_i||22 * 0.25) with choice of constant c optimized for better prediction using the fitting data (like for w).

In the paper, they use 32 core machine with 128G RAM to train the model. Thanks to the powerful machine, they do not need to parallelize the computation. But I do mot have this kind of machine. That is the problem. Using big data technologies can help to speed up the process of training the model (Bayesian regression), especially when not having powerful machines.

5. Some Details of Implementation

Pyspark.sql is used to load raw data from MongoDB by using org.mongodb.spark:mongo-spark-connector_2.11:2.2.0.

K-means in MLlib is used to cluster the raw data. And sample entropy calculated through RDD is used to choose effective centers which is not mentioned in the paper.

RDD is used to compute the average price change, but pyspark does not support bigfloat.exp; only supports math.exp; this may need to be improved for the Spark development team.

LinearRegressionWithSGD in MLlib is used to find parameters w.

6. Original Trading Strategy and Results

The trading strategy is very simple in the paper: at each time, either maintain position of +1 Bitcoin, 0 Bitcoin or −1 Bitcoin; if Δp > t, a threshold, and current bitcoin position is ≤ 0; then they buy a bitcoin. if Δp < −t, and current position is ≥ 0, then they sell a bitcoin, else do nothing.

Figure 3 shows the effect of different threshold and profit basing on the strategy above. The thresholds in the figure are timed 108 for plotting it better. The maximum profit is 164.75 $ when threshold set properly in only two days from 2018-11-08 20:00 to 2018-11-10.

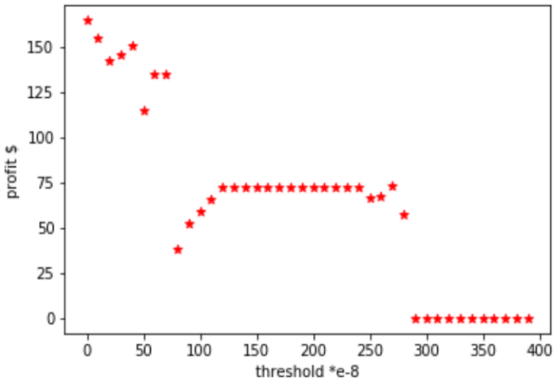


Figure 3: the effect of different threshold and profit basing on the strategy

Figure 4 shows the effect of different threshold and the number of trades basing on the strategy above.

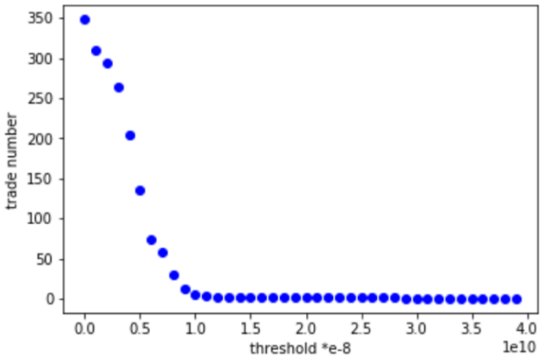


Figure 4: the effect of different threshold and the number of trades basing on the strategy

Figure 4 shows the effect of different threshold, profit and the number of trades basing on the strategy. The red star indicates the profit, and the blue circle indicates the the number of trades.

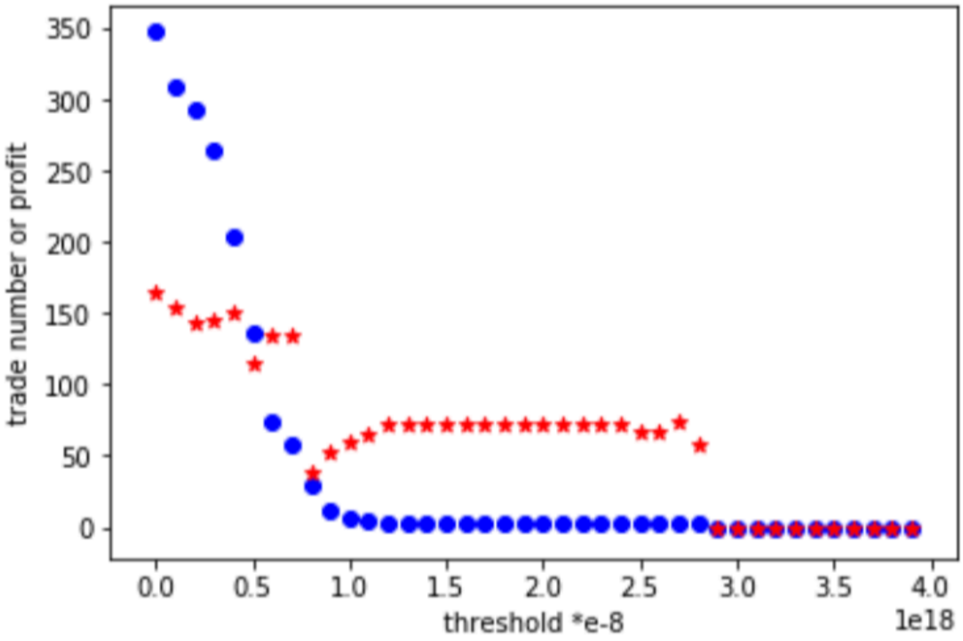


Figure 4: the effect of different threshold, profit and the number of trades basing on the strategy

7. Modified Trading Strategy and Results

The original trading strategy does not consider transection fees. The modified trading strategy consider transection fees. And the trading fee is 0.05%. The modified trading strategy is as follows: at each time, either maintain position of +1 Bitcoin, 0 Bitcoin or −1 Bitcoin; if $\Delta p > t$, a threshold, and current bitcoin position is ≤ 0, then we buy a bitcoin, and minus transaction fees, if $\Delta p < -t$, and current position is ≥ 0, then we sell a bitcoin, and minus transaction fees, else do nothing.

Figure 6 shows the effect of different threshold and profit basing on the modified strategy. The maximum profit is 66.8932150000004 $ when the threshold set properly from 2018-11-08 20:00 to 2018-11-10.

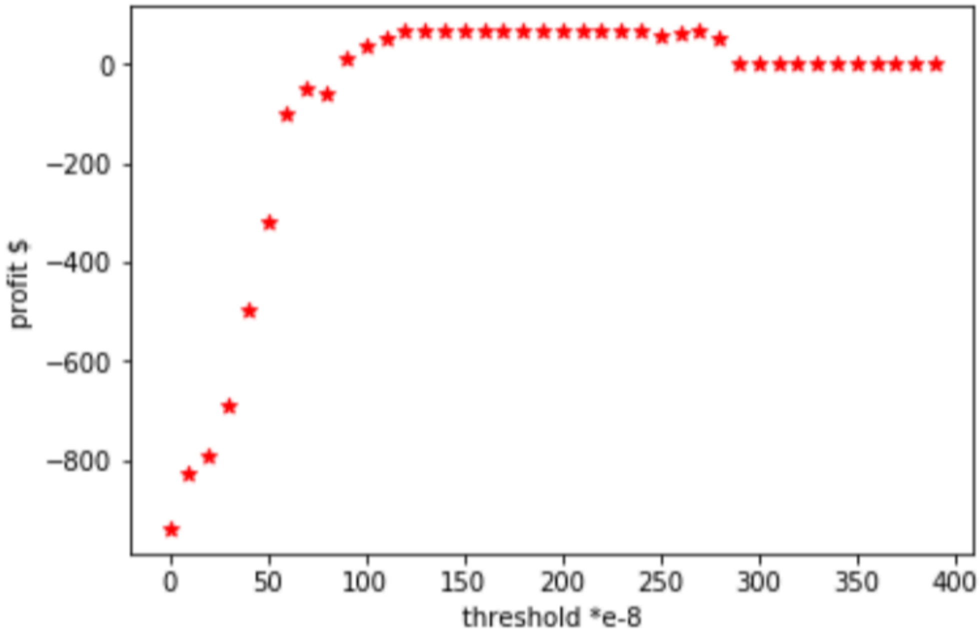


Figure 6: the effect of different threshold and profit basing on the modified strategy

Figure 7 shows the effect of different threshold and the number of trades basing on the modified strategy.

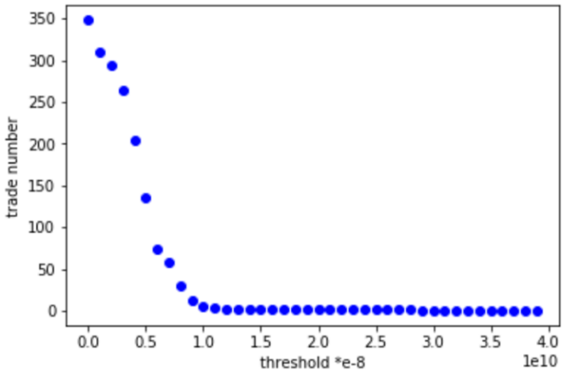


Figure 7: the effect of different threshold and the number of trades basing on the modified strategy

Figure 8 shows the effect of different threshold, profit and the number of trades basing on the modified strategy. The red star indicates the profit, and the blue circle indicates the the number of trades.

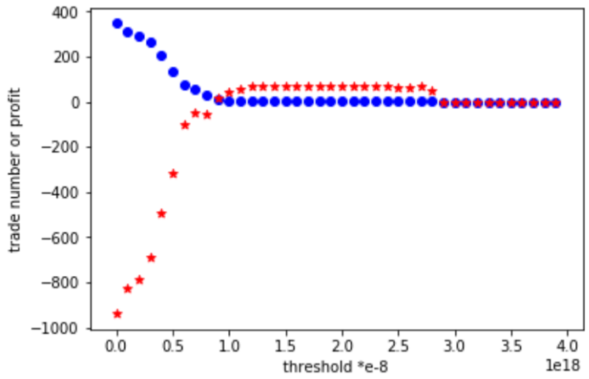


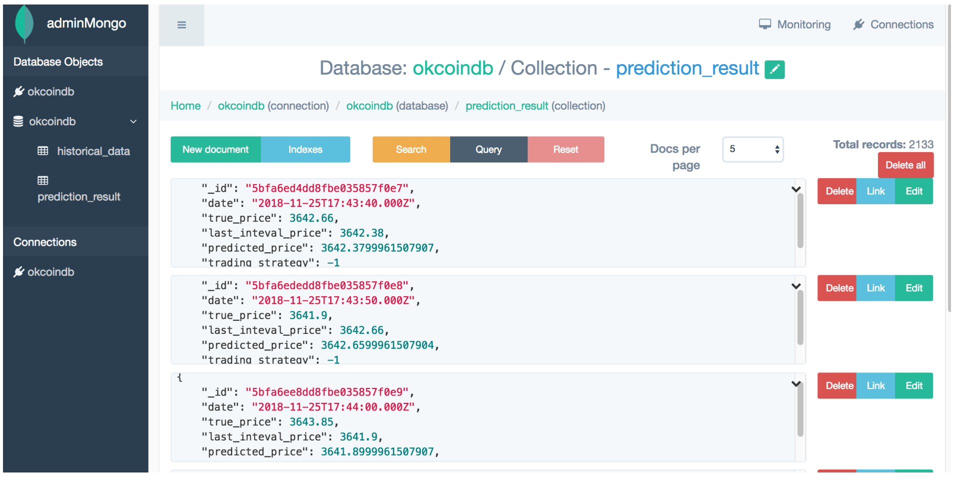
Figure 8: the effect of different threshold, profit and the number of trades basing on the modified strategy

8. Real-time Prediction

Data related to Bitcoin price and order book is obtained from Okcoin.com every 10 second interval in real time by its APIs, and is saved in MongoDB. Every 10 seconds, the latest 720 records are loaded from MongoDB and transformed into the input of the trained model. Also, the model can be retrained to get more effective parameters if necessary. The prediction result is the predicted Bitcoin price of next 10 seconds which is saved in MongoDB as well.

9. Visualization of Predicted Results

The predicted results can be visualized by a adminMongo (a web based user interface to handle all your MongoDB connections/databases needs [8].) Figure 9 shows the visualization of predicted results in MongoDB.



Database: okcoindb / Collection - prediction_result

Home / okcoindb (connection) / okcoindb (database) / prediction_result (collection)

Total records: 2133

```
    "_id": "5bfa6ed4dd8fbe035857f0e7",
    "date": "2018-11-25T17:43:40.000Z",
    "true_price": 3642.66,
    "last_inteval_price": 3642.38,
    "predicted_price": 3642.3799961507907,
    "trading strategy": -1
```

```
    "_id": "5bfa6ededd8fbe035857f0e8",
    "date": "2018-11-25T17:43:50.000Z",
    "true_price": 3641.9,
    "last_inteval_price": 3642.66,
    "predicted_price": 3642.6599961507904,
    "trading strategy": -1
```

```
{
    "_id": "5bfa6ee8dd8fbe035857f0e9",
    "date": "2018-11-25T17:44:00.000Z",
    "true_price": 3643.85,
    "last_inteval_price": 3641.9,
    "predicted_price": 3641.8999961507907,
```

Figure 9: shows the visualization of predicted results in MongoDB

10. Benefits of Using Big Data Technologies

In this real-time system of bitcoin price prediction, using big data technologies can help to speed up the process of the raw data preprocessing, model training, as well as the prediction, especially when we do not have powerful machines.

Alibaba cloud helps to collect data from Okcoin.com every 10 second interval by its APIs in real time, and makes it possible for this system to be a real-time system. It contains storage, database, virtual machine, cluster, etc.

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. The document model maps to the objects in the application code, making data easy to work with. What is more important, MongoDB is free and open-source. It is also supported by spark.

adminMongo is a web based user interface (GUI) to handle all MongoDB connections or databases needs. It is free and open-source as well. It helps to visualize the prediction results stored in MongoDB.

RDD is the primary data abstraction in Apache Spark and the core of Spark, which is a fault- tolerant collection of elements that can be operated on in parallel. The benefits of RDD are mainly from its features, including fault tolerance mechanism, distributed and in-memory storage. Because of these features, RDD can help to speed up the process of the raw data preprocessing, model training, as well as the prediction.

Spark SQL is a Spark module for structured data processing. It is very convenient and quick to load raw data from MongoDB by using org.mongodb.spark:mongo-spark-connector_2.11:2.2.0.

MLlib is Apache Spark's scalable machine learning library. It is easy to use. Spark excels at iterative computation, enabling MLlib to run fast. At the same time MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.

References

[1] https://en.wikipedia.org/wiki/Bitcoin

[2] https://www.okcoin.com/docs/en/#summary-README

[3] https://www.okcoin.com/

[4] https://www.mongodb.com/

[5] https://spark.apache.org/streaming/

[6] https://spark.apache.org/sql/

[7] https://spark.apache.org/mllib/

[8] https://github.com/mrvautin/adminMongo

[9] Devavrat Shah, Kang Zhang Bayesian regression and Bitcoin https://arxiv.org/abs/1410.1231