Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Credit scoring by feature-weighted support vector machines*

Jian SHI[†1,2], Shu-you ZHANG[1], Le-miao QIU[†1]

([1]The State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China)

([2]School of Electrical and Automatic Engineering, Changshu Institute of Technology, Changshu 215500, China)

[†]E-mail: {yievans, qiulm}@zju.edu.cn

Received June 28, 2012; Revision accepted Jan. 22, 2013; Crosschecked Feb. 25, 2013

Abstract: Recent finance and debt crises have made credit risk management one of the most important issues in financial research. Reliable credit scoring models are crucial for financial agencies to evaluate credit applications and have been widely studied in the field of machine learning and statistics. In this paper, a novel feature-weighted support vector machine (SVM) credit scoring model is presented for credit risk assessment, in which an *F*-score is adopted for feature importance ranking. Considering the mutual interaction among modeling features, random forest is further introduced for relative feature importance measurement. These two feature-weighted versions of SVM are tested against the traditional SVM on two real-world datasets and the research results reveal the validity of the proposed method.

Key words: Credit scoring model, Support vector machine (SVM), Feature weight, Random forest
doi:10.1631/jzus.C1200205 **Document code:** A **CLC number:** TP391.4

1 Introduction

Due to the recent USA subprime mortgage crisis and European debt crisis, credit risk management has become one of the most important issues in the financial area. The whole world has been suffering from economic recession recently, and granting of credit has become a common strategy for acquiring and retaining clients for economic entities (especially for commercial banks), thereby to gain competitive advantages in a severe economic environment. However, increasing defaults on debts result in great losses for financial institutions, and they cannot simply refuse all applications just to avoid credit risks. Effective credit scoring models are therefore considered essential for credit agencies to make sound credit granting decisions.

Credit scoring models have been developed to discriminate potential borrowers as creditworthy or default applicants based on their financial status and credit performance recorded in an application form or by a credit reference agency. Since even a fraction of improvement in accuracy of credit scoring might translate into noteworthy future savings (Huang *et al.*, 2007), a variety of different data mining and statistical techniques have been proposed to derive reliable credit scoring models over recent decades. Generally, these methods can be divided into statistical approaches and machine learning approaches. Statistical approaches, such as logistic regression (LR) and linear discriminant analysis (LDA), are relatively simple and explainable, and assume a threshold on the underlying probability of default to reject clients with a posterior probability below that threshold (Bellotti and Crook, 2009). However, their discrimination ability is still contentious due to the nonlinear relationship between default probability and credit patterns. Some emerging machine learning approaches, such as artificial neural networks (ANNs) and support vector machines (SVMs), are particularly well suited for coping with nonlinear problems and commonly

* Project supported by the National Basic Research Program (973) of China (No. 2011CB706506), the National Natural Science Foundation of China (No. 50905159), the Natural Science Foundation of Jiangsu Province (No. BK2010261), and the Fundamental Research Funds for the Central Universities (No. 2011XZZX005), China

operate in a data-driven way without the limitation of prior probability assumptions (Pang *et al.*, 2011; Wang *et al.*, 2011).

SVMs, motivated by statistical learning theory (Vapnik, 1995; 1998), have been applied to some credit scoring problems because of their solid theoretical foundation and appealing classification performance. Baesens *et al.* (2003) tested the performances of various classification algorithms on eight real-life credit scoring datasets and SVM outperformed most of the other techniques. Thomas *et al.* (2005) compared 17 consumer credit modeling methods and reported that SVM was the best, but the classification result was not good enough for practical applications. Huang *et al.* (2007) constructed hybrid SVM-based credit scoring models with three strategies for feature selection and model parameter setting. However, the elimination of features may cause information loss and affect the classification accuracy. Martens *et al.* (2007) extracted rules from a trained SVM to obtain both accurate and comprehensible credit scoring models. Nevertheless, the extracted rules lose a small percentage of accuracy compared with the black box model from which they are generated.

Though SVMs are efficient in dealing with a large number of explanatory attributes, called 'features', too many features may raise the risk of model over-fitting. Selection of appropriate features to represent the data has received significant attention in the data mining literature (Pal *et al.*, 2000; Guyon *et al.*, 2002). However, since for the credit scoring problem, there are relatively few features to begin with and the input features are mutually independent, it is inappropriate to simply remove or eliminate any features. Moreover, when building credit scoring models using traditional SVMs, all input features are treated equally and assigned the same modeling contribution despite their different impacts on the output grant decision. Recently, a new feature-weighting method has emerged in the field of machine learning (Blum and Langley, 1997; Yeung and Wang, 2002; Wang *et al.*, 2004). Instead of explicitly selecting subsets of features, feature-weighting assigns the input features with different degrees of perceived importance. In this study, a feature-weighted SVM for credit scoring modeling is proposed, to improve the performance of a nonlinear support vector classifier by scaling the input features, thus optimizing the separating hyperplane in the projected feature space. The effect of two weighting schemes, the *F*-score and random forest feature importance ranking strategies, are compared with traditional SVM using two real-world credit datasets.

2 Support vector machines (SVMs)

2.1 Traditional SVM

SVM tries to search for an optimal hyperplane to separate binary classified data, such that the margin between the hyperplane and the two classes is maximized. Given a training set of l sample points $(\boldsymbol{x}_1, y_1)$, $(\boldsymbol{x}_2, y_2)$, ..., $(\boldsymbol{x}_l, y_l)$, where $\boldsymbol{x}_i$ is a real vector with m-dimensional features and $y_i\in\{-1, 1\}$ as its corresponding class label, SVM seeks the classification mapping

$$Y(\boldsymbol{x})=\mathrm{sgn}(\boldsymbol{wx}+b) \tag{1}$$

by solving the following quadratic program with inequality constraints:

$$\begin{aligned} &\min \|\boldsymbol{w}\|^2 \\ &\text{s.t.}\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geq 1,\ i=1,2,\ldots,l, \end{aligned} \tag{2}$$

where the parameter pair $(\boldsymbol{w}, b)$ consists of the weight vector and the bias term.

For cases where data points are not rigidly separable, typical of practical applications, the constraints in Eq. (2) cannot be satisfied by all training examples. Thus, the previous form of Eq. (2) is generalized by introducing the non-negative variable ξ_i, such that some slack is allowed for the wrong classified input samples and the quadratic programming problem is modified as

$$\begin{aligned} &\min \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i \\ &\text{s.t.}\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geq 1-\xi_i,\ \xi_i\geq 0,\ i=1,2,\ldots,l, \end{aligned} \tag{3}$$

where the regularized constant C controls the trade-off between training error and classification margin.

However, in many real-world problems, the input points cannot be linearly separated in the original input space but can be restrictively classified in a higher dimensional feature space. Thus, nonlinear mapping φ: $\boldsymbol{x}\to\varphi(\boldsymbol{x})$ is introduced to derive a nonlinear version of Eq. (3). By incorporating the nonlinear constraints into a Lagrangian and setting its partial derivatives to zero, the QP problem turns into the following dual form:

$$\begin{aligned}&\min \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{i=1}^{l}\alpha_i\\ &\text{s.t.}\ \sum_{i=1}^{l}y_i\alpha_i=0,\quad 0\le\alpha_i\le C,\ i=1,2,...,l,\end{aligned} \tag{4}$$

where α is the Lagrange multiplier and $K(\boldsymbol{x}_i, \boldsymbol{x}_j)=\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x}_i)$ is the kernel function which satisfies Mercer's condition. A radial basis function (RBF) kernel is employed in this study because of its superiority, as experimentally demonstrated by van Gestel *et al.* (2004).

2.2 Feature-weighted SVM

In traditional support vector classifiers, a kernel function is introduced to compute the dot product of the data points in the feature space. This involves the Euclidean distance calculation of input samples with all selected modeling features treated equally. However, some features have tremendous impact on the final decision function while others may have little or no relevance to the separating surface. When the kernel function is dominated by irrelevant features, the classification accuracy and generalization ability of the SVM may be greatly contaminated. However, it is unreasonable to simply remove any features, especially when there are relatively few features in the modeling datasets. Thus, a feature-weighting scheme is necessary for such situations. The corresponding generalized Euclidean distance $d_{ij}^{(w)}$ is defined as follows (Wang *et al.*, 2004):

$$d_{ij}^{(\boldsymbol{w})}=\sqrt{\sum_{k=1}^{n}\boldsymbol{w}_k^2(\boldsymbol{x}_{ik}-\boldsymbol{x}_{jk})^2}, \tag{5}$$

where $d_{ij}^{(\boldsymbol{w})}$ is the weighted Euclidean distance between two sample points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with p-dimensional features, and $\boldsymbol{w}$ is the feature-weight vector with w_k, the measure of importance corresponding to the kth feature. Note that a larger w_k makes the kth feature more important in SVM. $\boldsymbol{w}$=(1, 1, …, 1) means that all features are treated equally and Eq. (5) turns into the usual Euclidean distance, while $\boldsymbol{w}\neq$(1, 1, …, 1) indicates that the axes in the transformed feature space would be extended with $\boldsymbol{w}_k<1$, or shrunk with $\boldsymbol{w}_k>1$.

In the framework of feature-weight learning, the nonlinear mapping φ: $\boldsymbol{x}\to\varphi(\boldsymbol{x})$ turns into $\varphi_{\boldsymbol{w}}$: $\boldsymbol{x}\to\varphi_{\boldsymbol{w}}(\boldsymbol{x})=\varphi(\boldsymbol{Px})$ and the kernel function is modified as $K_{\boldsymbol{w}}(\boldsymbol{x}_i, x_j)=K(\boldsymbol{Px}_i, \boldsymbol{Px}_j)=\varphi_{\boldsymbol{w}}(\boldsymbol{x}_i)\cdot\varphi_{\boldsymbol{w}}(\boldsymbol{x}_j)$, where $\boldsymbol{P}$=diag($\boldsymbol{w}_k$) is a diagonal matrix. Then, the dual problem (Eq. (4)) becomes

$$\begin{aligned}&\min \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K_{\boldsymbol{w}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{i=1}^{l}\alpha_i\\ &\text{s.t.}\ \sum_{i=1}^{l}y_i\alpha_i=0,\quad 0\le\alpha_i\le C,\ i=1,2,...,l.\end{aligned} \tag{6}$$

Note that the feature-weighted SVM is a little different from a traditional SVM in the form of kernel function. In a traditional SVM, linear non-separable input samples can be restrictively separated when mapped to a higher dimensional feature space. We expect that classification performance could be further improved in the transformed feature space by a feature-weight mapping scheme.

3 Feature-weighting strategies

Feature weight learning could be considered as a generalization of feature selection. Thus, those feature selection strategies concerned with feature importance estimation can be extended to feature weight assignment.

F-score is a simple technique which measures the discrimination between two sets of real numbers (Chen and Lin, 2006). Given a training set with input vectors $\boldsymbol{x}_k$, k=1, 2, …, m, where $\boldsymbol{x}_k\in\mathbb{R}^n$, the F-score of the ith feature is defined as follows, when the input vectors are first standardized:

$$F(i)=\frac{(\bar{\boldsymbol{x}}_i^{(+)}-\bar{\boldsymbol{x}}_i)^2+(\bar{\boldsymbol{x}}_i^{(-)}-\bar{\boldsymbol{x}}_i)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}(\boldsymbol{x}_{k,i}^{(+)}-\bar{\boldsymbol{x}}_i^{(+)})^2+\frac{1}{n_--1}\sum_{k=1}^{n_-}(\boldsymbol{x}_{k,i}^{(-)}-\bar{\boldsymbol{x}}_i^{(-)})^2}, \tag{7}$$

where n_+ is the number of positive instances in input vectors and n_- is the negative, $\bar{\boldsymbol{x}}_i$, $\bar{\boldsymbol{x}}_i^{(+)}$, and $\bar{\boldsymbol{x}}_i^{(-)}$ are the means of all, the positive, and the negative input vectors respectively on the ith feature, and $\boldsymbol{x}_{k,i}^{(+)}$ and $\boldsymbol{x}_{k,i}^{(-)}$ are the ith feature values of the kth positive and negative input points, respectively. Note that a larger F-score makes the corresponding feature more discriminative in modeling (Chen and Lin, 2006).

The F-score is a relatively simple but rather effective feature importance measurement strategy. However, it does not reveal the feature relevance among the whole set of features. As the importance of a feature is often affected by its interaction with other features, a random forest algorithm is sometimes used to measure the importance of a feature by calculating the percentage increase in the misclassification rate when compared to the out-of-bag (OOB) rate on un-permuted data (Breiman, 2001; Prinzie and Poel, 2008).

A random forest is a bagged classifier consisting of an ensemble of classification trees (CTs). Each tree is constructed using a different bootstrap dataset with randomly chosen instances, and each node is split using the best among randomly selected features. These two kinds of randomness make random forest more robust against model over-fitting and data noise. The random forest is constructed as follows (Archer and Kimes, 2008):

Obtain bootstrap dataset S_b from the original dataset S to construct CTs in the following way:

1. Randomly select n features out of all modeling features at node t.

2. For the kth feature, k=1, 2, …, n, search for the best split s_k.

3. Split data at node t using the best split s among n best splits from step 2.

4. Repeat steps 1–3 at every node to grow the tree to the maximum extent possible.

Input samples are then categorized to the class with the maximum number of votes over all trees in the forest.

Usually, there are four different feature importance measurements defined in the present implementation of random forest classification. In this study, the Gini index was employed which accumulates the Gini decrease at every split in the forest due to the given feature.

4 Results and discussion

To further evaluate the effectiveness of the SVM credit scoring models with feature weight assignment, experiments were conducted on two real-world credit datasets (German and Australian credit datasets) from the UCI Repository of Machine Learning Databases. The German credit dataset consists of 1000 loan applications, with 700 accepted and 300 rejected. Each applicant is described by 19 originally selected attributes, four of which are transformed into nine dummy variables, and 24 numeric input features are finally obtained for classifier modeling. Details of the input features of the German credit dataset are shown in Table 1. The Australian credit dataset is composed of 690 applicants, with 383 creditworthy and 307 default examples. Each instance contains eight numerical features, six categorical features, and one discriminant feature, with sensitive information being transferred to symbolic data for confidentiality reasons.

Two adjustable parameters should be determined before the learning of random forest. The number of CTs in random forests was set to 500 (default value) when run on the two credit datasets. The number of random features n at each node, to which random forest is more sensitive, should be chosen more carefully. Usually, the root mean square of the whole features is suggested when it is applied to classification problems. As there are relatively few features in this study, a grid search strategy was adopted to tune the parameter. Fig. 1 shows the OOB errors of random forest construction when built with different numbers of features n. The plots indicate that a random forest with 500 CTs and n=15 achieved the best performance for the German dataset, and n=3 was the optimal value for the Australian dataset.

Datasets were randomly split into training and test sets for five-fold cross validation. For each tree in the random forest, 4/5 of the data was bootstrapped and then used to fit a tree, and the OOB observations from this 4/5 dataset were used for estimating variable importance. Table 2 shows the F-score and Gini index of each feature based on different training sets of the Australian data. Feature 8 had the highest importance estimate based on its mean Gini index of 76.518, while feature 12 had the highest importance ranking based on its mean F-score of 9.753.

Table 1 Input variables for the German dataset

Original attribute	Input variable	Variable type	Attribute description
A1	V1	Qualitative	Status of the existing checking account
A2	V2	Numerical	Duration in month
A3	V3	Qualitative	Credit history
A4	V4, V5	Dummy	Purpose (V4: new car; V5: used car)
A5	V6	Numerical	Credit amount
A6	V7	Qualitative	Savings account/bonds
A7	V8	Qualitative	Present employment since
A8	V9	Qualitative	Personal status and sex
A9	V10, V11	Dummy	Other debtors/guarantors (V10: none; V11: co-applicant)
A10	V12	Numerical	Present residence since
A11	V13	Qualitative	Property
A12	V14	Numerical	Age in years
A13	V15	Qualitative	Other installment plans
A14	V16, V17	Dummy	Housing (V16: rent; V17: own)
A15	V18	Numerical	Number of existing credits at this bank
A16	V19, V20, V21	Dummy	Job (V19: unemployed/unskilled (non-resident); V20: unskilled (resident); V21: skilled employee/official)
A17	V22	Numerical	Number of people being liable to provide maintenance for
A18	V23	Qualitative	Telephone
A19	V24	Qualitative	Foreign worker

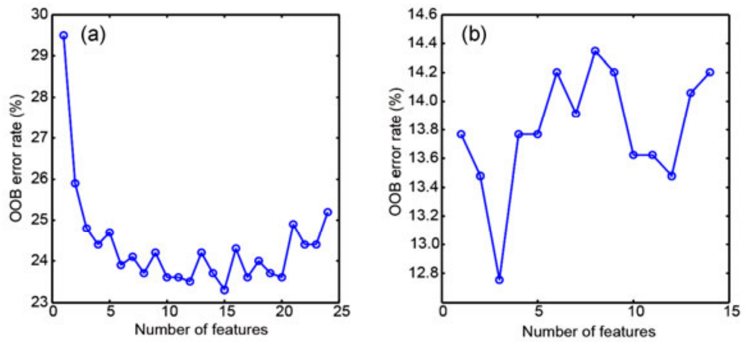

Fig. 1 Out-of-bag (OOB) error rates of random forest with respect to the number of features at each node
(a) German (credit) dataset;
(b) Australian (credit) dataset

Table 2 Feature importance measure of the Australian credit dataset by *F*-score and the Gini index

No.	Gini index					*F*-score				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	2.975	3.141	2.523	2.635	2.772	2.533	2.672	2.559	2.642	2.496
2	18.534	18.338	19.359	18.935	18.466	1.390	1.354	1.304	1.339	1.294
3	22.461	20.044	21.085	21.638	21.758	1.618	1.595	1.563	1.612	1.567
4	4.720	5.505	4.666	4.835	4.569	3.513	3.692	3.265	3.602	3.368
5	17.974	22.995	20.544	19.412	18.756	1.405	1.447	1.448	1.424	1.425
6	7.725	6.876	8.468	7.060	6.763	1.845	1.853	1.875	1.815	1.845
7	28.050	26.255	26.085	24.844	25.535	1.501	1.394	1.427	1.479	1.368
8	74.477	76.246	77.091	72.326	82.449	2.003	2.018	2.007	2.004	1.999
9	12.479	14.735	14.435	18.141	15.512	2.091	2.032	2.069	2.132	2.105
10	26.729	27.884	27.617	27.837	24.741	1.643	1.548	1.977	1.713	1.626
11	3.470	3.135	3.025	3.124	3.002	2.051	2.032	2.018	2.020	2.012
12	2.564	3.947	3.773	4.039	3.166	9.721	9.750	9.342	9.550	10.402
13	17.624	17.453	17.773	18.588	19.300	1.041	1.135	1.071	1.279	1.004
14	25.579	22.533	21.374	22.992	19.449	1.206	1.049	0.776	1.107	0.991

When the feature importance is sorted in ascending order, the corresponding feature number is derived (Table 3). The importance estimation of the F-score was more stable than that of random forest.

Similar results were found using the German credit dataset. The average values of relevant feature importance on a five-fold German dataset were calculated and the difference between the two methods is depicted in Fig. 2. The importance estimations of the F-scores among features appear to be less variable than those of random forest, especially for the first 10 features. The agreements between the random forest Gini variable importance and the F-score were further compared by plotting the ranks against one another for each fold, and estimating Spearman's correlation (Fig. 3). The feature importance rankings of the two methods are different, which can be explained by the fact that random forest concerns the interactions among features while the F-score makes individual calculations. Feature weights are then assigned according to the maximum values of the normalized Gini index and the F-score.

A detailed comparison among random forest feature-weighted SVM (RF-FWSVM), F-score feature-weighted SVM (FS-FWSVM), and traditional SVM is illustrated in Table 4. The programs were run under the MATLAB environment. The performances of the three methods were measured by their average classification accuracy (percentage correctly classified (PCC)) and the standard deviation of their classification accuracy (STD) of five-fold cross validation, defined as follows:

$$\begin{cases} A_i = \dfrac{C_i}{T_i}, \quad \mathrm{PCC} = \dfrac{1}{f}\sum_{i=1}^{f} A_i, \\ \mathrm{STD} = \sqrt{\dfrac{1}{f-1}\sum_{i=1}^{f}(A_i - \mathrm{PCC})}, \end{cases} \tag{8}$$

where f is the number of folds, C_i is the number of correctly classified instances, and T_i is the total number of instances in fold i.

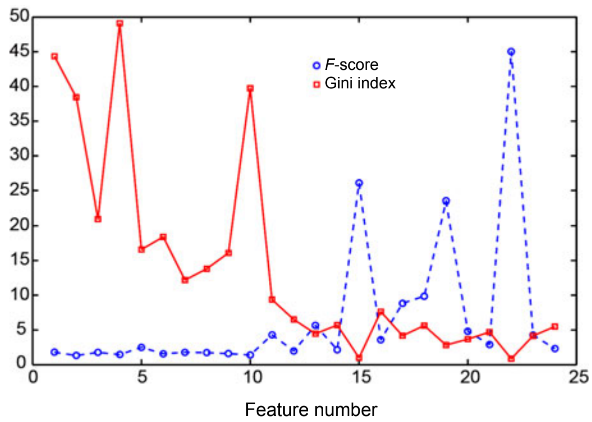

Fig. 2 Feature importance plot for the German credit dataset

Table 3 Feature importance sorting of the Australian credit dataset

Rank	Feature number of Gini					Feature number of F-score				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	12	11	1	1	1	13	14	14	14	14
2	1	1	11	11	11	14	13	13	13	13
3	11	12	12	12	12	2	2	2	2	2
4	4	4	4	4	4	5	7	7	5	7
5	6	6	6	6	6	7	5	5	7	5
6	9	9	9	9	9	3	10	3	3	3
7	13	13	13	13	2	10	3	6	10	10
8	5	2	2	2	5	6	6	10	6	6
9	2	3	5	5	13	8	8	8	8	8
10	3	14	3	3	14	11	9	11	11	11
11	14	5	14	14	3	9	11	9	9	9
12	10	7	7	7	10	1	1	1	1	1
13	7	10	10	10	7	4	4	4	4	4
14	8	8	8	8	8	12	12	12	12	12

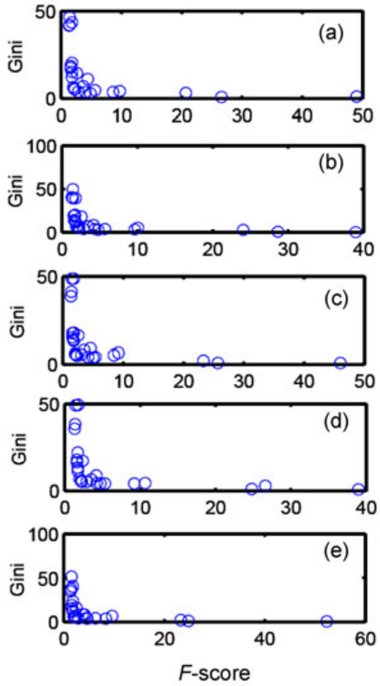

Fig. 3 Gini index plots against the *F*-score for the German credit dataset of each fold

(a) Fold 1: Spearman's correlation=−0.90435; (b) Fold 2: Spearman's correlation=−0.89304; (c) Fold 3: Spearman's correlation=−0.83826; (d) Fold 4: Spearman's correlation =−0.89739; (e) Fold 5: Spearman's correlation=−0.86435

Table 4 Performance for the German and Australian test datasets

Method	PCC (%)		STD (%)	
	German	Australian	German	Australian
RF-FWSVM	77.3	87.7	1.35	2.40
FS-FWSVM	76.5	86.7	1.66	2.83
SVM	74.6	85.7	2.53	4.02

SVM: support vector machine; RF-FWSVM: random forest feature-weighted SVM; FS-FWSVM: *F*-score feature-weighted SVM; PCC: percentage correctly classified; STD: standard deviation of the classification accuracy

Results in Table 4 indicate that RF-FWSVM achieved the best performance on both PCC and STD measures among the three approaches, which confirms its classification accuracy and stability. RF-FWSVM obtained the largest PCC of 77.3% on the German test dataset, compared with 76.5% from FS-FWSVM and 74.6% from SVM. RF-FWSVM also yielded the lowest value of STD. Similar results from the three credit scoring models were observed using the Australian dataset, except that the overall performance on the Australian dataset was better as the German dataset was generally considered to be more unbalanced and complicated.

5 Conclusions

Credit scoring on two real-word credit datasets using feature-weighted SVM was reported in this article. Compared with the traditional SVM, the feature-weighted SVM assigns input features with different degrees of importance to improve the separating hyperplane in the transformed high dimensional feature space. Two feature-weighting strategies, *F*-score and random forest, were adopted for relative feature importance measurements. The former calculates feature importance irrespective of their mutual relevance, while the latter takes this factor into consideration. Experimental results indicate that RF-FWSVM provides both high classification accuracy and good stability for practical credit scoring applications.

References

Archer, K.J., Kimes, R.V., 2008. Empirical characterization of random forest variable importance measures. *Comput. Stat. Data Anal.*, **52**(4):2249-2260. [doi:10.1016/j.csda.2007.08.015]

Baesens, B., van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J., 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. *J. Oper. Res. Soc.*, **54**(6):627-635. [doi:10.1057/palgrave.jors.2601545]

Bellotti, T., Crook, J., 2009. Support vector machines for credit scoring and discovery of significant features. *Expert Syst. Appl.*, **36**(2):3302-3308. [doi:10.1016/j.eswa.2008.01.005]

Blum, A.L., Langley, P., 1997. Selection of relevant features and examples in machine learning. *Artif. Intell.*, **97**(1-2):245-271. [doi:10.1016/S0004-3702(97)00063-5]

Breiman, L., 2001. Random forests. *Mach. Learn.*, **45**(1):5-32. [doi:10.1023/A:1010933404324]

Chen, Y.W., Lin, C.J., 2006. Combining SVMs with Various Feature Selection Strategies. Feature Extraction Studies in Fuzziness and Soft Computing, **207**:315-324. [doi:10.1007/978-3-540-35488-8_13]

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**(1-3):389-422. [doi:10.1023/A:1012487302797]

Huang, C.L., Chen, M.C., Wang, C.J., 2007. Credit scoring with a data mining approach based on support vector machines. *Expert Syst. Appl.*, **33**(4):847-856. [doi:10.

1016/j.eswa.2006.07.007]
Martens, D., Baesens, B., van Gestel, T., Vanthienen, J., 2007. Comprehensible credit scoring models using rule extraction from support vector machines. *Eur. J. Oper. Res.*, **183**(3):1466-1476. [doi:10.1016/j.ejor.2006.04.051]
Pal, S.K., De, R.K., Basak, J., 2000. Unsupervised feature evaluation: a neuro-fuzzy approach. *IEEE Trans. Neur. Network*, **11**(2):366-376. [doi:10.1109/72.839007]
Pang, H.X., Dong, W.D., Xu, Z.H., Feng, H.J., Li, Q., Chen, Y.T., 2011. Novel linear search for support vector machine parameter selection. *J. Zhejiang Univ.-Sci. C* (*Comput. & Electron.*), **12**(11):885-896. [doi:10.1631/jzus.C1100006]
Prinzie, A., Poel, D.V.D., 2008. Random forests for multiclass classification: random multinomial logit. *Expert Syst. Appl.*, **34**(3):1721-1732. [doi:10.1016/j.eswa.2007.01.029]
Thomas, L.C., Oliver, R.W., Hand, D.J., 2005. A survey of the issues in consumer credit modelling research. *J. Oper. Res. Soc.*, **56**(9):1006-1015. [doi:10.1057/palgrave.jors.2602018]
van Gestel, T., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., Moor, B.D., Vandewalle, J., 2004. Benchmarking least squares support vector machine classifiers. *Mach. Learn.*, **54**(1):5-32. [doi:10.1023/B:MACH.0000008082.80494.e0]
Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer Verlag, New York.
Vapnik, V., 1998. Statistical Learning Theory. John Wiley & Sons, New York.
Wang, D.L., Zheng, J.G., Zhou, Y., 2011. Binary tree of posterior probability support vector machines. *J. Zhejiang Univ.-Sci. C* (*Comput. & Electron.*), **12**(2):83-87. [doi:10.1631/jzus.C1000022]
Wang, X.Z., Wang, Y.D., Wang, L.J., 2004. Improving fuzzy *c*-means clustering based on feature-weight learning. *Pattern Recogn. Lett.*, **25**(10):1123-1132. [doi:10.1016/j.patrec.2004.03.008]
Yeung, D.S., Wang, X.Z., 2002. Improving performance of similarity-based clustering by feature weight learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(4):556-561. [doi:10.1109/34.993562]