A MODIFIED LEAST SQUARES SUPPORT VECTOR MACHINE CLASSIFIER WITH APPLICATION TO CREDIT RISK ANALYSIS∗

LEAN YU†,‡ SHOUYANG WANG† and JIE CAO§

†Institute of Systems Science
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
‡Research Center for Financial Engineering and Financial Management
Changsha 410114, China
§School of Economics and Management
Nanjing University of Information Science & Technology
Nanjing 210044, China
†yulean@amss.ac.cn

In this paper, a modified least squares support vector machine classifier, called the $C$-variable least squares support vector machine ($C$-VLSSVM) classifier, is proposed for credit risk analysis. The main idea of the proposed classifier is based on the prior knowledge that different classes may have different importance for modeling and more weight should be given to classes having more importance. The $C$-VLSSVM classifier can be obtained by a simple modification of the regularization parameter, based on the least squares support vector machine (LSSVM) classifier, whereby more weight is given to errors in classification of important classes, than to errors in classification of unimportant classes, while keeping the regularized terms in their original form. For illustration purpose, two real-world credit data sets are used to verify the effectiveness of the $C$-VLSSVM classifier. Experimental results obtained reveal that the proposed $C$-VLSSVM classifier can produce promising classification results in credit risk analysis, relative to other classifiers listed in this study.

Keywords: Least squares support vector machine classifier; regularization parameter; prior knowledge; credit risk analysis.

1. Introduction

Credit risk evaluation is one of the most important issues in financial risk management. The recent financial crisis has made more people more aware about the importance of credit risk. The result is that credit risk evaluation has become a major focus for academic researchers and business practitioners alike. In the past few
decades, different hard-computing techniques such as discriminant analysis,\textsuperscript{2} logit analysis,\textsuperscript{3} probit analysis,\textsuperscript{4} geometrical method\textsuperscript{5} and mathematical programming\textsuperscript{6} have been applied to credit risk evaluation. Due to the nonlinear relationship between default probability and credit patterns, hard-computing techniques do not generate good classification performance in credit risk analysis. For this purpose, some emerging soft-computing techniques, such as artificial neural networks (ANN),\textsuperscript{7,8} evolutionary algorithms (EA),\textsuperscript{9} and support vector machines (SVM)\textsuperscript{10–13} have also been used to classify credit risks and other risks. More references about soft computing for credit risk evaluation can refer to the monograph of Yu et al.\textsuperscript{13}

Empirical tests have revealed that soft computing techniques are advantageous compared to the traditional hard computing techniques in credit risk analysis, due to the flexibility of tolerable classification errors.

Among soft-computing techniques, SVM has been reported to be the best by many practical credit risk evaluation experiments.\textsuperscript{10–13} In terms of classification and regression problems, SVM applications can be categorized into support vector classification (SVC) and support vector regression (SVR).\textsuperscript{14} In this paper, SVC is used to judge whether a customer is likely to default or not. However, the solution of the standard SVC is obtained by solving a convex quadratic programming (QP) problem.\textsuperscript{14} An important shortcoming of QP is that it might lead to a high computational cost if a large-scale problem is required to be computed. To avoid this shortcoming, a least squares support vector machine (LSSVM) classifier, first proposed by Suykens and Vandewalle,\textsuperscript{15} is presented, where the solution can be obtained by solving a set of linear equations, instead of solving the QP problem.

In practical credit risk classification problems, bad customers’ classification may be more important than correct classification of good customers because bad customers usually lead to direct economic loss if the credit loan is approved by credit-granting institutions. The prior knowledge may be extremely important in credit risk management. If customers classified as bad are given more attention than those classified as good customers, classification accuracy may be greatly improved, thereby reducing financial losses of commercial banks and credit card companies. For this purpose, the prior knowledge that different classes need to be given different levels of importance should be taken into account in modeling. It is, therefore, advisable for decision-makers to give more weight to customers belonging to classes that need to be given more importance. In view of this idea, a modified version of LSSVM classifier, called \textit{C}-variable least squares support vector machine (\textit{C}-VLSSVM) classifier,\textsuperscript{1} which uses the variable regularization parameter in LSSVM to classify good or delinquent customers. The \textit{C}-variable LSSVM classifier can be obtained by a simple modification of regularization parameter \textit{C} in LSSVM, whereby the data of important classes are penalized more heavily than data of relatively less important classes of customers.

The primary objectives of this paper are to propose a modified LSSVM classifier called the \textit{C}-variable LSSVM (\textit{C}-VLSSVM) classifier that can significantly
reduce computational cost, and to examine whether the prior knowledge that data of important classes should be given more weight than data of less important classes can be utilized to improve the accuracy and the effectiveness of classification, especially in credit risk classification problems. In addition, the proposed C-VLSSVM classifier also provides a solution to an unsolved issue given in Eq. (11) in the paper of Zhou et al.16

The remainder of this paper is organized as follows. In Sec. 2, a least square support vector machine (LSSVM) classifier is briefly reviewed. Section 3 presents the formulation of the C-variable LSSVM (C-VLSSVM) classifier in detail. For further illustration, two typical credit data sets are used for verification and accordingly the computational results are reported in Secs. 4 and 5, respectively. Finally, some concluding remarks are drawn in Sec. 6.

2. Overview of Least Squares Support Vector Machine (LSSVM) Classifier

Assume that there is a training data set \( \{ x_i, y_i \} \) \( (i = 1, 2, \ldots, N) \) where \( x_i \in \mathbb{R}^N \) is the \( i \)th input pattern and \( y_i \) is its corresponding observed result, and it is a binary variable. In credit risk evaluation models, \( x_i \) denotes the attributes of applicants or debtors, and \( y_i \) is the observed outcome of repayment obligations. If the customer defaults, \( y_i = 1 \), or else \( y_i = -1 \). The SVM first maps the input data into a high-dimensional feature space through a mapping function \( \phi(\cdot) \) and finds the optimal separating hyperplane with minimal classification errors. The separating hyperplane can be represented as follows:

\[
z(x) = w^T \phi(x) + b = 0
\]

where \( w \) is the normal vector of the hyperplane and \( b \) is the bias, which is a scalar.

Suppose that \( \phi(\cdot) \) is a nonlinear function that maps the input space into a higher dimensional feature space. If the set is linearly separable in this feature space, the classifier should be constructed as follows:

\[
\begin{cases}
w^T \phi(x_i) + b \geq 1, & \text{if } y_i = 1 \\
w^T \phi(x_i) + b \leq -1, & \text{if } y_i = -1
\end{cases}
\]

which is equivalent to

\[
y_i(w^T \phi(x_i) + b) \geq 1, \quad i = 1, \ldots, N
\]

In order to deal with data that are not linearly separable, the previous analysis can be generalized by introducing some nonnegative variables \( \xi_i \geq 0 \), such that Eq. (3) is modified into

\[
\begin{cases}
y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, & i = 1, \ldots, N \\
\xi_i \geq 0, & i = 1, \ldots, N
\end{cases}
\]
The nonnegative variables $\xi_i$ in Eq. (4) are those for which data point $x_i$ does not satisfy Eq. (3). Thus the term $\sum_{i=1}^{N} \xi_i$ can be considered as a measure of the amount of misclassification, i.e., tolerable misclassification errors.

According to the structural risk minimization (SRM) or the margin maximization principle, the risk bound is minimized by formulating the following optimization problem:

$$
\begin{align*}
\text{Minimize} & \quad J(w, b; \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \\
\text{Subject to} & \quad y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, N
\end{align*}
$$

(5)

where $C$ is a regularization parameter controlling the trade-off between margin maximization and tolerable classification errors.

Searching the optimal hyperplane in Eq. (5) is a quadratic programming (QP) problem. When a large-scale QP problem is computed, it may lead to a high computational cost. For this purpose, Suykens and Vandewalle proposed a least squares version of support vector machines. In the least squares support vector machine (LSSVM) classifier, the following optimization problem can be formulated.

$$
\begin{align*}
\text{Minimize} & \quad J(w, b; \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i^2 \\
\text{Subject to} & \quad y_i (w^T \phi(x_i) + b) = 1 - \xi_i, \quad i = 1, \ldots, N
\end{align*}
$$

(6)

Using Eq. (6), one can define a Lagrangian function:

$$
L(w, b, \xi; \alpha_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} \alpha_i [y_i (w^T \phi(x_i) + b) - 1 + \xi_i]
$$

(7)

where $\alpha_i$ is the $i$th Lagrangian multiplier. The condition for optimality can be obtained from Eq. (7).

$$
\begin{align*}
\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\
\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \xi_i = \alpha_i / C, \quad i = 1, 2, \ldots, N \\
\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow y_i [w^T \phi(x_i) + b] - 1 + \xi_i = 0, \quad i = 1, 2, \ldots, N
\end{align*}
$$

(8)
After elimination of $w$ and $\xi_i$, the solution is given by the following set of linear equations:

\[
\begin{align*}
\sum_{i,j=1}^{N} \alpha_i y_i y_j \phi(x_i)^T \phi(x_j) + b y_i + (\alpha_i/C) - 1 &= 0 \\
\sum_{i=1}^{N} \alpha_i y_i &= 0
\end{align*}
\] (9)

Using the Mercer condition, the kernel function can be defined as $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ for $i, j = 1, 2, \ldots, N$. Typical kernel functions include linear kernel $K(x_i, x_j) = x_i^T x_j$, polynomial kernel $K(x_i, x_j) = (x_i^T x_j + 1)^d$, Gaussian or RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2)$, and MLP kernel $K(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$, where $d, \sigma, \beta$ and $\theta$ are kernel parameters, which are specified by users beforehand. Accordingly, using the matrix form, linear equations in Eq. (9) can be rewritten as

\[
\begin{bmatrix}
\Omega & Y \\
Y^T & 0
\end{bmatrix}
\begin{bmatrix}
\alpha \\
b
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0
\end{bmatrix}
\] (10)

where $b$ is a scalar, $\Omega$, $Y$, $\alpha$, and $1$ are Eqs. (11), (12) and (13), respectively, as follows:

\[
\Omega_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) + (1/C)I
\] (11)

\[
Y = (y_1, y_2, \ldots, y_N)^T
\] (12)

\[
\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T
\] (13)

\[
1 = (1, 1, \ldots, 1)^T
\] (14)

where $I$ is a unit matrix in Eq. (11). From Eq. (11), the $\Omega$ is positive definite, and solution of Lagrangian multiplier $\alpha$ can be obtained from Eq. (10), i.e.

\[
\alpha = \Omega^{-1}(1 - bY)
\] (15)

Substituting Eq. (15) into the second matrix equation of Eq. (10), we can obtain

\[
b = \frac{Y^T \Omega^{-1} 1}{Y^T \Omega^{-1} Y}
\] (16)

Here, since $\Omega$ is positive definite, $\Omega^{-1}$ is also positive definite. In addition, since $Y$ is a nonzero vector, $Y^T \Omega^{-1} Y > 0$. Thus, $b$ is always obtained. Substituting Eq. (16) into Eq. (15), $\alpha$ can be obtained. Accordingly, the solution of $w$ can be obtained from the first equation in Eq. (8). Using $w$ and $b$, the separating hyperplane in Eq. (1) can be determined. One distinct advantage of LSSVM lies in that the optimal solution of Eq. (1) can be found by solving a set of linear equations, instead of solving a quadratic programming (QP) problem, which is used in standard SVM. Thus the computational process will be simplified immensely.
and computational costs might be reduced when large-scale problems are needed to be computed.

3. C-Variable Least Squares Support Vector Machine (C-VLSSVM) Classifier

As mentioned in Sec. 2, the regularization parameter $C$ shown in Eqs. (5) and (6) determines the trade-off between the regularized term and the tolerable empirical classification errors. With the increase of $C$, the relative importance of empirical errors will increase, relative to the regularized term, and vice versa. Usually, in standard SVM, and in LSSVM, the empirical risk function’s weight $C$ is equal to weight of misclassification error function,$^{14}$ and of least squares error function.$^{15}$ That is, regularization parameter $C$ is a constant or a fixed value in standard SVM and LSSVM.

However, many practical applications have shown that a fixed regularization parameter $C$ is unsuitable for some classification tasks with some prior knowledge. Considering the prior knowledge that different classes might have different importance for classification tasks, more weight should be given to classes with more importance. In the case of LSSVM for credit risk analysis, more weight should be given to the class of delinquent customers, taking into account the prior knowledge that likely default customers might lead to more economic loss than the profit that good customers can generate. For this purpose, regularization parameter $C$ should be replaced by a variable regularization parameter $C_i$, to emphasize important classes in the sample data $\{(x_i, y_i)\}_{i=1}^N$. In terms of the prior knowledge that important classes might provide more actionable information than unimportant classes in practical classification tasks, variable regularization parameter $C_i$ should satisfy $C_i(I_A) > C_i(I_B)$, where $I_A$ and $I_B$ are the subscript sets of $A$th (important) and $B$th (unimportant) class of the training data, respectively. Since auto-adaptive regularization parameters $C_i$ will change with different classes automatically, $C_i$ is called the variable regularization parameter which will give more weight to important classes. In practical applications, the form of $C_i$ often depends on the prior knowledge we have.$^1$ In credit risk evaluation, a typical segment form is often used, which is defined as follows.

$$C_i = \sum_{l=1}^Z C^l \lambda_{il} \quad (i = 1, 2, \ldots, N; \ l = A, B, \ldots, Z) \quad (17)$$

where $C^l$ is a constant corresponding to the $l$th class of the training data, $Z$ is the number of classes, and $N$ is the number of training sample, and

$$\lambda_{il} = \begin{cases} 
1, & i \in I_l \\
0, & \text{otherwise} 
\end{cases} \quad (18)$$

where $I_l$ is the subscript set of the $l$th class of the training data.$^1$
Based on the variable regularization parameter \( C_i \) in Eq. (17), a new LSSVM called \( C \)-variable LSSVM (\( C \)-VLSSVM) can be introduced. Similar to Eq. (6), the optimization problem of \( C \)-VLSSVM for classification can be formulated as follows.

\[
\begin{align*}
\text{Minimize} & \quad J(w, b; \xi_i) = \frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^{N} C_i \xi_i^2 \\
\text{Subject to} & \quad y_i(w^T \phi(x_i) + b) = 1 - \xi_i, \quad i = 1, \ldots, N
\end{align*}
\]

Using the Lagrangian theorem, the final solution is similar to Eqs. (15) and (16). The only difference is the value of \( \Omega \) due to the introduction of variable regularization parameter \( C_i \). In the case of \( C \)-VLSSVM classification, the value of \( \Omega \) is calculated by

\[
\Omega_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) + (1/C_i)I
\]

According to Eqs. (19) and (20), the LSSVM classifier algorithm can still be used, except that the value of regularization parameter \( C_i \) is different for every training data point. Thus, computation and simulation procedures of LSSVM classification should be used by a simple modification of the regularization parameter, from a fixed value \( C \) to a variable parameter \( C_i \), in terms of Eqs. (17) and (18).

It is worth noting that the proposed \( C \)-VLSSVM classifier algorithm can provide a solution to an unsolved issue given in Eq. (11) in the paper of Zhou et al.\(^{16}\) In his paper, the optimization problem of LSSVM for classification can be given below.

\[
\begin{align*}
\text{Minimize} & \quad J(w, b; \xi_i) = \frac{1}{2} w^T w + \frac{C_1}{2} \sum_{i=1}^{N_1} \xi_i^2 + \frac{C_2}{2} \sum_{i=1}^{N_2} \xi_i^2 \\
\text{Subject to} & \quad y_i(w^T \phi(x_i) + b) = 1 - \xi_i, \quad i = 1, \ldots, N
\end{align*}
\]

where \( N_1 \) and \( N_2 \) are the number of cases in class 1 and class 2, \( C_1 \) and \( C_2 \) are the different regularization parameters respectively. In Eq. (21), Zhou et al.\(^{16}\) did not give any solution. Comparing with the \( C \)-VLSSVM algorithm proposed in this study, the \( C \)-VLSSVM algorithm does not only have more wide applicability, but also provide an efficient solution using least squares principle.

4. Experiment Design

In this paper, two real-world credit data sets (German and Australian credit data sets) are used to test the performance of the \( C \)-VLSSVM classifier. The two credit data sets are from UCI Machine learning Repository (http://archive.ics.uci.edu/ml/). The German credit data set consists of 1000 instances, which include 700 instances of creditworthy applicants (good) and 300 default instances (bad). Each instance has 24 input variables and one output variable. For the convenience of computation, all input variables are transformed into the numerical-type variables. In the Australian data set, all variables have been transferred to meaningless symbolic data to protect confidentiality. It contains 690 instances, of which 383
instances are of Good Class, which are supposed to be good applicants, and 307 of Bad Class, which are supposed to be bad applicants. Each instance has 14 explanatory variables and 1 observed variable.

For comparison purpose, several commonly used credit risk evaluation models, such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logit analysis (LogA), $k$-nearest neighbor (KNN), artificial neural networks (ANN), and least squares support vector machine (LSSVM) classifier are also applied.

In the experiments, we use 5-fold cross validation to test the performance of the model. For each fold, the original data set is split into two parts: training set and testing set. The final results are the average of results of five tests. In addition, the $k$ of KNN is set to ten. In the ANN model, a common three-layer feed-forward neural network (FNN) with sigmoidal hidden layer and linear output layer is employed. The network training algorithm is Leverberg–Marquardt (LM) algorithm and learning goal is 0.0001. For hidden nodes, its numbers depend on the number of input patterns. In LSSVM and $C$-VLSSVM models, RBF kernels are used to perform the classification task, and their parameters $C$ and $\sigma$ are obtained, using the direct search method. All experiments use identical training and testing sets with 5-fold cross validation. In addition, three evaluation criteria, Type I accuracy, Type II accuracy and Total accuracy are used, which are defined as follows:

\[
\text{Type I accuracy} = \frac{\text{number of both observed bad and classified as bad}}{\text{number of observed bad}} \quad (22)
\]

\[
\text{Type II accuracy} = \frac{\text{number of both observed good and classified as good}}{\text{number of observed good}} \quad (23)
\]

\[
\text{Total accuracy} = \frac{\text{number of correct classification}}{\text{the number of evaluation sample}} \quad (24)
\]

In order to test the capability of different models in discriminating potentially defaulting customers from good customers, the following five-step procedure is implemented.

1. **Data preprocessing.** In the modeling of this study, data preprocessing is first performed. All input variables are linearly scaled to $[0, 1]$, as in Eq. (25), to avoid the dominance of attributes with greater numeric values over those with small values. Note that in Eq. (25), $i = 1, 2, \ldots, N$ is the number of training samples and $k = 1, 2, \ldots, m$ denote the number of input variables.

\[
x'_{ik} = \frac{x_{ik} - \min_i \{x_{ik}\}}{\max_i \{x_{ik}\} - \min_i \{x_{ik}\}}, \quad i = 1, 2, \ldots, N; \quad k = 1, 2, \ldots, m \quad (25)
\]

2. **Feature processing.** Instead of conducting feature selection, a feature weighting strategy based on $t$-test is applied on the training data set, which is defined below.

\[
t_k = \frac{|\text{mean}(x_{ik}|y_i = 1) - \text{mean}(x_{ik}|y_i = -1)|}{\sqrt{\text{var}(x_{ik}|y_i = 1)/\text{num}(y_i = 1) + \text{var}(x_{ik}|y_i = -1)/\text{num}(y_i = -1)}} \quad (26)
\]
where \( \text{mean}(\cdot) \) and \( \text{var}(\cdot) \) are the mean and variance of a group of values, and \( \text{num}(\cdot) \) is the number of values. Using Eq. (26), the weight to each input feature can be represented by

\[
p_k = \frac{t_k}{\sum_{k=1}^{m} t_k}, \quad k = 1, 2, \ldots, m.
\]

(27)

(3) **Data division.** The sample data is randomly divided into two parts, training set and testing set, which are used for model training and model evaluation purposes, respectively.

(4) **Parameter selection.** This step is to specify model parameters. For example, in the C-VLSSVM model with RBF kernel function, two important parameters, the constant value \( C \) of the variable regularization parameter, and kernel parameter \( \sigma^2 \) should be specified before training.

(5) **Model training and testing.** In this step, the models are trained using training samples, and are applied to testing samples. The final performance of the models is evaluated.

Using the above five-step procedure, simulation experiments with different classifiers can be conducted normally, for comparison purpose.

### 5. Experimental Results

#### 5.1. Dataset 1: German credit data

After using the above data and implementing the computational procedure, computational results for the German credit data set are obtained, as shown in Table 1. Note that classification values of LDA, QDA, LogA, KNN, ANN and LSSVM are from Table 5 in Ref. 16.

As can be seen from Table 1, several interesting findings can be observed.

First of all, performance of the two soft computing models (LSSVM and C-VLSSVM) is generally better than hard computing classifiers, according to different performance measures, indicating that soft computing models have advantages over...
hard computing models. The main reason is that there is more flexibility in soft computing models. Interestingly, another soft computing technique, artificial neural network (ANN), performs worse in credit risk classification than in other classification tasks. One important reason may be that there is an overfitting problem when applying ANN to credit risk classification task.

Second, from the viewpoint of Total accuracy, the $C$-VLSSVM is superior to other classifiers listed in this study, followed by LSSVM classifier; QDA performs the worst. The main reason is that the variable regularization parameter increases the classification capability of LSSVM in capturing the complex characteristics of bad customers, while QDA is only a linear quadratic discriminant analysis model.

Third, there are some conflicts between classification results of Type I accuracy and Type II accuracy. Generally speaking, Type II accuracy is much better than Type I accuracy. But from the viewpoint of Type I accuracy, the performance of LDA is the best, while for Type II accuracy, the $C$-VLSSVM classifier performs the best. The main reason is that it is more difficult to classify bad customers from out of credit applicants, due to the complexity of credit risk.

Finally, there are some interesting observations about the performance of hard computing classifiers in terms of Type I accuracy and Type II accuracy. For example, KNN ranks the second in terms of Type II accuracy, while it ranks the last in terms of Type I accuracy. On the contrary, QDA ranks the second in terms of Type I accuracy while it ranks the last in terms of Type II accuracy. The reasons leading to these results are not known, which is worth exploring further.

From results in terms of the three measures (i.e. Type I accuracy, Type II accuracy and Total accuracy) shown in Table 1, we can judge which model is the best and which model is the worst. However, it is not clear what the differences between good and bad models are. For this purpose, McNemar’s test\textsuperscript{19} is conducted to examine whether the proposed $C$-VLSSVM classifier significantly outperforms the other six classifiers listed in this paper.

As a non-parametric test for two related samples, McNemar’s test is particularly useful for before–after measurement of the same subjects.\textsuperscript{20} Taking the Total accuracy results from Table 1, Table 2 shows the results of the McNemar’s test for the German credit data set, statistically comparing the performance of the seven classifiers. Note that the results listed in Table 2 are Chi squared values, and $p$ values are in brackets.

In terms of the results reported in Table 2, some important conclusions can be drawn in terms of McNemar’s statistical test.

(1) The proposed $C$-VLSSVM classifier outperforms the $k$-nearest neighbor (KNN), artificial neural network ANN, and quadratic discriminant analysis (QDA) model at 1% statistical significance level, and outperforms linear discriminant analysis (LDA) at 5% significance level. However, the proposed $C$-VLSSVM classifier does not significantly outperform the LSSVM and the Logit analysis (LogA) model. These results are consistent with those of Table 1.
Table 2. McNemar’s test of different models for the German credit data set.

<table>
<thead>
<tr>
<th>Model</th>
<th>LSSVM</th>
<th>LogA</th>
<th>LDA</th>
<th>KNN</th>
<th>ANN</th>
<th>QDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-VLSSVM</td>
<td>1.0110</td>
<td>1.3120</td>
<td>6.2170</td>
<td>12.051</td>
<td>18.721</td>
<td>26.124</td>
</tr>
<tr>
<td></td>
<td>[0.3146]</td>
<td>[0.2520]</td>
<td>[0.0126]</td>
<td>[0.0005]</td>
<td>[0.0001]</td>
<td>[0.0001]</td>
</tr>
<tr>
<td>LSSVM</td>
<td>0.0090</td>
<td>2.0860</td>
<td>5.8850</td>
<td>10.796</td>
<td>16.6050</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0.9258]</td>
<td>[0.1487]</td>
<td>[0.0153]</td>
<td>[0.0010]</td>
<td>[0.0001]</td>
<td></td>
</tr>
<tr>
<td>LogA</td>
<td>1.7020</td>
<td>5.2300</td>
<td>9.9050</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0.1920]</td>
<td>[0.0222]</td>
<td>[0.0016]</td>
<td></td>
<td>[0.0001]</td>
<td></td>
</tr>
<tr>
<td>LDA</td>
<td>0.8850</td>
<td>3.2550</td>
<td>6.7500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0.3469]</td>
<td>[0.0712]</td>
<td>[0.0094]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>0.6770</td>
<td>2.6190</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0.4107]</td>
<td>[0.1056]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>0.5710</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[0.4498]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(2) The LSSVM classifier and LogA model outperform ANN and LDA at 1% significance level and outperform KNN at 5% significance level. But the McNemar’s test does not conclude that the LSSVM classifier and LogA model perform better than the LDA model.

(3) In hard computing models, LDA is better than QDA and ANN at 1% and 10% significance level, respectively, but LDA is not better than KNN in the statistical sense.

(4) Comparing with KNN, ANN and QDA models, it is easy to find that there are no significant differences among the three models. All findings are consistent with results reported in Table 1.

5.2. Dataset 2: Australian credit data

Similar to the German credit data set, classification results for the Australian credit data set can be obtained, as shown in Table 3. Accordingly, Table 4 reports the results of McNemar’s test for Australian credit data set, to statistically compare the performance of the seven classifiers.

From Tables 3 and 4, we can have the following findings.

First, in terms of the three criteria, the proposed C-VLSSVM classifier obtained the best performance among the seven classifiers listed in this paper, suggesting that the C-VLSSVM classifier is an effective soft computing technique for credit risk evaluation. Furthermore, the results of McNemar’s test have also confirmed this conclusion because McNemar’s test shows C-VLSSVM classifier outperforms other classifiers at 1% significance level.

Second, different from the German data set, the worst classifier for the Australian data set is KNN, for Type II accuracy and Total accuracy, but QDA is still the worst classifier for Type I accuracy. Furthermore, McNemar’s test shows that the KNN classifier is worse than other classifiers at 1% significance level, in terms of Total accuracy.
Table 3. Performance comparison of different models for the Australian credit data set.

<table>
<thead>
<tr>
<th>Models</th>
<th>Type I accuracy</th>
<th>Type II accuracy</th>
<th>Total accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Rank</td>
<td>% Rank</td>
<td>% Rank</td>
</tr>
<tr>
<td>LDA</td>
<td>80.94</td>
<td>92.18</td>
<td>85.94</td>
</tr>
<tr>
<td>QDA</td>
<td>66.12</td>
<td>91.38</td>
<td>80.14</td>
</tr>
<tr>
<td>LogA</td>
<td>85.90</td>
<td>86.32</td>
<td>86.09</td>
</tr>
<tr>
<td>KNN</td>
<td>81.72</td>
<td>54.40</td>
<td>69.57</td>
</tr>
<tr>
<td>ANN</td>
<td>72.56</td>
<td>83.61</td>
<td>78.94</td>
</tr>
<tr>
<td>LSVM</td>
<td>85.12</td>
<td>89.25</td>
<td>86.96</td>
</tr>
<tr>
<td>C-VLSSVM</td>
<td>88.83</td>
<td>93.29</td>
<td>91.88</td>
</tr>
</tbody>
</table>

Table 4. McNemar’s test of different models for the Australian credit data set.

<table>
<thead>
<tr>
<th>Model</th>
<th>LSSVM</th>
<th>LogA</th>
<th>LDA</th>
<th>QDA</th>
<th>ANN</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSSVM</td>
<td>[0.0063]</td>
<td>0.1340 [0.7139]</td>
<td>0.0000 [1.0000]</td>
<td>6.5000 [0.0108]</td>
<td>0.1740 [0.6768]</td>
<td>11.538 [0.0007]</td>
</tr>
<tr>
<td>C-VLSSVM</td>
<td>7.4590 [0.0016]</td>
<td>10.4580 [0.0012]</td>
<td>33.161 [0.0023]</td>
<td>38.527 [0.0004]</td>
<td>88.004 [0.0001]</td>
<td>88.004 [0.0001]</td>
</tr>
<tr>
<td>LogA</td>
<td>0.1340 [0.7139]</td>
<td>0.1930 [0.6608]</td>
<td>9.3220 [0.0004]</td>
<td>12.409 [0.0001]</td>
<td>47.203 [0.0001]</td>
<td>47.203 [0.0001]</td>
</tr>
<tr>
<td>LDA</td>
<td>0.0000 [1.0000]</td>
<td>6.8670 [0.0088]</td>
<td>9.5600 [0.0020]</td>
<td>41.729 [0.0001]</td>
<td>41.729 [0.0001]</td>
<td>41.729 [0.0001]</td>
</tr>
<tr>
<td>QDA</td>
<td>6.5000 [0.0108]</td>
<td>9.1280 [0.0025]</td>
<td>40.860 [0.0001]</td>
<td>14.9390 [0.0001]</td>
<td>14.9390 [0.0001]</td>
<td>14.9390 [0.0001]</td>
</tr>
<tr>
<td>ANN</td>
<td>0.1740 [0.6768]</td>
<td>14.9390 [0.0001]</td>
<td>11.538 [0.0007]</td>
<td>11.538 [0.0007]</td>
<td>11.538 [0.0007]</td>
<td>11.538 [0.0007]</td>
</tr>
</tbody>
</table>

Third, similar to the German data set, the ANN performs much worse than LSSVM and C-VLSSVM classifiers in terms of the three evaluation criteria. In the statistical sense, C-VLSSVM, LSSVM, LogA and LDA perform better than ANN at 1% significance level. The possible reason is that ANN classifiers suffer from over-fitting or local minima problems.

In addition, according to the results from Tables 1–4, several important conclusions can be drawn, which are summarized as follows.

1. Generally, the performance of the German data set is worse than that of the Australian data set. There are two possible reasons. On the one hand, the credit market in Germany is more complex than Australia. On the other hand, there is more nonlinearity in the German data set than in the Australian data set.

2. According to the results of Type II accuracy and Total accuracy, the C-VLSSVM model performs the best in both credit data sets. This implies the strong classification capability of the proposed C-VLSSVM model, in credit risk classification. However, from the viewpoint of Type I accuracy, the C-VLSSVM model is the best of all the listed approaches, for the Australian data set. But
in the German data set, the LDA model is the best approach. The reason for this phenomenon is not known, and is worth exploring further.

(3) In the empirical results of the above two data sets, it is not hard to find that C-VLSSVM model can effectively improve the credit risk classification performance if partial prior knowledge is provided. This finding is consistent with the information theory. That is, the more we know, the less uncertainty we get. This confirms that the classification algorithms incorporating prior knowledge will obtain greater classification power.

6. Concluding Remarks

In this paper, a modified LSSVM classifier, called the C-variable LSSVM (C-VLSSVM) classifier, is proposed for credit risk classification. In terms of empirical results, we can find that across different models, for the test cases of two main credit data sets, in terms of three different evaluation criteria, the proposed C-VLSSVM classifier performs the best. In the presented two cases, Total accuracy is the highest, indicating that the proposed C-variable LSSVM classifier can be used as a promising tool for credit risk analysis. This implies that the proposed C-variable LSSVM classifier has great potential in its application to other classification problems.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments and suggestions. Their comments helped to improve the quality of the paper immensely. This work is partially supported by grants from the National Natural Science Foundation of China/RGC Joint Research Scheme, the Knowledge Innovation Program of the Chinese Academy of Sciences and the Open Fund Project of Key Research Institute of Philosophies and Social Sciences in Hunan Universities.

References