A Novel Support Vector Machine With Its Features Weighted by Mutual Information

Hong-Jie Xing, Ming-Hu Ha, Da-Zeng Tian, and Bao-Gang Hu

Abstract—A novel support vector machine (SVM) with weighted features is proposed. To assign appropriate weights for each feature, a mutual information (MI) based approach is presented. Although the calculation of feature weights may add an extra computational cost, the proposed method generally exhibits better generalization performance over the traditional SVM. The numerical studies on one synthetic and five existing benchmark classification problems confirm the benefits in using the proposed method.

I. INTRODUCTION

Support Vector Machine (SVM) is a machine learning algorithm based on statistical learning theory [1][2][3]. It has drawn much attention in recent years for its better generalization performance. In real-world applications, SVM has been shown higher performances than traditional learning machines [1]. Therefore, SVM has been widely used in pattern recognition [4][5] and function regression [6][7]. The main reason for its promising performance is that SVM can simultaneously minimize prediction error and model complexity [8].

As far as we know, the traditional SVM and its improvement versions [9] are all assumed that all the features of samples in a given training set have equal contributions to construct the optimal separating hyperplane. However, for a certain real-world data set, some features of it may possess more relevances to the classification information, while the other features may have less relevances. Thus, the features with more relevances are more important to train an optimal SVM than those with less relevances. Therefore, it is desirable and feasible to construct a kind of SVM with different features possessing different weights. However, the existing SVMs with weighted features [10][11] are all directly multiplying the weights upon their corresponding features of the given samples. In the paper, the certain weights are absorbed in the quadratic programming to obtain the dual solution of the original optimization problem. Meanwhile, the detailed theoretical deduction is provided. It should be noted that the proposed method is different from the weighted SVMs [12]-[14] because the weights of the latter are assigned to the samples other than their features. Moreover, the main contribution of the weighted SVMs is dealing with the imbalanced data classification.

To determine the weights for each feature towards a given data set, the relevances of these features to the given class labels should be considered. It is well known that mutual information (MI) can be used to measure the relevance between two random variables [15][16]. Moreover, it should be mentioned here that the MI based method for feature selection and feature ranking have achieved promising results [17][18]. Thus, we utilize the mutual information based method to compute the values of weights for each feature, and use the method in literature [17] to estimate the MI between each feature and the given class labels.

The rest of the paper is organized as follows. The traditional SVM is briefly reviewed in Section II and the proposed SVM is discussed in detail in Section III. The approach for determining the weights for the features through the MI based method is described in Section IV. The result concerning experiments are reported based on one synthetic and five benchmark data sets in Section V. Finally, some concluding remarks are given in Section VI.

II. SUPPORT VECTOR MACHINES

In this section, the theory of SVM for classification is briefly reviewed. Given a two-class training set \( \mathcal{X} = \{ (x^{(p)}, y^{(p)}) \}_{p=1}^l \), where \( x^{(p)} \in \mathbb{R}^d \) and \( y^{(p)} \in \{-1, 1\} \) is the class label of the training sample \( x^{(p)} \). To obtain the optimal separating hyperplane, the following problem is considered [3]:

\[
\begin{align*}
\min & \quad \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \\
\text{s.t.} & \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, l
\end{align*}
\]

(1)

where \( C \) is a constant which determines the trade-off between the maximization of margin and the amount of misclassification.

The above optimization problem can be solved by constructing a Lagrange function and transformed into the following dual optimization problem:

\[
\begin{align*}
\min & \quad \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} - \sum_{i=1}^l \alpha_i \\
\text{s.t.} & \quad \sum_{i=1}^l \alpha_i y^{(i)} = 0, \quad i = 1, \ldots, l
\end{align*}
\]
For the choice of kernel functions literatures [3][9]. Moreover, the decision function is the Lagrange multipliers and the training samples plus their class labels as
\[ \hat{\alpha} \hat{y}^{(i)} \hat{x}^{(i)}, \]
where \( x^{(i)} \) is the \( i \)th training sample, while \( y^{(i)} \) and \( \hat{\alpha} \) are its corresponding class labels and Lagrange multiplier, respectively. Moreover, the bias term \( b \) can be determined from the Karush-Kuhn-Tuker condition (3) as follows:
\[ \hat{b} = \frac{1}{N_s} \sum_{i \in S} \left( y^{(i)} - \sum_{j \in S} \alpha_j y^{(j)} (x^{(i)})^T x^{(j)} \right). \]
where \( N_s \) is the total number of support vectors and \( S \) denotes the set of indices of the support vectors.

For a test sample \( x \), its class label can be determined by the following function
\[ f(x) = \text{sign}(\hat{w}^T x + \hat{b}). \]

For the nonlinear SVM, the optimization problem (2) can be generalized for nonlinear kernels as follows:
\[ \min_{\beta} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^{(i)} y^{(j)} K(x^{(i)}, x^{(j)}) - \sum_{i=1}^{l} \alpha_i \]
\[ \text{s.t.} \sum_{i=1}^{l} \alpha_i y^{(i)} = 0, \quad \alpha_i \geq 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, l. \]

III. SUPPORT VECTOR MACHINES WITH WEIGHTED FEATURES

In this section, the proposed SVM with weighted features is formulated. Given a training set \( \{ x^{(p)}, y^{(p)} \}_{p=1}^{l} \), the feature weight vector \( \beta \in \mathbb{R}^d \) for the features of training samples \( x^{(p)} \) is predefined. The method for determining \( \beta = (\beta_1, \beta_2, \ldots, \beta_d)^T \) will be introduced in the following section. However, it should be noticed here that the elements of the feature weight vector \( \beta \) obey the following two conditions:
\[ 0 \leq \beta_k \leq 1 \quad (k = 1, \ldots, d) \]
and
\[ \sum_{k=1}^{d} \beta_k = 1. \]

Based on formula (11), it can be deduced from (10) that \( \beta_k \geq 0 \) (\( k = 1, \ldots, d \)). Therefore, towards the SVM with weighted features, the optimization problem (1) can be re-designed as follows:
\[ \min \frac{1}{2} \beta^T C \sum_{i=1}^{l} \xi_i + \frac{1}{2} \beta^T \beta \]
\[ \text{s.t.} \sum_{i=1}^{l} \alpha_i y^{(i)} [w^T \text{diag}(\beta)x^{(i)} + b] \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, l \]
\[ \sum_{k=1}^{d} \beta_k = 1, \quad k = 1, \ldots, d \]
\[ \beta_k \geq 0, \quad k = 1, \ldots, d. \quad (12) \]

To solve this optimization problem, we construct the Lagrange function as follows:
\[ L(w, \beta, \xi, \lambda) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \]
\[ - \sum_{i=1}^{l} \alpha_i \left( y^{(i)} [w^T \text{diag}(\beta)x^{(i)} + b] - 1 + \xi_i \right) \]
\[ - \sum_{i=1}^{l} \nu_i \xi_i + \gamma \sum_{k=1}^{d} \beta_k - 1 - \sum_{k=1}^{d} \lambda_k \beta_k \quad (13) \]

Therefore, the problem (12) can be solved by finding the saddle point of the above Lagrange function, by taking the derivative of the Lagrangian \( L(w, \beta, \xi, \lambda) \) with respect to the parameters, \( w, \beta, \xi, \lambda \), as follows:
\[ \frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \alpha_i [w^T \text{diag}(\beta)x^{(i)}] \]
\[ = 0 \quad (14) \]
\[ \frac{\partial L}{\partial \beta} = - \sum_{i=1}^{l} \alpha_i y^{(i)} = 0 \quad (15) \]
\[ \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \nu_i = 0 \quad (16) \]
\[ \frac{\partial L}{\partial \beta_k} = - \sum_{i=1}^{l} \alpha_i y^{(i)} w_k x^{(i)} + \gamma - \lambda_k = 0 \quad (17) \]

Moreover, from the Karush-Kuhn-Tucker conditions, the following expression can be obtained
\[ \lambda_k \beta_k = 0. \quad (18) \]

For the proposed SVMs with weighted features, through experiments we observed that the certain weights are always nonzero, i.e., \( \beta_k \neq 0 \) (\( k = 1, \ldots, d \)). Therefore, we can obtain that \( \lambda_k = 0 \) for \( k = 1, \ldots, d \). Then, from equation...
nonlinear kernels as follows:

SVM, the optimization problem (20) can be generalized by the entropy which is defined in terms of probability as

\[ H(Y) = - \sum_y P(y) \log(P(y)), \]

where the base of the logarithm in the formula is taken to be 2.

After giving a continuous random variable \(Y\), the remaining uncertainty of \(Y\) when knowing \(X\) is now the conditional entropy

\[ H(Y|X) = - \int_x P(y|x) \left( \sum_y P(y|x) \log(p(y|x)) \right) dx. \]

By definition, the amount by which the uncertainty is decreased is the MI between the two variables \(Y\) and \(X\):

\[ I(Y; X) = \sum_y \int_x p(y,x) \log \frac{p(y,x)}{P(y)p(x)} dx. \]

Equation (26) may also be expressed as a function of entropy and conditional entropy,

\[ I(Y; X) = H(Y) - H(Y|X). \]

To estimate the MI in equation (27), the Parzen window based approach proposed by Kwak and Choi [17] is utilized in this paper.

Given \(l\) input vectors \((x^{(p)})_{p=1}^l\) in \(N_Y\) classes, the conditional probability density function \(p(x|y)\) of class \(y \in \{1, 2, \ldots, N_Y\}\) can be estimated by Parzen window estimations as

\[ \hat{p}(x|y) = \frac{1}{J_y} \sum_{i \in I_y} \kappa(x - x^{(i)}, \Sigma_{x_y}), \]

with \(h\) is the window function parameter which is taken as [20]

\[ h = \left( \frac{4}{d + 2} \right)^{\frac{1}{d+1}} I^{-\frac{1}{d+1}}. \]

According to equation (27), the probability of \(Y\) taken as \(y\) can be estimated directly by

\[ \hat{P}(y) = \Pr\{Y = y\} = \frac{J_y}{I}. \]

Thus, the first term in the right-hand side of equation (27) can be computed by combining (24) and (31).

Similar to [17], suppose that each input vector has the same probability, i.e. \(\hat{p}(x^{(j)}) = \frac{1}{l}\) with \(j = 1, 2, \ldots, l\), the second term in the right-hand side of equation (27) can be estimated by

\[ \hat{I}(Y|X) = -\sum_j \sum_{y=1}^{N_y} \hat{p}(y|x^{(j)}) \log \hat{p}(y|x^{(j)}) \]

with

\[ \hat{p}(y|x^{(j)}) = \frac{\sum_{i \in I_y} G(x^{(j)} - x^{(i)}, \Sigma_{x_y})}{\sum_{i=1}^{N_Y} \sum_{k \in I_l} G(x^{(j)} - x^{(k)}, \Sigma_{x_k})}. \]

In the following, a method for determining weights for each feature will be presented based on the above statements of MI and its estimation method. For a given data set \(X\), it is assumed that it has \(d\) features which are denoted by \(f_1, f_2, \ldots, f_d\). Then, the MI between each feature \(f_k\) \((k = 1, \ldots, d)\) and the output class variable \(Y\) can be estimated as follows:

\[ I(Y; f_k) = H(Y) - H(Y|f_k), \quad k = 1, \ldots, d. \]
The first term of the right-hand side of the above equation can be computed with (24) and (31), while the second term can be estimated by (32).

Therefore, the elements of foresaid weight vector $\beta$ in Section III can be determined as follows:

$$\beta_k = \frac{I(Y; f_k)}{\sum_{s=1}^{d} I(Y; f_s)}, \quad k = 1, \ldots, d. \quad (35)$$

V. EXPERIMENTAL RESULTS

In the following experiments, the kernel functions for the traditional SVMs and the proposed SVMs are all linear kernel functions $K(x, x') = x^T x'$. The main reason for the choice is that for the other kernel functions, additional parameters, e.g., width parameter $\sigma$ for Gaussian RBF kernel function, have to be adjusted, which may bias the effects of weighting. Moreover, the trade-off constant $C$ for the two types of SVMs are all taken 100 in the paper. For the multi-class classification tasks, the one-against-one strategy [21] is utilized. Furthermore, each of the following experiments will run 10 times. For each data set, we take the mean of all the 10 training classification accuracy rates as the final training accuracy rate, while the same manner for the final testing accuracy rates.

A. Synthetic data set

Ripley’s synthetic data set [22] is generated from mixtures of two Gaussian distributions. There are 250 two dimensional samples in the training set, while 1000 samples in the test set. In the paper, 20% of the whole 1250 samples are randomly chosen as the training set and the rest 80% are utilized as the test set in each trial.

Fig. 1 illustrates the weights estimated by the MI based method for the two features. Fig. 2 and Fig. 3 show the classification results of the traditional SVM and the proposed SVM on the test set in a certain trial. For the two features of the Ripley’s synthetic data set, Fig. 1 shows that the weights are 0.264 and 0.736, respectively. It can be easily observed from Fig. 2 and Fig. 3 that the second feature contains more classification information than that of the first feature, so it can be concluded that the MI based method provides us with the appropriate weights for the two features.

![Fig. 1. Values of feature weights for the Ripley data set](image)

Table I concludes the training and testing accuracy rates of the traditional and the proposed SVMs on the synthetic data set. It is shown in Table I that both the training and the testing accuracy rates of the SVM with weighted features are higher than those of the traditional SVMs. The coincident conclusion for the test set in one trial can also be achieved by observing Fig. 2 and Fig. 3.

B. Benchmark data sets from the UCI Repository

The five benchmark data sets used in the following experiments are all chosen from the UCI machine learning repository [23]. They are (number of samples, number of classes, number of features): Iris (150, 3, 4), Wine (178, 3, 13), glass (214, 6, 9), Vowel (990, 11, 10), and New-thyroid (215, 3, 5). For training the aforementioned two models on each data set, we randomly choose 75% samples for the training data set and the rest 25% for the test data set in each trial.

It is shown in Fig. 4(a) that the obtained weights for the four features of the Iris data set. They are all estimated by the foresaid MI approach. Similarly, the weights for the other four data sets, i.e. Wine, Glass, Vowel, and New-thyroid, are
illustrated in Fig. 4(b)-4(e), respectively.

Table II shows the training and testing accuracy rates of the traditional SVMs and the proposed SVMs with weighted features on the five benchmark data sets. In comparison with the traditional SVM, it can be easily found from Table II that the proposed method upon the Wine, Vowel, and New-thyroid data sets, can obtain better training and testing performances. For the Iris and Glass data sets, though with less training accuracy rates by comparing with the traditional SVMs, the testing accuracy rates of the presented method are obviously better than those obtained by the traditional SVMs. Especially for the Glass and Vowel data sets, the testing accuracy rates of the proposed method are 5.44% and 5.06% higher than those of the traditional SVMs, respectively.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>SVM</th>
<th>WFSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>98.74</td>
<td>94.87</td>
</tr>
<tr>
<td>Wine</td>
<td>88.12</td>
<td>88.67</td>
</tr>
<tr>
<td>Glass</td>
<td>71.66</td>
<td>63.68</td>
</tr>
<tr>
<td>Vowel</td>
<td>70.68</td>
<td>67.79</td>
</tr>
<tr>
<td>New-thyroid</td>
<td>96.56</td>
<td>96.18</td>
</tr>
</tbody>
</table>

* Note: Acc\textsubscript{train}–Training accuracy rates; Acc\textsubscript{test}–Testing accuracy rates; WFSVM–SVM with weighted features.
VI. Final Remarks

In this paper, we propose a novel SVM with weighted features. To construct this model, the weights of each feature are estimated by the MI method. Then, the weight vector is adopted into the primal and dual optimization problems of the proposed SVM model. Experimental results show that the proposed method can significantly improve the generalization ability of the traditional SVMs.

To make our proposed SVM more promising, there are three tasks for future investigation. First, the other types of kernel functions for the two aforementioned SVMs will be considered, such as Gaussian RBF kernel, sigmoid kernel, polynomial kernel, and etc. Second, the other approaches to determining the weights of features will be considered, such as the feature weight learning method in reference [10][24] and especially the iterative RELIEF method proposed by Sun and Li [25][26]. Third but not the last, the conditions under which the proposed SVM outperforms the traditional SVM will be investigated in our future work.

References