Effect of Investor Attention on Stock Return

Group 61
Yuankang Xiong
Rui Xiao
Yuan Yin
Yifei Lu

Abstract

In this report, we analyzed the relationship between investor attention and stock return in China A-share market. We firstly divided stock returns into four classes by their quantile and changed our problem into a classfication problem. And then, we used extreme gradient boosting tree (XGBoost) model to predict stock returns. By adding investor attention factors, we improved roughly 7% model performance in stock return prediction compared with using only classic financial factors. It shows that investor attention factors act as a useful feature on explaining stock returns.

keywords: investor attention, XGBoost, stock return

1 Problem statement and motivation

Most financial models assume that all investors are rational, which means for those investors following the same principle, their reactions to the same specific event should be the same optimal one. However, it is often not the case in the real world. Studies have shown that stock returns do have some relationships with investors' behavior. Due to human nature, even a relatively rational investor can waver under consecutive bad news, regardless of the credibility of the news.

In general, investors can be classified into two types, which are industry investors and individual investors. For industry investors, they usually have their own research department or have relative reliable third-party information with regard to the stocks they invest in, hence their movements are more likely based on their own research result, rather than some unreliable news. For individual investors, upon hearing the news, they may not have the resource or ability to verify personally, hence they are more likely to make trading decision based on those news and become the main driver of sentiment based stock movements.

This makes us wonder, is there a relationship between the individual investors' behaviors and common stock returns? Does the investors' attention on news affect the movements of stocks and can we detect them?

These questions are valuable if we can make use of them during investment decision making process. For example if a certain magnitude of investor attention can potentially increase the fluctuation of stock price, or even push the stock price to a specific direction, then a detector can be built to signal or warn our shareholders to prevent a potential loss due to the quick move of stock price.

Mathematically, the above process can be simplified to a classification problem. Based on different attention features[1], if a classifier can be built to yield a prediction of stock returns, say "perform among the best 25% stocks" or "perform among the worst 25% stocks", then it can act as a tool during investment decision making and potentially create a considerable return.

[1]These features will be further discussed in section 3.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

The rest parts of this paper provide a more detailed analysis of above proposition. Section 2 explains the historical process of investors' behavior as well as the result of different related research. Later in Section 3, the exact model architecture will be provided along with all variables used inside the model. After that, the result will be evaluated and displayed in Section 4.

2 Literature review

In the past, investors usually get financial market information from financial news articles. Nowadays, news is easily spread and thus individual investors can analyze financial markets based on the most recent news on the stocks they care about. It's reasonable that the stock market will be affected by these fast-spread news since most individual investors will react on the news and thus affect the stock price. Concentrating on the potential relationship between observed news and returns of some specific assets in financial market would provide insightful and reasonable rules for individuals' speculative and investment behaviors[1].

"Sentiment Analysis" is what researchers provided to analyze the relationship between investor's behavior and investment return. When news is available to investors, they will first have their own interpretation on the news and react as market sentiments, then the financial market will be affected by these sentiments.

Recent study has shown that sentiment analysis could be applied to find the relationship between social media and stock return. By using sentiment analysis, Yangyu[2] pointed out that different types of social media would have various effects on the stock return. Besides, other studies have shown that it could also be used to analyze cryptocurrency market returns. Tianyu Ray[3] has mentioned that social media platforms such as Twitter could be used to capture investor sentiment, and would gain the early signal for the future price fluctuation for Bitcoin market. Furthermore, strong evidence for herding behaviors among individual investors in China A and B-share markets has been found few years ago[4]. It's not hard to notice that these studies have provided us a different perspective to explain the stock return.

To use technical tools to track the movements in stock markets, researchers found that the stock price is generally a dynamic, non-parametric, chaotic and noisy process. To fully describe the process with mathematical expression is rather difficult. However, in recent years researchers found that using machine learning algorithms to predict the movements in financial markets performed very well. Early models used in stock forecasting involved statistical methods such as time series modeling and multivariate analysis [5][6][7]. Shubharthi Dey mentioned that using XGBoost to predict the direction of stock price had outstanding performance on accuracy[8]. Also Tianyu Ray Li usesd XGBoost to predict the price of cryptocurrency based on sentiment analysis with outperformed results.

3 Material and methods

Based on what we have mentioned before, we will use individual investor's behavior data along with XGBoost to analyze the relationship between individual investor's behavior and investment return for the following reasons:

(i) XGBoost is a scalable tree boosting system, which could be used in most scenarios. Since the algorithm itself has the scalability property, it would provide us with unlimited freedom by adding more children nodes on the preexisting parents nodes when needed. Which means, we would not stagnate and meet limitations by the number of data. Now, we may track with the features in one trading day, but it is possible for us to deal with some features gathered in per hour. This makes our model more adaptable for other assets with larger daily information volumes.

(ii) Tree models are not sensitive to specific range of data and features. There is no need to do much preparations for the data before applying the XGBoost model as long as one has proper labels, reasonable features, and acquire plentiful and manageable size of data.

(iii) XGBoost is a rule based learning method, which could unveil the potential relationship among features.

3.1 Variable explanation

3.1.1 Return

The return r of stock i at time t is defined as follows

$$r_t^{(i)} = \frac{p_{t+1}^{(i)} - p_t^{(i)}}{p_t^{(i)}}$$

where $p_t^{(i)}$ is the stock price at time t. The return represents the percentage of money made or lost on an investment during a given period.

Stock return can be regarded as a continuous variable. However, making a point estimation of return value is often unreliable[9]. Therefore, we classified the return into 4 categories and the model provides only a general direction of stock movement. The four categories are defined based on quantiles, which are "return lies on bottom 25% among all stocks", "return lies on 25-50 quantile", "return lies on 50-75 quantile" and "return lies on top 25%".

3.1.2 Click

Click number $c_t^{(i)}$ represents the number of clicks at time t for stock i by all stock application users from our database.

3.1.3 Favorite percent

Favorite percent $f_t^{(i)}$ denotes the percentage number of investors following on stock i on day t. The word "follow" means investors would prefer some stocks and pay more attention to them, and then they added the stocks into their own watchlist.

3.1.4 Financial factors

There are several factors such as market return, company size effect, etc. that would be part of the explanation of stock return[10]. Fama and French [11][12] mentioned that a basket of factors could be used in explaining stock return.

To analyze whether investor attention features are useful, a reasonable approach is to extract the part of return that is not fully explained by the basket of factors. Or in other words, we would apply these common factors in our model as the original model, and then update the model using investor attention data. If the performance of the new model is better than that of the original model, then a further discussion can be conducted.

3.2 Model and algorithm

3.2.1 Model

Here we are dealing with a supervised learning problem, where we could use the training data which may contain several features to predict the target variable. The objective function includes the loss function estimating the error and regularization term to avoid overfitting.

For n samples each with d features, we define $\phi(\boldsymbol{x})$ below as our prediction function, each f_k represents a decision tree in forest $\mathcal{F}$.

$$\phi(\boldsymbol{x}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \quad f_k \in \mathcal{F}$$

Here $\mathcal{F} = \{f(\boldsymbol{x}) = w_{q(\boldsymbol{x})}\}$ is the set of trees (i.e. forest), K is the number of trees we use to predict; for the expression $f_k(\boldsymbol{x}) = w_{q_k(\boldsymbol{x})}$, $w_k \in \mathbb{R}^{T_k}$ is the leaf weight of tree k; T_k is the number of leaves in tree k and $q_k : \mathbb{R}^d \to T_k$ maps feature $\boldsymbol{x}$ to a specific leaf.

Here we use softmax classifier for our multi-label classification problem. The above process is repeated for m times where m stands for number of classes, which transfers a m-labels problem into

m binary problems. The output variable $\boldsymbol{a} = [\phi_1, \phi_2, ..., \phi_m]$ are then transferred into $S(\boldsymbol{a}) \in \mathbb{R}^m$, for j-th entry $S_j(\boldsymbol{a})$, it is calculated as follows

$$S_j(\boldsymbol{a}) = \frac{e^{a_j}}{\sum_j e^{a_j}}$$

and finally the prediction label is assigned as

$$\hat{y}_i = \arg\max_j S_j(\boldsymbol{a})$$

For the next step, we can choose reasonable loss function to minimize the objective function $\mathcal{L}$ which has form as follows:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

where $\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda ||w||^2$.

Here we use multi-label-log loss function[2], where Ω represents the regularization term, which helps to smooth the learned parameters and thus avoid overfitting. Note that $\mathcal{L}(\phi)$ includes functions as parameters and cannot be optimized using traditional convex optimization, hence we used the following iterative approach to update the parameters:

Let $\hat{y}_i^{(t)}$ be the prediction of the i - th instance at the t - th iteration. Then we greedily add f_t that most improves the model at next iteration:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

According to Tianqi[13], we could use the second order approximation to quickly acquire an optimal to the objective above

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

where,

$$g_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}}, h_i = \frac{\partial^2 l(y_i, \hat{y}^{(t-1)})}{\partial (\hat{y}^{(t-1)})^2}$$

Let $I_j = \{i | q(x_i) = j\}$ be the sample set of leaf j, we would rewrite $\tilde{\mathcal{L}}^{(t)}$ as the following

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Then for each structure $q(x)$, we could compute the optimal weight w_j^* of leaf j

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

and the corresponding optimal value

$$\tilde{\mathcal{L}}^{(t)} = -\frac{1}{2} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda}$$

This optimal value would be used as the score function for the algorithm to measure the quality of a tree structure q, which is similar as the impurity score for a simple decision tree.

[2] http://wiki.fast.ai/index.php/Log_Loss

3.2.2 Algorithm

Here is the detailed process for XGBoost model algorithm: The idea for XGBoost algorithm is firstly we use only one tree to learn a regression predictor, then we compute the error residual and add an additional tree to learn to predict the residual. The error rate is calculated using the parameters mentioned below. We repeat the steps above until error estimate is small enough.

One of the key problems in tree learning is to find the best split for one tree. In order to do so, a split finding algorithm enumerates over all the possible splits on all the features. We call this the exact greedy algorithm. It is computationally demanding to enumerate all the possible splits for continuous features. Here we attach the pseudo code for it:

Algorithm 1: Exact greedy algorithm for split finding used in our price prediction model.

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i, H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ *to m* **do**
  $G_L \leftarrow 0, H_L \leftarrow 0$
  for *j in sorted (I, by* $\boldsymbol{x_{jk}}$*)* **do**
    $G_L \leftarrow G_L + g_j, H_L \leftarrow H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$
    $score \leftarrow max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

With exact greedy algorithm for finding best split methods for each tree, we can construct the whole XGBoost algorithm. Here we attach pseudo code for XGBoost algorithm:

Algorithm 2: eXtreme Gradient Boosting

Input: $(x_i, y_i)_1^n$ /*the labeled training data*/
Input: $(\gamma, L, l, \lambda, N)$
Output: A tree which is configured to predict the class label of a test sample
$l \leftarrow l + 1$;
if $l \leq L$ **then**
  $t \leftarrow 0$; $f \leftarrow 0$;
  while $t < N$ **do**
    estimate f_t as a regressor function, density, or distribution;
    $f \leftarrow f + f_t$
  end
  initialize array of scores $S[:]$;
  $\boldsymbol{ApplyAlgorithm\ 1}$ **:** greedy algorithm for computing max score;
  $C_L, C_R \leftarrow \boldsymbol{MaxGain\ ((x_i, y_i)_1^n, S)}$;
  /*C_L, C_R are the left and right children of this node respectively*/
  if C_L *does not satisfy the desired level of purity* **then**
    GradientBoostedTree($C_L, \gamma, L, l, \lambda, N$);
  end
  if C_R *does not satisfy the desired level of purity* **then**
    GradientBoostedTree($C_R, \gamma, L, l, \lambda, N$);
  end
end

Note here γ is the minimum required structure score, L is the maximum number of levels in the tree, l is the current level of the tree, λ is the learning rate, N is the number of training steps.

4 Model evaluation

Features include the click numbers(representing as "CLICK"), favorite percentage(representing as "FAVORITE_PERCENT_x"). Moreover, classic financial factors are also included in features.

Using above model, the accuracy of predicting each stock's return varies, with an overall accuracy of 49.58% for all test data. As comparison, using only financial factors yields a relative poor result of 42.66%.

Figure 1 and figure 2 are two confusion matrices based on real label. The first figure uses only financial factors and second figure is generated through using additional investor attention features.

The labels range from 0 to 3, representing "return lies on bottom 25% among all stocks", "return lies on 25-50 quantile", "return lies on 50-75 quantile" and "return lies on top 25%" respectively. For each grid, the row label indicates the true label; column label indicates the predicted label and the numbers inside represent the accuracy. For example number 0.22 in second row and fourth column of figure 1 indicates that 22% of sample with real label 0 was classified into label 3 by our model.

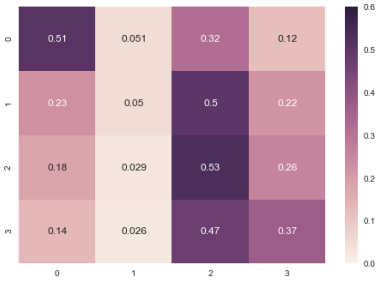

Figure 1: True label basis, no click data

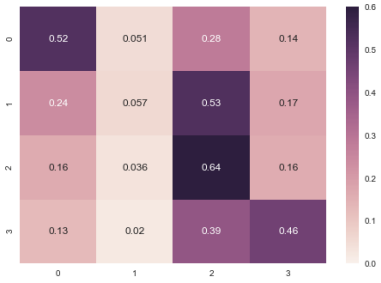

Figure 2: True label basis, with click data

Figure 3 and figure 4 are generated similarly, but instead of each row, now each column has a summation of 1. For example number 0.084 in second row and fourth column of figure 3 indicates that among all test data with predicted label 3, 8.4% of them are misclassified and have true label 1.

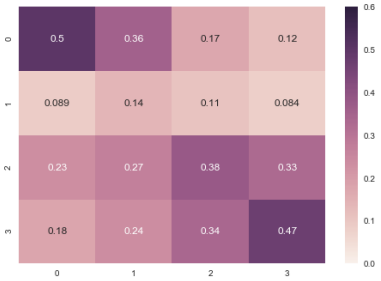

Figure 3: Predicted label basis, no click data

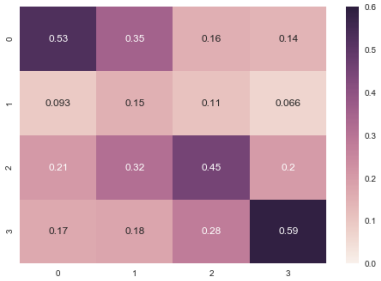

Figure 4: Pridicted label basis, with click data

The confusion matrix shed a light on how the "CLICK" and "FAVORITE_PERCENT_x" can help with explaining the stock return and its direction. This can be observed by having the increment of diagonal value of figure 2 and 4 in contrast with figure 1 and 3.

More importantly, from figure 4 one can observe that the accuracy among group 0 and 3 is better than group 1 and 2. That is to say, we can trust more on the model when the predicted label is 0 or 3.

However, these figures show one drawback of the model: it is not quite accurate when the true label is 1 or 2, or equivalently, the model performs only good at predicting the extreme situation(top 25% or botton 25%).

Based on the result of different feature combinations, we can regard "CLICK" as an essential feature explaining return.

From the feature importance graph in figure 5, we can see that "CLICK" is the most important feature here. Feature importance is calculated by taking the average of f-score among all stocks with respect to

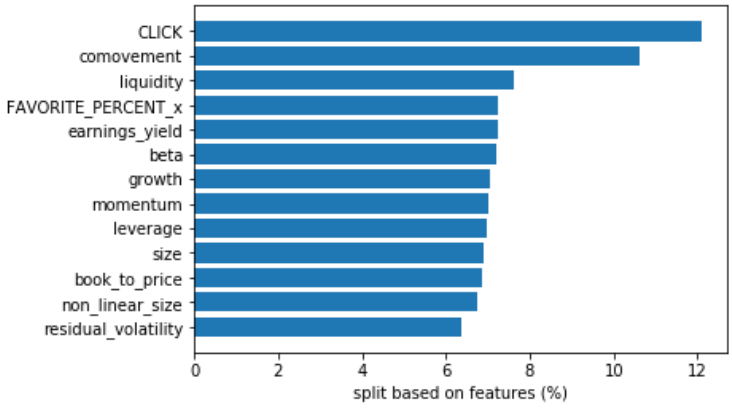

Figure 5: Feature Importance

each feature separately. "CLICK" contributes to 12% splits, which is the most important feature. The traditional factor "comovement" contributes to 11% splits. "FAVORITE_PERCENT_x" only takes around 7% in the splits feature selection. Feature importance varies among different stocks but the order of the features remains almost unchanged. In a word, "CLICK" and "FAVORITE_PERCENT_x" are features that can be added as new explanatory variables for stock return.

Below provides an example of our tree splitting process for stock with code "000968". One can observe that "CLICK" and "FAVORITE_PERCENT_x" are used extensively in tree structure building.

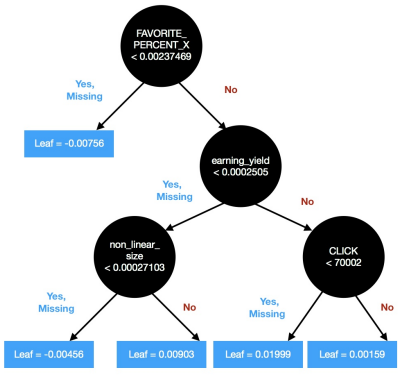

Figure 6: Tree structure example 1

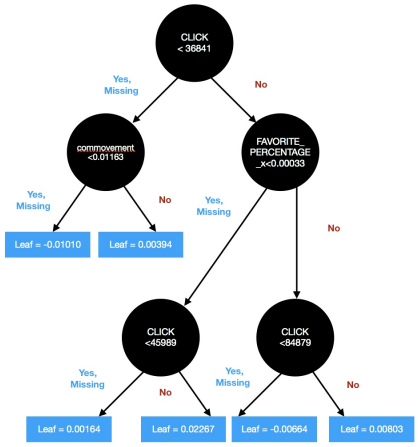

Figure 7: Tree structure example 2

5 Conclusion

In this project, we applied XGBoost algorithm to learn to fit the stocks' return by using features of investor attention and sentiment analysis such as click numbers and stocks favorite percentage by investors, and several other financial factors.

We found that model performs better at predicting extreme situation for stock return, which can potentially be used to signal or warn shareholders to prevent a potential loss due to the quick move of stock price. However, model performs poorly in predicting relative moderate return, which is one potential drawback. Further research could be conducted by solving this problem and improving the performance.

Besides, feature click number makes great efforts on predicting stocks' return. Favorite percentage also contribute as an important feature. Click number serves as the main engine to provide effective

explanatory power among those sentiment analysis and investor attention features. Also we can see that XGBoost provides with a great learning algorithm in financial market, which can be applied on other financial market prediction problems by later researchers.

All available data and code are provided here.[3]

6 Description of individual effort

Yuankang Xiong: Raw data cleaning and feature importance evaluation; algorithm design.
Rui Xiao: Raw data cleaning and feature importance evaluation; algorithm design.
Yuan Yin: Coding and parameter tuning; model evaluation.
Yifei Lu: Coding and parameter tuning; model evaluation.
The essay wrting are completed by the whole group with equal effort.

References

[1]O. Netzer, R. Feldman, J. Goldenberg, M. Fresko. "Mine your own business: market-structure surveillance through text mining." Marketing Science(05)(2012): vol. 31(3), pp. 521-543

[2]YangYu, Wenjing Duan, Qing Cao. "The impact of social and conventional media on firm equity value: A sentiment analysis approach Decision Support Systems." Decision Support Systems(11)(2013): vol. 55(4), pp. 919-926

[3]Li, Tianyu Ray, et al. "Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model." arXiv preprint arXiv:1805.00558 (2018).

[4]Yao, Juan, Chuanchan Ma, and William Peng He. "Investor herding behaviour of Chinese stock market." International Review of Economics & Finance 29 (2014): pp. 12-29

[5]R. Gencay. "Linear, non-linear and essential foreign ex- change rate prediction with simple technical trading rules." Journal of International Economics(1999): vol. 47, pp. 91-107.19

[6]Bao D., Yang Z. "Intelligent stock trading system by turning point confirming and probabilistic reasoning." Expert Systems with Applications(2008): vol. 34(1), pp. 620-627.

[7]Timmermann, A., Granger, C. W. "Efficient market hypothesis and forecasting." International Journal of Forecasting(2004): vol. 20(1), pp. 15-27

[8]Dey, S., Kumar, Y., Saha, S. and Basak, S., Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. Working paper. DOI: 10.13140/RG. 2.2. 15294.48968.

[9]Qiu, Mingyue, and Yu Song. "Predicting the direction of stock market index movement using an optimized artificial neural network model." PloS one 11.5 (2016): e0155133

[10]K. Geert Rouwenhorst. "Local Return Factors and Turnover in Emerging Stock Markets." Journal of Finance(1999) : vol. 54(4), pp.1439-1464

[11]Fama, Eugene F., and Kenneth R. French. "Multifactor explanations of asset pricing anomalies." Journal of Finance(1996): vol. 51, pp. 55?84

[12]Fama, Eugene F., and Kenneth R. French. "The cross?section of expected stock returns." Journal of Finance(1992): vol. 67, pp. 427?465

[13]Chen T, Guestrin C. "XGBoost: A scalable tree boosting system." In Proceedings of the 22nd ACM SIG KDD International conference on Knowledge Discovery and Data Mining (08)(2016): pp. 785-794

[3]Code: https://drive.google.com/open?id=0B3rYDUzGrMm4NXlrTmc1NHhZQVY3RW9yclprV1ROaHpPLVdr
Data: https://drive.google.com/open?id=1AynNVyb1D9y1DI0oVexFYJgBK6bxVkj1