A Cost-sensitive Intelligent Prediction Model for Outsourced Software Project Risk

Zhang Hongming  
*Business Intelligence and Knowledge Discovery, School of Business Guangdong University of Foreign Studies*

Feng Bin  
*Business Intelligence and Knowledge Discovery, School of Business Guangdong University of Foreign Studies*

Mo Xizhu  
*Business Intelligence and Knowledge Discovery, School of Business Guangdong University of Foreign Studies*

Su Lijun  
*Business Intelligence and Knowledge Discovery, School of Business Guangdong University of Foreign Studies*

Xiangzhou Zhang Xiangzhou  
*Business Intelligence and Knowledge Discovery, School of Business Guangdong University of Foreign Studies*

See next page for additional authors

Follow this and additional works at: [http://aisel.aisnet.org/whiceb2013](http://aisel.aisnet.org/whiceb2013)

Recommended Citation  
[http://aisel.aisnet.org/whiceb2013/72](http://aisel.aisnet.org/whiceb2013/72)

This material is brought to you by the Wuhan International Conference on e-Business at AIS Electronic Library (AISeL). It has been accepted for inclusion in WHICEB 2013 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Authors
Zhang Hongming, Feng Bin, Mo Xizhu, Su Lijun, Xiangzhou Zhang Xiangzhou, and Hu Yong
A Cost-sensitive Intelligent Prediction Model for Outsourced Software Project Risk

Hongming Zhang*, Xizhu Mo*, Lijun Su*, Bin Feng*, Xiangzhou Zhang, Yong Hu*
Business Intelligence and Knowledge Discovery, School of Business
Guangdong University of Foreign Studies, Guangzhou, 510006, China

Abstract: Outsourced software project is one of the main ways of software development, which is of high failure rate. Intelligent risk prediction model can help identify high risk project in time. However, the existing models are mostly based on such a hypothesis that all the cost of misclassification is equal, which is not consistent with the reality that in the domain of software project risk prediction, the cost of predicting a fail-prone project as a success-prone project is different from predicting a success-prone project as a fail-prone project. To the best of our knowledge, the cost-sensitive learning method has not yet been applied in the domain of outsourced software project risk management though it has been widely used in a variety of fields. Based on this situation, we selected five classifiers, and introduced cost-sensitive learning method to build intelligent prediction models respectively. This paper totally collected 292 real data of outsourced software project for modeling. Experiment results showed that, under cost-sensitive scenario, the polynomial kernel support vector machine is the best classifier for outsourced software project risk prediction among the five classifiers due to its high prediction accuracy, stability and low cost.

Keywords: Outsourced software project, Risk management, Cost-sensitive, Risk prediction

1. INTRODUCTION

Software outsourcing is one of the important parts of IT outsourcing since the 1990s [1]. According to the 2011 CHAOS Report of Standish Group, the total success rate of software project is only 37%, while the completely failure rate is 21% [2]. As the outsourced software project has a higher uncertainty, which may lead to a very high failure rate, the intelligent risk prediction for software outsourcing can help to identify high risk project timely.

The exist prediction models of outsourced software project are mostly based on Decision Trees (DT) [3], Neural Network (NN) [4], Bayesian Network (BN) [5, 6] and Support Vector Machine (SVM) [7]. Nowadays, most of the classification algorithms are based on such a hypothesis: all the cost of the misclassification is equal. However, in the real world that is not the case. For example, in the problem of outsourced software project risk prediction, the cost of “forecast the failure project as the success project” and “forecast the success project as the failure project” is obviously different. In the domain of outsourced software project risk management, we concern more about the misclassification of “forecast the failure project as the success project”. When we predict a fail-prone project as a success-prone project and invest a plenty of resources and money to it, we may counter immeasurable losses. Therefore, it is necessary to apply software project risk prediction model to identify this risk and minimize the cost of the project failure.

At present, the cost-sensitive learning method has been widely used in the domain of software engineering, such as error prediction [8], software defect prediction [4] or software quality prediction [9, 10]. To the best of

---

* Contribute equally, regarded as first authors.
+ Corresponding author. Email: henryhu200211@163.com (Yong Hu)
our knowledge, there is no research that applies the cost-sensitive learning method to the domain of outsourced software project risk management. As mention above, the outsourced software project risk is a typical cost-sensitive problem. Therefore, this paper is one of the first to introduce cost-sensitive learning method into the problem of outsourced software project risk prediction, so as to build a more effective model for the practice of outsourced software project.

Based on 292 real outsourced software project data, we select five classifiers and then introduce cost-sensitive learning method to establish the prediction models respectively. The experiment results indicated that among the five classifiers, the polynomial kernel support vector machine performed the best. In other words, it is a good classifier for outsourced software project risk prediction.

2. RELATED WORKS

There are generally three methods in researches of software project risk analysis, namely prediction and classification algorithm (including BN, DT, SVM, etc.), cluster analysis and correlation analysis.

Prediction and classification algorithm is used for early evaluation of project output. Hu et.al collected 154 real data of outsourced software project, and combined the expert knowledge and Bayesian Network structure learning method to build an intelligence analysis model for outsourced software project risk. The experimental results indicated that the intelligent model has important significance both in theory and practice, and it can give good guidelines for software project risk assessment and analysis [6]. Cluster analysis is applied to analyze the relevance of risk and output. Based on the previous researches, Wallace et.al divided software project risks into six dimensions, and applied a cluster analysis to identify the software risk projects. The experiment results showed that all the six dimensions of risks are influenced by the project scope [11]. Correlation analysis is mainly used to analyze the correlation of risk and output. García et.al used association rules to establish a model, which can forecast the influence of management policy factors on a variety of software project properties (such as software quality, software development time and investment etc.)[12].

At present, the cost-sensitive learning method has been widely used in the domain of software engineering, such as error prediction [8], software defect prediction [4] and software quality prediction [9].

Jiang and Cukic introduced the cost-sensitive learning method in the development of software fault prediction based on the project data from a public dataset. They found that although the cost-sensitive learning method did not increase the performance of the model, it made it easier for project managers to select appropriate model with the explicit information on misclassification cost [8]. Zheng used three algorithms (one was threshold-moving based and the other two were weight-updating based), to boost neural networks for software defect prediction according to the four datasets of NASA projects. The results indicated that the threshold-moving based algorithm was the most appropriate algorithm of the three for establishing a cost-sensitive software defect prediction model in neural networks [4]. Based on the data from high-assurance systems, Seliya and Khoshgoftar conducted an empirical study using two popular decision tree classification algorithms, namely the C4.5 and Random Forest. The result of this research offered project managers a clear procedure, that how to consider and analyze the misclassification cost in software quality prediction model [9].

In conclusion, there are two main research gaps in the research field of software project risk prediction model: First of all, there are seldom researches of risk prediction model specifically for outsourced software project. Secondly, although there are a plenty of researches for software project risk prediction, to our best knowledge, there are no researches about software project risk prediction that introduced the cost-sensitive learning method.
3. METHODOLOGY

3.1 Data collection

This research totally collected 292 outsourcing software project data. The data we collected are of high quality and enough to be representative: From the aspect of the basic information of the project, the samples cover the government, education, finance, information, health, manufacturing, business and many other industries. In the total 292 samples, most of the samples are from information (22.26%), the government (20.21%), business (17.12%) and manufacturing (10.27%). From the aspect of project scale, the team size more than ten people accounted for nearly 37%, and the development scale including projects that range from 200 function points to more than 10000 function points. From the aspect of the information quality, nearly 80% of the respondents are project manager (27.05%), leader (14.73%) or development team members (33.56%), who have 3 years or more project working experience. Therefore, they have enough qualification to provide the information we need.

3.2 Input and output variables

In this paper, we selected 25 risk factors (as shown in Table 1) as the input variables (we discretized each risk factor into three values, namely high, medium and low) of our outsourced software project risk model according to the research of Hu et.al [6].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Development Cost (DC)</td>
<td>[13]</td>
<td>1. Development Team (CDT)</td>
<td>[14-16]</td>
</tr>
<tr>
<td>2. Development Period (DP)</td>
<td>[17]</td>
<td>2. Project Manager (PM)</td>
<td>[15]</td>
</tr>
<tr>
<td>3. Function Point (FP)</td>
<td>[17]</td>
<td>3. Number of Team Members (TS)</td>
<td>[18]</td>
</tr>
<tr>
<td>5. Technology Complexity (TC)</td>
<td>[13]</td>
<td>5. Industry Experience (ESP)</td>
<td>[14, 15]</td>
</tr>
<tr>
<td>7. Schedule and Budget Management (TBR)</td>
<td>[13, 15]</td>
<td>7. Requirement Management (RM)</td>
<td>[18, 19]</td>
</tr>
<tr>
<td>8. Per 1000 lines of code (KLOC)</td>
<td>[17]</td>
<td>8. Development and Testing (IT)</td>
<td>[18]</td>
</tr>
<tr>
<td>Customer Risks</td>
<td>Ref.</td>
<td>9. Engineering Support (ES)</td>
<td>[15, 19]</td>
</tr>
<tr>
<td>2. Business Process (BP)</td>
<td>[14]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Top Management Support (TMS)</td>
<td>[14, 15, 18]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Client Department Support (CDS)</td>
<td>[14, 15, 18, 21]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Client Experience (ECM)</td>
<td>[14, 18]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Collaboration of Client Team (CTC)</td>
<td>[14, 15, 18]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Business Environment (BE)</td>
<td>[14]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Similarly, based on the research of Hu et.al [6], the measurement dimensions of software project output include two parts, namely product output and process output, which are subdivided into eight specific output indicators (as shown in Table 2). In this study, the project is regard as success when all the eight output indicators are “high”; otherwise the project is regard as failure.

3.3 Cost-sensitive learning

The cost-sensitive learning is proved to be an effective method of machine learning which can incorporate different misclassification costs into the classification process [23, 24]. The core idea of cost-sensitive learning method is to consider how to train the classifier when different classification errors lead to different cost.
This paper introduces cost-sensitive learning method which theoretical foundations are formed by Elkan [23] into five classifiers respectively, namely Random Forest (RF), DT, SVM1 (linear kernel support vector machine), SVM2 (polynomial kernel support vector machine) and SVM3 (radial basis function kernel support vector machine), in order to find an appropriate classifier to build an intelligent prediction model for outsourced software project risk.

<table>
<thead>
<tr>
<th>Performances</th>
<th>Attributes</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The users perceive that the system meets intended functional requirements.</td>
<td>[11]</td>
</tr>
<tr>
<td>2</td>
<td>The system meets user expectations with respect to ease of use, response time and reliability.</td>
<td>[11]</td>
</tr>
<tr>
<td>3</td>
<td>The application developed is easy to maintain.</td>
<td>[11]</td>
</tr>
<tr>
<td>4</td>
<td>The information quality which the system provide to users and organizations.</td>
<td>[22]</td>
</tr>
<tr>
<td>5</td>
<td>The users are satisfied with the developed application.</td>
<td>[11]</td>
</tr>
<tr>
<td>6</td>
<td>The overall quality of the developed application is high.</td>
<td>[17]</td>
</tr>
<tr>
<td>7</td>
<td>The system was completed within schedule.</td>
<td>[11]</td>
</tr>
<tr>
<td>8</td>
<td>The system was completed within budget.</td>
<td>[11]</td>
</tr>
</tbody>
</table>

### 3.4 Evaluation method

In this paper, we introduce four measures, namely accuracy, precision, recall and F-Measure to effectively evaluate the performance of the selected classifiers. Accuracy is the most common measures to evaluate the degree of veracity while precision evaluate the degree of reproducibility. The recall represents the probability of a true positive sample being retrieved, and F-Measure can be regard as the weighted average of precision and recall.

We define the fail-prone project as positive sample; while success-prone project is regard as negative sample. In the following discussion, TP and TN respectively represents the amount of true positive samples and the amount of true negative samples; while FP and FN respectively represents the amount of false positive and false negative samples.

The accuracy, precision, recall and F-Measure are formally defined as follows:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

\[
\text{F-Measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

### 4. EXPERIMENTS AND RESULTS

In order to build our prediction model, we select five classifiers (i.e., RF, DT, SVM1, SVM2 and SVM3), and compare the performance of them under different cost scenario (λ). The experiment results are shown in Table 3 and Figure 1.

We can see from Figure 1 that the total cost of SVM3 is the lowest among the five classifiers. And the results of the four measurements, namely accuracy, precision, recall and the F-Measure are better than RF, DT and SVM1. However, it is obviously not so good when compared with SVM2, and the overall stability of the
algorithm (under different $\lambda$) is also inferior to SVM2.

<table>
<thead>
<tr>
<th>Table 3. Performances of different classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
</tr>
</tbody>
</table>
| $\begin{array}{cccccccccccccc}
| 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 & 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \\
| RF & 76.4 & 75.7 & 76.4 & 75.3 & 75 & 74.3 & 76 & 75.3 & 74.7 & 77.4 & 74 & 74 & 72.9 & 72.6 \\
| DT & 72.9 & 74.3 & 72.9 & 74.7 & 72.9 & 73.3 & 75.3 & 75.3 & 77 & 77 & 74.3 & 75.7 & 74.7 & 75.7 & 72.9 \\
| SVM1 & 73.3 & 73.3 & 73.3 & 73.3 & 73.3 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 \\
| SVM2 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 78.4 \\
| SVM3 & 77.1 & 77.1 & 77.1 & 77.1 & 77.1 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 \\
| & & & & & & & & & & & & & & |
| Precision (%)                                |
| $\begin{array}{cccccccccccccc}
| 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 & 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \\
| RF & 74.3 & 73.5 & 74.1 & 72.8 & 72.4 & 71.4 & 73.7 & 72.6 & 71.6 & 75.4 & 70.4 & 70.7 & 71 & 69 & 68.8 \\
| DT & 71.3 & 72.1 & 70.5 & 72.6 & 70.5 & 70.5 & 73.1 & 73 & 75 & 72.5 & 71.1 & 72.9 & 71.5 & 73 & 69.4 \\
| SVM1 & 71 & 71 & 71 & 71 & 71 & 71.9 & 71.9 & 71.9 & 71.9 & 71.9 & 71.9 & 71.9 & 71.9 & 71.9 & 71.4 \\
| SVM2 & 76.7 & 76.7 & 76.7 & 76.7 & 76.7 & 76.1 & 76.1 & 76.1 & 76.1 & 76.1 & 76.1 & 76.1 & 76.1 & 76.1 & 76.8 \\
| SVM3 & 75 & 75 & 75 & 75 & 75 & 73 & 73 & 73 & 73 & 73 & 73 & 73 & 73 & 73 & 74.7 \\
| & & & & & & & & & & & & & & |
| Recall (%)                                   |
| $\begin{array}{cccccccccccccc}
| 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 & 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \\
| RF & 76.4 & 75.7 & 76.4 & 75.3 & 75 & 74.3 & 76 & 75.3 & 74.7 & 77.4 & 74 & 74 & 74 & 72.9 & 72.6 \\
| DT & 72.9 & 74.3 & 72.9 & 74.7 & 72.9 & 73.3 & 75.3 & 75.3 & 77.1 & 77.1 & 74.3 & 75.7 & 74.7 & 75.7 & 72.9 \\
| SVM1 & 73.3 & 73.3 & 73.3 & 73.3 & 73.3 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 & 74.7 \\
| SVM2 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 77.7 & 78.4 \\
| SVM3 & 77.1 & 77.1 & 77.1 & 77.1 & 77.1 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 & 75.7 \\
| & & & & & & & & & & & & & & |
| F-Measure (%)                                |
| $\begin{array}{cccccccccccccc}
| 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 & 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \\
| RF & 74.6 & 73.8 & 74.2 & 73 & 72.7 & 71.7 & 73.7 & 72.5 & 71.8 & 74.6 & 70.5 & 71 & 71.5 & 69.5 & 69.5 \\
| DT & 71.9 & 72.7 & 71.3 & 73.2 & 71.3 & 71.1 & 73.5 & 73.4 & 74.9 & 75.3 & 71.3 & 72.7 & 71.5 & 72.6 & 70 \\
| SVM1 & 71.7 & 71.7 & 71.7 & 71.7 & 71.7 & 72.2 & 72.2 & 72.2 & 72.2 & 72.2 & 72.2 & 72.2 & 72.2 & 72.2 & 71.3 \\
| SVM2 & 77 & 77 & 77 & 77 & 77 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.4 & 76.6 \\
| SVM3 & 73.7 & 73.7 & 73.7 & 73.7 & 73.7 & 71 & 71 & 71 & 71 & 71 & 71 & 71 & 71 & 71 & 68.8 \\
| & & & & & & & & & & & & & & |

Notes: $\lambda$ represents the cost scenario ($\lambda$ = 1.0−2.4, similarly hereinafter)

Although the total cost of SVM2 is slightly higher than SVM3, it is obviously lower than the other three classifiers. In addition, the performance of SVM2 in accuracy, precision, recall and F-Measure is significantly better than the other four classifiers, and the overall stability of the algorithm of is also the best.

In conclusion, SVM2 performs the best of the five classifiers, and it is a good classifier for outsourced
software project risk prediction.

![Graph showing comparison of total cost on different classifiers](image)

**Figure 1. Comparison of total cost on different classifiers**

5. CONCLUSIONS

This research is one of the first to introduce cost-sensitive learning method into outsourced software project risk prediction. The experiment results imply that the SVM2 outperforms other classifiers not only in accuracy, precision, recall and F-measure, but also has a good stability and lower cost. It is a good individual classifier for outsourced software project risk prediction.

In this study, we attempt to join the cost scenario to evaluate the risk of outsourced software project. In the practical applications, we can select model with appropriate $\lambda$ value according to the actual implementation situation of the outsourced software project in the early stages of the project. In this way, we can identify the risk better and achieve the goal of minimizing the loss of unknown outsourced software project.

The future researches include: Firstly, applies stronger classifiers to improve the prediction accuracy of outsourced software project risk. Secondly, investigates how to select the optimal model (i.e. optimal $\lambda$ value) according to the project experience or current status of the development team. Thirdly, considers using an ensemble method, namely integrating multiple individual classifiers (e.g. Decision Trees, SVM) to one classifier, to build the prediction model.

**ACKNOWLEDGEMENT**

This research was partly supported by National Natural Science Foundation of China (Project No.:71271061, 70801020), Science and Technology Planning Project of Guangdong Province, China (Project No.:2010B010600034, 2012B091100192), Business Intelligence Key Team of Guangdong University of Foreign Studies (TD1202), and Innovation Ability Cultivation Plan for graduate students of Guangdong University of Foreign Studies.

**REFERENCES**


