Research Article

Credit Risk Evaluation with a Least Squares Fuzzy Support Vector Machines Classifier

Lean Yu^{1,2}

¹ Alibaba Business College, Hangzhou Normal University, Hangzhou 310036, China

² School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China

Correspondence should be addressed to Lean Yu; yulean@amss.ac.cn

Received 12 February 2014; Accepted 27 May 2014; Published 12 June 2014

Academic Editor: Fenghua Wen

Copyright © 2014 Lean Yu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A least squares fuzzy support vector machine (LS-FSVM) model that integrates advantages of fuzzy support vector machine (FSVM) and least squares method is proposed for credit risk evaluation. In the proposed LS-FSVM model, the purpose of incorporating the concepts of fuzzy sets is to add generalization capability and outlier insensitivity, while the least squares method is adopted to reduce the computational complexity. For illustrative purposes, a real-world credit risk dataset is used to test the effectiveness and robustness of the proposed LS-FSVM methodology.

1. Introduction

Credit risk evaluation has been a major area of focus for financial and banking industries due to recent financial crises as well as the Basel III regulations. Since the seminal work of Altman [1] was published, many different techniques such as discriminant analysis [1], logit analysis [2], probit analysis [3], linear programming [4], integer programming [5], k -nearest neighbor (KNN) classifiers [6], and classification trees [7] have widely been applied to credit risk assessment tasks. With the advance of modern computing technology, some artificial intelligence (AI) tools such as artificial neural networks (ANN) [8, 9], genetic algorithms (GAs) [10, 11], self-organizing learning [12], support vector machines (SVMs) [13–18], and some variants of SVMs [19–22] have also been employed for credit risk evaluation. Empirical results have revealed that these AI techniques offer advantages over traditional statistical models and optimization techniques in credit risk evaluation.

According to the literature review, it is easy to find that almost all classification methods can be used for credit risk assessment. However, some hybrid and combined (or ensemble) classifiers, which integrate two or more single classification methods, have shown higher predictability than individual methods. Research on combined or ensemble classifiers is currently flourishing in credit risk evaluation.

Recent examples are neural discriminant technique [23], neurofuzzy [16, 24], fuzzy SVM [25], rough set based SVM [26], evolving neural network [27], neural network ensemble [28, 29], support vector machine based multiagent ensemble learning [30], and AI-based fuzzy group decision making (GDM) model [31]. Two recent surveys [32, 33] and one monograph [34] cover credit risk analysis in more detail.

In this study, a new credit classification technique, least squares fuzzy SVM (LS-FSVM), is proposed to discriminate good from bad customers in customer credit evaluation. In the existing studies, the fuzzy SVM (FSVM) proposed by Lin and Wang [35] has been shown to be suitable for customer credit assessment [34]. The main reason is that in credit risk evaluation we usually cannot label a customer as absolutely good one who is sure to repay in time or absolutely bad one who will default certainly, while the FSVM treats every sample as belonging to good and bad classes to some extent. This enables the FSVM to offer a higher generalization capability without losing the merits of insensitivity to outliers. Although the FSVM has good generalization capabilities and outlier insensitivity, computational complexity of the FSVM makes its use rather difficult because the final solution of FSVM is derived from a quadratic programming (QP) problem [35]. To reduce the computational complexity of FSVM, this study applies the least squares method to reduce the computational complexity of FSVM and to formulate a new classification

method—least squares FSVM (LS-FSVM). In the proposed LS-FSVM model, equality constraints instead of inequality constraints are used. As a result, the solution is obtained from a set of linear equations, instead of QP presenting in the classical FSVM approach [35], thereby reducing the computational complexity, relative to the FSVM.

From the above descriptions, the main advantage of the proposed LS-FSVM can be summarized with regard to the following two aspects. On the one hand, fuzzification processing can increase the generalization capability and improve the suitability of SVM due to the fact that the same uncertainties can be well treated by fuzzy membership. On the other hand, the least squares method can reduce the computational complexity of FSVM because the solution of least squares FSVM can be obtained from a set of linear equations instead of a QP problem, thus increasing the computational speed of FSVM which is attractive in solving fuzzy information engineering problems. In the existing literature, Tsujinishi and Abe [36] proposed a fuzzy LS-SVMs method to solve the multiclass problems. Regrettably, the performance of the proposed fuzzy LS-SVMs model is inferior to the fuzzy SVMs. On the contrary, the proposed LS-FSVM model in this paper reported the good performance in the two-class problems due to the above features of LS-FSVM model.

The main motivation of this study is to formulate the least squares version of FSVM for binary classification problems and to compare its performance with some typical classification techniques in the area of credit risk evaluation. The rest of this study is organized as follows. Section 2 illustrates the formulation of the LS-FSVM methodology. In Section 3, a real-world credit dataset is used to test the performance of the LS-FSVM to classify different samples. Section 4 concludes the paper.

2. Methodology Formulation

In this section, a brief introduction of SVM classifiers [37] is first presented. Then a fuzzy SVM (FSVM) model [35] is briefly reviewed. Finally, the least squares FSVM (LS-FSVM) model is formulated.

2.1. SVM for Binary Classification (By Vapnik [37]). Given a training dataset $\{x_i, y_i\}$ ($i = 1, \dots, N$), where $x_i \in R^N$ is the i th input pattern and y_i is its corresponding observed result, which is a binary variable. In credit risk evaluation models, x_i s denote the attributes of customers and y_i is the observed outcome of repayment obligations. If the customer defaults, $y_i = 1$, or else $y_i = -1$. The generic idea of SVM is first to map the input data into a high-dimensional feature space through a mapping function and then to find the optimal separating hyperplane with minimal classification errors. The separating hyperplane can be represented as

$$z(x) = w^T \phi(x) + b = 0, \quad (1)$$

where w is the normal vector of the hyperplane and b is the bias.

Suppose $\phi(\cdot)$ is a nonlinear function that maps the input space into a higher dimensional feature space. If a dataset

is linearly separable in this feature space, the classifier is constructed as

$$\begin{aligned} w^T \phi(x_i) + b &\geq 1 & \text{if } y_i = 1 \\ w^T \phi(x_i) + b &\leq -1 & \text{if } y_i = -1 \end{aligned} \quad (2)$$

which is equivalent to

$$y_i (w^T \phi(x_i) + b) \geq 1 \quad \text{for } i = 1, \dots, N. \quad (3)$$

In order to deal with dataset that is not linearly separable, the previous analysis can be generalized by introducing some nonnegative variables $\xi_i \geq 0$, such that (3) can be modified as follows:

$$y_i [w^T \phi(x_i) + b] \geq 1 - \xi_i \quad \text{for } i = 1, \dots, N. \quad (4)$$

The nonnegative ξ_i in (4) are those for which data point x_i does not satisfy (3). Thus, the term $\sum_{i=1}^N \xi_i$ can be considered as a measure of the amount of misclassification, that is, tolerable misclassification errors.

According to the structural risk minimization principle [37], the risk bound is minimized by solving the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \Phi(w, b, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \\ \text{Subject to:} \quad & y_i (w^T \phi(x_i) + b) = 1 - \xi_i \quad \text{for } i = 1, \dots, N \\ & \xi_i \geq 0 \quad \text{for } i = 1, \dots, N, \end{aligned} \quad (5)$$

where C is a free regularization parameter controlling a tradeoff between margin maximization and tolerable misclassification errors.

Searching the optimal hyperplane in (5) is a quadratic programming (QP) problem. After introducing a set of Lagrangian multipliers α_i and β_i for constraints in (5), the primal problem in (5) is to find out the saddle point of the Lagrangian function; that is,

$$\begin{aligned} \min_{w, b, \xi_i} L(w, b, \xi_i; \alpha_i, \beta_i) &= \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \\ &\quad - \sum_{i=1}^N \alpha_i [y_i (w^T \phi(x_i) + b) - 1 + \xi_i] \\ &\quad - \sum_{i=1}^N \beta_i \xi_i. \end{aligned} \quad (6)$$

Differentiating (6) with w , b , and ξ_i , one obtains

$$\begin{aligned}\frac{d}{dw}L(w, b, \xi_i; \alpha_i, \beta_i) &= w - \sum_{i=1}^N \alpha_i y_i \phi(x_i) = 0, \\ \frac{d}{db}L(w, b, \xi_i; \alpha_i, \beta_i) &= -\sum_{i=1}^N \alpha_i y_i = 0, \\ \frac{d}{d\xi_i}L(w, b, \xi_i; \alpha_i, \beta_i) &= C - \alpha_i - \beta_i = 0.\end{aligned}\quad (7)$$

From (7) one has $w = \sum_{i=1}^N \alpha_i y_i \phi(x_i)$. The key issue is how to determine the value of α_i . To obtain a solution, the dual problem of the primal problem (6) becomes

$$\begin{aligned}\text{Maximize } Q(\alpha) &= -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ &\quad + \sum_{i=1}^N \alpha_i\end{aligned}\quad (8)$$

$$\text{Subject to: } \sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N.$$

Function $\phi(x)$ in (8) is then related to $K(x_i, x_j)$ by imposing

$$\phi(x_i)^T \phi(x_j) = K(x_i, x_j), \quad (9)$$

which is motivated by the Mercer theorem [37]. $K(x_i, x_j)$ is the kernel function in the input space that determines the inner products of the two data points in the feature space. According to the Karush-Kuhn-Tucker (KKT) theorem [38], the KKT conditions are defined as

$$\alpha_i [y_i (w^T \phi(x) + b) - 1 + \xi_i] = 0, \quad i = 1, 2, \dots, N, \quad (10)$$

$$(C - \alpha_i) \xi_i = 0, \quad i = 1, 2, \dots, N. \quad (11)$$

From this equality, it is deduced that the only nonzero values, α_i , in (10), are those for which constraints in (4) are satisfied with the equality sign. Data points x_i ($i = 1, 2, \dots, s$), corresponding to $\alpha_i > 0$, are called support vectors (SVs). But there are two types of SVs in a nonseparable case. In the case of $0 < \alpha_i < C$, the corresponding support vector x_i satisfies equalities $y_k [w^T \phi(x_k) + b] = 1$ and $\xi_i = 0$. In the case of $\alpha_i = C$, the corresponding ξ_i is not zero and the corresponding support vector x_i does not satisfy (2), which considered such SVs as misclassification errors. In terms of the above processing, data points x_i ($i = 1, 2, \dots, s$), corresponding to $\alpha_i = 0$, are classified correctly.

Using the support vectors, the optimal solution for the weight vector in (7) can be given by

$$w = \sum_{i=1}^{N_s} \alpha_i y_i \phi(x_i), \quad (12)$$

where N_s is the number of SVs. Moreover, in the case of $0 < \alpha_i < C$, condition $\xi_i = 0$ applies to (11) in terms of

the KKT theorem [38]. Thus, one may determine the optimal bias b by taking any data point in the dataset. However, from the numerical perspective, it is better to take the mean value of b , from such data points in the dataset. Once the optimal parameter pair (w, b) is determined, the decision function of the SVM classifiers can be represented as

$$z(x) = \text{sign} \left(\sum_{i=1}^{N_s} \alpha_i y_i K(x_i, x_j) + b \right). \quad (13)$$

2.2. FSVM (by Lin and Wang [35]). SVM has been proved to be a powerful tool for solving classification problems [37], but it has some inherent limitations. From its formulation discussed above, each training point in the training dataset belongs to either one or the other class. But in many real-world applications, every training sample does not exactly belong to one of the two classes; it may belong to one class to the extent of 80 percent and 20 percent of it may be meaningless. That is to say, there is a membership grade $\{\mu_i\}_{i=1}^N \in [0, 1]$ associated with each training data point x_i . In this sense, FSVM is an extension of a SVM that takes into consideration the varying significance of the training samples. For FSVM, each training sample is associated with a membership value $\{\mu_i\}_{i=1}^N \in [0, 1]$. The membership value μ_i reflects the confidence degree of the data points. The higher the value, the higher the confidence degree of its class label. Similar to SVM, the optimization problem of the FSVM [35] is formulated as follows:

$$\begin{aligned}\text{Minimize } \Psi(w, b, \xi_i, \mu_i) &= \frac{1}{2} w^T w + C \sum_{i=1}^N \mu_i \xi_i \\ \text{Subject to: } y_i (w^T \phi(x_i) + b) &\geq 1 - \xi_i \quad \text{for } i = 1, \dots, N \\ \xi_i &\geq 0 \quad \text{for } i = 1, \dots, N.\end{aligned}\quad (14)$$

Similar to SVM, the solution of FSVM is obtained from the above quadratic programming (QP) problem. Note that error term ξ_i is scaled by membership value μ_i . The membership values used to weigh the soft penalty term reflect the relative confidence degree of the training samples during training. Important samples with larger membership values will have more impact in FSVM training than those with smaller values.

Similar to Vapnik's SVM [37], optimization problem of the FSVM can be transformed into the following dual problem:

$$\begin{aligned}\text{Maximize } W(\alpha) &= -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &\quad + \sum_{i=1}^N \alpha_i \\ \text{Subject to: } \sum_{i=1}^N \alpha_i y_i &= 0, \quad 0 \leq \alpha_i \leq \mu_i C, \quad i = 1, \dots, N.\end{aligned}\quad (15)$$

In the same way, KKT conditions are defined as

$$\begin{aligned}\alpha_i [y_i (w^T \phi(x) + b) - 1 + \xi_i] &= 0, \quad i = 1, 2, \dots, N, \\ (\mu_i C - \alpha_i) \xi_i &= 0, \quad i = 1, 2, \dots, N.\end{aligned}\quad (16)$$

Data point x_i , corresponding to $\alpha_i > 0$, is called a support vector. There are two types of SVs. The one corresponding to $0 < \alpha_i < \mu_i C$ lies on the margin of the hyperplane, and the other corresponding to $\alpha_i = \mu_i C$ is treated as misclassified.

Solving (15) leads to a decision function similar to (13), but with different support vectors and the corresponding weights α_i . An important difference between SVM and FSVM is that data points with the same value of α_i may be indicated as a different type of SVs in FSVM due to membership factor μ_i . Interested readers can refer to [35] for more details.

2.3. Least Squares FSVM. In both SVM and FSVM, the final solution can be described as an issue closely associated with quadratic programming (QP) problem. The main issue with the QP method is that it is a time-consuming process to find the solutions when handling some large-scale real-world application problems. Motivated by Lai et al. [14] and Suykens and Vandewalle [39], the least squares FSVM (LS-FSVM) model is introduced by formulating the following optimization problem:

$$\begin{aligned}\text{Minimize } \varphi(w, b, \xi_i, \mu_i) &= \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^N \mu_i \xi_i^2 \\ \text{Subject to: } y_i (w^T \phi(x_i) + b) &= 1 - \xi_i \quad \text{for } i = 1, \dots, N.\end{aligned}\quad (17)$$

One can define the Lagrangian function as

$$\begin{aligned}\min_{w, b, \xi_i} L(w, b, \xi_i; \alpha_i) &= \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^N \mu_i \xi_i^2 \\ &\quad - \sum_{i=1}^N \alpha_i [y_i (w^T \phi(x_i) + b) - 1 + \xi_i],\end{aligned}\quad (18)$$

where α_i is the i th Lagrangian multiplier, which can be either positive or negative, due to equality constraints in accordance with KKT conditions [38].

The optimal conditions are obtained by differentiating (18):

$$\begin{aligned}\frac{d}{dw} L(w, b, \xi_i; \alpha_i) &= w - \sum_{i=1}^N \alpha_i y_i \phi(x_i) = 0, \\ \frac{d}{db} L(w, b, \xi_i; \alpha_i) &= -\sum_{i=1}^N \alpha_i y_i = 0, \\ \frac{d}{d\xi_i} L(w, b, \xi_i; \alpha_i) &= \mu_i C \xi_i - \alpha_i = 0, \\ \frac{d}{d\alpha_i} L(w, b, \xi_i; \alpha_i) &= y_i [w^T \phi(x_i) + b] - 1 + \xi_i = 0.\end{aligned}\quad (19)$$

From (19), one can obtain the following equations:

$$\begin{aligned}w &= \sum_{i=1}^N \alpha_i y_i \phi(x_i), \\ \sum_{i=1}^N \alpha_i y_i &= 0, \\ \alpha_i &= \mu_i C \xi_i, \\ y_i [w^T \phi(x_i) + b] - 1 + \xi_i &= 0.\end{aligned}\quad (20)$$

Using a matrix form, optimal conditions in (20) can be expressed by

$$\begin{bmatrix} \mathbf{\Omega} & \mathbf{Y} \\ \mathbf{Y}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix},\quad (21)$$

where $\mathbf{\Omega}$, $\mathbf{Y}$, and $\mathbf{1}$ are (22), (23), and (24), respectively:

$$\mathbf{\Omega}_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) + (\mu_i C)^{-1} I \quad (22)$$

$$\mathbf{Y} = (y_1, y_2, \dots, y_N)^T, \quad (23)$$

$$\mathbf{1} = (1, 1, \dots, 1)^T. \quad (24)$$

From (22), $\mathbf{\Omega}$ is positive definite. Thus, α can be obtained from (21); that is,

$$\alpha = \mathbf{\Omega}^{-1} (\mathbf{1} - b \mathbf{Y}). \quad (25)$$

Substituting (25) into the second matrix equation in (21), we can obtain

$$b = \frac{\mathbf{Y}^T \mathbf{\Omega}^{-1} \mathbf{1}}{\mathbf{Y}^T \mathbf{\Omega}^{-1} \mathbf{Y}}. \quad (26)$$

Here, since $\mathbf{\Omega}$ is positive definite, $\mathbf{\Omega}^{-1}$ is also positive definite. In addition, since $\mathbf{Y}$ is a nonzero vector, $\mathbf{Y}^T \mathbf{\Omega}^{-1} \mathbf{Y} > 0$. Thus, b is always obtained. Substituting (26) into (25), α can be obtained.

Hence, the separating hyperplane of LS-FSVM can be found by solving the linear set of (21)–(24), instead of quadratic programming (QP), thereby reducing the computational complexity, especially for large-scale problems.

The main advantages of the LS-FSVM can be summarized into the following five aspects. First of all, the LS-FSVM requires fewer prior assumptions about the input data than those required in statistical approaches, such as normal distribution and continuity. Second, it can perform nonlinear mapping from an original input space into a high dimensional feature space, in which it constructs a linear discriminant function to replace the nonlinear function in the original input space. This characteristic also solves the dimension disaster problem because its computational complexity is not dependent on the samples' dimension. Third, it attempts to learn the separating hyperplane to maximize the classification margin, thereby implementing structural risk minimization and realizing good generalization capability.

TABLE 1: Variables of the dataset.

Number	Variables
1	Year of birth
2	Number of children
3	Number of other dependents
4	Is there a home phone
5	Applicant's income
6	Applicant's employment status
7	Spouse's income
8	Residential status
9	Value of home
10	Mortgage balance outstanding
11	Outgoings on mortgage or rent
12	Outgoings on loans
13	Outgoings on hire purchase
14	Outgoings on credit cards

Fourth, a distinct trait of LS-FSVM is that it can further lower computational complexity by transforming a quadratic programming problem into a linear equation set problem. Finally, an important advantage of LS-FSVM is that support values α_i in LS-FSVM are proportional to the membership degree, as well as misclassification errors at the data points, thus making LS-FSVM more suitable for some real-world problems, which is the main difference between the proposed LS-FSVM and the traditional SVM and LS-SVM models. These important characteristics also make LS-FSVM preferable in many practical applications. In the following section, some experiments are presented for verification purpose.

3. Experiment Analysis

In this section, a real-world credit dataset is used to test the performance of LS-FSVM. For comparison purposes, linear regression (LinR) [14], logistic regression (LogR) [2], artificial neural network (ANN) [8, 9], Vapnik's SVM [37], Lin and Wang's FSVM [35], and LS-SVM [39] are also used.

The dataset in this study comes from a financial services company of England, obtained from a CDROM published by Thomas et al. [40]. Each applicant has 14 characteristics or variables, listed in Table 1. The dataset includes detailed information of 1225 applicants, of which 323 are observed as bad customers.

In this experiment, LS-FSVM, FSVM, LS-SVM, and SVM models use RBF kernel for classification. In the ANN model, a three-layer back-propagation neural network with 10 sigmoidal neurons is used in the hidden layer, and one linear neuron is used in the output layer. The network training algorithm is the Levenberg-Marquardt (LM) algorithm. Besides, the learning and momentum rates are set to 0.1 and 0.15, respectively. The accepted average squared error is 0.05, and the training epochs are 1600. The above parameters are obtained by trial-and-error method. The experiment is run by MATLAB 6.1 with statistical toolbox, NNET toolbox, and

LS-SVM toolbox. In addition, three evaluation criteria are used to measure the efficiency of classification:

Type I accuracy

$$= \frac{\text{number of both observed bad and classified as bad}}{\text{number of observed bad}}$$

Type II accuracy

$$= \frac{\text{number of both observed good and classified as good}}{\text{number of observed good}}$$

$$\text{Total accuracy} = \frac{\text{number of correct classification}}{\text{the number of evaluation sample}}. \quad (27)$$

To show the classification capability of LS-FSVM in distinguishing potentially insolvent customers from good customers, we perform the test with LS-FSVM at the beginning. This testing process includes five steps.

First, the number of observed bad customers is tripled to make their number nearly equal to the number of observed good customers. The main purpose of such processing is to avoid the impact on performance of imbalanced samples. A similar processing method can be found in Wang et al. [25]. Of course, some other imbalanced data processing methods, for example, the information granulation based method [41], can also be used.

Second, the original data is preprocessed to impute the missing data and transform category data. In this study, interpolation method is used to impute the missing data. For category data, a numerical method is used for transformation.

Third, the original dataset is randomly separated into two parts, that is, training samples and testing samples. In this study, 1500 samples are used for training and the remaining 371 samples are used for holdout testing and performance evaluation.

Fourth, membership grades are generated by the linear transformation function proposed by Wang et al. [25], in terms of the initial score obtained by experts' experience and opinions. Of course, some new membership generation methods in [42] can also be adopted.

Finally, the LS-FSVM classifier is trained and accordingly the results can be evaluated. The above five steps are repeated 20 times to confirm the robustness of the proposed method. In this study, the efficiency and robustness of credit risk evaluation using the LS-FSVM model are shown in Table 2.

As can be seen from Table 2, the proposed LS-FSVM model exhibits significant classification capabilities. In the 20 experiments, type I accuracy, type II accuracy, and total accuracy are 81.34%, 93.41%, and 89.21%, respectively, in the mean sense. Furthermore, the standard deviation is rather small, revealing that robustness of the LS-FSVM classifier is good. These results imply that the LS-FSVM model is a promising credit risk evaluation technique.

For further illustration, the classification power of the LS-FSVM is also compared with six other commonly used classifiers: linear regression (LinR) [14], logistics regression

TABLE 2: Computational results of credit risk evaluation using LS-FSVM.

Experiment number	Type I (%)	Type II (%)	Total (%)
1	81.56	93.81	88.54
2	86.98	95.14	92.53
3	82.02	91.84	88.49
4	79.81	98.36	93.53
5	87.77	94.03	92.24
6	81.56	96.85	92.38
7	79.19	93.05	87.41
8	85.69	92.11	88.86
9	79.27	89.33	87.63
10	82.45	96.58	90.45
11	77.96	93.08	85.06
12	80.61	89.57	85.89
13	81.16	93.41	88.34
14	78.96	89.88	87.56
15	85.56	97.35	94.53
16	70.36	98.13	84.11
17	80.14	92.01	88.68
18	75.22	89.54	86.25
19	86.72	89.61	88.04
20	83.85	94.70	92.64
Mean	81.34	93.41	89.21
Stdev	4.20	2.98	3.02

TABLE 3: Results of comparisons of different classifiers.

Method	Type I (%)	Type II (%)	Overall (%)	CPU time (s)
LinR	52.87	43.48	50.22	102.45
LogR	60.08	62.29	60.66	112.87
ANN	56.57	78.36	72.24	318.72
SVM	70.13	83.49	77.02	306.33
LS-SVM	79.37	93.27	89.16	138.61
FSVM	80.08	92.86	88.38	335.24
LS-FSVM	81.34	93.41	89.21	149.49

(LogR) [2], artificial neural network (ANN) [8, 9], Vapnik's SVM [37], FSVM [35], and LS-SVM [39]. The results of the comparison are reported in Table 3.

From Table 3, several important results can be observed.

- (a) For type I accuracy, LS-FSVM is the best of all the listed approaches, followed by FSVM, LS-SVM, Vapnik's SVM, logistics regression, artificial neural network model, and linear regression model, implying that the LS-FSVM is a very promising technique in credit risk assessment. Particularly, performance of the two fuzzy SVM techniques (Lin and Wang's FSVM [35] and LS-FSVM) is better than other classifiers listed in this study, implying that the fuzzy SVM classifier may be more suitable for credit risk assessment tasks than other deterministic classifiers, such as linear regression (LinR) and logit regression (LogR).

- (b) From the viewpoint of type II accuracy, LS-FSVM and LS-SVM outperform the other five models, implying the strong capability of the least squares version of the SVM model in credit risk evaluation. In the meantime, the proposed LS-FSVM model seems to be slightly better than LS-SVM, revealing that the LS-FSVM is a feasible solution to improve the accuracy of credit risk evaluation. Interestingly, performance of the FSVM is slightly worse than that of the LS-SVM. The reasons for this phenomenon are worth exploring further.
- (c) According to the total accuracy, performances of the two statistical models (LinR and LogR) are much worse than those of the other five models. The main reason is that the five models can effectively capture the nonlinear patterns hidden in the credit data. As is known, there are many factors that affect customer credit. Usually, the relationships between customer default and these factors are very subtle and complex. Besides some linear relationships, some nonlinear relationships often exist in the credit data. Therefore, nonlinear intelligent models can offer advantage over traditional linear models.
- (d) From the perspective of computational time, two traditional classification models (i.e., LinR and LogR) are faster than all the intelligent models due to their simplicity. In all the intelligent models, LS-SVM and LS-FSVM are faster than other intelligent models due to the least squares principle. However, LS-FSVM is slightly slower than LS-SVM because the fuzzification needs some processing time, but LS-FSVM performs faster than the SVM and FSVM, indicating that the proposed LS-FSVM model is a very effective model in credit risk evaluation.
- (e) In the five intelligent models, the performance of ANN and SVM models is much worse than the other three intelligent models. The main reason is that the standard ANN and SVM models have their own shortcomings, such as the sensitivity of parameters and outliers, thus affecting their classification performance. For example, ANN models often get trapped into local minima and suffer from overfitting problems, while SVM models occasionally encounter overfitting problem [43]; moreover, some fuzzy information is not handled well by the standard SVM models.
- (f) LS-SVM, FSVM, and LS-FSVM models outperform the other four models listed in this study, implying that the variants of SVM have a strong classification potentiality for credit risk evaluation. The possible reason is that the improvement of SVM variants overcomes some inherent limitations of standard SVM proposed by Van Gestel [44], thereby increasing the generalization capability. From a general point of view, LS-FSVM outperforms the other six classifiers from the above three measurements, revealing that

TABLE 4: McNemar's test for pairwise performance comparison of different models.

Model	LS-SVM	FSVM	SVM	ANN	LogR	LinR
LS-FSVM	0.0000 [1.0000]	0.0480 [0.8262]	15.488 [0.0001]	26.881 [0.0001]	59.274 [0.0001]	92.160 [0.0001]
LS-SVM		0.0120 [0.9131]	14.675 [0.0001]	25.840 [0.0001]	57.840 [0.0001]	90.482 [0.0001]
FSVM			13.133 [0.0003]	23.842 [0.0001]	55.048 [0.0001]	87.197 [0.0001]
SVM				1.5370 [0.2501]	15.584 [0.0001]	36.300 [0.0001]
ANN					7.0840 [0.0078]	22.781 [0.0001]
LogR						4.3630 [0.0367]

the LS-FSVM is used as an effective tool for credit risk evaluation.

In terms of Table 3 and the three measurements, it is easy to judge which model is the best and which model is the worst. However, it is unclear what the differences between good models and bad ones are. For this purpose, McNemar's test [45] is conducted to examine whether the proposed LS-FSVM classifier significantly outperforms the other six classifiers listed in this study.

As a nonparametric test for the two related samples, McNemar's test is particularly useful for before-after measurement of the same subjects [46]. Taking the total accuracy results from Table 3, Table 4 shows the results of the McNemar's test for the credit dataset to statistically compare the performance in respect of testing data among the seven classifiers. It should be noted that the results listed in Table 4 are the Chi-squared values, and P values are in brackets.

According to the results reported in Table 4, some important conclusions can be drawn in terms of McNemar's statistical test.

- (1) The proposed LS-FSVM classifier outperforms the standard SVM, ANN, logit regression (LogR), and linear regression (LinR) models at 1% statistical significance level. However, the proposed LS-FSVM model does not significantly outperform the LS-SVM model and the FSVM model. These results are consistent with those of Table 3.
- (2) Similar to the LS-FSVM model, LS-SVM and FSVM models outperform the other four individual models (i.e., individual SVM, ANN, LogR, and LinR models) at 1% significance level. But the McNemar's test does not conclude that the LS-SVM model performs better than the FSVM model.
- (3) For SVM and ANN models, we can find that these two models perform much better than the two statistical models (i.e., LogR and LinR models) at 1% significance level. Interestingly, the SVM model does not outperform the ANN model at 10% significance level, although many applications have reported that the performance of the SVM was much better than that of ANN. The possible reason lies in data samples used in this study.
- (4) Comparing with LogR and LinR models, it is easy to find that the LogR model performs better than the LinR model at 5% significance level. All findings are consistent with results reported in Table 3.

In summary, according to the above experimental results and statistical testing, it is easy to conclude that the LS-FSVM model can significantly outperform some standard intelligent models (e.g., SVM and ANN) and some statistical models (e.g., LogR and LinR), revealing that the LS-FSVM can also be used as a competitive solution to credit risk evaluation.

4. Conclusions

In this paper, a powerful classification method—least squares fuzzy support vector machines (LS-FSVM)—is proposed to evaluate credit risks. Through the least squares method, a quadratic programming (QP) problem of SVM can be transformed into a set of linear equations successfully, thereby reducing the computational complexity. Furthermore, the fuzzification processing in the proposed LS-FSVM model adds generalization capability and insensitivity to outliers. Experiments with real-world dataset have produced good classification results and fast computational efficiency and have demonstrated that the proposed LS-FSVM model can provide a feasible alternative to credit risk assessment. Besides the credit risk evaluation problem, the proposed LS-FSVM model can also be extended to other applications, such as consumer credit rating and corporate failure prediction problems, which will be investigated in the future research.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author would like to express his sincere appreciation to two independent referees in making valuable comments and suggestions to the paper. His comments have improved the quality of the paper immensely. This work is partially supported by Grants from the National Science Fund for Distinguished Young Scholars (NSFC no. 71025005), the National Natural Science Foundation of China (NSFC nos. 90924024 and 91224001), Hangzhou Normal University, and the Fundamental Research Funds for the Central Universities in BUCT.

References

- [1] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *Journal of Finance*, vol. 23, pp. 89–609, 1968.

- [2] J. C. Wiginton, "A note on the comparison of logit and discriminant models of consumer credit behaviour," *Journal of Financial Quantitative Analysis*, vol. 15, pp. 757–770, 1980.
- [3] B. J. Grablowsky and W. K. Talley, "Probit and discriminant functions for classifying credit applicants: a comparison," *Journal of Economic Business*, vol. 33, pp. 254–261, 1981.
- [4] F. Glover, "Improved linear programming models for discriminant analysis," *Decision Science*, vol. 21, pp. 771–785, 1990.
- [5] O. L. Mangasarian, "Linear and nonlinear separation of patterns by linear programming," *Operations Research*, vol. 13, pp. 444–452, 1965.
- [6] W. E. Henley and D. J. Hand, "A k-nearest-neighbour classifier for assessing consumer credit risk," *Journal of the Royal Statistical Society Series D: The Statistician*, vol. 45, no. 1, pp. 77–95, 1996.
- [7] P. Makowski, "Credit scoring branches out," *Credit World*, vol. 75, pp. 30–37, 1985.
- [8] K. K. Lai, L. Yu, S. Y. Wang, and L. G. Zhou, "Neural network meta-learning for credit scoring," in *Intelligent Computing*, vol. 4113 of *Lecture Notes in Computer Science*, pp. 403–408, 2006.
- [9] R. Malhotra and D. K. Malhotra, "Evaluating consumer loans using neural networks," *Omega*, vol. 31, no. 2, pp. 83–96, 2003.
- [10] M.-C. Chen and S.-H. Huang, "Credit scoring and rejected instances reassigning through evolutionary computation techniques," *Expert Systems with Applications*, vol. 24, no. 4, pp. 433–441, 2003.
- [11] F. Varetto, "Genetic algorithms applications in the analysis of insolvency risk," *Journal of Banking and Finance*, vol. 22, no. 10–11, pp. 1421–1439, 1998.
- [12] Z. Zhu, H. He, J. A. Starzyk, and C. Tseng, "Self-organizing learning array and its application to economic and financial problems," *Information Sciences*, vol. 177, no. 5, pp. 1180–1192, 2007.
- [13] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: A Market Comparative Study," *Decision Support Systems*, vol. 37, no. 4, pp. 543–558, 2004.
- [14] 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 Artificial Intelligence*, pp. 490–495, 2006.
- [15] K. K. Lai, L. Yu, W. Huang, and S. Y. Wang, "A novel support vector machine metamodel for business risk identification," in *PRICAI 2006: Trends in Artificial Intelligence*, vol. 4099 of *Lecture Notes in Artificial Intelligence*, pp. 480–484, 2006.
- [16] R. Malhotra and D. K. Malhotra, "Differentiating between good credits and bad credits using neuro-fuzzy systems," *European Journal of Operational Research*, vol. 136, no. 1, pp. 190–211, 2002.
- [17] 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.
- [18] L. Zhou, K. K. Lai, and L. Yu, "Credit scoring using support vector machines with direct search for parameters selection," *Soft Computing*, vol. 13, no. 2, pp. 149–155, 2009.
- [19] L. Yu and X. Yao, "A total least squares proximal support vector classifier for credit risk evaluation," *Soft Computing*, vol. 17, no. 4, pp. 643–650, 2013.
- [20] L. Yu, X. Yao, S. Wang, and K. K. Lai, "Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15392–15399, 2011.
- [21] L. Yu, S. Wang, and J. Cao, "A modified least squares support vector machine classifier with application to credit risk analysis," *International Journal of Information Technology and Decision Making*, vol. 8, no. 4, pp. 697–710, 2009.
- [22] L. Yu, S. Wang, and K. K. Lai, "Credit risk evaluation using a c-variable least squares support vector classification model," *Communications in Computer and Information Science*, vol. 35, pp. 573–579, 2009.
- [23] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, "Credit scoring using the hybrid neural discriminant technique," *Expert Systems with Applications*, vol. 23, no. 3, pp. 245–254, 2002.
- [24] S. Piramuthu, "Financial credit-risk evaluation with neural and neurofuzzy systems," *European Journal of Operational Research*, vol. 112, no. 2, pp. 310–321, 1999.
- [25] Y. Wang, S. Wang, and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, 2005.
- [26] L. Yu, S. Wang, F. Wen, K. K. Lai, and S. He, "Designing a hybrid intelligent mining system for credit risk evaluation," *Journal of Systems Science & Complexity*, vol. 21, no. 4, pp. 527–539, 2008.
- [27] R. Smalz and M. Conrad, "Combining evolution with credit apportionment: a new learning algorithm for neural nets," *Neural Networks*, vol. 7, no. 2, pp. 341–351, 1994.
- [28] K. K. Lai, L. Yu, S. Y. Wang, and L. G. Zhou, "Credit risk analysis using a reliability-based neural network ensemble model," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN '06)*, vol. 4132 of *Lecture Notes in Computer Science*, pp. 682–690, 2006.
- [29] L. Yu, S. Wang, and K. K. Lai, "Credit risk assessment with a multistage neural network ensemble learning approach," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1434–1444, 2008.
- [30] L. Yu, W. Yue, S. Wang, and K. K. Lai, "Support vector machine based multiagent ensemble learning for credit risk evaluation," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1351–1360, 2010.
- [31] L. Yu, S. Wang, and K. K. Lai, "An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring," *European Journal of Operational Research*, vol. 195, no. 3, pp. 942–959, 2009.
- [32] L. C. Thomas, "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers," *International Journal of Forecasting*, vol. 16, no. 2, pp. 149–172, 2000.
- [33] L. C. Thomas, R. W. Oliver, and D. J. Hand, "A survey of the issues in consumer credit modelling research," *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1006–1015, 2005.
- [34] L. Yu, S. Y. Wang, K. K. Lai, and L. G. Zhou, *Bio-Inspired Credit Risk Analysis—Computational Intelligence with Support Vector Machines*, Springer, Berlin, Germany, 2008.
- [35] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.
- [36] D. Tsujinishi and S. Abe, "Fuzzy least squares support vector machines for multiclass problems," *Neural Networks*, vol. 16, no. 5–6, pp. 785–792, 2003.
- [37] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [38] R. Fletcher, *Practical Methods of Optimization*, A Wiley-Interscience Publication, John Wiley & Sons, 2nd edition, 1987.
- [39] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

- [40] L. C. Thomas, D. B. Edelman, and J. N. Crook, *Credit Scoring and Its Applications*, Society of Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2002.
- [41] M.-C. Chen, L.-S. Chen, C.-C. Hsu, and W.-R. Zeng, "An information granulation based data mining approach for classifying imbalanced data," *Information Sciences*, vol. 178, no. 16, pp. 3214–3227, 2008.
- [42] A. Celikyilmaz and I. B. Turksen, "Fuzzy functions with support vector machines," *Information Sciences*, vol. 177, no. 23, pp. 5163–5177, 2007.
- [43] L. Cao and F. E. H. Tay, "Financial forecasting using Support Vector Machines," *Neural Computing and Applications*, vol. 10, no. 2, pp. 184–192, 2001.
- [44] T. Van Gestel, B. Baesens, J. Garcia, and P. Van Dijcke, "A support vector machine approach to credit scoring," *Bank en Financiewezen*, vol. 2, pp. 73–82, 2003.
- [45] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
- [46] D. R. Cooper and C. W. Emory, *Business Research Methods*, Irwin, Chicago, Ill, USA, 1995.