Neural Network Metalearning for Credit Scoring

Kin Keung Lai[1,2], Lean Yu[2,3], Shouyang Wang[1,3], and Ligang Zhou[2]

[1] College of Business Administration, Hunan University, Changsha 410082, China
[2] Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
{mskklai, msyulean, mszhoulg}@cityu.edu.hk
[3] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China
{yulean, sywang}@cityu.edu.hk

Abstract. In the field of credit risk analysis, the problem that we often encountered is to increase the model accuracy as possible using the limited data. In this study, we discuss the use of supervised neural networks as a metalearning technique to design a credit scoring system to solve this problem. First of all, a bagging sampling technique is used to generate different training sets to overcome data shortage problem. Based on the different training sets, the different neural network models with different initial conditions or training algorithms is then trained to formulate different credit scoring models, i.e., base models. Finally, a neural-network-based metamodel can be produced by learning from all base models so as to improve the reliability, i.e., predict defaults accurately. For illustration, a credit card application approval experiment is performed.

1 Introduction

In the financial risk management field, the credit risk analysis is beyond doubt an important branch and credit scoring is one of the key techniques in the credit risk analysis. Especially for any credit-granting institution, such as commercial banks and certain retailers, the ability to discriminate good customers from bad ones is crucial. The need for reliable models that predict defaults accurately is imperative, in order to enable the interested parties to take either preventive or corrective action [1].

As Thomas [2] argued, credit scoring is a technique that helps organizations decide whether or not to grant credit to consumers who apply to them. The generic approach of credit scoring is to apply a classification technique on similar data of previous customers – both faithful and delinquent customers – in order to find a relation between the characteristics and potential failure. One important ingredient needed to accomplish this goal is to seek an accurate classifier in order to categorize new applicants or existing customers as good or bad. Therefore, many different models, including traditional methods, such as linear discriminant analysis [3] and logit analysis [4], and emerging artificial intelligence (AI) techniques, such as artificial neural networks (ANN) [5] and support vector machine (SVM) [1], were widely applied to credit scoring tasks and some interesting results have been obtained. A good recent survey on credit scoring and behavioral scoring is [2].

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCS 4113, pp. 403 – 408, 2006.

However, in the above approaches, it is difficult to say that the performance of one method is consistently better than that of another method in all circumstances, especially for data shortage leading to insufficient estimation. Furthermore, in realistic situation, due to competitive press and privacy, we can only collect few available data about credit risk, making the statistical approaches and intelligent inductive learning algorithm difficult to obtain a consistently good result for credit scoring. In order to improve the performance and overcome data shortage, it is therefore imperative to introduce a new approach to cope with these challenges. In this study, a neural-network based metalearning technique [6] is introduced to solve these problems.

The main motivation of this study is to take full advantage of the flexible mapping capability of neural network and inherent parallelism of metalearning to design a powerful credit scoring system. The rest of this study is organized as follows. In Section 2, a neural-network-based metalearning process is provided in detail. To verify the effectiveness of the proposed metalearning technique, a credit card application approval experiment is performed in Section 3. Finally, Section 4 concludes the paper.

2 The Neural-Network-Based Metalearning Process

Metalearning [6], which is defined as learning from learned knowledge, is an emerging technique recently developed to construct a metamodel that deals with the problem of computing a metamodel from data. The basic idea is to use intelligent learning algorithms to extract knowledge from several data sets and then use the knowledge from these individual learning algorithms to create a unified body of knowledge that well represents the entire knowledge about data. Therefore metalearning seeks to compute a metamodel that integrates in some principled fashion the separately learned models to boost overall predictive accuracy.

Broadly speaking, learning is concerned with finding a model $f = f_a[i]$ from a single training set $\{TR_i\}$, while metalearning is concerned with finding a global model or a metamodel $f = f_a$ from several training sets $\{TR_1, TR_2, \ldots, TR_n\}$, each of which has an

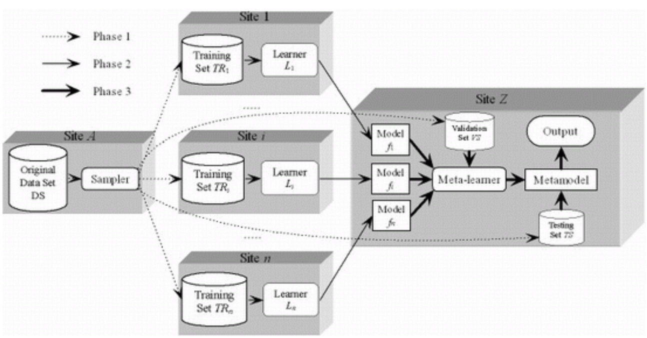

Fig. 1. The generic metamodeling process

associated model (i.e., base model) $f = f_a[i]$ (i =1, 2, …, n). The n base models derived from the n training sets may be of the same or different types. Similarly, the metamodel may be of a different type than some or all of the component models. Also, the metamodel may use data from a meta-training set (*MT*), which are distinct from the data in the single training set TR_i. Generally, the maim process of metalearning is first to generate a number of independent models by applying different learning algorithms to a collection of data sets in parallel. The models computed by learning algorithms are then collected and combined to obtain a metamodel. Fig. 1 shows a generic metalearning process, in which a global model or metamodel is obtained on *Site Z*, starting from the original data set *DS* stored on *Site A*.

As can be seen from Fig. 1, the generic metalearning process consists of three phases, which can be described as follows.

Phase 1: on *Site A*, training sets TR_1, TR_2, …, TR_n, validation set *VS* and testing set *TS* are extracted from *DS* with certain sampling algorithm. Then TR_1, TR_2, …, TR_n, *VS* and *TS* are moved from *Site A* to *Site* 1, *Site* 2, …, *Site n* and to *Site Z*.

Phase 2: on each *Site i* (i = 1, 2, …, n) the different models f_i is trained from TR_i by the different learners L_i. Then each f_i is moved from *Site i* to *Site Z*. It is worth noting that the training process of n different models can be implemented in parallel.

Phase 3: on *Site Z*, the f_1, f_2, …, f_n models are combined and validated on *VS* and tested on *TS* by the meta-learner *ML* to produce a metamodel.

A. Data set partitioning

Due to limitation of the number of data samples available in credit scoring analysis, some approaches, such as bagging [7] have been used for creating samples due to the feature of its random sampling with replacement. Bagging [7] is a widely used data sampling method in the machine learning. Given that the size of the original data set *DS* is *P*, the size of new training data is *N*, and the number of new training data items is *m*, the bagging sampling algorithm can be shown in Fig. 2.

```
Input: original data set DS
Output: The generated new training subsets {TR1, TR2, ..., TRm}
For t = 1 to m
  For i = 1 to N
    RandRow = P * rand ()
    If RandRow <= P
      Pt(i, AllColumns) = DS(RandRow, AllColumns)
    End If
  Next i
Next t
Output the final training subsets {TR1, TR2, ..., TRm}
```

Fig. 2. The bagging algorithm

B. Individual model creation

According to the principle of bias-variance trade-off [9], a metamodel consisting of diverse models (i.e., base models) with much disagreement is more likely to have a good performance. Therefore, how to create the diverse model is the key path to the creation of an effective metamodel. For neural network model, there are several methods for generating diverse models: (1) Initializing different starting weights for each neural network models; (2) Using different training subsets for training each neural network models; (3) Varying the architecture of neural network; and (4) Using different training algorithms. In this study, the single neural network models with different training subsets are therefore used as base learner L_1, L_2, …, L_n, as illustrated in Fig. 1. Through training, base models f_1, f_2, …, f_n can be formulated in a parallel way.

C. Neural-network-based metamodel generation

As Fig. 1 illustrated, the initial data set is first divided into subsets, and then these subsets are input to the different individual neural models which could be executed concurrently. These individual models are called "base models'. In this phase, the main task is to generate a metamodel to assimilate knowledge from different base models. Intuitively, the majority voting can produce a metamodel. But majority voting ignores the fact that some models that lie in a minority sometimes do produce the correct results. In metalearning, it ignores the existence of diversity that can reduce error variance. In this study, another single neural network model different from base neural network model is used to perform this task to generate a metamodel.

Concretely speaking, the base models can be generated based upon different training subsets in previous phase. Using the validation set *VS* and testing set *TS*, the performance of the base models can be assessed. Afterwards, the whole validation set *VS* is applied to these base models and corresponding results produced by these base models are used as input of another individual neural network model. By validation, a metamodel can be generated using the results generated by the base model as input, combined with their expected values. In this sense, neural network learning algorithm is used as a meta-learner (*ML*) shown in Fig. 1 for metamodel generation.

3 Experimental Analysis

The research data is about Japanese credit card application approval obtained from UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/). For confidentiality all attribute names and values have been changed to meaningless symbols. After deleting the data with missing attribute values, we obtain 653 data, with 357 cases were granted credit and 296 cases were refused. To delete the burden of resolving multi-category, we use the 13 attributes A1-A5, A8-A15. Because we generally should substitute k-class attribute with k-1 binary attribute, which will greatly increase the dimensions of input space, we don't use two attributes: A6 and A7.

In this empirical analysis, we randomly draw 400 data from the 653 data as the initial training set, 100 data as the validation set and the else as the testing set. In order to increase model accuracy for credit scoring, ten different training subsets are generated by bagging algorithm. Using these different training subsets, different neural network base models with different initial weights are presented. For neural network base models, a three-layer back-propagation neural network with 10 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer is used. The network training function is the TRAINLM. For the neural-network-based metamodel, a similar three-layer back-propagation neural network with 10 inputs neurons, 8 TANSIG neural in the second layer and one PURELIN neuron in the final layer is adopted for metamodel generation. Besides, the learning rate and momentum rate is set to 0.1 and 0.15. The accepted average squared error is 0.05 and the training epochs are 1600. The above parameters are obtained by trial and error.

For comparison, several typical credit scoring models, linear discriminant analysis (LDA), logit analysis, individual ANN and SVM, are selected as benchmark models. In addition, majority voting based metamodel is also adopted for further comparison. In the ANN model, a three-layer back-propagation neural network with 13 input nodes, 15 hidden nodes and 1 output nodes is used. The hidden nodes use sigmoid transfer function and the output node uses the linear transfer function. In the SVM, the kernel function is Gaussian function with regularization parameter C = 50 and σ^2=5. Similarly, the above parameters are obtained by trial and error. The classification accuracy (i.e., Type I accuracy and Type II accuracy [1]) in testing set is used as performance evaluation criterion. To overcome the bias of individual models, such a test is repeated ten times and the final Type I and Type II accuracy is the average of the results of the ten individual tests. The computational results are shown in Table 1.

Table 1. The prediction performance comparison results

Model	Type I (%)	Type II (%)
Linear discriminant analysis	79.79	81.05
Logit regression analysis	84.17	83.11
Single artificial neural network	81.34	83.78
Single support vector machine	80.58	82.36
Majority-voting-based metamodel	83.41	85.16
Neural-Network-based metamodel	89.56	91.19

As can be seen from Table 1, we can find the following conclusions. (1) For type I accuracy and Type II accuracy, the neural network based metamodel and the majority voting based metamodel outperforms the single credit scoring model, implying the strong capability of metamodel in credit scoring. (2) In the two metamodels, the performance of the neural-network-based metamodel is much better than that of the majority-voting-based metamodel. The main reason is that neural network has a flexible nonlinear learning capability that can capture subtle relationships between diverse base models. Inversely, the majority voting often ignores the existence of diversity of different base models, as earlier mentioned. (3) In the four individual models, the logit analysis surprisedly outperforms the linear discriminant analysis, the best artificial neural network and the best support vector machine from the view of Type I. For Type II, the artificial neural network is the best of the four individual models. For this example, Type II classification is more important than Type I classification. If a bad customer is classified as a good customer, it may lead to direct economic loss. In this sense, artificial neural network model is very promising approach to credit scoring. (4) Generally, the proposed neural-network-based metamodel perform the best in terms of both Type I accuracy and Type II accuracy, implying that the proposed neural network metalearning technique is a feasible solution to credit scoring.

4 Conclusions

In this study, a neural-network-based metalearning technique is proposed to solve the credit scoring with limit data. Through the practical data experiment, we have obtained good classification results and meantime demonstrated that the neural-network-based metamodel outperforms all the benchmark models listed in this study. These advantages imply that the proposed neural-network-based metalearning technique can be used a promising solution to credit scoring. Of course, this neural network metalearning method is also extended to other application areas.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (NSFC); Chinese Academy of Sciences (CAS) and Strategic Research Grant of City University of Hong Kong (SRG No. 7001677).

References

1. Lai, K.K., Yu, L., Zhou, L.G., Wang, S.Y.: Credit Risk Evaluation with Least Square Support Vector Machine. Lecture Notes in Computer Science. (2006)
2. Thomas, L.C.: A Survey of Credit and Behavioral Scoring: Forecasting Financial Risk of Lending to Consumers. International Journal of Forecasting .16 (2002) 149-172
3. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics. 7 (1936) 179-188
4. Wiginton, J.C.: A Note on the Comparison of Logit and Discriminant Models of Consumer Credit Behaviour. Journal of Financial Quantitative Analysis. 15 (1980) 757-770
5. Malhotra, R., Malhotra, D.K.: Evaluating Consumer Loans Using Neural Networks. Omega 31 (2003) 83-96
6. Chan, P., Stolfo, S.: Meta-Learning for Multistrategy and Parallel Learning. Proceedings of the Second International Workshop on Multistrategy Learning (1993) 150–165
7. Breiman, L.: Bagging Predictors. Machine Learning. 26 (1996) 123-140
8. Lai, K.K., Yu, L., Wang, S.Y., Huang, W.: A Bias-Variance-Complexity Trade-off Framework for Complex System Modeling. Lecture Notes in Computer Science. 3980 (2006) 518-527