

An Unsupervised Model on Outsourcing Software Projects Risks Based on T-S Fuzzy Neural Network

Z.Z. Ouyang

School of Finance
Guangdong University of Foreign Studies China

Z.H. Zhang

School of Economics and Trade
Guangdong University of Foreign Studies
China
Faculty of Business, Environment and Society
Coventry University
United Kingdom

J.Z. Chen

School of Economics & Finance
the University of Hong Kong
China

Y. Hu

School of Informatics
Guangdong University of Foreign Studies
China

Abstract-Supervised learning algorithm is widely used in current research of software project risks, while it cannot evaluate the risk without decision class. Considering this, we present an unsupervised model, which can solve the forecasting problem of outsourcing software projects risks in unsupervised situation, based on T-S fuzzy neural network (T-S FNN) and the concept of standard score in risk evaluation of outsourcing software project. In simulation, we use 200 samples to train and test the network repeatedly, helping us to get the accuracy rate of 76.2%. After that, the standard score of 0.558 is applied to the forecasting of 60 testing samples, and the accuracy of them is 75%, which shows the stability of the model. Moreover, the comparison with some supervised neural networks shows that the forecasting accuracy of T-S FNN is close to that of the supervised models above. Meanwhile, T-S FNN shows a more balanceable forecasting performance in both successful project and failure project.

Keywords-outsourcing software projects risks; unsupervised model; semi-supervised model; T-S fuzzy neural network; standard score

I. INTRODUCTION

In 2009, through a survey report on the global software outsourcing project development (CHAOS Summary, 2009, April), Standish Group suggested that overall success rate of outsourcing software project is only 32%, completely failure rate is 24% and the part of the failure rate is 44%. From this, we know that though the future of software outsourcing service industry is bright, high risks and high failure rate are accompanied with the development of outsourcing software projects. Thus, how to select suitable methods to identify and evaluate risk level based on the intelligent analysis and the data of outsourcing software projects is an important research field in outsourcing software projects risks analysis. In this research, Most researchers apply supervised learning algorithm to forecast the risk level, such as neural networks [1-2], decision tree[3], Bayesian networks [4-5], fuzzy recognition and fuzzy reasoning [6].

However, there are two main insignificances in the traditional forecasting models. First, current researchers

almost focus on supervised learning algorithms, which cannot deal with the situation lacking decision class. Second, supervised algorithms of the models occupy more time in training. To solve the problems above, we use feature selection to get the suitable factors and adopt T-S FNN and standard score, which can shortcut the training time, to evaluate the risk.

II. FORECASTING MODEL CONSTRUCTION

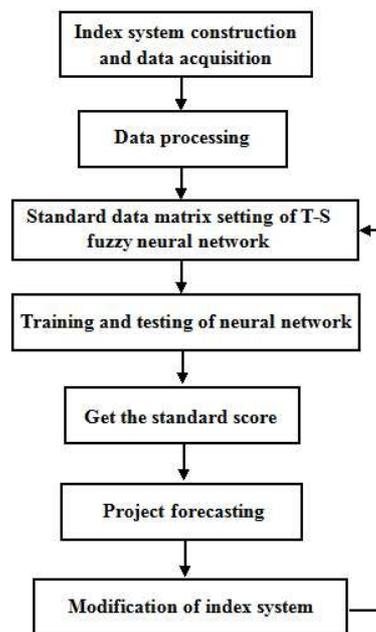


FIGURE 1. MODELLING STEPS.

We design an unsupervised model and keep high accuracy as far as possible.

From Figure 1, we first construct some project risk indexes, and collect experimental data from companies. In data processing, we remove the missing samples and select features for risk. After that, a standard data matrix would be set for T-S

FNN model. By training data and testing data over and over, T-S FNN can achieve the highest accuracy and the standard score would be obtained. Finally, we can use T-S FNN to predict the unknown risk project and get the prediction results by comparing the predicted score with the standard score. In addition, if there is any modification of index system, the standard data matrix of T-S FNN should be reset and then we perform all the steps above again.

A. Feature Selection

Feature selection refers to selecting a subset from all features to improve constructed model. Too many features might cause the following consequences: The model will be complicated, and we will spend much time to analyze the features and construct models; meanwhile, training data and selecting factors will be difficult; and the generalization ability of the model will decline. Moreover, features may be little correlated after the feature selection step.

B. Introduction to T-S Fuzzy Neural Network (T-S FNN)

According to fuzzy neural network (FNN) [7-9], we construct the unsupervised T-S FNN to evaluate each project and output the project scores.

Standard data matrix of T-S FNN: The evaluation of project risks is based on the setting evaluation criteria of each indicator and the selected indicator data. We can obtain comprehensive score of each project through mathematical models. Before training FNN, we set a standard data matrix on the basis of each indicator, which is divided into five grades in the index system.

T-S FNN model: T-S FNN is divided into 4 layers: input layer, fuzzy layer, fuzzy-rule-computing layer and output layer.

The input layer connects to the input vector x_i , the number of nodes is equal to the dimension of input vector, and the fuzzy layer employs the membership function.

$$\mu_{A_j} = \exp(-(x_j - c_j^i)^2 / b_j^i), j = 1, 2, \dots, k; i = 1, 2, \dots, n. \quad (1)$$

The fuzzy layer will fuzzify the input to get the fuzzy subjection degree. And the fuzzy-rule-computing layer finishes the calculation and we can get ω :

$$\omega^i = \prod_{k=1}^m u_{A_j^k}(x_k), i = 1, 2, \dots, n. \quad (2)$$

Lastly, the following formula is applied to get the result of the fuzzy neural network in the output layer:

$$y_i = \sum_{i=1}^n \omega^i (p_0^i + p_1^i x_1 + \dots + p_k^i x_k) / \sum_{i=1}^n \omega^i. \quad (3)$$

Error calculation is defined by:

$$e = \frac{1}{2} (y_d - y_c)^2 \quad (4)$$

Coefficient modification is defined by:

$$p_j^i(k) = p_j^i(k-1) - \alpha \frac{\partial e}{\partial p_j^i} \quad (5)$$

$$\frac{\partial e}{\partial p_j^i} = (y_d - y_c) \omega^i / \sum_{i=1}^m \omega^i x_j \quad (6)$$

p_j^i : coefficient of neural network; α : learning ratio of neural network; x_j : input parameter; ω^i : continued product of membership degree of input parameter.

Parameter modification is defined by:

$$c_j^i(k) = c_j^i(k-1) - \beta \frac{\partial e}{\partial c_j^i} \quad (7)$$

$$b_j^i(k) = b_j^i(k-1) - \beta \frac{\partial e}{\partial b_j^i} \quad (8)$$

c_j^i : centre of membership function; b_j^i : width of membership function.

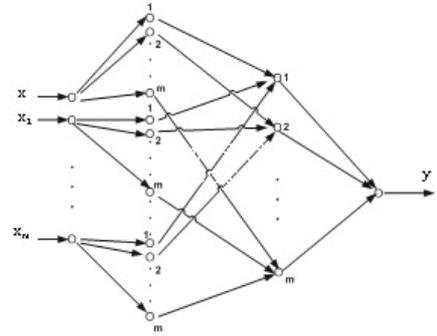


FIGURE 2. SCHEMATIC VIEW OF T-S FNN.

Dimension of the training sample determines the number of input/output nodes as well as the number of fuzzy membership functions, which is the basis of constructing FNN. The training of FNN: we generate the training sample by the means of evenly-spaced uniform distribution. The training sample is based on the data in standard data matrix. Then, we repeatedly train the original network with the training sample for 100 times to get the trained network.

C. Design of Standard Score

To improve the efficiency of the forecasting model, we present a definition of standard score. Standard score is produced by training and testing T-S FNN again and again. When the test of neural network achieves the highest accuracy, we transform all project scores into the numbers ranged from 0 to 1 by normalized process. And we select the score, which can recognize the outsourcing software project would be successful or not, as the standard score. Then the standard score will be applied to evaluate the project risk by comparing the project score and the standard score.

III. APPLICATION OF THE MODEL

A. Data Acquisition and Data Processing

Data acquisition: From the references of outsourcing software projects risks, we extract 3 risk categories and 25 risk factors. We own 293 samples, including small IT Company, medium-sized enterprise, and large enterprise, with 260 of

them are effective. And the costs of them are from 10000 to over 5 million Yuan. In all samples, 191 items are failure and 69 items are overall successful.

Data processing: We use feature selection to decline the number of factors from 25 to 16, which is: Development of Demand, Development Team, Plan and Control, Leader of the Project, Development and Test, Restriction of Time and Finance, Degree of Demand Change, Engineering Support, Industry Experience, Environment of Enterprise, Business Process, Leadership Support, Management of Demand, Cooperation of Customer Team, Support of Customer Department, Maturity of Customer Representative. Meanwhile, the following attributes are cancelled: Number of Team Members, Information Level, Lines of Code, Technical Complexity, Actual Development Time, Real-time and Security Requirements, Development Expense, Function Point, Number of Collaborators. In data processing, we apply Pearson correlation coefficient to select the relatively important features in modelling. By doing this, the dimension of the sample is decreased and the forecasting accuracy of the model is enhanced.

B. T-S FNN Model

We get a standard data matrix of T-S FNN and 16 factors with 1-5 grades.

TABLE I. STANDARD DATA MATRIX OF SOFTWARE PROJECT RISK EVALUATION.

Grades	I	II	III	IV	V
Development of Demand(\geq)	1	2	3	4	5
...
Restriction of Time and Finance(\geq)	1	2	3	4	5

After training T-S FNN for 100 times, we apply 200 random samples to the model and test repeatedly to obtain the corresponding score of each project. After that, the scores are transformed into the numbers ranged from 0 to 1 by normalized processing and ranked by ascending order.

Semi-supervised model: Assume that we obtain the success rate and failure rate for all training samples in the experiment. Since we have part information of decision attributes, this model is a Semi-supervised model. In 200 samples, there are 48 successful projects, which mean that overall success rate is 24%. The first 152 items are predicted as failure projects after ranking according to the scores, and the value of the 153rd score would be selected as standard score, which is 0.635. From the simulation results, the overall prediction accuracy is 75.8%.

Unsupervised model: According to Standish Group Survey Report, the percent of overall successful projects is 32% and that of failure projects is 68%, which means that 64 items can be predicted as successful projects and 136 items will be failure projects or partial failure projects. Then the standard score is 0.558. And we conclude that the overall prediction accuracy is 76.2%.

All results are shown in the Table 2, the overall forecasting accuracy rates of 6 models are from 75.8% to 78.1%, and the difference among them is very small. To be more specific, the overall forecasting accuracy and the accuracy of failure projects of Prune NN are both top the list, with 78.1% and 96.3% respectively, but the accuracy of successful projects is only 27.5%. In comparison, the overall accuracy of semi-supervised T-S NN and BP NN are both 75.8%, which is only 2.3% lower than that of Prune model. In regard of the accuracy of successful projects, unsupervised T-S model is the best model among all these models, with accuracy of 71%. From the comparison of results, we conclude that the overall forecasting accuracy of unsupervised T-S FNN is close to that of four supervised models. Besides, compared with supervised models, the forecasting performance of unsupervised T-S FNN is more stable and balanced in the prediction of failure projects and successful projects, which shows that the forecasting ability of unsupervised T-S FNN is as good as that of supervised model.

TABLE II. COMPARISON OF THE FORECASTING RESULTS OF NEURAL NETWORKS.

Prediction accuracy	Semi-supervised T-S FNN	Un-supervised T-S FNN	Dynamic NN	Prune NN	RBF NN	BP NN
Overall accuracy	75.8%	76.2%	76.9%	78.1%	77.7%	75.8%
Failure projects	82.7%	78.0%	80.1%	96.3%	92.1%	96.3%
Successful projects	56.5%	71.0%	68.1%	27.5%	37.7%	18.8%

IV. CONCLUSION

In the condition of keeping high accuracy, we introduce a semi-supervised T-S FNN model and an unsupervised T-S FNN model to evaluate the project risk. T-S FNN is an unsupervised neural network, and it can evaluate the project risk and output the project score, reducing the training time and the complexity of time and space. Besides, the use of standard score helps to avoid the retraining of neural network, helping to improve the forecasting efficiency and achieve the subjective effect. The experiment results show that the application of standard score can make the prediction accuracy of supervised learning models consistent with that of T-S FNN. Compared to some supervised algorithms, the overall accuracy of T-S FNN should be further improved. Thus, it is worth doing research to find the combinational model of T-S FNN and other algorithm.

ACKNOWLEDGMENT

This paper is funded by the National Natural Science Foundation of China (Grant No. 71271061), the Natural Science Foundation of Guangdong Province (Grant No. S2014010006386), the "Twelfth Five-Years" Philosophy and Social Sciences Planning Project of Guangdong Province (Grant No. 296-Z1313040, No GD12XGL14.), the Science

and Technology Innovation Project of Department of Education of Guangdong Province (Grant No. 296-GK132019, No. 2013KJCX0072), "Twelfth Five-Years" Philosophy and Social Sciences Planning Project of Guangzhou (Grant No. 14G41), the Teaching Quality and Teaching Reform Project for Higher Education of Guangdong Province (Grant No. 110-GK131021, No. 2013176), the "Twelfth Five- Years" Education Planning Project of Guangdong Province (Grant No. 2012JK129).

REFERENCE

- [1] Li H., Cao X.L. & Cheng J.R., Research on Risk Assessment of Software Project Based on BP Neural Network. *Computer Simulation*, 2011, 28(7): 374-478.
- [2] Hu Y., He X.J., Huang J.X., Rong Z.B. & Xie K., Risk Evaluation Model for Software Projects with Neural Network, *Chinese Journal of Management*, 2010, 7(3): 391-394.
- [3] Hu Y., Zhang H.Z., Chen G.H., Li Y.X. & Xie K., Intelligent Decision Tree Model for Software Outsourcing Project Risk Analysis. *Journal of Wuhan University*, 2010, 56(6): 729-734.
- [4] Tang A.G., Wang R.L. & Hu C.H., Application of Bayesian networks in software project risk assessment. *Computer Engineering and Application*, 2011, 46(7): 62-65.
- [5] Ren X.L., Application of Bayesian Network and Rough Sets in Software Risk Estimation. *Computer Systems & Applications*, 2010, 19(5): 202-204.
- [6] Pang L.J. & Jia G.J., Risk Evaluation of the Software Project Based on Fuzzy Simulation. *Journal of Shanxi Normal University (Nature Science Edition)*, 2011, 25(3): 45-49.
- [7] Hou Y. & Zhao H., Traffic flow prediction based on T-S fuzzy neural network optimized improved particle swarm optimization. *Computer Engineering and Applications*, 2014, 50(4): 236-239.
- [8] Wang W.Y., Chien Y.H., Leu Y.G. & Lee T.T. Adaptive T-S fuzzy-neural modeling and control for general MIMO unknownnonaffine nonlinear systems using projection update laws. *Automatica*, 2010, (46): 52-63.
- [9] Huang Z.W., Zhou J.Z., Li C.S., Li F.P. & Zhang Y.C., Fuzzy Neural Network Based on Improved T-S Model and Its Application. *Advances in Neural Networks*, 2009, (5552): 155-164.