MSBD 5013 Spring 2021

Final Report

Group Name: xyw666

HUANG, Yitian	WANG, Xiangyu	YANG, Yi
20718885	20711306	20710742
yhuangej@connect.ust.hk	xwanggb@connect.ust.hk	yyangeh@connect.ust.hk

1. Introduce

Research on the predictability of financial markets has a long history in the financial literature e.g. Although opin- ions differ regarding the efficiency of markets, many widely accepted studies show that financial markets are to some extent predictable. Two major classes of work which attempt to forecast financial time-series are, broadly speaking, statistical parametric models and data-driven machine learning approaches. Machine learning techniques can capture such arbitrary nonlinear relationships with little, or no, prior knowledge regarding the input data. Recently, more and more deep learning methods are also applied in forecasting financial time-series data.

In this work, we use XGBoost[1] regressor to predict the target of 4 encryption currency. The dataset consists of train data and test data, with the size of 97858 and 17252 respectively. Train data consists of 12 columns which are id, time, name, open, high, low, close, volume, quote asset volume, number of trades, taker buy base asset volume, taker buy quote asset volume and target *y*. The target *y* is masked for prediction in the testing set. The predicted target has a root mean square error (RMSE) of 0.00422, which means XGBoost regressor has a good performance. However, there is still a lot of room for improvement. In the future, we will try deep learning model to reduce the error of prediction.

2. Related Work & Methodology

2.1. Regression Tree [2]

Regression tree is a decision tree model. When predicting, it returns the average of corresponding leaf (or the score of corresponding leaf). For a regression tree *f*, the loss function of regression is:

$$L(f) = \|y - \hat{y}\|_2^2 + \Omega(f)$$

Where $\Omega(f) = \gamma \times |f.leaves| + \lambda\|f.scores\|^2$ is regular term to avoid overfitting.
Regression tree is trained by greedy method. In each loop, the tree finds a leaf *n*, an attitude *a*, and a threshold θ to split so that the loss function is minimized. If the loss is reduced, then *n* will be spitted on *x* with threshold θ, otherwise the training will be finished. The rough algorithm is shown as below:

```
Given $X, y, \gamma, \lambda, d$
while True:
  $reduce = 0, n = null, a = null, \theta = null$
  for $n'$ in $f.leaves$ with depth $\leq d$:
    $X', y'$ = samples in $n'$
    $m$ = the amount of samples in $X'$
    for $a'$ in $X'.attitude$:
        sort $X', y'$ by $a'$
        $l = 0, r = \|y'\|_1, s_l = 0, s_r = \|y'\|_2^2$
        $loss_{local} = s_r - \frac{r^2}{m} + \frac{\lambda r^2}{m^2}$
        for $i$ = 1 to $m$ - 1:
            $\theta' = y_i'$
            $l = l + \theta', r = r - \theta'$
            $s_l = s_l + \theta'^2, s_r = s_r - \theta'^2$
            $reduce_{new} = loss_{local} - s_r + \frac{r^2}{m-i} - \frac{\lambda r^2}{(m-i)^2} - s_l + \frac{l^2}{i} - \frac{\lambda l^2}{i^2} - \gamma$
            if $reduce_{new} > reduce$:
                $reduce = reduce_{new}$
                $n = n', a = a', \theta = \theta'$
if $n \neq null$:
    split $n$ on $x$ with threshold $\theta$
else:
    break
```

Algorithm 1: Regression Tree

2.2.XGBoost Regressor

However, when dealing with specific practical problems, it is certainly not sufficient to use a single regression tree. Thus, we need to improve and upgrade the regression tree model via XGBoost regressor. XGBoost regressor is a multiple regression tree model, the predicted value for input *x* is:

$$\hat{y} = \sum_{f \in M.tree} f(X)$$

The loss function of XGBoost regressor *M* is defined as:

$$L(M) = \|y - \hat{y}\|_2^2 + \sum_{f \in t(M)} \Omega(f)$$

The rough algorithm of XGBoost regressor is:

```
Give $X$ and $y$
$r, l = 0, \infty$
while True:
    train a regression tree $f$ for $y$ and $X$
    $y = y - f(X)$
    $r' = r + \Omega(f)$
    $l' = \|y\|^2 + r'$
    if $l' < l$:
        $l = l'$
        $r = r'$
        add $f$ to $M$
    else:
        break
```

Algorithm 2: Overall Logic of XGBoost Regressor

Let $y^{(t)}$ denote *y* after the *t*-th iteration, then *M* train a regression tree *f* to minimize the following term in the (*t*+1)-th iteration:

$$L\left(y^{(t+1)}\right) - L\left(y^{(t)}\right) \approx \left\langle \nabla L\left(y^{(t)}\right), y^{(t+1)} - y^{(t)} \right\rangle + \frac{1}{2}\left\langle \nabla^2 L\left(y^{(t)}\right), \left(y^{(t+1)} - y^{(t)}\right) \odot \left(y^{(t+1)} - y^{(t)}\right) \right\rangle + \Omega(f)$$

In the (*t*+1)-th iteration, XGBoost regressor regards this Taylor series [3] as the loss function and train a regression tree by following algorithm (adjusted regression tree algorithm):

```
Given $X, y, y^{(t)}, \gamma, \lambda, \nabla L(y^{(t)}), \nabla^2 L(y^{(t)}), d$
while True:
  $reduce = 0$, $n = null$, $a = null$, $\theta = null$
  for $n'$ in $f.leaves$ with depth $\leq d$:
    $X'$, $y'$ = samples in $n'$
    ${y^{(t)}}', \nabla L(y^{(t)})', \nabla^2 L(y^{(t)})'$ = corresponding parts in $y^{(t)}, \nabla L(y^{(t)}), \nabla^2 L(y^{(t)})$
    $m$ = the amount of samples in $X'$
    for $a'$ in $X'.attitude$:
      sort $X'$, $y'$, ${y^{(t)}}', \nabla L(y^{(t)})', \nabla^2 L(y^{(t)})'$ by $a'$
      $l = 0, r = \langle \nabla L(y^{(t)})', y' - {y^{(t)}}' \rangle$
      $l' = 0, r' = \|y'\|_1$
      $s_l = 0, s_r = \frac{1}{2}\langle \nabla^2 L(y^{(t)})', (y' - {y^{(t)}}') \odot (y' - {y^{(t)}}') \rangle$
      $loss_{local} = s_r + r + \frac{\lambda {r'}^2}{m^2}$
      for $i = 1$ to $m - 1$:
        $\theta' = y'_i, g = \nabla L(y^{(t)})'_i, h = \frac{1}{2}\nabla^2 L(y^{(t)})'_i$
        $t = \theta' - {y^{(t)}}'_i$
        $l = l + gt, r' = r' - gt$
        $l' = l' + \theta', r' = r' - \theta'$
        $s_l = s_l + ht^2, s_r = s_r - ht^2$
        $reduce_{new} = loss_{local} - s_r - r - \frac{\lambda {r'}^2}{(m-i)^2} - s_l - l - \frac{\lambda {l'}^2}{i^2} - \gamma$
        if $reduce_{new} > reduce$:
          $reduce = reduce_{new}$
          $n = n', a = a', \theta = \theta'$
  if $n \neq null$:
    split $n$ on $x$ with threshold $\theta$
  else:
    break
```

Algorithm 3: Training of Each Regression Tree in a XGBoost Regressor

3. Feature Engineering

Encryption currency is a transaction medium that uses cryptographic principles to ensure transaction security and control the creation of transaction units. The price trend of currencies essentially shows the changes in the relationship between supply and demand in the market. Its price changes are affected by many factors, such as time, type of currency, policy factors and so on. The difference in time, such as working days and holidays, different time periods of the day, etc., will have different effects on investors' behavior, thereby affecting investors' investment behavior, and thus the price of currency. In addition, the different types of currencies represent different developers and different mining methods. These factors will also affect investors' choice of currency types, and thus affect the price of currencies. Therefore, we choose the type of currency and time, including period, weekend and season, which are two types of quantifiable indicators to be the feature values to predict the price of currencies.

3.1.Currency

There are 4 cash currencies in the dataset: BTCUSDT, ETHUSDT, LTCUSDT and XRPUSDT. They have different developers and different mining methods. Investors have different levels of confidence in different developers, leading to different price levels of these four cryptocurrencies. Thus, these currencies are one-hot encoded.

3.2.Period

The trading hours of currencies are different from stocks. Its trading hours are 24 hours a day. Therefore, we divide the daily time period into the following 4 segments according to the normal life of human beings for one-hot encoding. Investors have different levels of investment activity in different time periods. For example, the two time periods of ‘*Morning and Noon*’ and ‘*Afternoon*’ are times of normal human activities, and they are also the most active time for most investors to invest.

Early Morning	Morning and Noon	Afternoon	Night
3 AM to 8AM	9AM to 2PM	3PM to 8PM	8PM to 2AM

Table 1: Periods of a day

3.3.Weekend?

For most investors the active level of investment behavior will be different on weekdays and holidays. Thus, we classify time into the following two categories for one-hot encoding:

Weekend	Working day
Friday 8PM to Sunday 8PM	Sunday 9PM to Friday 7PM

Table 2: Weekend or Not?

3.4.Season

Different quarters will have different effects on investors. As shown in the table below, we have adopted the quarterly time in the United States.

Spring	Summer	Fall	Winter
March to May	June to August	September to November	December to February

Table 3: Seasons of a year

3.5.Final Form

The final form of x is a 22-dimensional vector, as follows:

Currency Type Column 1 to 4	9 quantified features in original data Column 5 to 13	Period Column 14 to 17	Weekend? Column 18	Season Column 19 to 22

Table 4: The Structure of Preprocessed Data

4. Result and Future Work

4.1.Configuration

In this work, the parameter of XGBoost is set as below:

γ	λ	d
0	0	6

Table 5: The parameter of the XGBoost regressor

4.2.Result

First, we train a XGBoost regressor on 85% random samples of training set, then use it to predict othter 15% samples to validate. Then, we use whole training set to train another XGBoost regressor. Finally, this regressor is used to predict the testing set. The result is as shown as below:

	Training	Validation	Testing
RMSE	1.08×10^{-6}	1.08×10^{-6}	0.00422

Table 6: The performance of the XGBoost regressor

The RMSE on training and validation samples are both 1.08×10^{-6}, which indicates that there is no overfitting. The RMSE on the testing set is 0.00422, which means the XGB regressor works well.

4.3.Discussion & Future Work

Although XGBooster regressor works well on testing set, its performance on the testing set is far worse than that on the training set. Like other decision tree models, XGBooster is poor at predicting the data with the attribute outside the range of training set. To solve this problem, we plan to use some deep learning models in the future. In addition, the raw feature may not consist of much information. In future, we will try to make better features, e.g., introduce some financial factors.

5. Reference

[1] Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM. pp. 785–794. arXiv:1603.02754. doi:10.1145/2939672.2939785

[2] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8

[3] Taylor, Brook, Methodus Incrementorum Directa et Inversa [Direct and Reverse Methods of Incrementation] (London, 1715), pages 21-23 (Proposition VII, Theorem 3, Corollary 2). Translated into English in D. J. Struik, A Source Book in Mathematics 1200-1800 (Cambridge, Massachusetts: Harvard University Press, 1969), pages 329-332.