Filomat 30:15 (2016), 4015–4026
DOI 10.2298/FIL1615015X

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Clustering and Selection Based Transfer Ensemble Model for Customer Credit Scoring

Jin Xiao[a], Ling Xie[a], Dunhu Liu[b], Yi Xiao[c], Yi Hu[d]

[a]*Business School, Sichuan University, Chengdu 610064, China*
[b]*Management Faculty, Chengdu University of Information Technology, Chengdu 610103, China*
[c]*School of Information Management, Central China Normal University, Wuhan 430079, China*
[d]*School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China*

Abstract. Customer credit scoring is an important concern for numerous domestic and global industries. It is difficult to achieve satisfactory performance by traditional models constructed on the assumption that the training and test data are subject to the same distribution, because the customers usually come from different districts and may be subject to different distributions in reality. This study combines ensemble learning with transfer learning, and proposes a clustering and selection based transfer ensemble (CSTS) model to transfer the instances from related source domains to target domain for assisting in modeling. The experimental results in two customer credit scoring datasets show that CSTE model outperforms two traditional credit scoring models, as well as three existing transfer learning models.

1. Introduction

Over the past decades, the global credit businesses have developed rapidly, and the credit institutions are faced with more and more credit frauds, which results in huge losses. The value of outstanding consumer credit (excluding residential mortgage loan) in the US and UK at the end of 2009 was \$2.5 trillion and £171 billion respectively [1, 2]. At the same time, annualized write-off rates for credit cards and personal loans in the US and UK were 5.4% [2] and 1.5% [1, 3] respectively. Therefore, it is significant to build a scientific customer credit scoring model which can provide important decision support for the related people, and decrease the losses.

The first application of quantitative methods to customer credit scoring was undertaken by Durand, who adopted quadratic discriminant analysis to classify credit applications [4]. Since then, the most popular approaches to consumer risk assessment have continued to treat loan decisions as binary classification

2010 *Mathematics Subject Classification*. Primary 62P25

Keywords. customer credit scoring, transfer ensemble model, selective transfer learning, eliminate noise, multiple classifier ensemble

Received: 12 February 2015; Accepted: 24 February 2016

Communicated by Dr. Alex Maritz and Dr. Charles Xie

Research supported by the Natural Science Foundation of China under Grant Nos. 71471124, 71301160 and 71273036, National Social Science Foundation of China under Grant No. 14BGL175, Sichuan Province Social Science Planning Project under Grant No. SC14C019, Youth Foundation of Sichuan Province under Grant No. 2015RZ0056, Excellent Youth fund of Sichuan University under Grant No. skqx201607 and skzx2016-rcrw14.

Corresponding author: Yi Hu.

Email addresses: xiaojin@scu.edu.cn (Jin Xiao), xieling0101@126.com (Ling Xie), 264885613@qq.com (Dunhu Liu), xybill@amss.ac.cn (Yi Xiao), huyi@ucas.ac.cn (Yi Hu)

problems [5], i.e., divide the applicants into two categories based on their different default risks: applicants with good credit who always successfully fulfill the agreements and applicants with bad credit who have defaulted. At present, the commonly used credit scoring models include logistic regression [6], artificial neural network (ANN) [7], support vector machine (SVM) [8], VAR mode [9], etc.

The above models do not take the issue that the distributions of customer data are often highly imbalanced into account in customer credit scoring. The so-called class imbalance means that the applicants with bad credit constitute only a very small minority of the data (usually 2% of the total customers) [10]. For example, the Council of Mortgage Lenders, UK, reported that in the second quarter of 2010, the number of mortgages three or more months in arrears, i.e., default applicants with bad credit stood at 2.17% of total outstanding mortgages [11]. When the class distribution of the data is imbalanced, the misclassification rate of the above classification models for bad credit applicants is much higher than that of good credit applicants [12]. However, the value of accurate classification for a bad credit applicant is often higher than that of a good credit applicant [13].

At present, two types of approaches are proposed to deal with the class imbalance issue in customer credit scoring [14]: data-level and algorithm-level solutions. Data-level solutions mainly use resampling techniques, such as random over-sampling for the minority customers and random down-sampling for the majority customers, to balance the class distribution of the training set and construct the classification model. For example, Marqus et al. [15] investigated the suitability and performance of several resampling techniques when applied in conjunction with statistical and artificial intelligence prediction models over five real-world credit datasets, and their experimental results demonstrated that the use of resampling methods consistently improved the performance given by the original imbalanced data. Besides, oversampling techniques performed better than any under-sampling approach. Algorithm-level solutions attempt to adapt existing classification algorithms to strengthen learning with regard to the minority class. Such solutions mainly introduce cost sensitive learning technique and assign different misclassification costs to the customers from different classes. For instance, Zou et al. [16] combined support vector machine with cost-sensitive learning to construct cost-sensitive support vector machine for credit scoring. In addition, in recent years, some scholars have introduced multiple classifiers ensemble (MCE) technique to customer credit scoring for improving the generalization ability [13, 17]. For example, Paleologo et al. [17] proposed an ensemble classification technique, Subagging for highly imbalanced credit scoring data and demonstrated its effectiveness through experiments. The common characteristic of the two types of methods above is that they only use the original information in the inner system (target domain) to handle the class imbalance issue, and do not generate new information. According to the statistics learning theory, under certain sample information capacity, model accuracy has an upper limit [18]. Therefore, to improve the prediction accuracy of the minority customers for both types of solutions is usually on the condition that the prediction accuracy of the majority customers is sacrificed.

A popular phenomenon exists in the real customer credit scoring. There are a large number of customer data in related source domains, which may be from different districts, businesses, periods, or enterprises in the same industry. Although the customer data in the source and target domains are very similar, they are often subject to different distributions. It is difficult to achieve satisfactory performance in this case for most traditional models, because they are all based on the assumption that the training data and the test data are subject to the same distribution [19]. Therefore, it is expected to improve the credit scoring performance with imbalanced class distribution through integrating the data from the source and target domains effectively.

The transfer learning proposed in machine learning area provides a new idea for this issue, and its main idea is to utilize the data of related source domain tasks to assist in modeling for target task [19, 20]. In recent years, transfer learning has been applied to many areas such as text mining, image recognition, and so on. However, it is seldom applied to the customer credit scoring.

Combining the transfer learning with multiple classifiers ensemble (MCE) [21], this study proposes a clustering and selection based transfer ensemble (CSTS) model, and applies it to customer credit scoring. The experimental results in two customer credit scoring datasets show that CSTE can achieve better performance compared with the traditional credit scoring models, as well as some existing transfer learning models.

The structure of this study is organized as follows: it introduces the methodology proposed in this study in Section 2; presents the experimental design and detailed results analysis in Section 3. Finally, the conclusions are included in Section 4.

2. Methodology

2.1. Multiple classifiers ensemble

Classification is one of the key technologies in data mining, and has been applied to many areas such as speech recognition, text classification, image processing. Many classification learning algorithms generate a single classifier (e.g., a decision tree or neural network) that can be used to predict the class labels of new patterns. However, because the data in real classification issues include much noise, it is difficult to classify accurately in the whole pattern space with single classifier [22]. If we can integrate the classification results of some classifiers with MCE technique, and each classifier plays role in its dominant area, then it is hopeful to improve the classification accuracy [23]. A successful MCE system should have the following two characteristics [24]: firstly, the base classifiers for ensemble system have higher classification accuracy, at least greater than 0.5 [25]; secondly, the classification results of base classifiers should be diverse [26].

The construction of classifier ensemble strategies is a key step in multiple classifiers ensemble. The existing ensemble strategies can be divided into two types: 1) static classifier ensemble (SCE) [22], which selects a unified ensemble scheme for all test patterns; 2) dynamic classifier ensemble (DCE) [26]. In fact, different test patterns usually have different classification difficulties. Intuitively, if we adopt different classifiers for different test patterns, the classification performance may be better than that by SCE. This is also the basic idea of DCE. However, the time complexity of DCE is often much higher than that of SCE. Note that the CSTE model proposed in this study belongs to SCE strategy.

2.2. Transfer learning theory

In the past decades, research on transfer learning has attracted more and more attention in different names: learning to learn, life-long learning, knowledge transfer, inductive transfer, multi-task learning, knowledge consolidation, context sensitive learning, knowledge-based inductive bias, meta learning, and incremental/cumulative learning [20]. At present, there is no uniform definition about the transfer learning in academia. In 2005, the Broad Agency Announcement (BAA) 05-29 of Defense Advanced Research Projects Agency (DARPA)s Information Processing Technology Office (IPTO) gave a new mission of transfer learning: the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks. Figure 1 shows the difference between the traditional learning processes and transfer learning techniques [19].

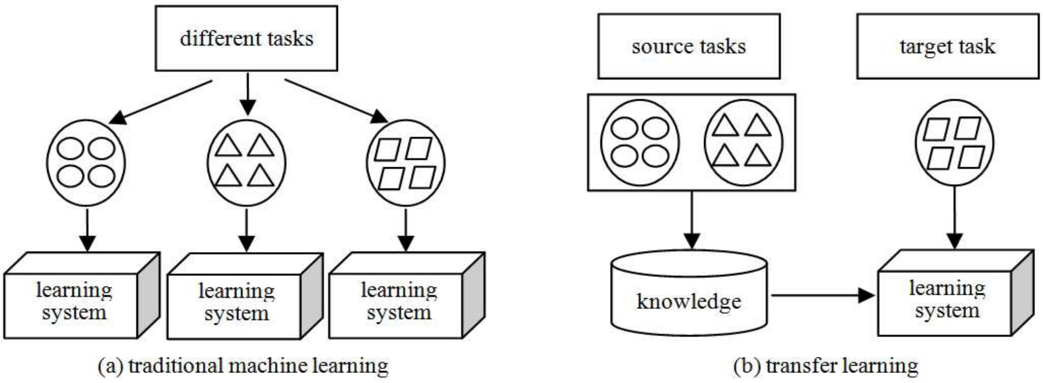

Figure 1: The comparison between traditional machine learning and transfer learning.

In general, the existing transfer learning strategies can be classified into four classes [19]: instance-based, feature-based, model-based, and related-knowledge-based. In recent years, many scholars have focused on transfer learning strategies, and the representative researches are as follows: the feature-based transfer component analysis (TCA) strategy [27], the instance-based TrBagg strategy [28], and the instance-based TrAdaBoost strategy [29]. Although these strategies have their own advantages, neither of them takes the imbalanced class distribution of data into consideration. So they can hardly conduct the credit scoring correctly when they are involved in CRM field. Whats more, the existing transfer learning strategies are seldom utilized in CRM.

2.3. Clustering and Selection Based Transfer Ensemble Model

2.3.1. The basic idea of the model

The CSTE model aims to transfer some useful information from the related source domain to assist in modeling for target domain. In real credit scoring issues, the source domains usually contain lots of noises. Thus, if we transfer the applicant samples from the source domain to the target domain indiscriminately or improperly, it may result in negative transfer due to introducing too many noises [30]. Therefore, it is necessary to consider how to avoid the negative transfer in constructing transfer learning model.

In Reference [31], we proposed a feature-selection-based dynamic transfer ensemble model, where we supposed that there was only one source domain related to the target domain. However, in real customer credit scoring issues, there may be many source domains, and how to construct transfer ensemble model in this case is the focus of this study. Suppose that T is the target domain dataset of a credit scoring issue, and there are p source domain datasets $S_{ri}(i = 1, 2, ..., p)$ related to T. At the same time, both T and S_{ri} contain two types of samples: bad credit samples with class label 1 and good credit samples with class label 2. Further, the target domain T is divided into two subsets: target training set T_1 and target test set T_2.

In order to avoid negative transfer effectively, the CSTE model proposed in this study contains 3 phases (see Fig. 2): 1) Transfer the source domain datasets selectively. It first utilizes k-means algorithm to divide T_1 into k cluster numbers, and obtains the initial clustering program C_I. In this study, we suppose the class label of the dataset only contains two classes: good credit and bad credit, therefore we let $k = 2$. Further, it combines each source domain dataset $S_{ri}(i = 1, 2, ..., p)$ with T_1, and clusters again with k-means algorithm (the cluster number is still 2) to get new clustering program $Cl_i(i = 1, 2, ..., p)$. Finally, it calculates the consistence between Cl_i and the initial clustering program C_I, and then transfers a half of source domains with higher consistence into the target training set to form the new training set T_R; 2) Eliminate the noise data in the new training set. Firstly, like in Phase 1, the new training set T_R is clustered into 2 clusters by k-means. If the samples belong to two classes in one cluster, then subdivide this cluster further. Meanwhile, it excludes the isolated cluster with fewer samples. At last, it numbers all clusters and regards the numbers as the samples new class label in the final training set T_f; 3) Train base classifiers and classify target test set T_2. It first selects N training subsets randomly with replacement from T_f. At this moment, the class distribution of the training subsets and then balances each subset with oversampling technology. Further, it trains a classifier in each balanced subset and classifies T_2 with each classifier. At last, it gets the final ensemble classification results by weighted voting.

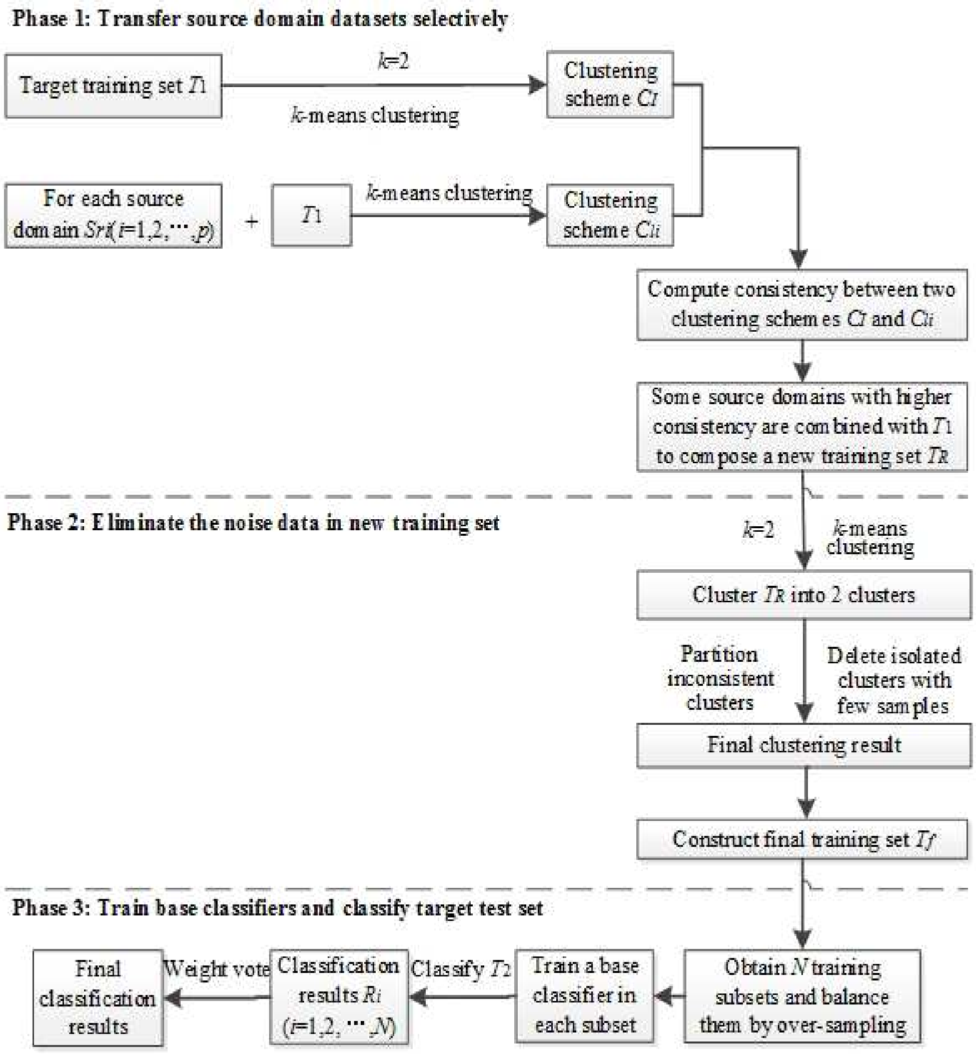
Phase 1: Transfer source domain datasets selectively
Target training set T1
k=2
k-means clustering
Clustering scheme CT
For each source domain Sri(i=1,2,⋯,p)
+
T1
k-means clustering
Clustering scheme Ci
Compute consistency between two clustering schemes CT and Ci
Some source domains with higher consistency are combined with T1 to compose a new training set TR
Phase 2: Eliminate the noise data in new training set
k=2
k-means clustering
Cluster TR into 2 clusters
Partition inconsistent clusters
Delete isolated clusters with few samples
Final clustering result
Construct final training set Tf
Phase 3: Train base classifiers and classify target test set
Final classification results
Weight vote
Classification results Ri (i=1,2,⋯,N)
Classify T2
Train a base classifier in each subset
Obtain N training subsets and balance them by over-sampling

Figure 2: The flow chart of CSTE model.

2.3.2. *Measure the consistence of two clustering programs with mutual information*

When transferring the source domain datasets selectively, it is important to measure the consistence of two clustering programs. In this study we regard the mutual information based method [32] as the measurement. Its theoretical description is described as following.

Firstly, suppose that we cluster the dataset with n samples by k-means clustering method, and obtain two labeled vectors with k clusters $\lambda^{(a)} = \{C_1^{(a)}, C_2^{(a)}, ..., C_k^{(a)}\}$ and $\lambda^{(b)} = \{C_1^{(b)}, C_2^{(b)}, ..., C_k(b)\}$ respectively. Further, suppose that there are n_i samples in the cluster $C_i^{(a)}$ and n_j samples in $C_j^{(b)}$, and there are n_{ij} same samples between $C_i^{(a)}$ and $C_j^{(b)}$. Then, the mutual information can be defined as follows:

$$\Phi^{NMI}(\lambda^{(a)}, \lambda^{(b)}) = \frac{2}{n}\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij} log_{k^2}(\frac{n_{ij}n}{n_i n_j}). \tag{1}$$

The mutual information value is between 0 and 1. The larger the value is, the more consistent two clustering programs are.

2.3.3. Detailed description of CSTE model

CSTE model contains four phases, and its pseudo-code is as follows:

Phase 1: Transfer the source domain datasets selectively

1) Divide T_1 into 2 clusters by k-means method and get the initial clustering program C_I;

2) Combine each source domain dataset $S_{ri}(i = 1, 2, ..., p)$ with T_1 respectively, cluster again by k-means algorithm (the cluster number is still 2), and suppose the new clustering program is $Cl_i(i = 1, 2, ..., p)$;

3) Calculate the consistence Φ_i between the initial clustering program and the new clustering program Cl_i based on Eq. (1), and then select a half of the source domains with higher Φ_i . Finally, add them to the target training set T_1 and construct the new training set T_R;

Phase 2: Eliminate the noise data in new training set

4) Divide T_R into 2 clusters with k-means method to get the clustering results $Clus_1, Clus_1$;

5) For each cluster $Clus_i(i = 1, 2)$, if it contains some samples with two different class labels, it is called inconsistent cluster, and we divide it into two sub-clusters according to the samples class labels: $Clus_{i1}, Clus_{i2}$; For each cluster $Clus_i$, if it only contains a few samples, i.e., isolated cluster, it is deleted directly. Finally, we number all the remaining clusters;

6) Set the cluster number as the new class label of samples in each cluster, and combine all the samples to get the final training set T_f;

Phase 3: Train the base classifiers and classify the target test set

7) Sample N training subsets from T_f randomly with replacement, and balance the class distribution of each training subset with the random over-sampling which is recommended by Marqus et al. [15];

8) Train a classifier in each training subset and get the base classifier set $C = C_1, C_2, ..., C_N$;

9) Classify the test set T_2 in the target domain with each base classifier $C_i(i = 1, 2, ..., N)$, and suppose the classification results are R_i. Further, integrate the classification results of N base classifiers with weighted voting and get the final ensemble classification results for T_2.

3. Empirical Analyses

In order to analyze the credit scoring performance of CSTE proposed in this study, we experimented in two datasets. Meanwhile, we compared CSTE model with the following five strategies: 1) traditional customer credit scoring model Subagging [17] by utilizing all data (Subagging), which trains N classifiers by uniting the data in the source domains with those in the target domain without distinction; 2) traditional Subagging by utilizing the target domain data only (Subagg-OT), which trains N classifiers by using the data in the target domain; 3) feature-based TCA strategy [27]; 4) instance based transfer learning strategy TrBagg [28]; and 5) instance based transfer learning strategy TrAdaBoost [29].

3.1. Datasets and data processing

(1) PAKDD2009 dataset

The first dataset is from Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2009 Data Mining Competition (http://sede.neurotech.com.br:443/PAKDD2009). The dataset of this year is a credit scoring problem that comes from the private label credit card operation of a major Brazilian retail chain. The client was labeled as bad (target variable=1) if, for 11 months after the first bill, he / she had any payment default (a delay longer than 60 days). Otherwise, the client was labeled as good (target variable=0). All the data from PAKDD2009 are divided into three datasets: modeling dataset with 50,000 samples, leaderboard dataset with 10,000 samples and prediction dataset with 10,000 samples. However, only the samples in the modeling dataset have the class label, therefore, we just select the modeling dataset in experiment for convenient analysis.

Table 1: Description of selected features in PAKDD2009 dataset.

Features	Description	Features	Description
x_1	Id_Shop	x_{11}	Flag_Mothers_Name
x_2	Sex	x_{12}	Flag_Fathers_Name
x_3	Marital_Status	x_{13}	Flag_Residence_Town=Working_Town
x_4	Age	x_{14}	If the applicant works in the same state where lives
x_5	Flag_Residencial_Phone	x_{15}	Time in the current job in months
x_6	Area_Code_Residencial_Phone	x_{16}	Profession_Code
x_7	Payment_Day	x_{17}	Mate_Income
x_8	Shop_Rank	x_{18}	Flag_Residencial_Address=Postal_Address
x_9	Residence_Type	x_{19}	Personal_Net_Income
x_{10}	Months_In_Residence	x_{20}	Cod_Application_Booth

There are 31 features in the data. After preliminary data cleaning, there are 49,904 samples and 20 features in the modeling dataset (see Table 1). We need to divide the dataset into target domain and source domain to take the transfer learning for experimenting. It is worth noting that x_6 means the modified residential phone area code. Its different values imply that the customers are from different regions, and these customer data may be subject to different distributions. Therefore, we divide 49,904 customer samples into 67 subsets according to different values of x_6 (from 1-70, but excluding 16, 55 and 66). We regard the subset $x_6 = 5$ (it contains 2,471 samples, where there are 341 customers with bad credit, and 2,310 ones with good credit) as the target domain dataset. Further, most of the remaining 66 subsets contain few samples (one or two), which lose the meaning of transferring. So we just select 10 subsets data that contain 10 samples at least as the source domain (the first 3 columns in Table 2 show 10 source domains).

Table 2: Description of the source domains and the multivariate two-sample testing results on PAKDD2009 dataset.

Number	Source domains	Number of samples	$\|\hat{t}\|$	$\|\hat{t}'_{950}\|$
1	$x_6 = 23$	994	96.832	1.8650
2	$x_6 = 24$	120	73.663	1.9766
3	$x_6 = 27$	27	78.688	1.3293
4	$x_6 = 31$	34,992	78.365	1.8924
5	$x_6 = 32$	33	94.545	2.4056
6	$x_6 = 38$	12	70.654	1.7283
7	$x_6 = 42$	14	83.675	1.9342
8	$x_6 = 49$	48	92.328	1.8476
9	$x_6 = 50$	11,071	76.437	1.8564
10	$x_6 = 56$	10	65.433	2.0341

(2) UK credit dataset

This dataset comes from the monograph of Thomas [33], and it is used to evaluate the UK credit. There are 14 attributes for the dataset – three nominal attributes, and eleven continuous attributes (see Table 3). It contains 1,225 customer samples that are divided into good credit with 902 samples and bad credit with 323 samples. The sample proportion is 2.79:1, so it belongs to highly class imbalanced dataset.

Table 3: Description of the features in UK credit dataset.

Features	Description	Features	Description
x_1	Year of birth	x_8	Residential status
x_2	Number of children	x_9	Value of home
x_3	Number of other dependents	x_{10}	Mortgage balance outstanding
x_4	Is there a home pone	x_{11}	Outgoings on mortgage or rent
x_5	Spouses income	x_{12}	Outgoings on loans
x_6	Applicants employment status	x_{13}	Outgoings on hire purchase
x_7	Applicants income	x_{14}	Outgoings on credit cards

In order to take the transfer learning for experiment, we notice that x_6 which means the applicants employment status has 11 different values from 1 to 11. The customers with different employment status may have different consumption habits, and the corresponding data may be subject to different distributions. Thus, we divide all the customer samples into 11 subsets according to different values of x_6. We regard the subset $x_6 = 1$ (it contains 231 samples, where there are 44 customers with bad credit, and 187 ones with good credit) as the target domain dataset, and the other 10 subsets as the source domains. However, the numbers of samples in the 3 source domains $x_6 = 9$, 10 and 11 are fewer than that in $x_6 = 10$, thus we delete them directly (the first 3 columns in Table 4 show the seven source domains), which is similar to the PAKDD2009 dataset.

Table 4: Description of the source domains and the multivariate two-sample testing results on UK credit dataset.

Number	Source domains	Number of samples	$\|\hat{t}\|$	$\|\hat{t}'_{950}\|$
1	$x_6 = 2$	37	257.42	3.8217
2	$x_6 = 3$	23	227.31	0.7554
3	$x_6 = 4$	531	711.97	33.834
4	$x_6 = 5$	30	301.68	21.446
5	$x_6 = 6$	104	603.46	87.222
6	$x_6 = 7$	124	202.56	19.466
7	$x_6 = 8$	123	528.73	74.325

To determine whether the distributions of the source domains and target domain are different, we introduced the multivariate two-sample testing procedure proposed in [34]. It can be roughly divided into the following steps: 1) Create a predictor variable training set $\{u_i\}_1^{m_1+m_2} = \{t_i\}_1^{m_1} \bigcup \{s_i\}_1^{m_2}$ by pooling the two samples, i.e., the target domain T and the source domain S, and assign a response value $y_i = 1(1 \leq i \leq m_1)$ to the observations originated from the first sample while assign $y_i = -1(m_1 + 1 \leq i \leq m_1 + m_2)$ to those from the second sample; 2) A binary classification learning machine (e.g., the support vector machine is selected in this study) is applied to this training data to produce a scoring function $L_m(u)$, and then this function is used to score each observation $\{score_i = L_m(u_i)\}_1^{m_1+m_2}$; 3) Generate two sets of score values $Score_+ = \{score_i\}_1^{m_1}$ and $Score_- = \{score_i\}_{m_1+1}^{m_1+m_2}$, regard the sets of numbers $Score_\pm$ as a random sample from respective probability distributions with densities $p_+(score)$ and $p_-(score)$, apply a univariate two-sample test (e.g., the two independent samples t-test is introduced in this study) for the equality of these densities $p_+(score) = p_-(score)$, and compute the test statistic $\hat{t}$; 4) Let $\{j(i)\}_1^{m_1+m_2}$ represent a random permutation of the integers $\{i\}_1^{m_1+m_2}$, and construct a dataset $\{y_{j(i)}, u_i\}_1^{m_1+m_2}$ in which the actual response values $\{y_i\}_1^{m_1+m_2}$ are randomly permuted among the predictors $\{u_i\}_1^{m_1+m_2}$; 5)Train a support vector machine with these data, score the observations, and compute the t-test statistic $\hat{t}'_1$; 6) Repeat Steps 4-5 1000 times to generate a set of test statistic values $\{\hat{t}'_i\}_1^{1000}$, sort them in ascending order according to their absolute values; 7) Giving a significance level α, one can reject the null hypothesis $p_+(score) = p_-(score)$ if $|\hat{t}| > |\hat{t}'_{1000*(1-\alpha)}|$. In this study, we let $\alpha = 0.05$, and the last two columns in Tables 2 and 4 show the test results for PAKDD2009 dataset and UK credit dataset respectively, where each row is to test whether there is difference between the distributions of target domain and the corresponding source domain. It can be seen that there is significant difference between the distributions of the target domain and the 10 source domains for PAKDD2009 dataset, and also significant difference between the distributions of the target domain and the 7 source domains for UK credit dataset.

3.2. Experimental setup

Before training the models, we need to partition the target domain T into the target training set T_1 and the target test set T_2. In this study, we adopted the random sampling without replacement method to select 30% patterns from T to construct T_2, and the remaining patterns composed T_1.

Many classification algorithms can be used to generate the base classifiers, in this study we choose support vector machine (SVM) [35] for its popularity and immense success in various customer classification tasks. When training SVM, the choice of kernel function is very important. We found that the classifier based on the radial basis kernel (RBK) could obtain the best performance through experimental comparison; thus, we designated it as the kernel function of SVM. The kernel parameter of the RBK and the regularization parameter were set as the default values. We did not optimize the parameters of the SVMs because we concerned more the relative performance of the compared ensemble models, rather than their absolute performance.

As for the CSTE model as well as other five models referred in the experiment, the number of the base classifiers is an important parameter. Note that all the six models belong to SCE model. Tsymbal et al. [36] found that the SCE models usually could achieve their best performance when the number of the base classifiers for ensemble equaled 50. Therefore, we let the size of base classifier pool for the six models be 50. For the other parameters in TCA, TrBagg, and TrAdaBoost models, we let them be the values which make the models perform best by repeated experiments. Meanwhile, except CSTE, Subagging and Subagg-OT models, the other three models do not consider the impact of class imbalance on the performance. To ensure the fairness of comparison, we balanced the class distribution of data by the random over-sampling technique before training the base classifiers. In addition, all experiments were performed on the MATLAB 6.5 platform with a dual-processor 2.1 GHz Pentium 4 Windows computer. For each model, the final classification result was the average of the results from 10 iterations of the experiment.

3.3. Evaluation criteria

In this study, the ability of the models to discriminate between "good" and "bad" applicants is evaluated by Receiver Operating Characteristic (ROC) curve analysis [8]. The ROC curves can also be used to compare the separated performance of two or more classifiers. Before we explain the ROC curve, we first introduce the confusion matrix in Table 5. For an issue of two classes, the ROC graph is a true positive rate – false positive rate graph, where $Y - axis$ is true positive rate ($TP/(TP + FN) \times 100\%$) and $X - axis$ is false positive rate ($FP/(FP + TN) \times 100\%$). The closer the curve follows the left and the top borders of the ROC space, the more accurate the model is. However, sometimes it is difficult to compare ROC curves of different models directly, so the area under the receiver operating characteristic curve (AUC) is more convenient and popular.

Table 5: Confusion matrix for credit scoring.

	Predicted positive	Predicted negative
Actual positive (bad credit customer)	TP (the number of True Positives)	FN (the number of False Negatives)
Actual negative (good credit customer)	FP (the number of False Positives)	TN (the number of True Negatives)

3.4. Impact of selective transferring, noise elimination and over-sampling on the performance of CSTE model

In CSTE model, selective transferring some most suitable source domains to the target domain and eliminating the noise samples in the final training set are two key steps. In addition, random over-sampling is adopted to balance the class distribution of the training set. To assess the impacts of selective transferring, noise elimination, and random over-sampling on the performance of CSTE model, we experimented with the following four strategies: 1) CSTE; 2) CSTE without random over-sampling, which is similar to CSTE except that it does not balance the final training subset with random over-sampling (called CSTE1); 3) CSTE without selective transferring, which transfers all of the source domain datasets without distinction to the target training set to obtain a new training set. Then it eliminates the noise data in the new training set, randomly samples some training subsets and balances each training subset with random over-sampling to train a classifier (called CSTE2); and 4) CSTE without noise elimination, which selectively transfers some most relevant source domain datasets to target training set, and then randomly samples some training subsets directly without the noise elimination as the Phase 2 of CSTE model. Finally, it balances all the training subsets with random over-sampling (called CSTE3).

Table 6: Experimental results in both datasets.

The results in PAKDD2009 dataset			**The results in UK credit dataset**		
Models	AUC	s.d.	Models	AUC	s.d.
CSTE	0.6689	0.0345	CSTE	0.6321	0.0675
CSTE1	0.6472	0.0541	CSTE1	0.5956	0.0734
CSTE2	0.6558	0.0522	CSTE2	0.6013	0.0843
CSTE3	0.6584	0.0481	CSTE3	0.6145	0.0692

Table 6 shows the experimental results in the two datasets. For each strategy, the average value and standard deviation (s.d.) of the AUC from 10 experiment runs are displayed. It can be seen that the performances of four strategies show the similar characteristics in both datasets: the CSTE model outperforms the other three strategies, after which come CSTE3, CSTE2, and finally CSTE1. Therefore, we can conclude that the impact of random over-sampling on the performance of CSTE model is the largest, followed by those of selective transferring and noise elimination, which also demonstrates that the imbalanced class distribution of customer data has a great impact on the performance of credit scoring model. However, the results above do not mean that the selective transferring and noise elimination are not important because the former can ensure transferring some most suitable source domains to the target domain and the latter can effectively eliminate the redundant samples in the final training set. In particular, the AUC value of CSTE2 model without selective transferring decreases by 0.0131 in PAKDD 2009 dataset than that of CSTE model, and 0.0308 in UK credit dataset. The AUC values of CSTE 3 without noise elimination decrease by 0.0105 and 0.0176 in PAKDD 2009 and UK credit respectively than those of CSTE model.

3.5. *Performance comparison with other models*

Tables 7 and 8 show the classification results of six models in PAKDD2009 dataset and UK dataset respectively. Based on the two tables, the following conclusions can be drawn: 1) The AUC values of CSTE model proposed in this study are the largest in both datasets, which demonstrates that the whole customer credit scoring performance of CSTE model is the best. 2) In six models, the standard deviation of CSTE model is always the smallest. The smaller the standard deviation of the model is, the more stable its performance is. Therefore, CSTE model shows better stability than the other models. 3) The credit scoring performance of Subagg-OT model which only utilizes the training set of target domain to model is poorer than that of four transfer learning strategies CSTE, TrBagg, TrAdaBoost and TCA, which demonstrates that it is important to transfer the source domains to the target domain. 4) In PAKDD2009 dataset, the credit scoring performance of the four transfer learning strategies including CSTE, TCA, TrBagg and TrAdaBoost is better than that of Subagging model. Further, in UK dataset, the transfer learning strategies CSTE, TCA and TrBagg still outperform Subagging model, and only the performance of transfer learning strategy TrAdaBoost is poorer than that of Subagging model. The Subagging model adds all source domain samples to the target domain without distinction, and then contracts multiple classifiers ensemble model. The results imply that most transfer learning strategies can eliminate some noise data in source domains through different internal mechanisms, and achieve better customer credit scoring performance than traditional multiple classifiers ensemble model Subagging.

Table 7: Customer credit scoring performance of six models in PAKDD2009 dataset.

Criteria	CSTE	Subagging	Subagg-OT	TCA	TrBagg	TrAdaBoost
AUC	0.6689	0.6548	0.6456	0.6582	0.6593	0.6581
s.d.	0.0345	0.0462	0.0732	0.0532	0.0492	0.0583

Table 8: Customer credit scoring performance of six models in UK credit dataset.

Criteria	CSTE	Subagging	Subagg-OT	TCA	TrBagg	TrAdaBoost
AUC	0.6321	0.6228	0.5956	0.6272	0.6233	0.6199
s.d.	0.0675	0.0662	0.0832	0.0678	0.0722	0.0783

4. Conclusions

Customer credit scoring is an important concern for numerous domestic and global industries. This study combines transfer learning with multiple classifier ensemble and proposes CSTE for customer credit scoring. Unlike the traditional research paradigm in customer credit scoring, which only utilizes the customer data in target domain, CSTE not only uses the data in target domain, but also utilizes the data in related source domains to assist in modeling. The experimental results in a customer credit scoring dataset show that CSTE not only outperforms two traditional credit scoring strategies, but also outperforms three existing transfer learning strategies.

References

[1] England, B. O. Trends in Lending June 2010. Bank of England, 2010.

[2] Board, TFR. Federal Reserve Statistical Release G.19. The Federal Reserve Board, 2010.

[3] Wen, F. H., He, Z., Dai, Z., Yang, X. Characteristics of investors' risk preference for stock markets, Economic Computation and Economic Cybernetics Studies and Research 48 (2014) 235–254.

[4] Durand, D. Risk Elements in Consumer Instatement Financing. National Bureau of Economic Research, New York, 1941.

[5] Lessmann, S., Baesens, B., Seow, H., Thomas, L. C. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research, European Journal of Operational Research (2015) doi: 10.1016/j.ejor.2015.05.030.

[6] Desai, V. S., Crook, J. N., Overstreet, G. A. A comparison of neural networks and linear scoring models in the credit union environment, European Journal of Operational Research 95 (1996) 24–37.

[7] Sustersic, M., Mramor, D., Zupan, J. Consumer credit scoring models with limited data, Expert Systems with Applications 36 (2009) 4736–4744.

[8] Chen, F. L., Li, F. C. Combination of feature selection approaches with SVM in credit scoring, Expert Systems with Applications 37 (2010) 4902–4909.

[9] Dai, Z., Wen, F. H. Robust CVaR-based portfolio optimization under a genal affine data perturbation uncertainty set, Journal of Computational Analysis & Applications 16 (2014) 93–103.

[10] Falangis, K. The use of MSD model in credit scoring, Operational Research, 7 (2007) 481–503.

[11] Lenders, C. M. CML reports decline in arrears and repossessions, http://www.cml.org.uk/ cml/media/ press/2680, 2009.

[12] Brown, I., Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Systems with Applications 39 (2012) 3446–3453.

[13] Xiao, J., Xie, L., He, C. Z., Jiang, X. Y. Dynamic classifier ensemble model for customer classification with imbalanced class distribution, Expert Systems with Applications 39 (2012) 3668–3675.

[14] He, H., Garcia, E. A. Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 1263–1284.

[15] Marqus, A., Garcła, V., S¢nchez, J. On the suitability of resampling techniques for the class imbalance problem in credit scoring, Journal of the Operational Research Society 64 (2013) 1060–1070.

[16] Zou, P., Hao, Y. Y., Li, Y. J. Customer value segmentation based on cost-sensitive learning support vector machine, International Journal of Services Technology and Management 14 (2010) 126–137.

[17] Paleologo, G., Elisseeff, A., Antonini, G. Subagging for credit scoring models, European Journal of Operational Research 201 (2010), 490–499.

[18] Vapnik, V. Statistical Learning Theory. John Wiley & Sons , New York, 1998.

[19] Pan, S. J., Yang, Q. A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359.

[20] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G. Transfer learning using computational intelligence: a survey, Knowledge-Based Systems 80 (2015) 14–23.

[21] Kittler, J., Hatef, M., Duin, R. P. W., Matas, J. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 226–239.

[22] Ranawana, R., Palade, V. Multi-classifier systems: review and a roadmap for developers, International Journal of Hybrid Intelligent Systems 3 (2006) 35–61.

[23] Mikel, G., Alberto, F., Edurne, B., Humberto, B. Francisco, H. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 42 (2012) 463–484.

[24] Kuncheva, L., Whitaker, C. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2003) 181–207.

[25] Hansen, L. K., Salamon, P. Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 993–1001.

[26] Xiao, J., He, C. Z., Jiang, X. Y., Liu, D. H. A dynamic classifier ensemble selection approach for noise data, Information Sciences 180 (2010) 3402–3421.

[27] Pan, S. J., Tsang, I. W., Kwok, J. T., Yang, Q. Domain adaptation via transfer component analysis, IEEE Transactions on Neural Networks 22 (2011) 199–210.

[28] Kamishima, T., Hamasaki, M., Akaho, S. TrBagg: A simple transfer learning method and its application to personalization in collaborative tagging, In: 2009 Ninth IEEE International Conference on Data Mining, pp. 219–228.

[29] Dai, W. Y., Yang, Q., Xue, G. R., Yu, Y. Boosting for transfer learning, In: 2009 Proceedings of the 24th international conference on Machine learning, pp. 193–200.

[30] Rosenstein, M. T., Marx, Z., Kaelbling, L. P. To transfer or not to transfer, In: NIPS 2005 Workshop on Transfer Learning.

[31] Xiao, J., Xiao, Y., Huang, A. Q., Liu, D. H., Wang, S. Y. Feature-selection-based dynamic transfer ensemble model for customer churn prediction, Knowledge and information systems 43 (2015) 29–51.

[32] Tang, W., Zhou, Z. Bagging-based selective clusterer ensemble, Journal of Software 16 (2005) 496–502.

[33] Thomas, L.C., Edelman, D.B., Crook, J.N. Credit Scoring and Its Applications, Philadelphia: Siam, 2002.

[34] Friedman, J. H. On multivariate goodness-of-fit and two-sample testing, In: 2003 Proceedings of Phystat, pp. 1–3.

[35] Cortes, C., Vapnik, V. Support vector networks, Machine Learning 20 (1995) 273–297.

[36] Tsymbal, A., Puuronen, S., Patterson, D. W. Ensemble feature selection with the simple Bayesian classification, Information Fusion 4 (2003) 87–100.