Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 379690, 13 pages
http://dx.doi.org/10.1155/2013/379690

Research Article

Weight-Selected Attribute Bagging for Credit Scoring

Jianwu Li, Haizhou Wei, and Wangli Hao

Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Jianwu Li; ljw@bit.edu.cn

Received 21 March 2013; Accepted 29 April 2013

Academic Editor: Yudong Zhang

Assessment of credit risk is of great importance in financial risk management. In this paper, we propose an improved attribute bagging method, weight-selected attribute bagging (WSAB), to evaluate credit risk. Weights of attributes are first computed using attribute evaluation method such as linear support vector machine (LSVM) and principal component analysis (PCA). Subsets of attributes are then constructed according to weights of attributes. For each of attribute subsets, the larger the weights of the attributes the larger the probabilities by which they are selected into the attribute subset. Next, training samples and test samples are projected onto each attribute subset, respectively. A scoring model is then constructed based on each set of newly produced training samples. Finally, all scoring models are used to vote for test instances. An individual model that only uses selected attributes will be more accurate because of elimination of some of redundant and uninformative attributes. Besides, the way of selecting attributes by probability can also guarantee the diversity of scoring models. Experimental results based on two credit benchmark databases show that the proposed method, WSAB, is outstanding in both prediction accuracy and stability, as compared to analogous methods.

1. Introduction

The assessment of credit risk has become increasingly crucial for financial institutions because high risks associated with inappropriate credit decisions may result in great losses [1]. In the latest financial crisis, many financial institutions suffered heavy losses from numerous customers' defaults on loans. Therefore, effective methods for evaluating credit risk are needed to help financial institutions to avoid losses [2]. The objective of credit scoring is to assign credit applicants to either a "good credit" group with high possibility to repay their financial obligation or a "bad credit" group with high possibility to default on their financial obligation. When considering the application for a large loan, the lender tends to evaluate the risk by a loan officer or even a committee to investigate the applicant in detail. Nevertheless, with the rapid growth and increasing competition in credit industry, it is necessary to perform fast automatic decisions on credit risk evaluations for financial institutions, especially when facing millions of applications for credit cards or consumer loans simultaneously. This demand leads to the birth of quantitative credit scoring method.

Quantitative credit scoring has gained more and more attention in recent years because an improvement in accuracy, even a fraction of a percent, can translate into significant future savings for the credit institutions [3]. Quantitative credit scoring models are developed based on the observations of historical data including income, age, and profession for good and bad examples, respectively. An excellent model should allow accurate classification of new applicants as good or bad.

Numerous models have been developed to evaluate consumer loans and improve credit scoring accuracy [4]. Initially, some statistical techniques are widely used to build credit scoring models such as linear discriminate analysis (LDA) [5] and logistic regression (LR) [6]. Although the two methods are relatively simple and explainable, the ability to discriminate good credit applicants from bad ones is still disputed. LDA is criticized because it needs a strong hypothesis such as the categorical property of the data and the variance homogeneity. In reality, the covariance matrices of good credit data are considerably distinct from those of bad credit data. Besides, both LDA and LR are linear classifiers, such that they are not suitable for complex nonlinear classification problems featured in credit scoring. In recent years, some new methods from the field of artificial intelligence have also been applied to the credit scoring, such as decision trees [7, 8], k-nearest neighbor (KNN) [9], artificial neural networks [10, 11], genetic algorithms [12–14], genetic programming [15, 16], artificial immune algorithms [17], and support vector machines (SVM) [18, 19]. Among these artificial intelligence methods, decision trees, artificial neural networks, and support vector machines are generally regarded as the most efficient individual models [20]. Furthermore, in order to improve the prediction accuracy and overcome the shortcoming of individual scoring model, two-stage scoring models [16, 21], hybrid scoring models [22, 23] and ensemble scoring models [20, 24] are also introduced. Experimental results show that these models perform better than individual classifiers.

Ensemble learning that combines outputs from multiple individual classifiers is one of the most important techniques for improving classification accuracy in machine learning [25, 26]. For example, an ensemble of multiple least square SVMs can obtain higher accuracy than the individual least square SVM in credit scoring and bankruptcy prediction [24]. Among ensemble learning models, bagging (short for "bootstrap aggregating") and boosting are two kinds of popular and widely used methods. Standard bagging (SB) [27, 28] is based on data partitioning which produces k working sets, each one with the same size as the original training set through randomly sampling from the original training set with replacement. Then, each working set is used to train a child classifier independently. During test phase, a test instance is evaluated by all child classifiers simultaneously and a collective decision is obtained based on some aggregation strategy. Boosting [25, 26, 29] is also based on data partitioning. Boosting produces a series of child classifiers, and the training dataset of each child classifier is generated according to the classification errors of previously created child classifiers. Test examples are classified by combining the predictions of all child classifiers according to a special aggregation strategy. AdaBoost is the most frequently used boosting method [25, 26].

Theoretical and experimental results suggest that combining classifiers can give effective improvement in accuracy if classifiers within an ensemble are not correlated with each other [30, 31]. One of the most effective methods of achieving such independence is to train the members in the ensemble by using different attribute subsets [32]. In other words, attribute partitioning methods can obtain better performance than data partitioning methods in ensemble learning [30]. Ensemble learning methods based on attribute partitioning are called as attribute bagging (AB) and have been investigated in many publications [33, 34]. Also, some AB models have been used to construct credit scoring systems and show promising results [35, 36].

For attribute bagging models, the selection of optimal attribute subsets plays an important role. Usually, attributes are selected randomly to construct attribute subsets. This method is called randomly selected attribute bagging (RSAB) [30]. However, RSAB has a deficiency that elements of some attribute subsets may contribute little to classification. The individual classifiers trained by such subsets perform badly in terms of accuracy. These individual classifiers lead to bad bagging results.

To overcome the shortcomings of RSAB, we propose a new attribute ensemble learning method, namely, weight-selected attribute bagging (WSAB). WSAB is based on the fact that some attributes are more important for the classification problem than others [37, 38]. Therefore, the more important attributes should appear more frequently in attribute subsets of AB model so as to guarantee that all individual classifiers perform well. In order to achieve this, the probabilities of the important attributes to be selected into attribute subsets should be larger than those of the unimportant attributes. Besides, the individual classifier that uses only a subset of original attributes will become more accurate after eliminating some redundant and uninformative features. On the other hand, given a certain size of attribute subsets (smaller than the total number of original attributes), selecting attributes by probability can result in some differences between different attribute subsets. Therefore, the diversity between different classifiers in an ensemble can be ensured. In other words, the WSAB can still keep the independence to some extent among different classifiers.

The implementation of the WSAB model contains two phases. In the first phase, weights of attributes need to be calculated using some attribute evaluation method. The weight w_i expresses the importance extent of the ith attribute for a given classification problem. In the second phase, an appropriate attribute subset size is firstly decided. Then, weights of attributes are used to construct attribute subsets such that the attributes with the larger weights will be selected into attribute subsets with the larger probability. In this way, attribute subsets contain the attributes that are important to classification, with the large probability, so that the accuracy of individual classifiers can be guaranteed. Then, like normal attribute bagging (AB), projections of training examples onto every attribute subset are created, respectively. Individual classifiers are trained based on every projection, and test instances are classified by combining the predictions of all individual classifiers according to a specific aggregation strategy, usually voting by majority. This paper attempts to introduce the WSAB model to solve credit scoring problem. Experimental results, based on two credit datasets from the UCI (University of California, Irvine, CA, USA) Machine Learning Repository [39], show that the WSAB is outstanding in both the prediction accuracy and stability, compared with RSAB, SB, AdaBoost, and single classifier.

The rest of this paper is organized as follows. The related research work is reviewed in Section 2. The WSAB model is described in Section 3. Subsequently, experimental results and empirical analysis are given in Section 4. The last section concludes this paper and addresses the future research task.

2. Related Work

2.1. Data Partitioning Ensemble Methods. Bagging [27] is a kind of important ensemble learning methods for improving prediction accuracy in machine learning. Standard bagging

(SB) is based on data partitioning. Given a training set S with size n, training examples are randomly sampled with replacement to generate m new sample subsets $S_1, S_2, \ldots, S_m$, each with the same size as the original training set. Due to sampling with replacement, some training instances may be repeated several times in a new training set, and also some may not appear at all. Then, standard bagging trains m models, $M_1, M_2, \ldots, M_m$, based on m training subsets, respectively. For a test instance, the final result is obtained via combining the outputs of m models by a specific strategy (voting for classification or averaging for regression). According to theoretical and empirical results, standard bagging can give a significant improvement in classification accuracy as well as stability [24]. In standard bagging, each classifier may be less accurate than the classifier using all training examples. However, after these classifiers are combined, the ensemble result is more accurate than the single classifier using all training samples. The diversity among individual classifiers compensates for the accuracy loss of individual classifiers in ensemble and hence improves prediction performance.

Boosting [29] is another effective method to improve the accuracy of any given learning algorithm. Boosting produces a series of classifiers, and the training dataset of each classifier is generated based on the accuracy of previously created classifiers. The samples misclassified by previously created classifiers will have larger probability to be selected into the new training dataset. By doing this, the new classifier can pay more attention to the samples that are difficult for previously created classifiers. Boosting can be implemented in several different ways. Arcing [28] and AdaBoost [40] are two important representatives. In Arcing, the classifiers' votes are weighted equally, while AdaBoost weights the predictions based on classifiers' training accuracies. It is noted that the effectiveness of boosting depends more on the data set than on base classifiers. Though boosting can significantly improve performance of weak classifiers, the ensemble is easy to focus on several special training examples that are difficult to be classified. Therefore, boosting is not stable.

Standard bagging and AdaBoost do not need too many rounds in training. Experimental results [34] show that the improvement of the performance of the learning model occurs often in the first several rounds.

2.2. Attribute Partitioning Ensemble Methods. Compared to data partitioning ensemble methods, attribute partitioning ensemble methods can make individual classifiers within an ensemble more "independent" [32] and then can obtain better effectiveness [30].

The bagging method based on attribute partitioning can be called attribute bagging (AB). The AB method generates attribute subsets through selecting attributes from the whole attribute set without replacement. Then, projections of training examples onto attribute subsets are created. Each child classifier is trained based on each projection, respectively, and all child classifiers are aggregated by some combination strategy. During the test phase, a test instance is fed to all child classifiers simultaneously and a collective decision is obtained based on the aggregation strategy. In conventional attribute bagging methods, attribute subsets are generated through randomly selecting attributes from the whole attribute set. This method is called randomly selected attribute bagging (RSAB). For RSAB, all attributes have the same probability to be selected into one attribute subset. However, some attributes are very important but the others are not important for classification problems. As mentioned before, RSAB has a deficiency that some attribute subsets may only contain the attributes that contribute less to classification. Such classifiers are prone to resulting in bad bagging results.

To overcome the deficiency of RSAB, some optimization methods are used to select optimal attribute subsets. Guerra-Salcedo and Whitley use a genetic algorithm (GA) to explore the space of all possible feature subsets [41]. Their experiments compare two data partitioning ensemble methods including bagging and AdaBoost with three different AB models: complete, random and genetic, search. Experimental results show that attribute subsets selected by GA perform best, followed by RSAB. Optiz also uses GA to search for attribute subsets for ensembles [33], and experimental results also demonstrate the fact that genetic ensemble feature selection (GEFS) performs better than standard bagging (SB) and AdaBoost. However, using GA to select attribute subsets is very computationally intensive, and GA needs to evaluate each classifier after combining two objectives—accuracy and diversity in a subjective manner. Another method of selecting optimal attribute subsets for ensemble is proposed by Bryll et al. [30]. They only used the best random attribute subsets for voting. Their experiments show that ranking attribute subsets by classification accuracy and then only using the best subsets further improve the classification performance of ensemble. However, to determine which individual classifier is more accurate is very difficult because test dataset is not known beforehand. Besides, only using the best classifiers to perform ensemble may reduce the diversity among classifiers. Therefore, it is difficult to reach a balance between accuracy and diversity of individual classifiers.

3. Weight-Selected Attribute Bagging (WSAB)

3.1. Evaluating Weights of Attributes. In the first phase of WSAB modeling, weights of attributes need to be computed using attribute (or feature) evaluation method. In practice, many methods can be used to obtain weights of attributes such as linear SVM (LSVM), principal component analysis (PCA), correlation analysis, F-score model, LDA, and multivariate adaptive regression splines (MARS). Different approaches to decide weights of attributes are of different characteristics. In this paper, we attempt to employ LSVM and PCA, respectively, to evaluate weights of attributes.

3.1.1. Evaluating Weights of Attributes via LSVM. The SVM, proposed by Vapnik [42], is based on the statistical learning theory and has showed state-of-the-art performance for many classification problems. Basic SVM is designed to solve binary classification problems and implements the structural risk minimization theory by seeking a maximum-margin

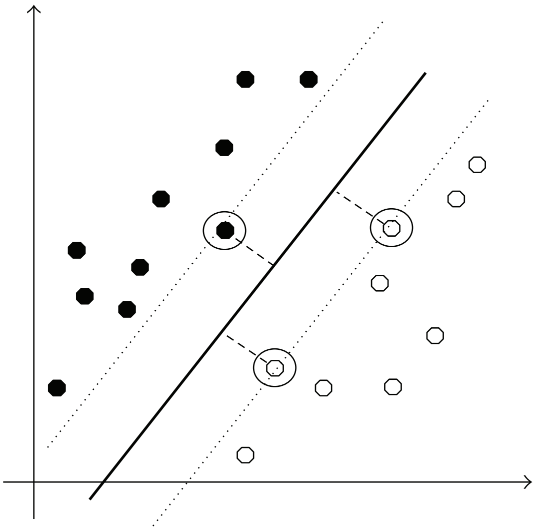

Figure 1: A linear SVM in 2-dimensional space with a maximum-margin hyperplane.

hyperplane between positive examples and negative examples in original space (linear SVM) or a high-dimensional feature space (nonlinear SVM with kernel trick). Figure 1 illustrates a linear SVM in two-dimensional space, where the examples on the boundary (two dashed lines with a maximum margin between two classes) are called support vectors, and the middle thick real line between these two dashed lines is the separator. An interesting result is that SVM can only use support vectors (a fraction of the set of all training examples) to form sparse solution and, thereby, has a fast classification speed.

Assume that there exists a training example set $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where $x_i \in R^d$, $y_i \in \{-1, +1\}$, and y_i represents class labels of training examples x_i, $i = 1, 2, \dots, n$. For seeking the maximum-margin hyperplane $y = w^T \cdot x$, the optimization objective of SVM is to minimize

$$F(w, \xi_i) = \frac{1}{2}\left(w^T \cdot w\right) + C\sum_i \xi_i, \quad (1)$$

subject to the constraints

$$y_i\left(w^T \cdot x_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \; i = 1, 2, \dots, n, \quad (2)$$

where C is a penalty factor. To solve this quadratic optimization problem, we need to find the saddle point of the Lagrange function:

$$\begin{aligned} L_p(w, b, \alpha) = {} & \frac{1}{2}w^T \cdot w + C\sum_{i=1}^{n}\xi_i \\ & - \sum_{i=1}^{n}\alpha_i\left[y_i\left(w^T \cdot x_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{n}\beta_i\xi_i, \end{aligned} \quad (3)$$

where α_i, β_i are Lagrange multipliers and $\alpha_i \geq 0$, $\beta_i \geq 0$. By differentiating with respect to w and b, and using the Karush Kuhn-Tucker (KKT) condition, L_p is transformed to the dual Lagrangian $L_D(\alpha)$:

$$\begin{aligned} & \operatorname*{Max}_{\alpha} \quad L_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\left(x_i^T \cdot x_j\right), \\ & \text{subject to:} \quad 0 \leq \alpha_i \leq C, \\ & \qquad i = 1, 2 \dots, n, \\ & \qquad \sum_{i=1}^{n}\alpha_i y_i = 0. \end{aligned} \quad (4)$$

The solution $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$ can be obtained by solving this quadratic optimization problem. Those $\{x_i\}$ corresponding to nonzero α_i^* are called support vectors. The parameters w^* and b^* of the optimal hyperplane can be obtained as follows:

$$\begin{aligned} w^* &= \sum_{i=1}^{N_s}\alpha_i^* y_i x_i, \\ b^* &= y_i - \sum_{i=1}^{N_s}\alpha_i^* y_i\left(x_i^T \cdot x_j\right), \end{aligned} \quad (5)$$

where N_s denotes the total number of support vectors.

Then, the optimal hyperplane decision function $f(x) = \operatorname{sgn}((w^{*T} \cdot x) + b^*)$ can be written as

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N_s} y_i\alpha_i^*\left(x_i^T \cdot x\right) + b^*\right). \quad (6)$$

SVM with linear kernel can be used to evaluate the weights of attributes. The decision function can be rewritten as

$$f(x) = \operatorname{sgn}\left(w^{*T} \cdot x + b^*\right). \quad (7)$$

According to [43], the vector $w^* = (w_1, w_2, \dots, w_d)^T$ can be used as weights of attributes, where d represents the total number of attributes. LSVM categorizes new data instances by testing whether the linear combination of the components of the vector $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$, $w_1x_{i1} + w_2x_{i2} + \cdots + w_dx_{id}$, is above or below some threshold $-b^*$. It is easy to find that the larger the $|w_j|$ the larger the impact of the corresponding jth attribute on the linear combination $w_1x_{i1} + w_2x_{i2} + \cdots + w_dx_{id}$. This means that the final classification result is sensitive to the jth attribute, and then the jth attribute can be considered as being important for classification problem.

Attribute (or feature) evaluating using LSVM has a strict theoretical foundation [43]. Obviously, an attribute is considered as being important if it significantly influences the width of the margin between two classes. According to the theory of SVM, the margin is inversely proportional to the length $\|w\|$ of w. For the solution $w = \sum_i \alpha_i y_i x_i$ obtained by linear SVM (for the convenience of expression, the stars above w and α_i are omitted), $\|w\|^2$ can be regarded as a function of the training vectors $x_1, x_2, \ldots, x_{N_s}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, and thus the influence of feature j on $\|w\|^2$ can be evaluated via absolute values of partial derivatives of $\|w\|^2$ with respect to x_{ij}. This approach can provide an approximate analysis on importance extent of attributes although it neglects the fact that the values of the multipliers α_i will change with training vectors changing [43].

For linear SVM, it turns out that

$$\sum_i \left| \frac{\partial \|w\|^2}{\partial x_{ij}} \right| = k \left| w_j \right|, \tag{8}$$

where the sum is over all support vectors and k is a constant independent of j. Thus, the attributes with the higher $|w_j|$ are of the more important role in determining the width of the margin. Intuitively, this type of feature weighting seems to be appealing because features with small value of $|w_j|$ do not have large influence on the output of SVM. Thus, these features can be considered as being unimportant for classification.

Support vector machine is based on the structural risk minimization theory and is an outstanding classification method. Meanwhile, the credit scoring problem is also a classification problem in essence. Hence, it is natural and reasonable that linear support vector machine is adopted to evaluate weights of attributes. In other words, the weights decided by LSVM are closely related to classification ability of classifiers.

3.1.2. Evaluating Weights of Attributes via PCA. As an alternative method, PCA is also used to evaluate weights of attributes in this paper. The main idea of PCA is to find the principal directions which best describe the distribution of credit samples within the entire credit sample space. The original variables are transformed to a set of new variables which are uncorrelated with each other and can be ranked from large to small in terms of variance such that the first several variables retain most of the variation in the entire original data.

Considering the set $S' = \{x_1, x_2, \ldots, x_n\}$, which only contains input vectors of training examples in $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. The vectors in S' are centered by $c_i = x_i - a$, where $a = (1/n)\sum_{i=1}^{n} x_i$, and then principal component analysis is performed to seek d orthonormal vectors, u_k $(k = 1, 2, \ldots, d)$, which best describe the distribution of original data. The kth vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{n} \sum_{i=1}^{n} \left(u_k^T c_i \right)^2 \tag{9}$$

is maximized, subject to

$$u_g^T u_k = \delta_{gk} = \begin{cases} 1, & \text{if } g = k, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Thus, the vectors u_k and the scalars λ_k are the eigenvectors and the eigenvalues, respectively, of the covariance matrix

$$C = \frac{1}{n} \sum_{i=1}^{n} c_i c_i^T = \frac{1}{n} A A^T, \tag{11}$$

where $A = [c_1, c_2, \ldots, c_n]$.

The eigenvectors can be ranked via their corresponding eigenvalues from large to small to reflect their importance extent in characterizing the variation of original data. These eigenvectors span a new space, and all training samples are projected into the new space.

An example x is transformed to $v = (v_1, v_2, \ldots, v_h)$ in the new space by the following operation,

$$v_k = u_k^T (x - a), \quad k = 1, 2, \ldots, h, \tag{12}$$

where h is the total number of the reserved eigenvectors.

The eigenvalue λ_k is the variance of original data in the direction of the eigenvector u_k and hence reflects the capability of the kth attribute in the new space in describing original data. As such, $\alpha_k = \lambda_k / \sum_{i=1}^{h} \lambda_i$ can be defined as the importance percentage of the kth attribute in the new space.

PCA and LSVM provide two different ways of evaluating weights of attributes. LSVM method selects attributes in the original space while PCA method selects attributes in the transformed feature space. Additionally, weights of attributes decided by LSVM reflect the classification ability of the corresponding attributes, and those obtained by PCA reflect the description capability for original data distribution.

3.2. Weight-Selected Attribute Bagging. After obtaining weights of attributes, we can select attributes based on probabilities decided by weights of attributes and train multiple classifiers using different attribute subsets, respectively, to perform attribute bagging.

The essence of weight-selected attribute bagging (WSAB) lies in the way of selecting attributes for each individual classifier in an ensemble (bagging). Concretely speaking, weights of attributes are used to construct many different attribute subsets such that the attributes with the larger weights have the larger probabilities to be selected into each attribute subset. The selection of attributes for each single attribute subset does not permit repetition of attributes that is, there are no repeated attributes in each attribute subset, but the same attributes can be chosen into different attribute subsets. Thus, the subsets containing unimportant features only can be avoided with a larger probability, compared to randomly selected attribute bagging (RSAB), so that the classification accuracies of individual classifiers can be guaranteed. On the other hand, the diversity of individual classifiers can be still ensured because attributes are chosen by probabilities and there exist differences among different attribute subsets.

Subsequently, like standard attribute bagging (AB), projections of training examples onto these attribute subsets are created. Individual scoring models are trained based on each projection, respectively, and all individual scoring models are aggregated by a specific strategy for test instances.

The appropriate size of attribute subset can be determined by cross-validation technique. The original training set is divided into two parts—a new training set and a validation set. With the attribute subset size changing from 1 to d (weights are decided by linear SVM) or h (weights are decided by PCA), d (or h) WSAB models with different attribute subset sizes are created, respectively. Then, the validation set is used to test these WSAB models to find out the optimal attribute subset size.

The main steps of WSAB are as follows.

Step 1. Compute weights of attributes, $w_1, w_2, \dots, w_d$ (or w_h), via attribute evaluating method.

Step 2. Decide an appropriate attribute subset size, m, by cross-validation.

Step 3. Generate a series of attribute subsets through repeating the following substeps.

Substep 1. Construct an array with l elements (l is large enough). The ith attribute takes $w_i / \sum_j w_j$ part of the array, $i = 1, 2, \dots, d$ or h.

Substep 2. Perform the following cycle to construct an attribute subset with m attributes:

(A) $t = 0$;

(B) randomly select an element of the array into the subset;

(C) delete all positions of the chosen element from the array;

(D) $t = t + 1$;

(E) if $t = m$, one attribute subset is created; else go to (B).

Step 4. Create projections of training examples onto the selected attribute subsets.

Step 5. Train individual classification models based on each projection, respectively, and use all individual scoring models to vote for test instances.

The modeling process of WSAB is illustrated in Figure 2.

4. Experimental Results and Comparisons

4.1. Credit Datasets. Two datasets are described in Table 1, which are German credit dataset and Australian credit dataset from the UCI Machine Learning Repository [39]. In German credit dataset, there are 1000 instances which contain 700 instances of creditworthy applicants and 300 instances regarded as bad credit applicants. Each example consists of 24 predictive attributes. Australian credit dataset includes 690 observers which record 307 good credit applicants and 383 bad credit applicants, and every instance consists of 14 predictive attributes.

4.2. Experimental Settings. We randomly divided the whole dataset into two parts—training set and test set. Training set takes two-thirds of the whole dataset and testing set takes one third of the whole dataset. The SVM with a Gaussian kernel was chosen as the basic classifier. Grid search in training set was used to decide the best parameters of SVM. The PR_tools [44] developed by MATLAB language was utilized as experimental platform. Each bagging model was repeated for 30 times, and then the average accuracy and the average standard deviation of accuracy were computed.

4.3. Evaluating Weights of Attributes

4.3.1. Evaluating Weights of Attributes via LSVM. The LSVM was performed on training dataset to obtain the weight of the jth attribute, w_j. Then the importance percentage of the jth attribute was computed as $w_j / \sum_i w_i$. The results are shown in Figure 3 for Australian dataset and Figure 4 for German dataset, respectively.

For Australian dataset, the top four important attributes make up 61.30% of the sum of all attribute weights, while the four most unimportant attributes only make up 3.82%. In addition, the most important one makes up 28.91% of the sum of all attribute weights and the most unimportant one makes up only 0.37% of the sum. The statistical results mean that, for Australian dataset, a few attributes are very important yet some other attributes contribute less to classification.

For German dataset, the top seven important attributes make up 56.33% of the sum of all attribute weights, while the four most unimportant attributes only make up 8.46%. Meanwhile, the most important attribute makes up 12.76% of the sum of all attribute weights and the most unimportant one makes up only 0.71% of the sum. Compared to Australian dataset, we can conclude that the importance extent of each attribute in German dataset is more evenly distributed.

4.3.2. Evaluating Weights of Attributes via PCA. We also performed PCA to calculate weights of attributes. Just as described in Section 3, weights of attributes obtained by linear SVM reflect the classification ability of attributes, but those obtained by PCA mainly describe the distribution of the original data.

The eigenvalues λ_i and their corresponding eigenvectors of the covariance matrix for each dataset were calculated. Then, all training instances were projected onto the space spanned by h orthonormal eigenvectors. The importance percentage of each new attribute can be presented as $\lambda_j / \sum_{i=1}^{h} \lambda_i$. Analogous to the analysis on weights of attributes from LSVM, the distribution of the attribute importance obtained by PCA is shown in Figure 5 for Australian dataset and Figure 6 for German dataset.

For Australian dataset, the top four important attributes make up 78.3% of the sum of all attribute weights, while the four most unimportant attributes only make up 1.80%. Additionally, the most important attribute makes up 30.2% of the sum of all attribute weights and the most unimportant one makes up only 0.23%. The results reflect the fact that several main attributes contribute more to the description of the original data for Australian dataset, yet some other attributes provide less information for the original data.

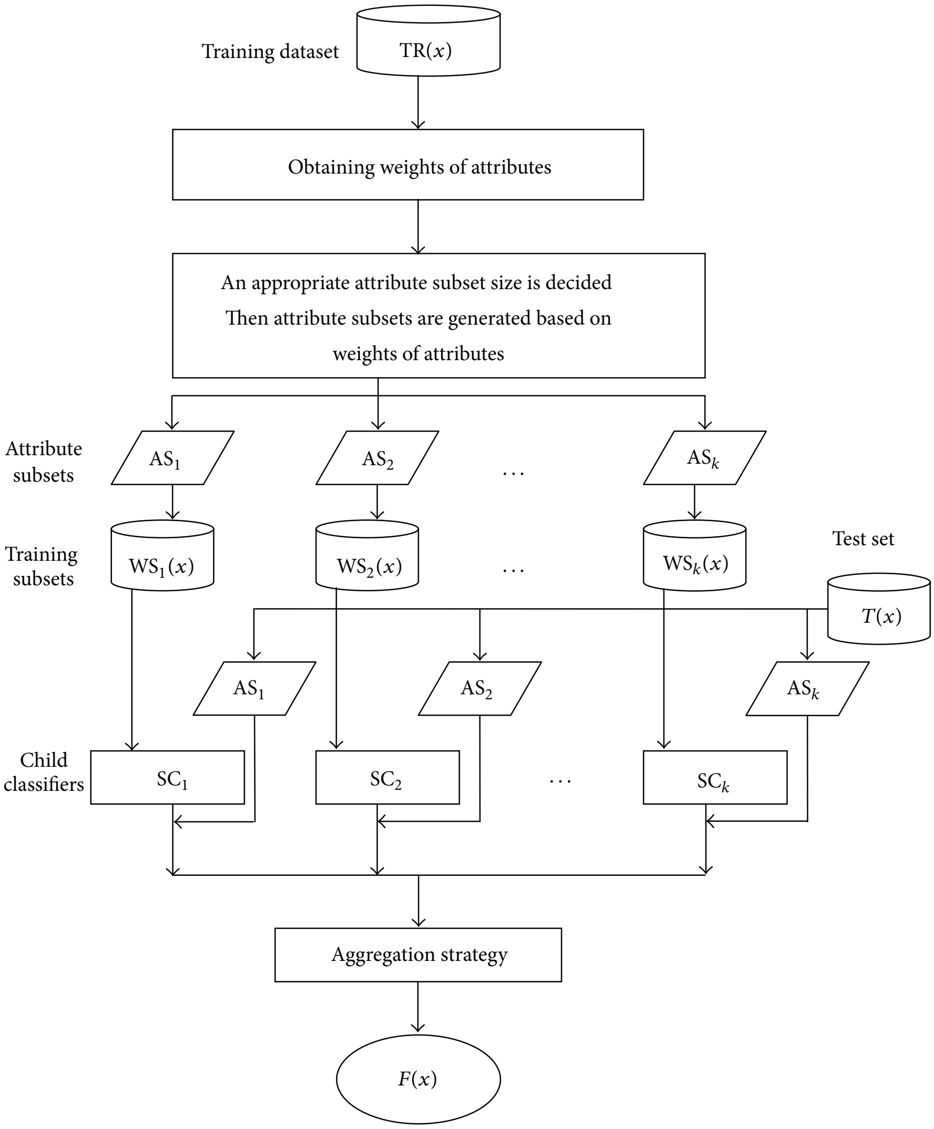

Figure 2: Weight-selected attribute bagging model.

Table 1: Description of credit datasets.

	German credit	Australian credit
Number of observers	1000	690
Number of predictive attributes	24	14
Percentage of good credit	70%	44.5%
Percentage of bad credit	30%	55.5%

For German dataset, the top seven important attributes make up 63.25% of the sum of all attribute weights, while the seven most unimportant attributes only make up 5.85%. The most important attribute makes up 13.5% of the sum of all attribute weights and the most unimportant attribute makes up only 0.33%. Therefore, the importance of each attribute for German dataset is more evenly distributed, compared to Australian dataset.

Interestingly, the results obtained via PCA are similar to those via LSVM not only for Australian dataset but also for German dataset, although LSVM and PCA adopt different approaches to calculating the weights of attributes one for data classification, and the other for data description.

4.4. Performance Comparison on Different Attribute Bagging Methods with Different Attribute Subset Sizes

4.4.1. Comparison on Accuracy. The size of attribute subset is critical for attribute bagging. Hence, this section will evaluate classification accuracy of the WSAB with different sizes of attribute subsets. Meanwhile, several other related methods were also compared.

For the convenience of expression, the WSAB using LSVM to determine weights of attributes is abbreviated as LSVM-WSAB; the WSAB using PCA to calculate weights of attributes is denoted as PCA-WSAB; the randomly selected attribute bagging is written as RSAB. For each bagging method as well as for each size of attribute subset ranging from 1 to d (or h), 45 classifiers were built to perform voting for test instances. Concretely speaking, for each attribute bagging method, 45 attribute subsets with the same subset size m were created, and then training examples and test examples were, respectively, projected onto the m selected attributes. Subsequently, 45 SVMs corresponding to 45 attribute subsets, respectively, were trained to vote for test instances. For each attribute subset size and each bagging method, 30 trials of the above process were performed and their results were averaged to evaluate classification accuracy. Furthermore, in order to prove the superiority of the voting, each attribute bagging method was also compared with the best single classifier. The best single classifier is represented as BS-SVM. The final results are given in Figure 7 for Australian dataset and Figure 8 for German dataset.

From Figures 7 and 8, the accuracy of WSAB is not high for small size of attribute subset, but rises gradually as the size of attribute subset becomes larger; then the accuracy tends to decrease when the size of attribute subset becomes large enough. Meanwhile, RSAB has the same behavior as WSAB. However, when the size of attribute subset is small, LSVM-WSAB and PCA-WSAB are more accurate than RSAB. In addition, LSVM-WSAB and PCA-WSAB are able to use smaller size of attribute subset to reach the maximum accuracy than RSAB. Meanwhile, the maximum accuracies of LSVM-WSAB and PCA-WSAB are higher than that of RSAB. We can provide a reasonable explanation for this phenomenon. For too small attribute subsets, individual classifiers used for voting have low accuracies since the information used for classification is lost too much. On the other side, for too large attribute subsets, the diversity among all members in ensemble decreases, so that the ensemble effect is affected. Additionally, for RSAB, the attributes in attribute subsets are selected randomly. Hence, RSAB needs larger sizes of attribute subsets to acquire sufficient information in order to reach its highest classification accuracy. However, WSAB selects attributes in terms of weights, such that WSAB can use smaller sizes of attribute subsets to contain most of important attributes and then to achieve its highest accuracy.

LSVM-WSAB, PCA-WSAB and RSAB are more accurate than BS-SVM for large size of attribute subset. The maximum accuracies of LSVM-WSAB, PCA-WSAB, and RSAB are higher than that of BS-SVM. The results prove that attribute bagging can improve effectively the performance of single classifier.

Additionally, PCA-WSAB needs less attributes to reach the maximum accuracy than LSVM-WSAB. The reason lies in the fact that PCA eliminates the correlation among attributes and the important attributes are more concentrated on several eigenvectors.

Moreover, an interesting finding for WSAB is that small size of attribute subsets can achieve high accuracy for Australian dataset, whereas for German dataset, accuracy of WSAB rises slowly with the size of attribute subset increasing. As we mentioned before, the important attributes in Australian dataset are concentrated on only several variables, and the remaining attributes contribute less to classification. Hence, WSAB model can acquire enough information using a small size of attribute subset for Australian dataset. However, for German dataset, the importance of attributes is more evenly distributed, and thus WSAB needs larger size of attribute subset to obtain enough information for classification.

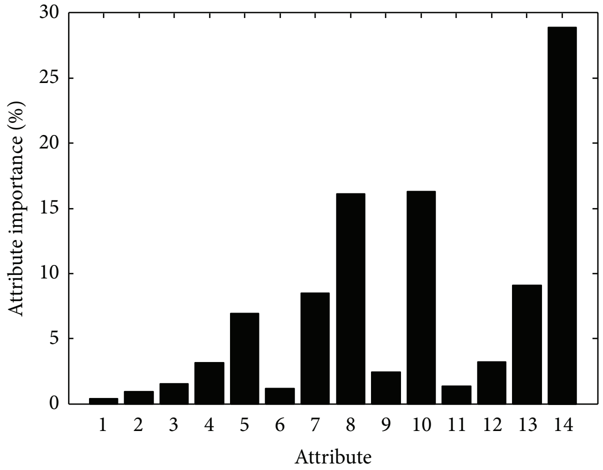

Figure 3: Importance percentage of each attribute for Australian dataset.

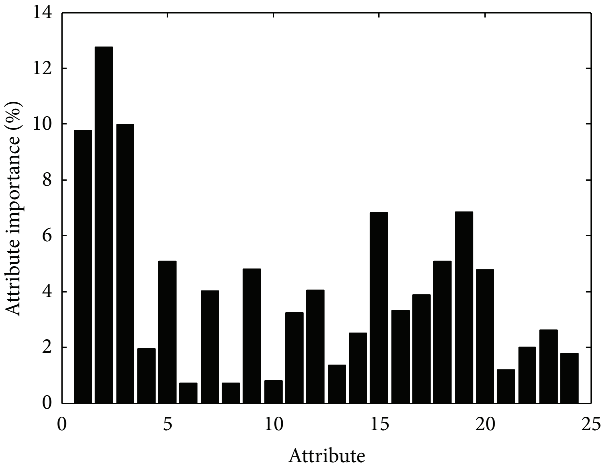

Figure 4: Importance percentage of each attribute for German dataset.

4.4.2. Comparison on Stability. When computing average accuracy of 30 trials for each attribute bagging model as well as for each attribute subset size, the standard deviation of accuracy was also computed to evaluate the classification stability of attribute bagging models. The standard deviations are shown in Figure 9 for Australian dataset and Figure 10 for German dataset.

From Figures 9 and 10, when the size of attribute subset is small, the standard deviation of accuracy of WSAB is larger than RSAB; when the size of attribute subset becomes larger gradually, the standard deviation of accuracy of WSAB

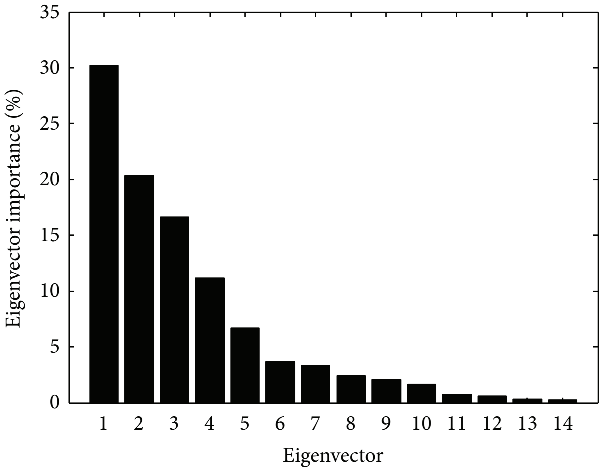

Figure 5: Importance percentage of each new attribute for Australian dataset.

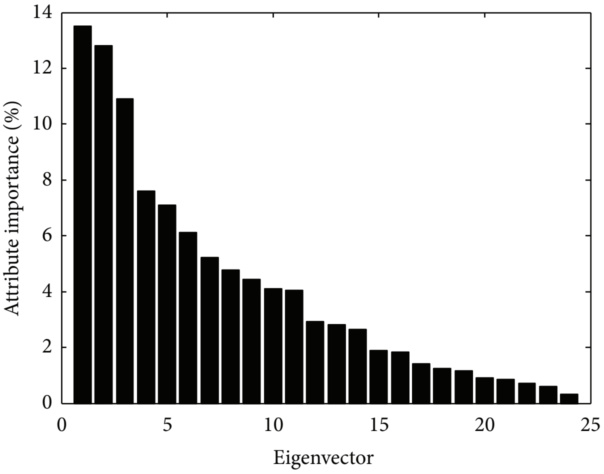

Figure 6: Importance percentage of each new attribute for German dataset.

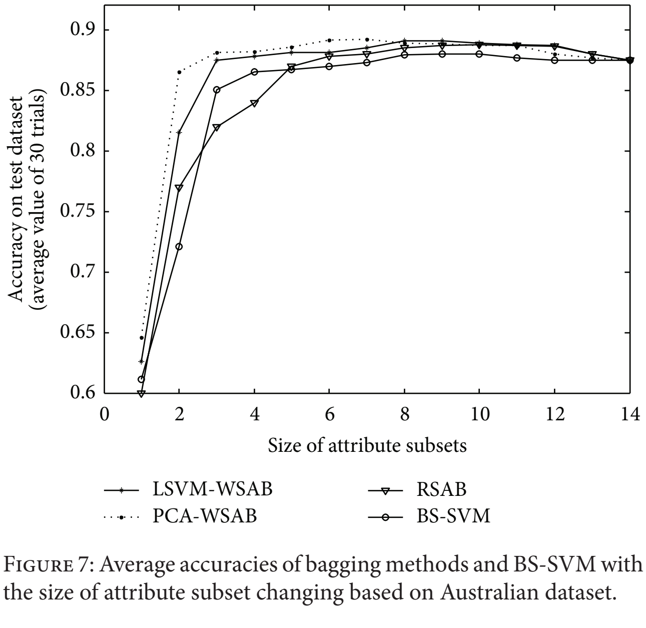

Figure 7: Average accuracies of bagging methods and BS-SVM with the size of attribute subset changing based on Australian dataset.

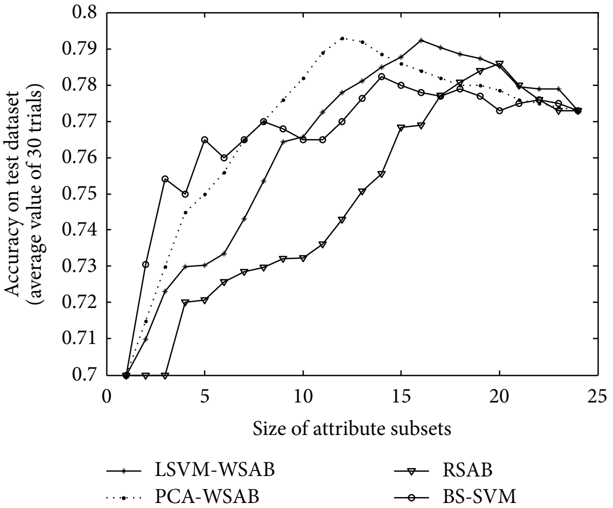

Figure 8: Average accuracies of bagging methods and BS-SVM with the size of attribute subset changing based on German dataset.

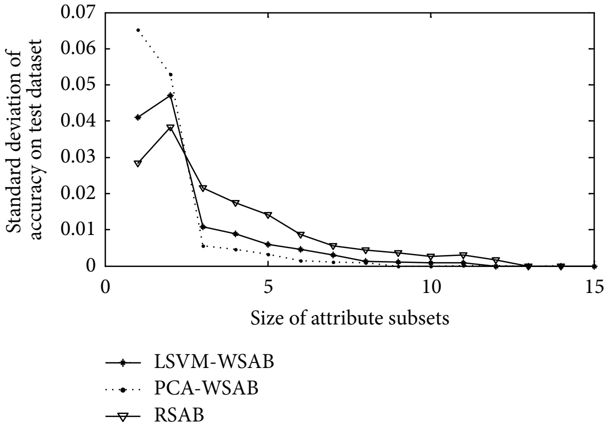

Figure 9: Standard deviations for three attribute bagging methods: LSVM-WSAB, PCA-WSAB, and RSAB based on Australian dataset.

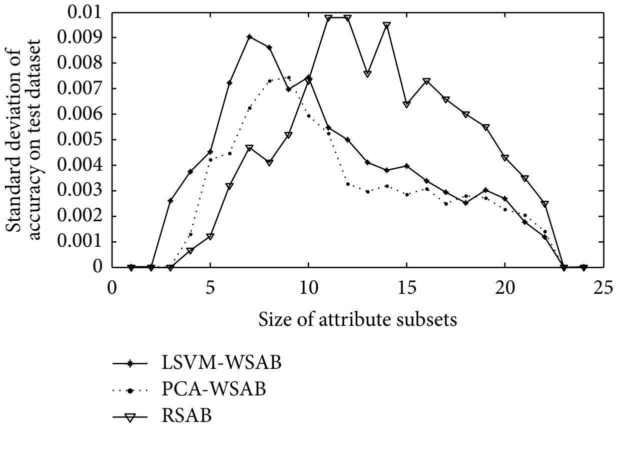

Figure 10: Standard deviations for three attribute bagging methods: LSVM-WSAB, PCA-WSAB, and RSAB based on German dataset.

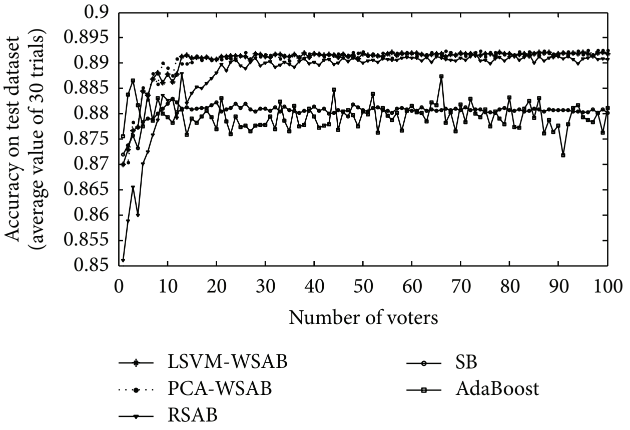

Figure 11: The average accuracies of attribute bagging, standard bagging (SB), and AdaBoost with the number of voters changing for Australian dataset.

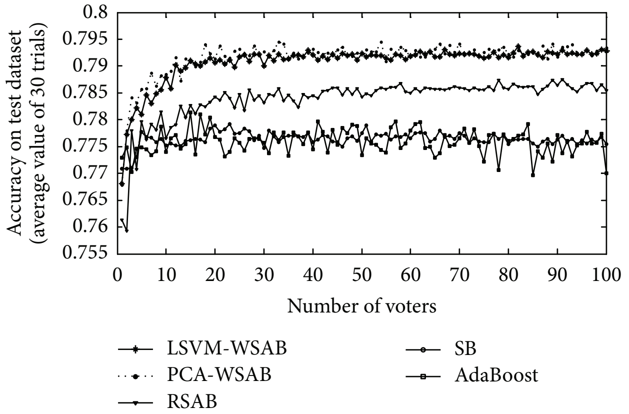

Figure 12: The average accuracies of attribute bagging, standard bagging (SB), and AdaBoost with the number of voters changing for German dataset.

become smaller than RSAB. This is because when the size of attribute subset is small, most of attribute subsets may not contain important attributes, so the accuracy of RSAB is always low. Meanwhile, for small attribute subsets, some of them for WSAB may contain important attributes, yet the other some may not contain any important attributes; so the performance of WSAB model is not stable. But for large attribute subsets, most of important attributes can be chosen into attribute subsets with large probability, resulting in the stable classification performance of WSAB. On the other hand, RSAB randomly chooses attributes into attribute subsets, each attribute with the same probability; so larger difference exists between different attribute subsets for RSAB than for WSAB. Therefore, the performance of RSAB becomes less stable than WSAB when the size of attribute subset becomes larger. Furthermore, when WSAB and RSAB adopt their optimal sizes of attribute subsets, respectively, the standard deviation of accuracy of WSAB is smaller than that of RSAB.

The highest accuracy of each model and the corresponding standard deviation (Std) are shown in Table 2 for Australian dataset and Table 3 for German dataset.

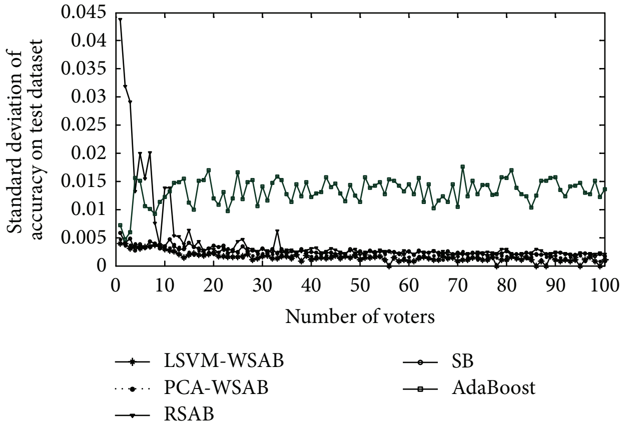

Figure 13: Standard deviations of accuracy of attribute bagging, standard bagging, and AdaBoost with the number of voters changing for Australian dataset.

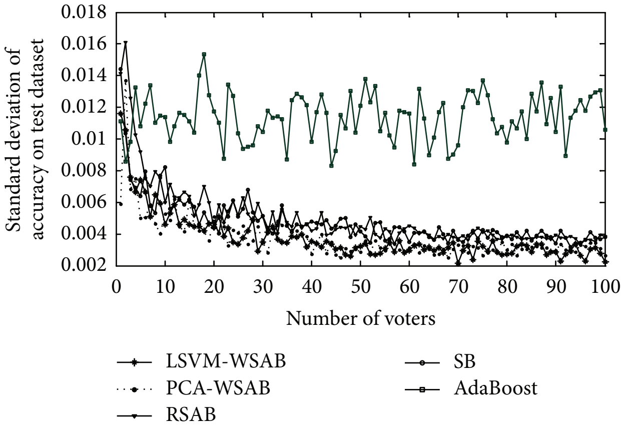

Figure 14: Standard deviations of accuracy of attribute bagging, standard bagging, and AdaBoost with the number of voters changing for German dataset.

From Tables 2 and 3, WSAB model performs better than RSAB both in accuracy and stability, and all attribute bagging methods have higher accuracies than BS-SVM.

4.5. Performance Comparison on Different Ensemble Methods with Different Numbers of Voters

4.5.1. Comparison on Accuracy. In this section, each attribute bagging model adopts its optimal attribute subset size, which is calculated by cross-validation. Then, we compare the accuracies of attribute bagging models including LSVM-WSAB, PCA-WSAB, and RSAB, as well as data partitioning ensemble models including standard bagging (SB) and AdaBoost, with the number of voters changing. For each number of voters and for each model, the experiments were also repeated 30 times, and the average accuracy was computed. The results are illustrated in Figure 11 for Australian dataset and Figure 12 for German dataset.

From Figures 11 and 12, WSAB has higher accuracy than SB and AdaBoost for almost each number of voters.

Table 2: The highest accuracy of each model and the corresponding standard deviation (Std) for Australian dataset.

	LSVM-WSAB (8 attributes)	PCA-WSAB (6 attributes)	RSAB (10 attributes)	BS-SVM (9 attributes)	SVM (14 attributes)
Accuracy	0.8912	0.8918	0.8880	0.8800	0.8750
Std	0.0013	0.0015	0.0030	0	0

Table 3: The highest accuracy of each model and the corresponding standard deviation (Std) for German dataset.

	LSVM-WSAB (16 attributes)	PCA-WSAB (12 attributes)	RSAB (20 attributes)	BS-SVM (14 attributes)	SVM (24 attributes)
Accuracy	0.7924	0.7930	0.7860	0.7824	0.7729
Std	0.0034	0.0033	0.0043	0	0

This is because WSAB can reserve important attributes and eliminate some redundant and uninformative attributes by large probability. When the number of voters is small, SB and AdaBoost have higher accuracies than RSAB. But when the number of voters is more than 10, RSAB has higher accuracy than standard bagging and AdaBoost. The reason is that RSAB needs more voters in order to include most of important attributes.

Moreover, the accuracy of standard bagging model increases gradually before the number of voters reaches 20, and its accuracy maintains at a certain level after the number of the voters is larger than 20. The accuracy of AdaBoost fluctuates most sharply with the number of voters changing. Meanwhile, the accuracies of attribute bagging models increase quickly before the number of voters reaches 20 and then their accuracies also maintain at certain levels. This is because standard bagging and AdaBoost sample from training dataset with all attributes for each voter, whereas attribute bagging only uses part of attributes. Therefore, attribute bagging needs to use more voters to "cover" all attributes, and with the number of voters increasing, more information is integrated into bagging model. The higher accuracies achieved by attribute bagging models support the conclusion that attribute bagging models are superior to data partitioning ensemble models.

For small number of voters, both LSVM-WSAB and PCA-WSAB perform better than RSAB. This further proves our idea that WSAB model can utilize important attributes to obtain better classification results. For large numbers of voters, WSAB model performs slightly better than RSAB for Australian dataset and much better than RSAB for German dataset. Therefore, the conclusion can be made that WSAB outperforms RSAB.

4.5.2. Comparison on Stability. Besides computing average accuracies of 30 trials for each number of voters, we also computed the standard deviation of accuracy to evaluate the classification stability of different methods. The standard deviations of classification accuracy for each model are shown in Figure 13 for Australian dataset and Figure 14 for German dataset.

From Figures 13 and 14, for small number of voters, WSAB has almost the same standard deviation as SB, and RSAB has higher standard deviation than SB. But for large number of voters, WSAB has lower standard deviation than SB, and SB has nearly the same standard deviation as SB. Additionally, when the number of voters is more than 10, the standard deviation of accuracy of AdaBoost is much larger than other methods. This supports the conclusion that boosting is easy to bias on several samples which are difficult to be classified. Therefore, boosting is not stable for the credit scoring problem; sometimes effective and sometimes not.

Furthermore, WSAB is more stable than RSAB. The reason is that WSAB can select important attributes for each child classifier, such that the accuracies of child classifiers in WSAB fluctuate less than those of RSAB. Therefore, from the viewpoint of the whole results, WSAB is more stable than RSAB.

When the number of voters is larger than 50, the accuracy and stability of each ensemble model maintain certain levels. Therefore, in order to compare the performance of all ensemble models, we show the accuracies of all ensemble models in Table 4 for Australian dataset and Table 5 for German dataset when the number of voters is 50.

From Tables 4 and 5, WSAB model performs best both on accuracy and stability, and RSAB model follows.

5. Conclusions and Future Research

This paper presents the WSAB for credit risk evaluation. The implementation of WSAB includes two steps. The first step is to determine weights of attributes. During the second step, attributes are selected into attribute subsets according to the probabilities determined by attribute weights. This method of modeling makes the WSAB have two advantages, namely, improving the accuracy of each individual classifier in ensemble and increasing the diversity among all individual classifiers. For the first merit, the more important attributes can be incorporated into each attribute subset with the larger probabilities so that each individual classifier can acquire high classification accuracy. For the second merit, the way of selecting attributes by probability makes different attribute subsets have different unimportant attributes which are of small weights, and consequently the diversity among different classifiers can be still guaranteed. In fact, accuracy and diversity are two critical factors for bagging. Experimental

Table 4: The accuracy of each model and the corresponding standard deviation for Australian dataset.

	LSVM-WSAB	PCA-WSAB	RSAB	SB	AdaBoost
Accuracy	0.8919	0.8922	0.8890	0.8807	0.8779
Std	0.0011	0.0013	0.0020	0.0026	0.0114

Table 5: The accuracy of each model and the corresponding standard deviation for German dataset.

	LSVM-WSAB	PCA-WSAB	RSAB	SB	AdaBoost
Accuracy	0.7918	0.7924	0.7842	0.7775	0.7731
Std	0.0032	0.0029	0.0045	0.0041	0.0121

results also confirm the superiority of WSAB over randomly selected attribute bagging (RSAB), especially over standard bagging, AdaBoost, and individual classifier.

Broadly speaking, the WSAB provides a framework of evaluating credit risk. In this framework, any attribute weighting method and any basis classifier can be combined. This paper adopts two completely different ways to compute weights of attributes: LSVM and PCA. The weights obtained by LSVM emphasize the classification ability of attributes, and the weights from PCA reflect the description ability of attributes for original data. However, credit scoring is just considered as a classification problem, for which LSVM seems to be more suitable than PCA, and experimental results also demonstrate the conclusion.

The next work will attempt to combine other approaches of computing weights of attributes and other basis classifiers to perform credit risk evaluation and then to compare their performances in terms of accuracy and stability. Additionally, the WSAB can also be applied to other practical systems, such as stock market prediction [45] and MRI brain image classification [46].

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions which have led to great improvement on this paper. This work is supported by the National Natural Science Foundation of China (no. 61271374) and the Beijing Natural Science Foundation (no. 4122068).

References

[1] H. L. Chen, B. Yang, G. Wang et al., “A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method,” *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1348–1359, 2011.

[2] B. Yang, L. X. Li, Q. Xie, and J. Xu, “Development of a KBS for managing bank loan risk,” *Knowledge-Based Systems*, vol. 14, no. 5-6, pp. 299–302, 2001.

[3] L. C. Thomas, D. B. Edelman, and J. N. Crook, *Credit Scoring and Its Applications*, SIAM Monographs on Mathematical Modeling and Computation, SIAM, Philadelphia, Pa, USA, 2002.

[4] J. N. Crook, D. B. Edelman, and L. C. Thomas, “Recent developments in consumer credit risk assessment,” *European Journal of Operational Research*, vol. 183, no. 3, pp. 1447–1465, 2007.

[5] J. H. Myers and E. W. Forgy, “The development of numerical credit evaluation systems,” *Journal of the American Statistical Association*, vol. 58, no. 303, pp. 799–806, 1963.

[6] J. C. Wiginton, “A note on the comparison of logit and discriminant models of consumer credit behavior,” *Journal of Financial Quantitative Analysis*, vol. 15, no. 3, pp. 757–770, 1980.

[7] P. Makowski, “Credit scoring branches out,” *Credit World*, vol. 74, no. 2, pp. 30–37, 1985.

[8] X. Y. Zhou, D. F. Zhang, and Y. Jiang, “A new credit scoring method based on rough sets and decision tree,” in *Advances in Knowledge Discovery and Data Mining*, vol. 5012 of *Lecture Notes in Computer Science*, pp. 1081–1089, Springer, New York, NY, USA, 2008.

[9] W. E. Henley and D. J. Hand, “Construction of a k-nearest-neighbor credit-scoring system,” *IMA Journal of Management Mathematics*, vol. 8, no. 4, pp. 305–321, 1997.

[10] H. L. Jensen, “Using neural networks for credit scoring,” *Managerial Finance*, vol. 18, no. 6, pp. 15–26, 1992.

[11] D. West, “Neural network credit scoring models,” *Computers and Operations Research*, vol. 27, no. 11-12, pp. 1131–1152, 2000.

[12] V. S. Desai, D. G. Conway, J. N. Crook, and G. A. J. R. Overstreet, “Credit-scoring models in the credit-union environment using neural networks and genetic algorithms,” *IMA Journal of Management Mathematics*, vol. 8, no. 4, pp. 323–346, 1997.

[13] M. B. Yobas, J. N. Crook, and P. Ross, “Credit scoring using neural and evolutionary techniques,” *IMA Journal of Mathematics Applied in Business and Industry*, vol. 11, no. 2, pp. 111–125, 2000.

[14] D. F. Zhang, H. Y. Huang, Q. S. Chen, and Y. Jiang, “A comparison study of credit scoring models,” in *Proceedings of the 3rd International Conference on Natural Computation (ICNC ’07)*, vol. 1, pp. 15–18, Haikou, China, August 2007.

[15] H. A. Abdou, “Genetic programming for credit scoring: the case of Egyptian public sector banks,” *Expert Systems with Applications*, vol. 36, no. 9, pp. 11402–11417, 2009.

[16] J. J. Huang, G. H. Tzeng, and C. S. Ong, “Two-stage genetic programming (2SGP) for the credit scoring model,” *Applied Mathematics and Computation*, vol. 174, no. 2, pp. 1039–1053, 2006.

[17] K. Leung, F. Cheong, and C. Cheong, “Consumer credit scoring using an artificial immune system algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’07)*, pp. 3377–3384, Singapore, September 2007.

[18] K. K. Lai, L. Yu, L. G. Zhou, and S. Y. Wang, “Credit risk evaluation with least square support vector machine,” in *Rough Sets and Knowledge Technology*, vol. 4062 of *Lecture Notes in*

Computer Science, pp. 490–495, Springer, New York, NY, USA, 2006.

[19] K. B. Schebesch and R. Sleeking, "Support vector machines for classifying and describing credit applicants: detecting typical and critical regions," *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1082–1088, 2005.

[20] D. Zhang, X. Y. Zhou, S. C. H. Leung, and J. Zheng, "Vertical bagging decision trees model for credit scoring," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7838–7843, 2010.

[21] S. L. Lin, "A new two-stage hybrid approach of credit risk in banking industry," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8333–8341, 2009.

[22] N. C. Hsieh, "Hybrid mining approach in the design of credit scoring models," *Expert Systems with Applications*, vol. 28, no. 4, pp. 655–665, 2005.

[23] D. Zhang, M. Hifi, Q. Chen, and W. Ye, "A hybrid credit scoring model based on genetic programming and support vector machines," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, vol. 7, pp. 8–12, Jinan, China, October 2008.

[24] L. Zhou, K. K. Lai, and L. Yu, "Least squares support vector machines ensemble models for credit scoring," *Expert Systems with Applications*, vol. 37, no. 1, pp. 127–133, 2010.

[25] T. G. Dietterich, "Machine-learning research: four current directions," *AI Magazine*, vol. 18, no. 4, pp. 97–136, 1997.

[26] R. O. Duda, P. H. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2000.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] L. Breiman, "Bias, variance, and arcing classifiers," Tech. Rep. 460, University of California, Department of Statistics, Berkeley, Calif, USA, 1996.

[29] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th International Conference on Machine Learning*, pp. 148–156, Bari, Italy, 1996.

[30] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.

[31] K. Tumer and N. C. Oza, "Decimated input ensembles for improved generalization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '99)*, pp. 3069–3074, Washington, DC, USA, July 1999.

[32] K. Tumer and J. Ghosh, "Classifier combining: analytical results and implications, Working notes from the workshop 'integrating multiple learned models'," in *Proceedings of the 13th National Conference on Artificial Intelligence*, Portland, Ore, USA, August 19961996.

[33] D. Opitz, "Feature selection for ensembles," in *Proceedings of the 16th AAAI National Conference on Artificial Intelligence*, pp. 379–384, Orlando, Fla, USA, 1999.

[34] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[35] L. Nanni and A. Lumini, "An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3028–3033, 2009.

[36] G. Wang, J. Ma, L. Huang, and K. Xu, "Two credit scoring models based on dual strategy ensemble trees," *Knowledge-Based Systems*, vol. 26, pp. 61–68, 2012.

[37] P. Ravisankar and V. Ravi, "Financial distress prediction in banks using group method of data handling neural network, counter propagation neural network and fuzzy ARTMAP," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 823–831, 2010.

[38] C. F. Tsai, "Feature selection in bankruptcy prediction," *Knowledge-Based Systems*, vol. 22, no. 2, pp. 120–127, 2009.

[39] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, School of Information and Computer Science, University of California, Irvine, Calif, USA, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[40] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[41] C. Guerra-Salcedo and D. Whitley, "Genetic approach to feature selection for ensemble creation," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*, pp. 236–243, Morgan Kaufmann, 1999.

[42] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, Germany, 1995.

[43] B. Janez, G. Marko, M. Natasa, and M. Dunja, "Feature selection using linear support vector machines," Tech. Rep. MSR-TR-2002-63, Microsoft Research Microsoft Corporation, 2002.

[44] R. P. W. Duin, P. Juszczak, P. Paclik et al., PRTools4. 1, A Matlab Toolbox for Pattern Recognition, 2007, http://www.prtools.org/.

[45] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2009.

[46] Y. Zhang, Z. Dong, L. Wu, and S. Wang, "A hybrid method for MRI brain image classification," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10049–10053, 2011.