Mathematical and Computer Modelling 54 (2011) 1093–1099

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Learning SVM with weighted maximum margin criterion for classification of imbalanced data

Zhuangyuan Zhao, Ping Zhong *, Yaohong Zhao

College of Science, China Agricultural University, Beijing, 100083, PR China

ARTICLE INFO

Keywords:
Support vector machine
Imbalanced data learning
Kernel optimization
Weighted maximum margin criterion

ABSTRACT

As a kernel-based method, whether the selected kernel matches the data determines the performance of support vector machine. Conventional support vector classifiers are not suitable to the imbalanced learning tasks since they tend to classify the instances to the majority class which is the less important class. In this paper, we propose a weighted maximum margin criterion to optimize the data-dependent kernel, which makes the minority class more clustered in the induced feature space. We train support vector classification with the optimal kernel. The experimental results on nine benchmark data sets indicate the effectiveness of the proposed algorithm for imbalanced data classification problems.

1. Introduction

Learning from imbalanced data sets is an important and on-going issue in machine learning research. The class imbalance problem corresponds to domains for which one class is represented by a large number of instances while the other is represented by only a few. There are many class imbalanced problems in real-world applications [1–6]. Conventional classifiers that seek accuracy over a full range of instances are not suitable to deal with imbalanced learning tasks, since they tend to be overwhelmed by the majority class which is usually the less important class.

There are roughly two types of approaches to deal with imbalanced data classification problems. One is to sample data, either randomly or intelligently, to obtain an altered class distribution. These approaches consist of under-sampling the majority class or over-sampling the minority class, such as randomly under-sampling, randomly over-sampling, one-sided selection, cluster-based over-sampling, Wilson's editing, SMOTE, and borderline-SMOTE [7–11]. The other is to modify the standard learning algorithms. These approaches include cost-sensitive methods [12,13], margin calibration method [14], unsupervised self-organizing method [15], minimax probability machines [16,17], and one-class support vector machine [18].

Support vector machine (SVM) is an excellent kernel-based tool for classification and regression [19]. Within a few years after its introduction, SVM has already outperformed most other systems in a wide variety of applications, which include a wide spectrum of research areas ranging from pattern recognition, text categorization, biomedicine, brain–computer interface, and financial regression. However, the conventional SVM performs poorly on imbalanced learning because they pay less attention to the minority class. Classification rules for predicting the minority class tend to be fewer and weaker than those for the majority class. Consequently, testing instances in the minority class are misclassified more often than those in the majority class. However, the minority class is often the more important class in applications.

As we know, kernel plays an important role in SVM. Whether the kernel matches the data determines its performance. A kernel optimization algorithm [20] was proposed to maximize the class separability of the data in the empirical feature

* Corresponding author. Tel.: +86 10 62736511.
E-mail addresses: zping@cau.edu.cn, pingsunshine@yahoo.com.cn (P. Zhong).

0895-7177/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.11.040

space. Later, the kernel function was optimized through the maximum margin criterion [21]. However, the classifier still tends to be overwhelmed by the majority class even after the kernel transformation.

In this paper, based on the data-dependent kernel, we propose a weighted maximum margin criterion to optimize the kernel. The new optimization rule pays more attention to the minority class. The optimal data-dependent kernel makes the minority class more clustered in the induced feature space. Then we develop an SVM algorithm with the optimal kernel for imbalanced learning tasks. The experimental results on benchmark data sets show that the proposed method is effective.

The paper is organized as follows. The backgrounds including the soft margin SVM and the data-dependent kernel are introduced in Section 2. In Section 3, we propose a weighted maximum margin criterion to optimize the data-dependent kernel. Following that, we evaluate SVM with the optimal kernel on a series of experiments in Section 4. Section 5 is the conclusion.

2. Backgrounds

Let a set of m vector $\boldsymbol{x}^i \in R^d$, $i = 1, \ldots, m$ denote the instances to be trained. Let $y_i \in \{\pm 1\}$ denote the corresponding target of the training instance $\boldsymbol{x}^i$, $i = 1, \ldots, m$.

2.1. Support vector machine

SVM represents the novel learning technique that has been introduced in the framework of structural risk minimization and the theory of VC bounds [19]. In the binary classification tasks, SVM uses a linear separating hyperplane to create a classifier with maximal margin. The soft margin SVM solves the following mathematical programming problem:

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

where C is the penalty parameter, ξ_i's are slack variables, and $\phi : R^d \to \mathcal{F}$ is a nonlinear mapping. In general, under the Mercer theorem [19], it is possible to use some kernel $k(\boldsymbol{u}, \boldsymbol{v})$ to represent the inner product in feature space $\mathcal{F}$, i.e., $k(\boldsymbol{u}, \boldsymbol{v}) = \phi(\boldsymbol{u})^T\phi(\boldsymbol{v})$, such as RBF kernel $k(\boldsymbol{u}, \boldsymbol{v}) = \exp(-\gamma\|\boldsymbol{u} - \boldsymbol{v}\|^2)$ with parameter $\gamma > 0$. In practice, rather than solving (1) to get the appropriate separating hyperplane, we solve its dual problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}^i, \boldsymbol{x}^j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i\alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, \ i = 1, \ldots, m \end{aligned} \tag{2}$$

where α_i's are Lagrange multipliers.

2.2. Data-dependent kernel

The widely used kernels, such as RBF kernels and polynomial kernels, can achieve satisfactory performance in many applications. However, there is no general kernel function that is suitable to all data sets. So it is reasonable to construct the data-dependent kernel. The data-dependent kernel function is defined as follows:

$$k(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})q(\boldsymbol{z})k_0(\boldsymbol{x}, \boldsymbol{z}) \tag{3}$$

where $\boldsymbol{x}, \boldsymbol{z} \in R^d$, $k_0(\boldsymbol{x}, \boldsymbol{z})$ is the basic kernel, and $q(\boldsymbol{x})$ is a factor function of the form

$$q(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n}\beta_i k_1(\boldsymbol{x}, \boldsymbol{a}_i) \tag{4}$$

where $k_1(\boldsymbol{x}, \boldsymbol{a}_i) = \exp(-\gamma_1\|\boldsymbol{x} - \boldsymbol{a}_i\|^2)$, $\boldsymbol{a}_i \in R^d$, $\boldsymbol{a}_i$'s are empirical cores that can be chosen from the training data or determined according to the distribution of the training data, β_i's are the combination coefficients. According to [22], the data-dependent kernel satisfies the Mercer condition. Let $K = [k(\boldsymbol{x}^i, \boldsymbol{x}^j)]_{m\times m}$ and $K_0 = [k_0(\boldsymbol{x}^i, \boldsymbol{x}^j)]_{m\times m}$ be the kernel matrices corresponding to $k(\boldsymbol{x}, \boldsymbol{z})$ and $k_0(\boldsymbol{x}, \boldsymbol{z})$, respectively. It follows from (3) that

$$K = [q(\boldsymbol{x}^i)q(\boldsymbol{x}^j)k_0(\boldsymbol{x}^i, \boldsymbol{x}^j)]_{m\times m} = QK_0Q \tag{5}$$

where $Q = \mathrm{diag}(q(\boldsymbol{x}^1), q(\boldsymbol{x}^2), \ldots, q(\boldsymbol{x}^m))$ is a diagonal matrix. Denote the vectors $(q(\boldsymbol{x}^1), \ldots, q(\boldsymbol{x}^m))^T$ and $(\beta_1, \beta_2, \ldots, \beta_n)^T$ by $\boldsymbol{q}$ and β, respectively. Then we have

$$\boldsymbol{q} = \begin{pmatrix} 1 & k_1(\boldsymbol{x}^1, \boldsymbol{a}_1) & \cdots & k_1(\boldsymbol{x}^1, \boldsymbol{a}_n) \\ 1 & k_1(\boldsymbol{x}^2, \boldsymbol{a}_1) & \cdots & k_1(\boldsymbol{x}^2, \boldsymbol{a}_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(\boldsymbol{x}^m, \boldsymbol{a}_1) & \cdots & k_1(\boldsymbol{x}^m, \boldsymbol{a}_n) \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \triangleq K_1\beta \tag{6}$$

where K_1 is an $m \times (n+1)$ matrix.

3. Weighted maximum margin criterion

In this section, we propose a weighted maximum margin criterion that maximizes the weighted average margin between the majority and minority classes to learn an optimal data-dependent kernel.

Denote c_1 and c_2 the majority class and the minority class, respectively. Let the number of instances in the majority class be m_1, and the number of instances in the minority class be m_2. We define the weighted margin between two classes as follows:

$$d(c_i, c_j) = (\boldsymbol{e}_i - \boldsymbol{e}_j)^T(\boldsymbol{e}_i - \boldsymbol{e}_j) - \rho_i \mathrm{tr}S_i - \rho_j \mathrm{tr}S_j \tag{7}$$

where $\boldsymbol{e}_i$ is the mean vector of instances in class c_i, S_i is the scatter matrix of class c_i, tr denotes the trace of a matrix, and ρ_i is a weighted value. Define

$$J = \frac{1}{2m}\sum_{i=1}^{2}\sum_{j=1}^{2} m_i m_j d(c_i, c_j) \tag{8}$$

where $m = m_1 + m_2$. Formulation (8) can be simplified by substituting (7) into (8):

$$\begin{aligned} J &= \frac{1}{2m}\sum_{i=1}^{2}\sum_{j=1}^{2} m_i m_j[(\boldsymbol{e}_i - \boldsymbol{e}_j)^T(\boldsymbol{e}_i - \boldsymbol{e}_j) - \rho_i \mathrm{tr}S_i - \rho_j \mathrm{tr}S_j] \\ &= \frac{1}{2m}\sum_{i=1}^{2}\sum_{j=1}^{2} m_i m_j(\boldsymbol{e}_i - \boldsymbol{e}_j)^T(\boldsymbol{e}_i - \boldsymbol{e}_j) - \frac{1}{2m}\sum_{i=1}^{2}\sum_{j=1}^{2} m_i m_j(\rho_i \mathrm{tr}S_i + \rho_j \mathrm{tr}S_j) \\ &= \sum_{i=1}^{2} m_i(\boldsymbol{e}_i - \boldsymbol{e})^T(\boldsymbol{e}_i - \boldsymbol{e}) - \frac{1}{m}\sum_{i=1}^{2}\sum_{j=1}^{2} m_i m_j(\boldsymbol{e}_i - \boldsymbol{e})^T(\boldsymbol{e} - \boldsymbol{e}_j) - \sum_{i=1}^{2}\rho_i m_i \mathrm{tr}S_i \\ &= \mathrm{tr}S_b - (\rho_1 \mathrm{tr}S_{w_1} + \rho_2 \mathrm{tr}S_{w_2}) \end{aligned} \tag{9}$$

where $\boldsymbol{e}$ is the mean vector of the total data set, $\mathrm{tr}S_{w_i} = m_i \mathrm{tr}S_i$, $i = 1, 2$, and $\mathrm{tr}S_b = \sum_{i=1}^{2} m_i(\boldsymbol{e}_i - \boldsymbol{e})^T(\boldsymbol{e}_i - \boldsymbol{e})$. Here, S_b is the between-class scatter matrix. Denote $\mathrm{tr}S_w = m_1 \mathrm{tr}S_1 + m_2 \mathrm{tr}S_2$, where S_w is the within-class scatter matrix, and S_{w_1} and S_{w_2} are the parts of S_w corresponding to class c_1 and class c_2, respectively.

We call (8) the weighted maximum margin criterion (WMMC). In WMMC, we can adjust the scatter scales of two classes by setting ρ_1 and ρ_2. When the minority class is more clustered than the majority class, we can get better separating hyperplane. So we set $\rho_2 > \rho_1$. The reason is that when we maximize J, the scatter scale of minority class may get smaller than that of majority class.

In the following, we will maximize J in the feature space. Considering a mapping $\phi(\cdot)$ which corresponds to the data-dependent kernel, we maximize

$$J = \mathrm{tr}S_b^{\phi} - (\rho_1 \mathrm{tr}S_{w_1}^{\phi} + \rho_2 \mathrm{tr}S_{w_2}^{\phi}) \tag{10}$$

where S_b^{ϕ}, $S_{w_1}^{\phi}$ and $S_{w_2}^{\phi}$ are matrices in the feature space, which are similar to the S_b, S_{w_1} and S_{w_2} in the input space:

$$S_b^{\phi} = \sum_{i=1}^{2} m_i(\boldsymbol{e}_i^{\phi} - \boldsymbol{e}^{\phi})(\boldsymbol{e}_i^{\phi} - \boldsymbol{e}^{\phi})^T \tag{11}$$

with $\boldsymbol{e}_i^{\phi} = \sum_{p=1}^{m_i} \frac{1}{m_i}\phi(\boldsymbol{x}_i^p)$, $\boldsymbol{x}_i^p \in c_i$, $i = 1, 2$, $\boldsymbol{e}^{\phi} = \sum_{i=1}^{2} m_i \boldsymbol{e}_i^{\phi}$.

$$S_{w_i}^{\phi} = \sum_{p=1}^{m_i}(\phi(\boldsymbol{x}_i^p) - \boldsymbol{e}_i^{\phi})(\phi(\boldsymbol{x}_i^p) - \boldsymbol{e}_i^{\phi})^T, \quad i = 1, 2, \qquad S_w^{\phi} = \sum_{i=1}^{2} S_{w_i}^{\phi}.$$

Denote

$$\phi_i(\boldsymbol{X}) = (\phi(\boldsymbol{x}_i^1), \ldots, \phi(\boldsymbol{x}_i^{m_i})), \quad i = 1, 2$$
$$\phi(\boldsymbol{X}) = (\phi(\boldsymbol{x}_1^1), \ldots, \phi(\boldsymbol{x}_1^{m_1}), \phi(\boldsymbol{x}_2^1), \ldots, \phi(\boldsymbol{x}_2^{m_2}))$$
$$K_{ij} = \phi_i(\boldsymbol{X})^T \phi_j(\boldsymbol{X}), \quad i, j = 1, 2$$
$$K = \phi(\boldsymbol{X})^T \phi(\boldsymbol{X}) = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

where K is the kernel matrix, K_{11} is the kernel matrix corresponding to class c_1, and K_{22} that for the data in class c_2. Let us call the following matrices between-class and weighted within-class kernel scatter matrices, and denote them by B and W, respectively

$$B = \begin{pmatrix} \frac{1}{m_1} K_{11} & 0 \\ 0 & \frac{1}{m_2} K_{22} \end{pmatrix} - \frac{1}{m} K, \qquad W = \begin{pmatrix} \rho_1 W^{(1)} & 0 \\ 0 & \rho_2 W^{(2)} \end{pmatrix}$$

with

$$W^{(1)} = \begin{pmatrix} k_{11} & 0 & \cdots & 0 \\ 0 & k_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{m_1,m_1} \end{pmatrix} - \frac{1}{m_1} K_{11}$$
$$W^{(2)} = \begin{pmatrix} k_{m_1+1,m_1+1} & 0 & \cdots & 0 \\ 0 & k_{m_1+2,m_1+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{m,m} \end{pmatrix} - \frac{1}{m_2} K_{22}.$$

We also denote by B_0 and W_0 the between-class and the weighted within-class kernel scatter matrices corresponding to the basic kernel k_0. Using (5), we can easily see that

$$B = QB_0Q, \qquad W = QW_0Q. \tag{12}$$

In addition, $\mathrm{tr}S_b^\phi$, $\mathrm{tr}S_{w_1}^\phi$ and $\mathrm{tr}S_{w_2}^\phi$ can be further calculated as follows:

$$\begin{aligned} \mathrm{tr}S_b^\phi &= \sum_{i=1}^{2} m_i (\boldsymbol{e}_i^\phi - \boldsymbol{e}^\phi)^T (\boldsymbol{e}_i^\phi - \boldsymbol{e}^\phi) \\ &= \sum_{i=1}^{2} \frac{1}{m_i} \mathbf{1}_{m_i}^T \phi_i(\boldsymbol{X})^T \phi_i(\boldsymbol{X}) \mathbf{1}_{m_i} - \frac{1}{m} \mathbf{1}_m^T \phi(\boldsymbol{X})^T \phi(\boldsymbol{X}) \mathbf{1}_m \\ &= (\mathbf{1}_{m_1}^T, \mathbf{1}_{m_2}^T) \begin{pmatrix} \frac{1}{m_1} K_{11} & 0 \\ 0 & \frac{1}{m_2} K_{22} \end{pmatrix} \begin{pmatrix} \mathbf{1}_{m_1} \\ \mathbf{1}_{m_2} \end{pmatrix} - \frac{1}{m} \mathbf{1}_m^T K \mathbf{1}_m \\ &= \mathbf{1}_m^T B \mathbf{1}_m \end{aligned} \tag{13}$$

where $\mathbf{1}_{m_i}$ (or $\mathbf{1}_m$) is m_i (or m)-dimensional vectors with all entries being equal to unity.

$$\begin{aligned} \mathrm{tr}S_{w_1}^\phi &= \sum_{p=1}^{m_1} (\phi(\boldsymbol{x}_1^p) - \boldsymbol{e}_1^\phi)^T (\phi(\boldsymbol{x}_1^p) - \boldsymbol{e}_1^\phi) \\ &= \sum_{p=1}^{m_1} \phi(\boldsymbol{x}_1^p)^T \phi(\boldsymbol{x}_1^p) - m_1 \boldsymbol{e}_1^{\phi T} \boldsymbol{e}_1^\phi \\ &= \sum_{i=1}^{m_1} k_{ii} - \frac{1}{m_1} \mathbf{1}_{m_1}^T K_{11} \mathbf{1}_{m_1} \\ &= \mathbf{1}_{m_1}^T W^{(1)} \mathbf{1}_{m_1} \end{aligned} \tag{14}$$

$$\begin{aligned}\mathrm{tr}S_{w_2}^{\phi} &= \sum_{p=1}^{m_2}(\phi(\boldsymbol{x}_2^p)-\boldsymbol{e}_2^{\phi})^T(\phi(\boldsymbol{x}_2^p)-\boldsymbol{e}_2^{\phi})\\ &= \sum_{p=1}^{m_2}\phi(\boldsymbol{x}_2^p)^T\phi(\boldsymbol{x}_2^p)-m_2\boldsymbol{e}_2^{\phi T}\boldsymbol{e}_2^{\phi}\\ &= \sum_{i=m_1+1}^{m}k_{ii}-\frac{1}{m_2}\mathbf{1}_{m_2}^T K_{22}\mathbf{1}_{m_2}\\ &= \mathbf{1}_{m_2}^T W^{(2)}\mathbf{1}_{m_2}. \end{aligned} \tag{15}$$

By (10) and (13)–(15), we have

$$\begin{aligned}J &= \mathbf{1}_m^T B\mathbf{1}_m-(\rho_1\mathbf{1}_{m_1}^T W^{(1)}\mathbf{1}_{m_1}+\rho_2\mathbf{1}_{m_2}^T W^{(2)}\mathbf{1}_{m_2})\\ &= \mathbf{1}_m^T B\mathbf{1}_m-\mathbf{1}_m^T\begin{pmatrix}\rho_1 W^{(1)} & 0\\ 0 & \rho_2 W^{(2)}\end{pmatrix}\mathbf{1}_m\\ &= \mathbf{1}_m^T B\mathbf{1}_m-\mathbf{1}_m^T W\mathbf{1}_m. \end{aligned} \tag{16}$$

By (16), (12), (6), and noticing that $Q\mathbf{1}_m=\boldsymbol{q}$, we can write

$$J=\boldsymbol{q}^T B_0\boldsymbol{q}-\boldsymbol{q}^T W_0\boldsymbol{q}=\boldsymbol{\beta}^T K_1^T M_0 K_1\boldsymbol{\beta} \tag{17}$$

where $M_0 = B_0 - W_0 = B_0 - \begin{pmatrix}\rho_1 W_0^{(1)} & 0\\ 0 & \rho_2 W_0^{(2)}\end{pmatrix}$. To maximize J by the normalized vector $\boldsymbol{\beta}$, we obtain the following optimization problem:

$$\begin{aligned}&\max \quad \boldsymbol{\beta}^T K_1^T M_0 K_1\boldsymbol{\beta}\\ &\text{s.t.} \quad \boldsymbol{\beta}^T\boldsymbol{\beta}=1. \end{aligned} \tag{18}$$

To solve the optimization problem, we introduce the Lagrangian

$$\mathcal{L}(\boldsymbol{\beta},\lambda)=\boldsymbol{\beta}^T K_1^T M_0 K_1\boldsymbol{\beta}-\lambda(\boldsymbol{\beta}^T\boldsymbol{\beta}-1) \tag{19}$$

with multiplier λ. The Lagrangian $\mathcal{L}$ has to be maximized with respect to $\boldsymbol{\beta}$ and λ. The derivative of $\mathcal{L}$ with respect to $\boldsymbol{\beta}$ must vanish

$$\frac{\partial\mathcal{L}(\boldsymbol{\beta},\lambda)}{\partial\boldsymbol{\beta}}=(K_1^T M_0 K_1-\lambda I)\boldsymbol{\beta}=0$$

which leads to

$$K_1^T M_0 K_1\boldsymbol{\beta}=\lambda\boldsymbol{\beta} \tag{20}$$

It is shown by (20) that λ is the eigenvalue of $K_1^T M_0 K_1$ and $\boldsymbol{\beta}$ is the corresponding eigenvector. Thus

$$J=\boldsymbol{\beta}^T K_1^T M_0 K_1\boldsymbol{\beta}=\lambda\boldsymbol{\beta}^T\boldsymbol{\beta}=\lambda. \tag{21}$$

From (21), we can see that the maximal value of J is equal to the largest eigenvalue of the matrix $K_1^T M_0 K_1$. Therefore, the optimal solution $\boldsymbol{\beta}^*$ is the eigenvector of $K_1^T M_0 K_1$ corresponding to the largest eigenvalue. So we obtain the optimal combination coefficients β_i^* in data-dependent kernel (3), and thereby obtain the optimal data-dependent kernel.

4. Numerical experiments

We employ *g-means*, *sensitivity*, and *specificity* [11,23] which are the popular measures for imbalanced learning to evaluate the performance of algorithms in our experiments.

(1) *g-means* $= \sqrt{\mathrm{acc}^+\times\mathrm{acc}^-}$ (22)

(2) *sensitivity* $= \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ (23)

(3) *specificity* $= \mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$ (24)

where acc^+ indicates the *sensitivity* and acc^- the *specificity*. TP and TN denote true positives and true negatives, respectively. FN and FP denote false negatives and false positives, respectively. *Sensitivity* is defined as the accuracy on the positive class and *specificity* is the accuracy on the negative class. The value of *g-means* is high when both acc^+ and acc^- are high as well as the difference between acc^+ and acc^- is small.

Table 1
Nine benchmark data sets from UCI.

Data set	BC	Glass	Vehicle	German	Wine	SH	LD	Haberman	Fourclass
#ats	10	9	18	24	13	13	6	3	2
#neg	444	205	428	700	130	150	200	225	550
#pos	239	9	168	300	48	120	145	81	305
#tol	683	214	596	1000	178	270	345	306	855

Note: #ats, the number of attributes of instances; #neg, the number of negative instances; #pos, the number of positive instances; #tol, the total number of instances.

Table 2
Result comparisons of SVM_{rbf}, SVM_{mmc}, and SVM_{wmmc} on benchmark data sets.

Data set	Algorithm	g-means (%)	Sensitivity (%)	Specificity (%)	C	γ_0	γ_1	ρ_1	ρ_2
BC	SVM_{rbf}	96.84	96.24	**97.51**	10^4	10^{-5}	–	–	–
	SVM_{mmc}	96.93	96.63	97.28	10^4	5×10^{-5}	10^{-2}	–	–
	SVM_{wmmc}	**97.68**	**98.33**	97.06	10^4	5×10^{-5}	10^{-2}	1	1.8
Glass	SVM_{rbf}	92.40	88.00	**99.51**	10^4	10^{-5}	–	–	–
	SVM_{mmc}	95.18	**100**	90.73	10^7	5×10^{-5}	1	–	–
	SVM_{wmmc}	**99.01**	**100**	98.05	10^7	5×10^{-5}	1	1	2
Vehicle	SVM_{rbf}	85.92	77.37	**95.55**	10^2	10^{-3}	–	–	–
	SVM_{mmc}	88.25	88.03	88.57	10^4	1	1	–	–
	SVM_{wmmc}	**88.31**	**88.64**	88.10	10^4	1	1	1	1.2
German	SVM_{rbf}	63.57	47.33	**85.71**	10^4	10^{-3}	–	–	–
	SVM_{mmc}	70.04	**77.33**	63.43	10^4	5×10^{-5}	10^{-3}	–	–
	SVM_{wmmc}	**70.35**	**77.33**	64.00	10^4	5×10^{-5}	10^{-3}	1	1.05
Wine	SVM_{rbf}	83.62	73.89	**96.15**	10	0.5	–	–	–
	SVM_{mmc}	75.96	77.78	74.62	10^0	5×10^{-3}	10^{-3}	–	–
	SVM_{wmmc}	**84.86**	**83.89**	86.92	10^3	5×10^{-4}	0.5	1	6
SH	SVM_{rbf}	**85.47**	**82.50**	88.67	10^4	10^{-5}	–	–	–
	SVM_{mmc}	85.38	81.67	**89.33**	10^3	5×10^{-5}	0.5	–	–
	SVM_{wmmc}	85.38	81.67	**89.33**	10^3	5×10^{-5}	0.5	1	2
LD	SVM_{rbf}	**72.13**	71.03	74.50	10^5	10^{-2}	–	–	–
	SVM_{mmc}	71.07	67.59	**76.50**	10^4	5×10^{-4}	0.5	–	–
	SVM_{wmmc}	70.72	**73.79**	72.00	10^0	10^{-5}	10^{-3}	1	2
Haberman	SVM_{rbf}	48.44	31.18	**86.67**	10^3	1	–	–	–
	SVM_{mmc}	**59.79**	59.78	68.89	10^4	1	10^{-3}	–	–
	SVM_{wmmc}	58.67	**62.28**	65.78	10^4	1	10^{-3}	1	1.2
Four class	SVM_{rbf}	79.34	66.89	94.18	10^4	10^{-1}	–	–	–
	SVM_{mmc}	**97.12**	97.70	**96.55**	10^6	1	1	–	–
	SVM_{wmmc}	96.55	**98.03**	95.09	10^6	1	1	1	1.2

We train the data-dependent kernel by our WMMC. For comparison, we also train the data-dependent kernel by the MMC [21]. We denote SVM with RBF kernel by SVM_{rbf}, SVM with the kernel learned by MMC as SVM_{mmc}, and SVM with the kernel learned by WMMC as SVM_{wmmc}, respectively. In our experiments, RBF kernel $k_0(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma_0\|\boldsymbol{x} - \boldsymbol{z}\|^2)$ was chosen as the basic kernel, and the data-dependent kernel has the form of $k(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})q(\boldsymbol{z})k_0(\boldsymbol{x}, \boldsymbol{z})$, where $q(\boldsymbol{u}) = \beta_0 + \sum_{i=1}^{n} \beta_i \exp(-\gamma_1\|\boldsymbol{u} - \boldsymbol{a}_i\|^2)$. We took the centers of positive class and negative class as the empirical cores. In addition, we set the value of ρ_1 unity and the value of ρ_2 greater than unity.

The benchmark data sets used for the numerical experiments are taken from the UCI machine learning database repository [24]: Breast Cancer (BC), Glass, Vehicle, German, Wine, Stalog Heart (SH), Liver Disorder (LD), Haberman, and Four class. Since Glass, Vehicle, and Wine data sets are multi-class classification problems, we used the fifth class of Glass, the third class of Vehicle, and the third class of Wine as the minority classes, respectively. The remainder classes of each data set are assembled as the majority class. The other data sets belong to two-class imbalanced classification problems. Table 1 gives the description of the nine benchmark data sets.

In the experiments, we compared SVM_{wmmc} with SVM_{rbf} and SVM_{mmc}. We used five fold cross validation to evaluate the performance of the algorithms. The optimal parameters γ_0, γ_1, C were selected from the sets of values $\{10^{-5}, 5\times10^{-5}, 10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 1\}$, $\{10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 1\}$, and $\{1, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7\}$, respectively. The experimental results and optimal parameters are summarized in Table 2.

We can see from Table 2 that our algorithm achieves the best sensitivity accuracy rates on all data sets expect Stalog Heart. Our algorithm and SVM_{mmc} achieve the comparable second best sensitivity accuracy rates on Stalog Heart. For g-means index, our algorithm has the best values on five of nine data sets. Although our algorithm does not achieve the

best g-means values on the other four data sets, they are comparable to the best values. As for specificity accuracy rates, SVM_{rbf} has the best performance on six data sets, while SVM_{wmmc} gets the comparable results on four data sets including Breast Cancer, Glass, Stalog Heart, and Four class.

5. Conclusion

SVM solves the linearly inseparable classification problems by kernel trick. The selected kernel plays an important role because it determines the performance of SVM. In this work, we have proposed the weighted maximum margin criterion to optimize the data-dependent kernel. The proposed optimization rule can make the minority class more clustered in the induced feature space by maximizing the weighted average margin between the majority and minority classes. This kernel transformation is more suitable to the imbalanced learning tasks as the scatter scales of two classes can be adjusted by setting different weighted values. Compared with the conventional SVM and maximum margin criterion, our algorithm achieves best accuracy on positive minority class and comparable accuracy on negative majority class, which makes the g-means accuracy of our algorithm best.

Further research is required for discussing the relationship between the performance of the algorithm and the weighted values of ρ_1 and ρ_2, and developing the parameter search algorithm to search for the best setting.

Acknowledgements

The work is supported by the National Science Foundation of China (Grant No. 70601033) and Innovation Fund for Graduate Student of China Agricultural University (Grant No. KYCX2010105). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceeding of ECML-98 10th European Conference on Machine Learning, 1998.
[2] I. Maglogiannis, E. Zafiropoulos, I. Anagnostopoulos, An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers, Applied Intelligence 30 (2009) 24–36.
[3] T. Fawcett, F. Provost, Adaptive fraud detection, Data Mining and Knowledge Discovery 1 (1997) 291–316.
[4] R. Ji, D. Li, L. Chen, W. Yang, Classification and identification of foreign fibers in cotton on the basis of a support vector machine, Mathematical and Computer Modelling 51 (11–12) (2010) 1433–1437.
[5] S. Cai, R. Zhang, L. Liu, D. Zhou, A method of salt-affected soil information extraction based on a support vector machine with texture features, Mathematical and Computer Modelling 51 (11–12) (2010) 1319–1325.
[6] J.M. Matías, J. Taboada, C. Ordóñez, W. González-Manteiga, Partially linear support vector machines applied to the prediction of mine slope movements, Mathematical and Computer Modelling 51 (3–4) (2010) 206–215.
[7] R. Barandela, R.M. Valdovinos, J.S. Sanchez, F.J. Ferri, The imbalanced training sample problem: under or over sampling? in: Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition, SSPR/SPR'04, in: Lecture Notes in Computer Science, vol. 3138, 2004, pp. 806–814.
[8] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
[9] H. Han, W. Wang, B. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, ICIC'05, in: Lecture Notes in Computer Science, vol. 3644, 2005, pp. 878–887.
[10] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, SIGKDD Explorations 6 (2004) 40–49.
[11] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proceeding of the 14th International Conference on Machine Learning, 1997.
[12] D. Margineantu, T.G. Dietterich, Bootstrap methods for the cost-sensitive evaluation of classifiers, in: Proceeding of International Conference on Machine Learning, 2000.
[13] Y. Sun, M. Kamela, A. Wongb, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (2007) 3358–3378.
[14] C. Yang, J. Yang, J. Wang, Margin calibration in SVM class-imbalanced learning, Neurocomputing 73 (1–3) (2009) 397–411.
[15] Y.Y. Nguwi, S.Y. Cho, An unsupervised self-organizing learning with support vector ranking for imbalanced datasets, Expert Systems with Applications 37 (12) (2010) 8303–8312.
[16] G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, M.I. Jordan, A robust minimax approach to classification, Journal of Machine Learning Research 3 (2002) 555–582.
[17] K. Huang, H. Yang, I. King, et al., The minimum error minimax probability machine, Journal of Machine Learning Research 5 (2004) 1253–1286.
[18] L.M. Manevitz, M. Yousef, One-class SVMs for document classification, Journal of Machine Learning Research 2 (1) (2001) 139–154.
[19] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[20] H. Xiong, M.N. Swamy, M.O. Ahmad, Optimizing the kernel in the empirical feature space, IEEE Transactions on Neural Networks 16 (2) (2005) 460–474.
[21] J. Li, S. Chu, J. Pan, A criterion for learning the data-dependent kernel for classification, in: Advanced Data Mining and Applications, Springer, 2007, pp. 365–376.
[22] B. Schölkopf, A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[23] G. Wu, E. Chang, Class-boundary alignment for imbalanced dataset learning, in: ICML 2003 Workshop on Learning from Imbalanced Data, Sets II, Washington, 2003.
[24] P.M. Murphy, D.W. Aha, UCI machine learning database repository, 1985. http://www.ics.uci.edu/mlearn/MLRepository.html.