Least Squares One-class Support Vector Machine on Fuzzy Set

Jingjing Zhang\(^{1}\) and Ping Zhong\(^{1}\)*

\(^{1}\)College of Science, China Agricultural University, 100083 Beijing, China
zhjingjing09@163.com, *Corresponding author: zping@cau.edu.cn

Abstract

In this paper, we formulate a least squares version of the one-class support vector fuzzy machine (LS one-class SVFM) which is combined with the fuzzy set theory. The parameters in the proposed algorithm, such as weight vector and bias term, are fuzzy numbers. Our model only needs to solve a system of linear equations, instead of a complex quadratic programming problem (QPP) solved in one-class SVFM. Our experiments on publicly available datasets indicate that our model has comparable classification accuracy to that of the one-class SVFM but with remarkably less computational time.

Keywords: Support vector machine; Fuzzy system; Least squares; Class imbalanced; One-class SVM

1. Introduction

Learning from imbalanced or skewed datasets is an important issue in machine learning research. There are many class imbalanced problems in real-world applications [1,2,3]. For two-class imbalanced problems, one class is made up of a large number of samples while the other one is made up of only a few samples. The traditional algorithms are not suitable to deal with imbalanced dataset, since the separating hyperplane tends to the minority samples, which makes them misclassify easily. However, the minority is always the more important class. Improving the accuracy of minority class is a meaningful research.

To deal with the imbalanced dataset, there are probably two types of approaches. One type mainly contains under-sampling from majority class [4,5] or over-sampling from minority class [6,7]. The other type is to modify the learning algorithms such as adjusting the cost method [8,9], kernel-based method [10], margin calibration [11]. One-class support vector machine (one-class SVM) is widely used in the heavily class imbalanced problems, such as the context of information retrieval [12], hidden information [13], outlier detection [14] and so on. The one-class SVM aims to distinguish one class of samples from the rest. It constructs a maximal margin hyperplane to separate the training samples from the origin [15,16]. Many efforts have been made to reduce the time complexity or to raise the classification accuracy. The authors merged the rough set theory with one-class SVM to enhance the performance of the classifier [17]. In [18], the author proposed the least squares one-class SVM reducing the training time. In [19], the authors proposed the reduced least squares one-class SVM in empirical feature space.

In some systems, we consider a fuzzy structure when the available information is uncertain. The fuzzy structure is described by a fuzzy function whose parameters are given by fuzzy set [20]. Since the excellent of the SVM, many researches have been made apply the fuzzy set theory to SVM for classification and regression problems. In [21], the author gave the model of SVM with fuzzy set theory for two class problem. In [22], the authors incorporated the concept of fuzzy set theory into the support vector regression (SVR). In [23], the authors proposed the regularized LS SVR in fuzzy set theory.
Recently, Hao proposed the one-class SVM with fuzzy model, called one-class SVFM [24]. It constructs a fuzzy hyperplane to distinguish the target class from the rest. An effective combination of unseen samples and fuzzy set theory is useful to find a fuzzy structure in an evaluation system. However, the one-class SVFM needs to solve a time-consuming quadratic programming problem (QPP).

In this paper, we propose a least squares one-class support vector fuzzy machine, called LS one-class SVFM. Different from the one-class SVFM, the proposed algorithm introduces a quadratic loss function in objective function and equalities in the constraints. To solve the optimization problem of the proposed algorithm only requires solving a set of linear equations, instead of solving a complex QPP. Thus our algorithm owns low time complexity. Computational comparisons for linear kernel and nonlinear kernel between the proposed LS one-class SVFM and one-class SVFM on both generalization performance and training time have been made on several benchmark datasets. The experimental results show that the proposed method has a fast training speed, and it also acquires comparable accuracies with the one-class SVFM.

This paper is organized as follows. Section 2 briefly dwells on the one-class SVFM. Section 3 gives our least squares one-class SVFM. Experiments are presented in Section 4. Section 5 concludes this paper.

2. The One-Class SVFM

In this paper, the training samples are denoted by a set \( T = \{x_1, x_2, ..., x_l\} \), where \( x_i \in \mathbb{R}^n \).

The one-class SVFM is constructed based on one-class SVM and fuzzy theory [20]. It searches for a fuzzy hyperplane that separates the target class from the fuzzy origin. The parameters, such as the weight vector and bias term, are fuzzy numbers. For computational simplicity, the symmetric triangular fuzzy numbers are chosen [20]. The fuzzy hyperplane is described as:

\[
f(x) = W^T x + B, \quad \text{for } W \in T(R)^n, \quad B \in T(R)^n
\]

(1)

where \( T(R) \) denotes the space of all symmetric triangular fuzzy numbers. The fuzzy weight vector \( W = (w, c) \) is described by center \( w \) and width \( c \), while \( B = (b, d) \) is the fuzzy bias term described by center \( b \) and width \( d \). For a sample \( x \), \( f(x) = W^T x + B \) is a symmetric triangular fuzzy number with center \( w^T x + b \) and width \( c^T |x| + d \), where \( |x| = (|x_1|, ..., |x_n|) \).

To find \( f(x) \), the one-class SVFM considers the following QPP

\[
\min_{w,b,c,x_i} \frac{1}{2} \|w\|^2 + b + M \left( \frac{1}{2} \|x\|^2 + d \right) + C \sum_{i=1}^{l} \mu_i \xi_i
\]

s.t. \( W^T x_i + B \geq \Theta - \xi_i, \quad i = 1, ..., l \)

(2)

where \( 0 < \mu_i \leq 1 \) is the fuzzy membership. The parameter \( C > 0 \) is to balance the model complexity and the empirical risk. The symbol “\( \geq \)" denotes the fuzzy larger than, and \( \Theta \) represents the fuzzy number with center zero and width \( O_w \). The parameters \( 0 < M, O_w \leq 1 \) control the fuzzy range. Minimization the term \( \|w\|^2 \) is equivalent to the maximization of the margin between the target class from the origin. The slack variable
\( \xi_i \) represents the error caused by a training sample \( x_i \) with respect to the fuzzy hyperplane.

\[
\frac{1}{2} \left( \|e\|^2 + d \right)
\]
represents the vagueness of the model. The more vague of model, the more inaccurate results obtain. Problem (2) can be written as

\[
\min_{w, c, \xi, \varepsilon} \frac{1}{2} \|e\|^2 + b + M \left( \frac{1}{2} \|e\|^2 + d \right) + C \sum_{i=1}^{l} \mu_i (\xi_{i1} + \varepsilon_{i2})
\]

s.t. \( (w^T x_i + b) + (c^T |x_i| + d) \geq O_w - \xi_{i1} \),

\( (w^T x_i + b) - (c^T |x_i| + d) \geq -O_w - \xi_{i2} \),

\( d \geq 0, \xi_{i1}, \xi_{i2} \geq 0, i = 1, \ldots, l \)

where \( |\cdot| \) is to ensure the width hyperplane \( c^T |x_i| + d > 0 \).

The dual problem (2) can be formulated as follows by KKT conditions.

\[
\max_{\alpha_i, \alpha_{i2}} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_{ij} + \alpha_{i2}) (\alpha_{ij} + \alpha_{i2}) x_i^T x_j
\]

\[
- \frac{1}{2M} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_{ii} - \alpha_{i2}) (\alpha_{ij} - \alpha_{i2}) |x_i| |x_j| + \sum_{i=1}^{l} (\alpha_{ii} - \alpha_{i2}) O_w
\]

s.t. \( \sum_{i=1}^{l} (\alpha_{ii} + \alpha_{i2}) = 1 \)

\( \sum_{i=1}^{l} (\alpha_{ii} - \alpha_{i2}) \leq M \)

\( 0 \leq \alpha_{ii} \leq C \mu_i, 0 \leq \alpha_{i2} \leq C \mu_i, i = 1, \ldots, l \)

Once the solution of \( (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{i2}, \alpha_{i2})^T \) is obtained, the weight vectors \( w \) and \( c \) can be obtained as follows.

\[
w = \sum_{i=1}^{l} (\alpha_{i1} + \alpha_{i2}) x_i, \quad c = \frac{1}{M} \sum_{i=1}^{l} (\alpha_{ii} - \alpha_{i2}) |x_i|.
\]

For some \( \alpha_i \in (0, C \mu_i) \) and \( \alpha_j \in (0, C \mu_j) \), according to the KKT conditions, we have bias terms \( b \) and \( d \).

\[
b = -\frac{1}{2} \left( w^T x_i + w^T x_j + c^T |x_i| - c^T |x_j| \right)
\]

\[
d = -\frac{1}{2} \left( w^T x_i - w^T x_j + c^T |x_i| + c^T |x_j| - 2O_w \right)
\]

For a new sample \( x \), we evaluate it falls on which side of the hyperplane by defining the fuzzy partial ordering relation. Given two symmetric triangular fuzzy numbers \( P = (m_p, m_p) \) and \( Q = (m_q, m_q) \), we have that

\[
R_{\lhd} P = R(P, Q) = \begin{cases} 
1 & \text{if } A > 0 \& B > 0, \\
0 & \text{if } A < 0 \& B < 0, \\
0.5(1 + (A + B) / \max(|A|, |B|)) & \text{if } o.w.
\end{cases}
\]

where \( A = (m_p + c_p) - (m_q + c_q) \) and \( B = (m_p - c_p) - (m_q - c_q) \).
For (5), we observe that
\[ R(P, Q) = \begin{cases} > 0.5 & \text{if } m_P > m_Q, \\ = 0.5 & \text{if } m_P = m_Q, \\ < 0.5 & \text{if } m_P < m_Q. \end{cases} \]  

Hence, the decision function of the one-class SVFM can be expressed as
\[ f(x_i) = R_{sto}(W^T x_i + B) = R(W^T x_i + B, \Theta). \]  

3. Least Squares One-Class Support Vector Fuzzy Machine

3.1. Linear Kernel Case

Inspired by the one-class SVFM, we propose a least squares one-class SVFM. The LS one-class SVFM can be derived by solving the following optimization problem
\[ \min \frac{1}{2} \|w\|^2 + b + M \left( \frac{1}{2} \|c\|^2 + d^2 \right) + \frac{C}{2} \sum_{i=1}^{l} \mu_i \xi_i^2 \]
\[ s.t. \quad W^T x_i + B = \Theta - \xi_i, \quad i = 1, ..., l \]  

where \( 0 < \mu_i \leq 1 \) is the fuzzy membership. The parameter \( C > 0 \) is the regularization parameter which determines the penalty weight, and \( \xi \) is the slack vector. The parameter \( 0 < M \leq 1 \) controls the fuzzy origin. The symbol \( \mu \) represents the fuzzy equal.

We rewrite the primal problem (8) as
\[ \min \frac{1}{2} \|w\|^2 + b + M \left( \frac{1}{2} \|c\|^2 + d^2 \right) + \frac{C}{2} \sum_{i=1}^{l} \mu_i \xi_i^2 \]
\[ s.t. \quad W^T x_i + B = \Theta + \xi_i, \quad i = 1, ..., l \]
\[ (w^T x_i + b) + (c^T |x_i| + d) = O_w - \xi_i, \quad i = 1, ..., l \]
\[ (w^T x_i + b) - (c^T |x_i| + d) = -O_w - \xi_i, \quad i = 1, ..., l \]  

In problem (9), the first term \( \frac{1}{2} \|w\|^2 \) in the objective function is to control the model complexity for finding the fuzzy hyperplane. \( \frac{1}{2} \left( \|c\|^2 + d^2 \right) \) is the term which characterizes the vagueness of the model. The slack variables \( \xi_i \) represent the loss caused by the fuzzy hyperplane to the fuzzy region. \( 0 < O_w \leq 1 \) represents width of fuzzy region \( \Theta \). Different from (3), we reformulate our method by using a quadratic loss function and equality constraints. In order to solve problem (9), we construct its Lagrangian function
\[ L = \frac{1}{2} \|w\|^2 + b + M \left( \frac{1}{2} \|c\|^2 + d^2 \right) + \frac{C}{2} \sum_{i=1}^{l} \mu_i \xi_i^2 \]
\[ - \sum_{i=1}^{l} \alpha_i \left( w^T x_i + b + c^T |x_i| + d - O_w + \xi_i \right) \]
\[ - \sum_{i=1}^{l} \beta_i \left( w^T x_i + b - c^T |x_i| - d + O_w + \xi_i \right) \]  

Online Version Only.

Book made by this file is ILLEGAL.
where $\alpha = (\alpha_1, \ldots, \alpha_l)^T$ and $\beta = (\beta_1, \ldots, \beta_l)^T$ are the Lagrangian multiplier vectors. Differentiating $L$ with respect to $w, c, b, d, \xi_i, \xi_{2i}, \alpha, \beta$ and setting the results to zero, we obtain

$$
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} (\alpha_i + \beta_i) x_i = 0 \implies w = \sum_{i=1}^{l} (\alpha_i + \beta_i) x_i
$$

$$
\frac{\partial L}{\partial c} = Mc - \sum_{i=1}^{l} \alpha_i \mid x_i \mid + \sum_{i=1}^{l} \beta_i \mid x_i \mid = 0 \implies c = \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i) \mid x_i \mid
$$

$$
\frac{\partial L}{\partial b} = 1 - \sum_{i=1}^{l} \alpha_i - \sum_{i=1}^{l} \beta_i = 0 \implies \sum_{i=1}^{l} (\alpha_i + \beta_i) = 1
$$

$$
\frac{\partial L}{\partial d} = Md - \sum_{i=1}^{l} \alpha_i + \sum_{i=1}^{l} \beta_i = 0 \implies d = \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i)
$$

$$
\frac{\partial L}{\partial \xi_{ii}} = C\mu_i \xi_{ii} - \alpha_i = 0 \implies \xi_{ii} = \frac{\alpha_i}{C\mu_i}
$$

$$
\frac{\partial L}{\partial \xi_{2ii}} = C\mu_i \xi_{2ii} - \beta_i = 0 \implies \xi_{2ii} = \frac{\beta_i}{C\mu_i}
$$

$$
\frac{\partial L}{\partial \alpha_i} = w^T x_i + b + c^T |x_i| + d - O_x + \xi_{ii} = 0
$$

$$
\frac{\partial L}{\partial \beta_i} = w^T x_i + b - c^T |x_i| - d + O_x + \xi_{2ii} = 0
$$

From the above conditions for optimality, $b, \alpha$ and $\beta$ can be solved by following linear equations

$$
\begin{bmatrix}
0 & I \\
I^T & K + \frac{K}{M} + E + V & \frac{K}{M} + \frac{K}{M} & \frac{K}{M} + \frac{K}{M} + \frac{E}{M} + V
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha \\
\beta
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
O_w \\
O_w
\end{bmatrix}
$$

where $K = (K_{ij})_{ij}$ is a matrix with $K_{ij} = x_i^T x_j$, and $K1 = (K1_{ij})_{ij}$ is also a matrix with $K1_{ij} = x_i^T \mid x_j \mid$. $\alpha$ and $\beta$ denote the column vectors of Lagrangian multipliers. $I$ represents the all one row vector of $n$ dimension. $E$ denotes the identity matrix. $V = \text{diag}\left(\frac{1}{C\mu_1}, \ldots, \frac{1}{C\mu_l}\right)$ is a diagonal matrix. Utilizing the solution of (19), the optimal hyperplane of our algorithm can be expressed as

$$
f(x) = W^T x + B
$$

where $W = (\sum_{i=1}^{l} (\alpha_i + \beta_i) x_i, \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i) \mid x_i \mid)$ and $B = (b, \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i))$.

3.2. Nonlinear Kernel Case

In order to extend our results to the nonlinear case, we consider the kernel-generated fuzzy hyperplane as

$$
f(x) = W^T \varphi(x) + B
$$
where $\varphi(\cdot)$ is to map a dataset $\chi$ to high dimensional feature space $H$, and $k(\cdot, \cdot)$ is the kernel function. Similar to the linear case, the nonlinear LS one-class SVFM can be derived through solving the following optimization problem:

$$\min \frac{1}{2} \|w\|^2 + b + M (\frac{1}{2} \|e\|^2 + d^2) + \frac{C}{2} \sum_{i=1}^{l} \mu_i (\xi_i^2 + \xi_{2i}^2)$$

s.t. $(w^T \varphi(x_i) + b) + (c^T \varphi(x_i) + d) = O_w - \xi_i$, $i = 1, ..., l$

$(w^T \varphi(x_i) + b) - (c^T \varphi(x_i) + d) = -O_w - \xi_{2i}$, $i = 1, ..., l$

Similar with the linear case, the solution of problem (22) is

$$\begin{pmatrix}
0 & I \\
I^T & K + \frac{1}{M} + V & I \\
I^T & K - \frac{1}{M} - V & K - \frac{1}{M} - E \\
\end{pmatrix}
\begin{pmatrix} b \\
\alpha \\
\beta \\
\end{pmatrix} =
\begin{pmatrix} 1 \\
-O_w \\
O_w \\
\end{pmatrix}$$

where $I = (1, 1, ..., 1)$, $V = \text{diag} \{\frac{1}{C\mu_1}, ..., \frac{1}{C\mu_l}\}$, and $E$ denotes the identity matrix.

$K = (K_{ij})_{l \times l}$ is the kernel matrix with $K_{ij} = \varphi(x_i)^T \varphi(x_j)$, and $K = (K_{1i})_{l \times l}$ is also the kernel matrix with $K_{1i} = \varphi(x_i)^T \varphi(x_i)$.

Utilizing the solution of (23), the optimal hyperplane of nonlinear LS one-class SVFM is expressed as

$$(x) = W^T \varphi(x) + B$$

where $W = (\sum_{i=1}^{l} (\alpha_i + \beta_i) \varphi(x_i), \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i) \varphi(x_i))$ and $B = (b, \frac{1}{M} \sum_{i=1}^{l} (\alpha_i - \beta_i))$.

4. Experiments

To check the validity of the proposed LS one-class SVFM in dealing with the class imbalanced problem, we compare it with the one-class SVFM on several benchmark datasets [25]. Both the algorithms are implemented on MATLAB 2012a under Windows 7 running on a PC. We utilize “quadprog.m” for one-class SVFM, “inv.m” for LS one-class SVFM. The parameters in the linear case, $(C, O_w, M)$ are employed. In the nonlinear case, gaussian kernel function $K(x, y) = \exp(-p \|x - y\|^2)$ is considered for all datasets, and the parameters $(p, C, O_w, M)$ are employed. We select the values of parameters $O_w, M \in \{0.1, 0.2, ..., 1\}$ and $(p, C) \in \{2^i | i = -10, -9, ..., 10\} \times \{10^i | i = -4, -3, ..., 5\}$. To search for the optimal parameters, we derive $(C, O_w, M)$ linear case and $(p, C, O_w, M)$ for nonlinear case on each training dataset by cross-validation approach. For LS one-class SVFM, since the optimal hyperplane is a distance function, we discriminate a test sample $x$ as positive (i.e., it belongs to the target class) if it satisfies $\varepsilon^- \leq f(x) \leq \varepsilon^+$, where $\varepsilon^+ = \max_k f(x_k)$, $\varepsilon^- = \min_k f(x_k)$ with $x_k$ being a training sample.
Table 1. Description of Seven Data Sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Minority</th>
<th>Majority</th>
<th>feature</th>
<th>Data size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>29</td>
<td>185</td>
<td>9</td>
<td>214</td>
</tr>
<tr>
<td>Wine</td>
<td>59</td>
<td>119</td>
<td>13</td>
<td>178</td>
</tr>
<tr>
<td>Vehicle</td>
<td>168</td>
<td>428</td>
<td>18</td>
<td>596</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>239</td>
<td>444</td>
<td>10</td>
<td>683</td>
</tr>
<tr>
<td>Segment</td>
<td>323</td>
<td>1879</td>
<td>18</td>
<td>2202</td>
</tr>
<tr>
<td>Abalone</td>
<td>391</td>
<td>3786</td>
<td>7</td>
<td>4177</td>
</tr>
<tr>
<td>fourclass</td>
<td>305</td>
<td>550</td>
<td>2</td>
<td>855</td>
</tr>
</tbody>
</table>

We evaluate the performance of LS one-class SVFM and one-class SVFM on seven UCI datasets, which consist of Glass, Wine, Vehicle, Breast Cancer, Segment, Abalone, and fourclass. In these datasets, Vehicle, Breast Cancer, and fourclass are two-class imbalanced datasets. The minority of the dataset is regarded as the positive class, and the majority one is regarded as the negative class. The others are multi-class datasets. For Glass, Wine, Segment and Abalone, we use the 6th, 1st, 1st and 7th class as the positive class, respectively. Table 1 shows the detail descriptions of these datasets. Here several performance criteria based on the confusion matrix are utilized to give a more comprehensive and objective assessment. The confusion matrix consists of true positive (TP), false positive (FP), false negative (FN) and true negative (TN). To evaluate the performance of the algorithms on imbalanced datasets, we take sensitivity, specificity, G-mean, AUC and training time as performance criteria as follows.

(i) sensitivity = \[ \frac{TP}{TP + FN} \]

(ii) specificity = \[ \frac{TN}{TN + FP} \]

(iii) G-mean = \[ \sqrt{\text{sensitivity} \times \text{specificity}} \]

Sensitivity and specificity are defined as the accuracies on the positive class and the negative class, respectively. The value of G-mean is high when both sensitivity and specificity values are high. Higher value of G-mean implies that the algorithm has better performance. The area under the curve of receiver operating characteristics defined as AUC. It is used to measure the algorithm performance in a quantitative manner. The larger AUC, the better the algorithm performs. Note that all the samples are normalized as in \([0,1]\) before learning.

We give the criterion to associate the fuzzy membership to each sample [20,26]. Given the training samples \((x_1,...,x_l)\). Let the mean of target class as \(x_{\text{mean}}\), and the radius of the target class is defined as \(r_{\text{target}} = \max_i |x_i - x_{\text{mean}}|, i = 1,...,l\). The fuzzy membership \(\mu_i\) is defined as \(\mu_i = 1 - \frac{|x_i - x_{\text{mean}}|}{(r_{\text{target}} + \delta)}, \text{ where } \delta > 0 \) is used to avoid \(\mu_i = 0\).
Table 2. Results of Linear Classifiers on Benchmark Datasets. I and II Stand for One-Class SVFM and LS One-Class SVFM Respectively

<table>
<thead>
<tr>
<th>Data set</th>
<th>Alg</th>
<th>sen</th>
<th>spe</th>
<th>G-mean</th>
<th>AUC</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>I</td>
<td>0.5714</td>
<td>0.6605</td>
<td>0.6143</td>
<td>0.6286</td>
<td>0.0130</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.6667</td>
<td>0.6995</td>
<td>0.6829</td>
<td>0.6831</td>
<td>0.0001</td>
</tr>
<tr>
<td>Wine</td>
<td>I</td>
<td>0.8494</td>
<td>0.4790</td>
<td>0.6378</td>
<td>0.6642</td>
<td>0.0671</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.7714</td>
<td>0.7613</td>
<td>0.7664</td>
<td>0.7577</td>
<td>0.0002</td>
</tr>
<tr>
<td>Vehicle</td>
<td>I</td>
<td>0.9909</td>
<td>0.3458</td>
<td>0.5854</td>
<td>0.6684</td>
<td>0.3017</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.9644</td>
<td>0.5075</td>
<td>0.4082</td>
<td>0.7359</td>
<td>0.0008</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>I</td>
<td>0.8640</td>
<td>0.9851</td>
<td>0.9226</td>
<td>0.9245</td>
<td>1.6106</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.8940</td>
<td>1.0000</td>
<td>0.9455</td>
<td>0.9470</td>
<td>0.0005</td>
</tr>
<tr>
<td>Segment</td>
<td>I</td>
<td>0.8865</td>
<td>0.3035</td>
<td>0.5187</td>
<td>0.6034</td>
<td>1.6257</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.9815</td>
<td>0.6621</td>
<td>0.8061</td>
<td>0.8218</td>
<td>0.0019</td>
</tr>
<tr>
<td>Abalone</td>
<td>I</td>
<td>0.7558</td>
<td>0.4660</td>
<td>0.9236</td>
<td>0.6109</td>
<td>3.7459</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.8100</td>
<td>0.8994</td>
<td>0.7727</td>
<td>0.8547</td>
<td>0.0009</td>
</tr>
<tr>
<td>fourclass</td>
<td>I</td>
<td>0.5588</td>
<td>0.7055</td>
<td>0.6279</td>
<td>0.6321</td>
<td>2.9559</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.9804</td>
<td>0.3824</td>
<td>0.6389</td>
<td>0.6984</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

Table 3. Results of Nonlinear Classifiers on Benchmark Datasets. I and II Stand for One-Class SVFM and LS One-Class SVFM Respectively

<table>
<thead>
<tr>
<th>Data set</th>
<th>Alg</th>
<th>sen</th>
<th>spe</th>
<th>G-mean</th>
<th>AUC</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>I</td>
<td>0.6381</td>
<td>0.9676</td>
<td>0.7857</td>
<td>0.8028</td>
<td>0.0051</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.5143</td>
<td>0.9914</td>
<td>0.7140</td>
<td>0.7528</td>
<td>0.0001</td>
</tr>
<tr>
<td>Wine</td>
<td>I</td>
<td>0.7195</td>
<td>1.0000</td>
<td>0.8110</td>
<td>0.8058</td>
<td>0.0055</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.8961</td>
<td>0.4553</td>
<td>0.6246</td>
<td>0.6657</td>
<td>0.0001</td>
</tr>
<tr>
<td>Vehicle</td>
<td>I</td>
<td>0.9470</td>
<td>0.8042</td>
<td>0.8727</td>
<td>0.8756</td>
<td>0.0171</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.8932</td>
<td>0.9416</td>
<td>0.9171</td>
<td>0.9174</td>
<td>0.0010</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>I</td>
<td>0.9148</td>
<td>0.9577</td>
<td>0.9360</td>
<td>0.9362</td>
<td>0.2814</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.9742</td>
<td>0.6743</td>
<td>0.8105</td>
<td>0.8243</td>
<td>0.0024</td>
</tr>
<tr>
<td>Segment</td>
<td>I</td>
<td>0.9674</td>
<td>0.9989</td>
<td>0.9831</td>
<td>0.9832</td>
<td>1.6257</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.9395</td>
<td>0.9993</td>
<td>0.9689</td>
<td>0.9694</td>
<td>0.0040</td>
</tr>
<tr>
<td>Abalone</td>
<td>I</td>
<td>0.8731</td>
<td>0.6908</td>
<td>0.7766</td>
<td>0.7819</td>
<td>2.9833</td>
</tr>
<tr>
<td></td>
<td>II</td>
<td>0.6600</td>
<td>0.9047</td>
<td>0.7727</td>
<td>0.7824</td>
<td>0.0009</td>
</tr>
</tbody>
</table>
Table 2 and Table 3 report the comparison results of these algorithms with linear and nonlinear kernels, respectively. They mainly contain the sensitivity (sen), specificity (spe), G-mean, the AUC and the training time (trT) on seven datasets. In Table 2, it has been seen that for the sensitivity index, the proposed method has better values than those of one-class SVFM on five datasets except Wine and Vehicle datasets. It indicates that the accuracy of positive class performs better in the proposed algorithm on these datasets. For the specificity, our method performs better on six of seven datasets. For G-mean, our method has better values on six datasets except Vehicle dataset. As for AUC, our method performs better on all datasets. Concerned with nonlinear one-class SVFM and nonlinear LS one-class SVFM, we observe that the values of sensitivity in our method perform better on Wine and Breast Cancer datasets, and it has comparable values with one-class SVFM on Segment and fourclass datasets. For specificity index, our algorithm owns bigger values on five of the seven datasets. As for G-mean, our algorithm has comparable values on five datasets except Wine and Breast Cancer datasets. As for evaluation criteria AUC, our method has comparable values on Segment, Ablaone and fourclass datasets, and has satisfactory results on Vehicle dataset. The training CPU time shown in Table 2 and Table 3 indicate that the proposed LS one-class SVFM is almost dozens of times faster than the one-class SVFM, since the algorithm solves a system of linear equations instead of a complex QPP in the one-class SVFM. In short, the proposed algorithm has comparable generalization performance with the one-class SVFM, and the training time is reduced greatly.

5. Conclusions

In this paper we propose a novel least squares (LS) one-class SVFM to deal with imbalanced problem. The proposed LS one-class SVFM incorporates the concept of fuzzy set theory and the least squares technology into the one-class SVM. Different from solving a dual QPP in one-class SVFM, we only need to solve a system of linear equations in LS one-class SVFM. It makes the LS one-class SVFM a fast speed while the one-class SVFM needs a high training time. Computational results on several UCI benchmark datasets demonstrate that LS one-class SVFM obtains the classification accuracy comparable to that of one-class SVFM algorithm with a fast learning speed.

Acknowledgment

The work is supported by national Natural Science Foundation of China Grant No.11171346 and Chinese Universities Scientific Fund No.2013YJ010.

References


Authors

Jingjing Zhang, he has received her B.S. degree in College of Mathematics and Information Science from Shaanxi Normal University, 2010, and received Master's degree in College of Science from China Agricultural University, China, in 2012. Currently, she is currently pursuing the Ph.D. degree at the College of Science of China Agricultural University. Her research interests include optimization methods, machine learning and data mining.
Ping Zhong. He has received the B.S degree and Master degree in mathematics in 1994 and 1997 both from Qufu Normal University, China, and her Ph. D degree in 2002 from China Agricultural University. Now she is a professor of China Agricultural University. She has published more than 30 papers cited by SCI and EI. Her research interest includes optimization theory, machine learning and support vector machines.