PRIOR KNOWLEDGE NEURAL NETWORK FOR AUTOMATIC FEATURE CONSTRUCTION IN FINANCIAL TIME SERIES

Jie Fang\textsuperscript{1}, Jianwu Lin\textsuperscript{2}, Yong Jiang\textsuperscript{3}, Shutao Xia\textsuperscript{*}

Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
fangx18@mails.tsinghua.edu.cn, \{lin.jianwu, jiangy, xiast\}@sz.tsinghua.edu.cn

ABSTRACT

In quantitative finance, useful features are constructed by human experts. However, this method is of low efficient. Thus, automatic feature construction algorithms have received more and more attention. The state-of-the-art technic in this field is to use reverse polish expression to represent the features, and then use genetic programming (GP) to reconstruct it. In this paper, we propose a new method, alpha discovery neural network (ADN), which can automatically construct features by using neural network. In this work, we made several contributions. Firstly, we put forward new object function by using empirical knowledge in financial signal processing, and we also fixed its undifferentiated problem. Secondly, we use model stealing technic to learn from other prior knowledge, which can bring enough diversity into our network. Thirdly, we come up with a method to measure the diversity of different financial features. Experiment shows that ADN can produce more diversified and higher informative features than GP. Besides, if we use GP’s output to serve as prior knowledge, its’ final achievements will be significantly improved by using ADN.

Index Terms—Quantitative finance, genetic programming, prior knowledge, financial signal processing, neural network

1. INTRODUCTION

Feature construction involves transforming a given set of input features to generate new and more powerful features, and these constructed features should be useful for prediction [1][2]. Most approaches rely on human experts to construct features, because it needs human to give an assessment on what is good and what is bad, and different industries have different criterions. What’s more, it’s easier to construct features from empirical data, other than raw data. However, constructing features by human experts is of low efficient. Thus, many researchers hope to find an automatic feature construction method. Sometimes feature construction and feature selection happens in the same procedure, so there are some methods can both do feature selection and feature construction. These methods are wrapper, filtering and embedded [3]. Filter is easy but of bad performance, it only uses some criteria to choose the feature, and sometimes it can help us to monitor the feature construction process. Wrapper performs well by directly using the model’s results to serve as object function. So we can regard an individual trained model as a new constructed feature. However, it costs a lot of computation resource and time. Embedded is a method, which uses generalization factor and pruning technic to select or combine features. In real practice, people normally prefer wrapper and embedded. The most frequently used wrapper method for automatic feature construction is genetic programming (GP) [4]. It uses tree structure to express the data and operators of explicit formulas. In each step, these trees will evolve, multiply and be selected, until they end up this process. Then the algorithm dynamically chose the right expression according to object function. Such as Lensen [5], Tran [6], Vafaie H [7], and Hindmarsh [8], uses GP to solve high dimension pixel data, and then produce new features for object detection task. They all uses the tree structure to represent a region of pixel, and then use GP to produce and selected trees. Finally, the last pixel tree is the produced feature.

There are also some researches which use neural network to represent new constructed features. Such as La [9], Liang [10] and Güvenir [11], uses convolutional neural network to represent pixel information in some area of a picture, then they use the trained classifier to serve as new feature. Virgolin [12] uses evolutionary meta learning approach in place of GP to construct features from medical data. Botsis T [13] leverages recurrent neural networks to build rule-based classifier among text data, each classifier represents a part of the text. They all have successfully used deep neural network to construct new features. With the development of deep neural network, we may have more opportunity to use wrapper approach. For some industries, such as health care, education and financial business, the empirical models are still much more powerful than only
using deep learning algorithms [14][15][16]. Because these industries highly rely on people’s behaviors and decisions, and deep neural network will easily over fit this nonstationary noise. Relatively, prior knowledge is more robust and reliable, because they really have tried to learn and understood this business. This point is what we highly valued in our work.

In quantitative investment area, people normally dig out financial features by their own thinking, economic knowledge or countless trials. They truly need automatic feature construction tools to increase their investment opportunity and cut down the costs. A very famous hedge fund company is the pioneer in this field. They successfully used genetic programming learner (GP) to automatically construct financial features [17]. Although they didn’t officially publish papers to illustrate their algorithms, many people have tried to replicate this work. And similar approaches have been published. They use reverse polish expression to represent feature’s formula, and use GP to produce new expressions. [18]. This method truly works, it have produced useful features to find right trading signals. However, the drawback is that it produced a lot of similar features and these features do not contain more information than the features produced by human experts.

In this paper, we regard this method as our baseline, and we want to find the pros and cons of our algorithms compared with this work. We made several contributions compared with the works mentioned above. Firstly, we use deep neural network’s wrapper structure to replace tree structure, and construct a reasonable object function to conduct this construction process. What’s more, we come up with a kernel function to fix the object function’s undifferentiated problem. Secondly, we use model stealing technic to bring enough diversity into our network, which haven’t tried by other researchers. Thirdly, we come out with a method to measure the diversity of different financial features. All these works make sure that we can automatically produce useful, highly informative and diversified features.

2. PRIOR KNOWLEDGE IN FINANCIAL SIGNAL PROCESSING

2.1. Correlation object function

We put prior knowledge into the object function. In industry, people cared about the relative strength of each stock in each trading day. Thus, we group the input and output data by trading day, which makes each batch meaningful. The, we use spearman correlation to calculate the correlation of feature value and future’s return. This correlation can tell us, if the neural network proposes a feature value, how much money we can win or lose in the near future according to this value. The object function is shown in formula (1).

\[
\text{Correlation}(x, y) = \frac{\text{cov}(\text{rank}(x-\bar{x}), \text{rank}(y-\bar{y}))}{\sqrt{\text{var}(\text{rank}(x-\bar{x})) \cdot \text{var}(\text{rank}(y-\bar{y}))}}
\]

Here, we let \(x\) denotes feature value and \(y\) denotes future’s return. To make the gradient descent process more robust, we calculate the spearman coefficient in several randomly picked trading days. And use its mean value multiply -1, to serve as loss is each iteration.

However, the operator \(\text{rank()}\) can’t be differentiated. And this operator is essential, because it can get rid of anomalies in data. Take an example, we have a list [-20,-2,-1,1,2,3,4,5,6,7,20]. Obviously, -20 and 20 are the anomalies in this data. The ideal situation is to project this series into [1,2,3,4,5,6,7,8,9,10,11]. We hope to construct a kernel function \(g(x)\) which can do this projection, and meanwhile this function can be differentiated. We have tried \(\text{softmax}\) with ‘high temperature’. But its’ performance is not as competitive as \(\text{sigmoid}\) in this problem. The kernel function is shown in formula (2).

\[
g(x) = \frac{1}{1 + \exp(-p \frac{x - s}{2 \cdot \text{std}(x)})} (2)
\]

At first, we project \(x\) into a normal distribution which is zero-centralized. Next, we use hyper-parameter \(p\) to make sure that the majority of data between \([\text{mean} - 2 \cdot \text{std}, \text{mean} + 2 \cdot \text{std}]\). These normal points should take place 95% of space in the range of new projected series. In this way, we can get \(p=1.83\). Here is the projection performance based on the easy example mentioned above.

![Fig. 1. Use kernel function \(g(x)\) to replace operator \(\text{rank()}\).](image)

Fig. 1. Use kernel function \(g(x)\) to replace operator \(\text{rank()}\).

As we can see from Fig. 1, if we don’t do any operations on series \(x\), the proportion of normal points’ range to anomalies’ range is not reasonable. The ideal proportion is shown as rank’s curve. Although our kernel function is not as good as \(\text{rank()}\), it has improved the initial situation a lot.

2.2. Diversity among features

We use model stealing to let our network learn from different kinds of prior knowledge. Delete anomalies of data, or change the gap among data will be helpful to learn from the distribution of prior knowledge. This principle is similar to Hinton’s knowledge distillation [19]. After pre-train with
different prior knowledge, each time the network will hold different kinks of knowledge, which is the source of diversity.

What if our network produced thousands of similar features, it will not be so competitive compared with traditional methods. Thus we need to measure our produced features’ diversity. Taking an example, we use our network to produce m inexplicit formulas. If there are n stocks, each formula will produce a 1 * n array to represent feature distribution in a given trading day. As mentioned before, we only care about the features’ relative strength in each trading day. Thus, we do softmax on feature’s distribution in each trading days. This step can help us ignore data’s scale without losing any information. Secondly, we use cross entropy to measure the distance between two features’ distribution. m formulas can produce a m * m distance matrix. Thirdly, we use k-means to do clustering on these formulas [20]. At last, we calculate the average distance between each cluster center. This value can represent our method’s diversity in a given trading day. Different methods’ diversity can be compared during the same trading days. The diversity for this simplified example is shown in Fig. 2.

![Fig. 2. The diversity is calculated from the average distance of each cluster center.](image)

To sum up, we constructed a new object function based on prior knowledge in this field, and we also pre-train with prior knowledge to serve as the source of diversity. This alpha discovery neural network’s structure (ADN) is shown in Fig. 3.

![Fig. 3. Alpha discovery neural network’s structure, with its time grouped object function, and diversity resource.](image)

More specifically, we didn’t get much improvements by using more sophisticated network structure (residual layer, convolution layer and pooling layer), compared with a simple network structure, which only uses fully connected layer. There are two possible reasons to account for this result. The first is that we haven’t found the complicated and powerful structure although we have done a lot of experiments. The second is that, most of the trading signals are linear, because people use linear logic to do decision. Thus, we don’t have need to use complicated network structure. For further research in this filed, we will try to do experiments with all sophisticated network structures, and find whether there is a more powerful feature extractor will be suitable for this task. But in this paper, we propose a general network structure which is only made up by several fully connected layers. It’s simple but truly worked, and it has remarkably beaten the SOAT in this task.

### 3. EXPERIMENTS

We use A-share market’s day-frequency stock data. The input data is open, high, low, close and volume in the past 30 trading days. Because in each batch, we need to put more than 37000 samples into memory, the input data should be no longer than the past 30 trading days. (Use 20g GPU to conduct this experiment). Thus, more computing resource or good parallel computing technic will be helpful to this task.

We calculate the mean value and standard deviation of each feature in training set. Then use (x-mean)/std to standardize the data. In this way, our input is a batch*150 array, whose value ranges from -1 to 1. The future return is calculated by the next 5 trading days. For each back-propagation, we calculate the average value of spearman correlation from 10 randomly selected trading days. For each experiment, 250 trading days serve as training set, the following 30 trading days serve as validation set, and the following 30 trading days serve as testing set. We use same setting for the genetic programming algorithm (GP). In order to assess our algorithms’ performance, we give three schemes as following:

| Table 1. Three different schemes to show AND’s contributions, compared with baseline. |
|---------------------------------------------------------------|
| **Scheme A:** Only use GP to construct 100 features.          |
| **Scheme B:** Use GP to construct 100 features at first, and  |
| then uses these features as prior knowledge to pre-train our  |
| network. Finally, we get 100 features by ADN.                 |
| **Scheme C:** Only use ADN to construct 100 features. Each    |
| time, we pre-train the network with randomly produced         |
| distribution.                                                 |

For each trading day, we compare the average spearman coefficient and diversity among 100 produced features. In the same time, we use the same experiment settings to run GP. And get its results for Scheme B and Scheme C. The performance for each scheme is shown in Fig. 4.
Although these features have different explicit formulas, they can be easily classified into the same category. ADN can produce relatively more diversified features, and we think model stealing is the right method to add diversity. For Scheme B > Scheme C, because Scheme B is pre-trained with randomly produced distribution. Most of these distributions can’t serve as useful financial features. Thus, ADN can’t carry on their initial direction which let it can’t enjoy the diversity brought by different distributions. For the features produced by GP, although their diversity is relatively low, they are truly useful features in financial time series. Thus, if we pre-train with these features, these features’ differences can positively be passed to our network.

Due to the fact that financial time series is extremely non-stationary [21], although our method works well from 2018 to 2019, it can’t promise to work well in the other time. Thus, we repeated our experiments several times. Here is Scheme B’s average performance in the last four years, shown in Table 3.

Table 3. Scheme B’s performance in the last four years, value comes from testing set. We give the time interval of training set, and test interval is the next 30-60 trading day.

| Scheme B       | Test IC | Test Diversity |
|----------------|---------|----------------|
| Jan 2015 to Dec 2015 | 0.117   | 20.93          |
| Jan 2016 to Dec 2016 | 0.132   | 21.22          |
| Jan 2017 to Dec 2017 | 0.081   | 30.05          |
| Jan 2018 to Dec 2018 | 0.122   | 25.44          |

As we can see from Table 3, the performance in Jan - Dec 2017 is not good. It’s acceptable, because the majority features constructed by using price and volume in this year doesn’t work well. But all the test spearman coefficients are higher than 0.05, and all the test diversity are higher than Scheme A shown in Table 2. Thus, we can regard ADN as a reliable method which can be put into industry use.

4. CONCLUSION

Different from previous works, we highly associate prior knowledge in quantitative finance with neural network. In this work, we use wrapper approach to build a neural network, Alpha discovery neural network (ADN), which can automatically construct features. In our experiments, we find that ADN can construct more informative and diversified features than the state-of-the-art technic GP. Pre-train with high quality prior knowledge will make ADN’s final achievements even farther. Although the feature extractor we use in this paper is simple, but the whole structure truly worked and have beaten the SOTA technic is this task. For further research, we will try to use more sophisticated feature extractors. And we want to see whether different feature extractors can produce more diversified and more informative features.
5. REFERENCES

[1] Sondhi P. Feature construction methods: a survey[J]. Sifaka. cs.uiuc.edu, 2009, 69: 70-71.
[2] Janecek A, Ganssterer W, Demel M, et al. On the relationship between feature selection and classification accuracy[C]. New challenges for feature selection in data mining and knowledge discovery, 2008: 90-105.
[3] Chandrashekar G, Sahin F. A survey on feature selection methods[J]. Computers & Electrical Engineering, 2014, 40(1): 16-28.
[4] Mi G, Zhang P, Tan Y. A multi-resolution-concentration based feature construction approach for spam filtering[C]. The 2013 International Joint Conference on Neural Networks (IJCNN). IEEE, 2013: 1-8.
[5] Lensen A, Al-Sahaf H, Zhang M, et al. Genetic programming for region detection, feature extraction, feature construction and classification in image data[C]. European Conference on Genetic Programming. Springer, Cham, 2016: 51-67.
[6] Tran B, Xue B, Zhang M. Genetic programming for feature construction and selection in classification on high-dimensional data[J]. Memetic Computing, 2016, 8(1): 3-15.
[7] Vafaie H, De Jong K. Genetic algorithms as a tool for restructuring feature space representations[C]. Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence. IEEE, 1995: 8-11.
[8] Hindmarsh S, Andreae P, Zhang M. Genetic programming for improving image descriptors generated using the scale-invariant feature transform[C]. Proceedings of the 27th Conference on Image and Vision Computing New Zealand. ACM, 2012: 85-90.
[9] La Cava W, Moore J. A General Feature Engineering Wrapper for Machine Learning Using epsilon Lexicase Survival[C]. European Conference on Genetic Programming. Springer, Cham, 2017: 80-95.
[10] Liang Y, Zhang M, Browne W N. Wrapper Feature Construction for Figure-Ground Image Segmentation Using Genetic Programming[C]. Australasian Conference on Artificial Life and Computational Intelligence. Springer, Cham, 2017: 111-123.
[11] Güvenir H A, Çakır M. Voting features based classifier with feature construction and its application to predicting financial distress[J]. Expert Systems with Applications, 2010, 37(2): 1713-1718.
[12] Virgolin M, Alderliesten T, Bel A, et al. Symbolic regression and feature construction with GP-GOMEA applied to radiotherapy dose reconstruction of childhood cancer survivors[C]. Proceedings of the Genetic and Evolutionary Computation Conference. ACM, 2018: 1395-1402.
[13] Botis S, Nguyen M D, Woo E J, et al. Text mining for the Vaccine Adverse Event Reporting System: medical text classification using informative feature selection[J]. Journal of the American Medical Informatics Association, 2011, 18(5): 631-638.
[14] Ching T, Himmelstein D S, Beaulieu-Jones B K, et al. Opportunities and obstacles for deep learning in biology and medicine[J]. Journal of The Royal Society Interface, 2018, 15(141): 20170387.
[15] Xiao C, Choi E, Sun J. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review[J]. Journal of the American Medical Informatics Association, 2018, 25(10): 1419-1428.
[16] Gan L, Wang H, Yang Z. Machine Learning Solutions to Challenges in Finance: An Application to the Pricing of Financial Products[J]. Available at SSRN 3446042, 2019.
[17] Kakushadze Z. 101 formulaic alphas[J]. Wilmott, 2016, 2016(84): 72-81.
[18] Tran B, Xue B, Zhang M. Genetic programming for feature construction and selection in classification on high-dimensional data[J]. Memetic Computing, 2016, 8(1): 3-15.
[19] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.
[20] Barsony M, Schombert J M, Kis-Halas K. The LkH-alpha 101 infrared cluster[J]. The Astrophysical Journal, 1991, 379: 221-231.
[21] Cheng C, Sa-Ngasoongsong A, Beyca O, et al. Time series forecasting for nonlinear and non-stationary processes: a review and comparative study[J]. IIE Transactions, 2015, 47(10): 1053-1071.