A Support Vector Machine Classifier with Automatic Confidence and Its Application to Gender Classification

Ji Zheng[a], Bao-Liang Lu[a,b,*]

[a] Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering
Shanghai Jiao Tong University
800 Dong Chuan Road, Shanghai 200240, China
[b] MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University
800 Dong Chuan Road, Shanghai 200240, China

Abstract

In this paper, we propose a support vector machine with automatic confidence (SVMAC) for pattern classification. The main contributions of this work to learning machines are twofold. One is that we develop an algorithm for calculating the label confidence value of each training sample. Thus, the label confidence values of all of the training samples can be considered in training support vector machines. The other one is that we propose a method for incorporating the label confidence value of each training sample into learning and derive the corresponding quadratic programming problems. To demonstrate the effectiveness of the proposed SVMACs, a series of experiments are performed on three benchmarking pattern classification problems and a challenging gender classification problem. Experimental results show that the generalization performance of our SVMACs is superior to that of traditional SVMs.

Keywords: Label confidence, Pattern classification, Support vector machine, Gender classification.

[*] Corresponding author
Email address: `bllu@sjtu.edu.cn` (Bao-Liang Lu)

1. Introduction

In the last several years, support vector machine (SVM) has become one of the most promising learning machines because of its high generalization performance and wide applicability for classification as well as for regression [1]. SVM maximizes its margin of separation and obtains an optimal decision boundary determined by a set of particular training samples called support vectors. Although SVM can find an optimal boundary, it is known to us that the information of the SVM decision boundary is only contained in the support vector training samples and is not considered in non-support vector training samples [2].

To improve the generalization performance of traditional SVMs, it is very important for us to consider the problems of how to search and utilize the information and distribution of the whole training samples, how to encode human prior knowledge widely existing in training samples [3], and how to incorporate prior knowledge into learning [4]. Recently various learning machines for pattern classification have been proposed. For instance, Jiang *et al* [5] developed a perturbation-resampling procedure to obtain the confidence interval estimates centered at *k*-fold cross-validated point for the prediction error and apply them to model evaluation and feature selection. Liu [6] investigated the effects of confidence transformation in combining multiple classifiers using various combination rules, where classifier outputs are transformed to confidence measures. Feng *et al* [7] proposed a scaled SVM, which is to employ not only the support vectors but also the means of the classes to reduce the mean of the generalization error. Graf *et al* [8] presented a method for combining human psychophysics and machine learning, in which human classification is introduced. These methods, nevertheless, do not consider how to use the label confidence of each training sample which may be regarded as human prior knowledge and how to incorporate the label confidence value of each training sample into learning.

Inspired by the ideas from Feng *et al* [7] and Graf *et al* [8], we proposed a support vector machine with confidence (SVMC) in our previous work [3]. We theoretically analyzed the decision boundary of SVMCs and shown that the generalization performance of SVMCs is superior to that of traditional SVMs. For SVMCs, however, the confidence value of each training sample must be labeled by the user manually before training. When the number of training samples is very large, much time for labeling these confidence values is required. Furthermore, we can not guarantee all these labeled confidence values are reasonable because of subjectivity. To overcome these deficiencies of SVMCs and to explore how to label rational confidence value of each training sample automatically, we propose a support vector machine with automatic confidence (SVMAC). The flowchart of training SVMACs and SVMCs[1] is illustrated in Fig. 1. The main difference between SVMACs and SVMCs is that the confidence value of each training sample is calculated by using an algorithm, instead of labeling the confidence value for each training sample by the user manually. For training SVMACs, we use both the labels and the label confidence values of all of the training samples.

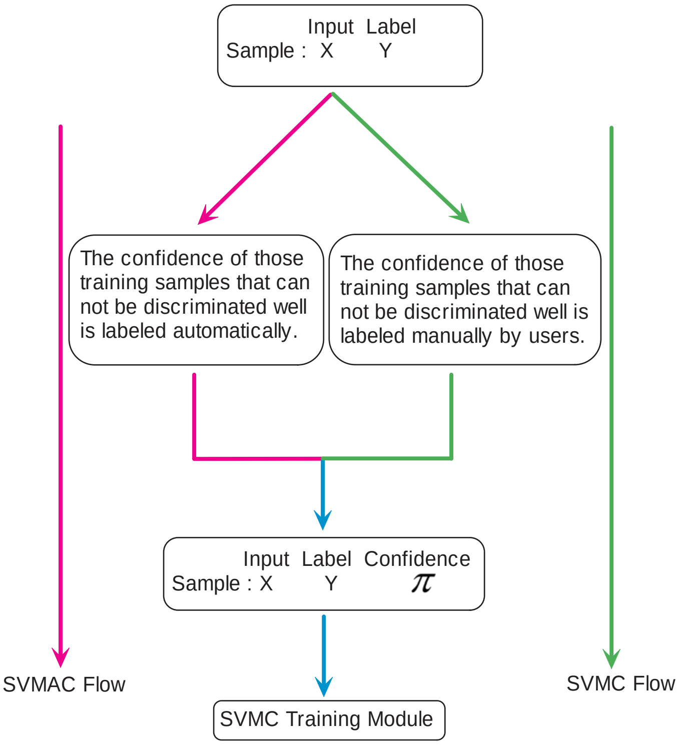

Input Label
Sample : X Y
The confidence of those training samples that can not be discriminated well is labeled automatically.
The confidence of those training samples that can not be discriminated well is labeled manually by users.
Input Label Confidence
Sample : X Y π
SVMAC Flow
SVMC Flow
SVMC Training Module

Figure 1: The flowchart of training SVMACs and SVMCs.

To evaluate the effectiveness of the proposed SVMAC, we apply SVMAC to gender classification problem as a case study. Gender classification based on facial images is a complicated and challenging two-class pattern classification problem, because the principle that the human brain can identify the gender from a facial image is still understood little. Although we can determine the gender of each facial image, sometimes we have not enough confidence for the real-life doubtless gender of some facial images. In other words, the gender of each facial image in a given face database can be confirmed, but in

[1] http://bcmi.sjtu.edu.cn/ jizheng/

reality we have some misgivings for identifying gender of some face images. From the viewpoint of learning machines, these misgivings can be expressed as the label confidence values of these facial images for discriminating the gender. Experimental results on a total of 10788 facial images indicate that the generalization performance of our SVMACs is superior to that of traditional SVMs and kNN, regardless of features used.

The remaining part of this paper is organized as follows. In Section 2, a method for incorporating the label confidence value of each training sample into learning is described and the corresponding quadratic programming problems are derived. In Section 3, a new algorithm for calculating the label confidence value for each training sample is described and an illustrative example is presented to demonstrate the performance of SVMACs. In Section 4, experimental results on three benchmarking pattern classification problems are described, and an application of SVMACs to gender classification is presented. Conclusions and future work are outlined in Section 5.

2. Support Vector Machine with Confidence

In this section, we present how to incorporate the label confidence value of each training sample into training support vector machines and derive the corresponding quadratic programming problem.

2.1. Traditional Support Vector Machine

The quadratic programming problems for the standard and soft margin forms of a traditional SVM [2] can be, respectively, expressed as

$$\begin{array}{cc} \min\limits_{\mathbf{w}} & \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \\ \text{s.t.} & \forall i, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0 \end{array} \tag{1}$$

and

$$\begin{array}{cc} \min\limits_{\mathbf{w}} & \frac{1}{2}||\mathbf{w}||^2 + D\sum_i \xi_i^2 \\ \text{s.t.} & \forall i, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0 \end{array} \tag{2}$$

where $\mathbf{w}$ is an adjusable weight vector, C is the parameter which is used to control the size of norm $||\mathbf{w}||$, the parameter D is to keep kernel matrix positive, x_i is the i-th sample vector, y_i is the label of sample vector ($y_i \in \{-1, 1\}$), b is the bias, and ξ_i is to measure the cost of generalization error on the i-th training sample.

The difference between Eqs. (1) and (2) is that the modes of measuring the cost of generalization error are distinct. Specifically, in Eq. (1), the parameter C can determine the optimal choice for $||w||_2$, and make $||\xi||_1$ (the first-order norm of ξ, where $\xi = (\xi_1, \cdots, \xi_i, \cdots)$) minimal. In Eq. (2), the parameter D can generate the best $||w||_2$, keep kernel matrix positive, and make $||\xi||_2$ (the second-order norm of ξ) least.

2.2. Incorporation of Label Confidence into Learning

One way of incorporating confidence values into learning is to re-scale the soft margin as follows,

$$\begin{aligned} \min_{\mathbf{w}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + D\textstyle\sum_i \xi_i^2 \\ \text{s.t.} \quad & \forall i, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant t(\pi_i) - \xi_i, \\ & \xi_i \geqslant 0 \end{aligned} \tag{3}$$

where $t(\pi_i)$ is a monotonic function to scale the confidence value.

In this paper, we select $t(\pi_i)$ as the following linear function,

$$t(\pi_i) = h \cdot \pi_i, \;\; \frac{1}{2} \leqslant \pi_i < 1, \tag{4}$$

where h is the scale parameter. The meaning of introducing this scale parameter h is to map the confidence values of training samples into another subspace, where we seek an optimal decision boundary maximizing the margin. Therefore, it is very important that we need understand how the decision boundary is influenced by the scale parameter h. For a two-class problem, the confidence value of each training sample should not be less than $\frac{1}{2}$ because each training sample has a determined label.

Many researchers reported that support vectors obtained by traditional support vector machines tend to be those training samples that people can not discriminate well [8, 9]. Based on this fact, we proposed a support vector machine with confidence in our previous work [10]. First, we divide the given training sample set $\mathcal{T}$ into two disjointed subsets $\mathcal{U}$ and $\mathcal{V}$ ($\mathcal{T} = \mathcal{U} \cup \mathcal{V}$), which are later treated in a different way in the training process. Then, we put the training samples in $\mathcal{U}$ with confidence π_i less than 1, and the remaining training samples in $\mathcal{V}$ with confidence π_i equal to 1. In essence, $\mathcal{U}$ contains the training samples that tend to be support vectors after training. In the following, we denote the number of training samples in $\mathcal{U}$ and $\mathcal{V}$ by n_u and n_v, respectively.

According to Eq. (3) for training subset $\mathcal{U}$ and Eq. (1) for training subset $\mathcal{V}$, we can express the quadratic programming problem for soft margin form as follows:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + D\sum_{i=1}^{n_u}\sigma_i^2 + C\sum_{j=1}^{n_v}\xi_j, \\ \text{s.t.} \quad & \forall 1 \leqslant i \leqslant n_u, \\ & \quad y_i^u(\mathbf{w}^T\mathbf{u}_i + b) = t(\pi_i) - \sigma_i, \\ & \forall 1 \leqslant j \leqslant n_v, \\ & \quad y_j^v(\mathbf{w}^T\mathbf{v}_j + b) \geqslant 1 - \xi_j, \quad \xi_j \geqslant 0, \end{aligned} \tag{5}$$

where $\mathbf{u}_i$ is the i-th vector in $\mathcal{U}$, $\mathbf{v}_j$ is the j-th vector in $\mathcal{V}$, $\mathbf{y}_i^u$ is the label of the i-th vector in $\mathcal{U}$, $\mathbf{y}_j^v$ is the label of the j-th vector in $\mathcal{V}$, and ξ_i in Eq. (3) is substituted by σ_i.

Using the standard Lagrangian dual technique, we obtain the Lagrangian function of Eq. (5) as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \xi, \lambda, \alpha, \beta) &= \frac{1}{2}||\mathbf{w}||^2 + D\sum_{i=1}^{n_u}\sigma_i^2 + C\sum_{j=1}^{n_v}\xi_j \\ & - \sum_{i=1}^{n_u}\lambda_i[y_i^u(\mathbf{w}^T\mathbf{u}_i + b) - t(\pi_i) + \sigma_i] \\ & - \sum_{j=1}^{n_v}\alpha_j[y_j^v(\mathbf{w}^T\mathbf{v}_j + b) - 1 + \xi_j] \\ & - \sum_{j=1}^{n_v}\beta_j\xi_j. \end{aligned} \tag{6}$$

At the saddle point, we get

$$\frac{\partial\mathcal{L}}{\partial\sigma_i} = 0 = \quad 2D\sigma_i - \lambda_i \tag{7}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_j} = 0 = \quad C - \alpha_j - \beta_j \tag{8}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 = \quad \mathbf{w} - \sum_{i=1}^{n_u}\lambda_i y_i^u\mathbf{u}_i - \sum_{j=1}^{n_v}\alpha_j y_j^v\mathbf{v}_j, \tag{9}$$

$$\frac{\partial\mathcal{L}}{\partial b} = 0 = \quad \sum_{i=1}^{n_u}\lambda_i y_i^u + \sum_{j=1}^{n_v}\alpha_j y_j^v. \tag{10}$$

From Eqs. (7), (8), (9) and (10), we have

$$\sigma_i = \frac{\lambda_i}{2D},$$

$$C = \alpha_j + \beta_j,$$

$$\mathbf{w} = \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j,$$

and

$$\sum_{i=1}^{n_u} \lambda_i y_i^u + \sum_{j=1}^{n_v} \alpha_j y_j^v = 0,$$

and take the place of σ_i and C. Thus, the following dual form is obtained

$$\begin{aligned}
\mathcal{L}(\mathbf{w}, b, \xi, \lambda, \alpha, \beta) &= \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} (\alpha_j + \beta_j)\xi_j \\
&\quad - \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{w}^T \mathbf{u}_i - \sum_{i=1}^{n_u} \lambda_i y_i^u b + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{2D}\sum_{i=1}^{n_u} \lambda_i^2 \\
&\quad - \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{w}^T \mathbf{v}_j - \sum_{j=1}^{n_v} \alpha_j y_j^v b + \sum_{j=1}^{n_v} \alpha_j - \sum_{j=1}^{n_v} \alpha_j \xi_j - \sum_{j=1}^{n_v} \beta_j \xi_j \\
&= \frac{1}{2}||\mathbf{w}||^2 - \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} (\alpha_j + \beta_j)\xi_j \\
&\quad - \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{w}^T \mathbf{u}_i - (\sum_{i=1}^{n_u} \lambda_i y_i^u + \sum_{j=1}^{n_v} \alpha_j y_j^v) b + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i \\
&\quad - \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{w}^T \mathbf{v}_j + \sum_{j=1}^{n_v} \alpha_j - \sum_{j=1}^{n_v} (\alpha_j + \beta_j)\xi_j \\
&= \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{w}^T \mathbf{u}_i - \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{w}^T \mathbf{v}_j \\
&\quad + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} \alpha_j. \qquad (11)
\end{aligned}$$

By substituting **w** with

$$\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j,$$

in above Lagrangian function (11), we have

$$\begin{aligned}
\mathcal{L}(\mathbf{w}, b, \xi, \lambda, \alpha, \beta) &= \frac{1}{2}(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j) \\
&\quad - \sum_{i=1}^{n_u} \lambda_i y_i^u(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \mathbf{u}_i \\
&\quad - \sum_{j=1}^{n_v} \alpha_j y_j^v(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \mathbf{v}_j \\
&\quad + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} \alpha_j \\
&= \frac{1}{2}(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j) \\
&\quad -[(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i)^T \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + (\sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i \\
&\quad +(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i)^T \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j + (\sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j] \\
&\quad + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} \alpha_j \\
&= -\frac{1}{2}(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j) \\
&\quad + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{4D}\sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} \alpha_j. \qquad (12)
\end{aligned}$$

Therefore, from Eq. (12), Eq. (5) can be rewritten by

$$\begin{aligned}
\max_{\lambda,\alpha} \quad & -\frac{1}{2}(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \\
& (\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j) \\
& + \sum_{i=1}^{n_u} t(\pi_i)\lambda_i - \frac{1}{4D} \sum_{i=1}^{n_u} \lambda_i^2 + \sum_{j=1}^{n_v} \alpha_j \qquad (13) \\
\text{s.t.} \quad & \forall 1 \leqslant i \leqslant n_u, \quad 0 \leqslant \lambda_i < +\infty, \\
& \forall 1 \leqslant j \leqslant n_v, \quad 0 \leqslant \alpha_j \leqslant C, \\
& \sum_{i=1}^{n_u} \lambda_i y_i^u + \sum_{j=1}^{n_v} \alpha_j y_j^v = 0,
\end{aligned}$$

namely,

$$\begin{aligned}
\min_{\lambda,\alpha} \quad & \frac{1}{2}(\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j)^T \\
& (\sum_{i=1}^{n_u} \lambda_i y_i^u \mathbf{u}_i + \sum_{j=1}^{n_v} \alpha_j y_j^v \mathbf{v}_j) \\
& - \sum_{i=1}^{n_u} t(\pi_i)\lambda_i + \frac{1}{4D} \sum_{i=1}^{n_u} \lambda_i^2 - \sum_{j=1}^{n_v} \alpha_j \qquad (14) \\
\text{s.t.} \quad & \forall 1 \leqslant i \leqslant n_u, \quad 0 \leqslant \lambda_i < +\infty, \\
& \forall 1 \leqslant j \leqslant n_v, \quad 0 \leqslant \alpha_j \leqslant C, \\
& \sum_{i=1}^{n_u} \lambda_i y_i^u + \sum_{j=1}^{n_v} \alpha_j y_j^v = 0.
\end{aligned}$$

For simplicity of description, we can express Eq. (14) in a matrix form as follows:

$$\begin{aligned}
\min_{\mathbf{W}} \quad & \tfrac{1}{2}\mathbf{W}^T\mathbf{H}\mathbf{W} + \mathbf{G}^T\mathbf{W} \qquad (15) \\
\text{s.t.} \quad & \mathbf{W}^T e = 0, \\
& \forall 1 \leqslant i \leqslant n_{n_u}, 0 \leqslant \lambda_i < +\infty \\
& \forall 1 \leqslant j \leqslant n_{n_v}, 0 \leqslant \alpha_j \leqslant C
\end{aligned}$$

where e denotes the vector with each component equal to 1, and **W**, **G** and **H** are defined as follows,

$$\mathbf{W} = \begin{bmatrix} \lambda_1 y_1^u & \cdots & \lambda_{n_u} y_{n_u}^u & \alpha_1 y_1^v & \cdots & \alpha_{n_v} y_{n_v}^v \end{bmatrix}^T, \tag{16}$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{t}(\pi_1) y_1^u & \cdots & -\mathbf{t}(\pi_{n_u}) y_{n_u}^u & -y_1^v & \cdots & -y_{n_v}^v \end{bmatrix}^T, \tag{17}$$

$$\mathbf{H} = \begin{bmatrix} U^T U + B & U^T V \\ V^T U & V^T V \end{bmatrix} \tag{18}$$

where $U = \begin{bmatrix} u_1 & \cdots & u_{n_u} \end{bmatrix}$, $V = \begin{bmatrix} v_1 & \cdots & v_{n_v} \end{bmatrix}$, and $B = \frac{1}{2D} I_{n_u \times n_u}$.

3. Support Vector Machine with Automatic Confidence

In this section, we will answer the question of how to automatically calculate the confidence values of training samples, and present an illustrative example to demonstrate the generalization performance of our proposed SVMACs.

3.1. Algorithm for Labeling Confidence

Although we have shown that the generalization performance of SVMCs is superior to that of traditional SVMs [3], SVMCs face the following three problems: a) we must spend much time to manually label the confidence value for each of the training samples, especially when the number of the training samples is large; b) we can not guarantee that all the labeled confidence values are reasonable because people's action on determining the confidence values is very subjective; and c) it is hard for the user to determine the label confidence values of training samples in some pattern classification problems such as text categorization and patent classification. To deal with these problems, we introduce a novel logical method for dividing the training sample set into two subsets $\mathcal{U}$ and $\mathcal{V}$, and propose an algorithm for labeling the confidence (ALC) automatically. The ALC algorithm is described in Algorithm 1.

In comparison with traditional SVMs, the main additional cost of training SVMACs is to construct a decision boundary γ for labeling the confidence value of each training sample. If a traditional SVM is used to label the confidence value of each training sample, the distances between support vector samples and the decision boundary γ are smaller than those between non-support vector samples and the decision boundary γ for the training samples with the same class label. Thus, Δ needn't be calculated in the ALC algorithm 1.

As a matter of fact, the distance between a training sample and the decision boundary γ suggests whether the sample can be discriminated well or not. Obviously, the training sample which is far from the decision boundary can tend to be discriminated and should be appended into $\mathcal{V}$. Otherwise, it needs to be added to $\mathcal{U}$. Therefore, the confidence values calculated automatically by the ALC algorithm is consistent with the confidence values labeled by the user manually.

Algorithm 1 ALC

```
Step 1: Train a pattern classifier such as SVM and multi-layer perceptron on a given training sample set $\mathcal{T} = \{(x_i, y_i)|1 \leqslant i \leqslant N\}$ and obtain a decision boundary $\gamma$
Step 2: Calculate the distances between all of the samples in $\mathcal{T}$ and the decision boundary $\gamma$ and form the distance set $\Omega = \{d_i \mid \text{the distance between the } i\text{-th sample and } \gamma\}$;
Step 3: Set a threshold value $\Delta$,
    for all $i$ from 1 to $N$
        if $d_i < \Delta$,
            Add the sample $(x_i, y_i)$ to $\mathcal{U}$;
        else
            Add the sample $(x_i, y_i)$ to $\mathcal{V}$;
        end if
    end for
Step 4: The confidence values of the samples in $\mathcal{V}$ are set to 1.0 while the confidence values of the samples in $\mathcal{U}$ are projected onto the confidence space $[\frac{1}{2}, 1)$ according to their distances and a linear mapping principle;
Step 5: Train an SVMAC on these training samples with both labels and label confidence values and obtain an SVMAC classifier
```

3.2. An Illustrative Example

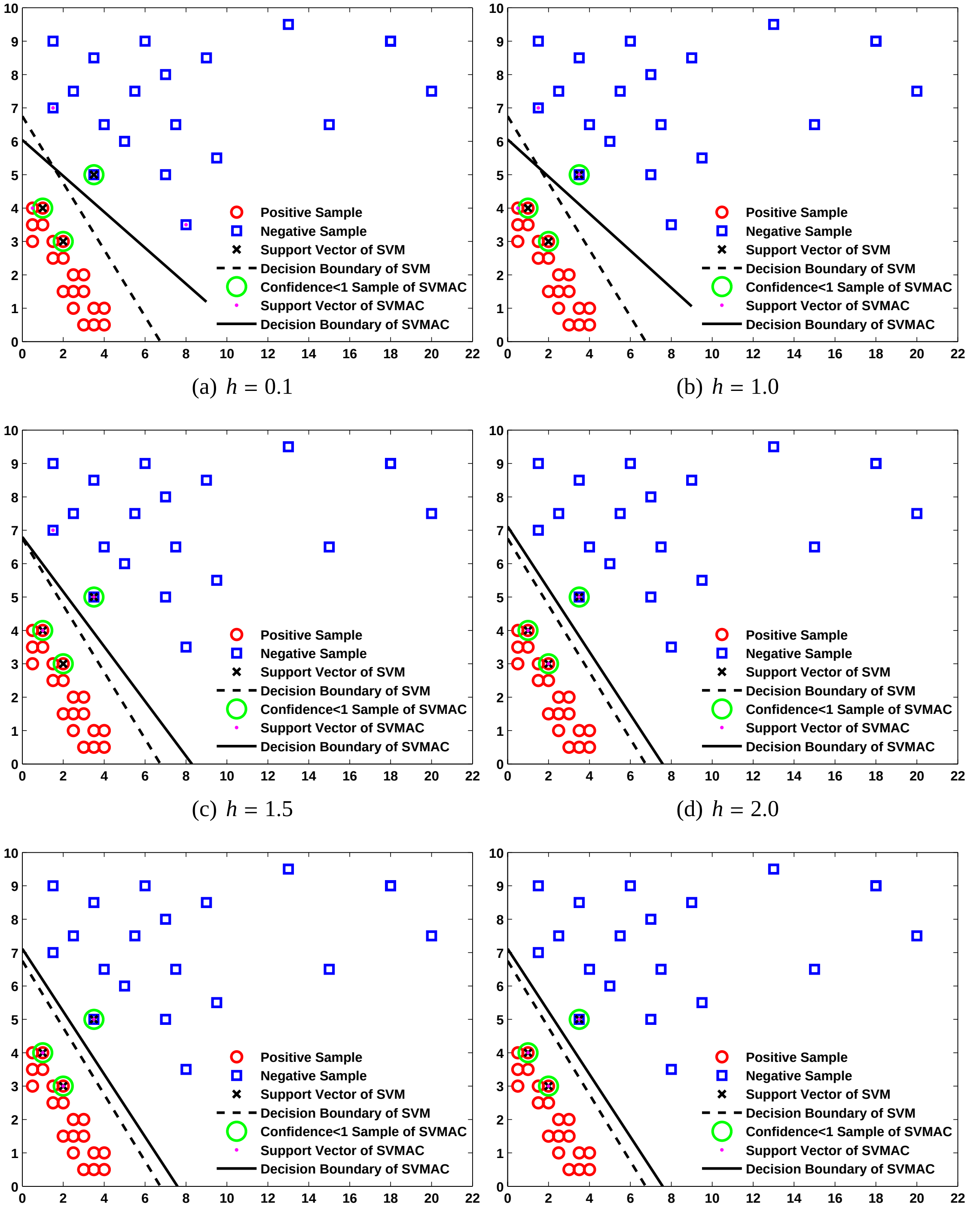
Now we examine the performance of SVMACs by using an illustrative example [11]. According to the ALC algorithm and SVMACs defined in Eq. (5), we set the confidence values of training samples in $\mathcal{U}$ less than 1. Those training samples are marked by small circles (green) shown in Fig. 2, and the right figure in Fig. 3. From these figures, we can see that the decision boundaries are changed if the confidence values of the support vector training samples in $\mathcal{U}$ are assigned by using the ALC algorithm. Here a traditional SVM is trained on the training sample set, $\mathcal{T} = \{(x_i, y_i)|1 \leqslant i \leqslant N\}$, to form the decision boundary γ.

From Fig. 2, we can observe that the change of the decision boundaries of SVMACs is negligible when the scale parameter h changes from 0.1 to 1 or from 2 to a very large value. This phenomenon suggests that a small variation in h ($h \in (0, 1]$) or a large variation in h ($h \in (2, 10^8)$ is hardly to affect the performance of SVMACs. Figs. 2 (e) and (f) show the decision boundaries of SVMACs, where the scale parameter h is set to 5.1 and

10^8, respectively. Although these two decision boundaries of SVMACs are quite different from those of SVMACs shown in Figs. 2 (a) and (b), they also moves from the side of dense training samples (lower left area) to that of sparse training samples (the upper right area). According to the discussions mentioned above, we can conclude that the movement of the decision boundaries formed by SVMACs is reasonable, and is identical to that of SVMCs [3]. Besides, because in Fig. 2 the different ranges of the scale parameter h, i.e., the range from 0.1 to 1.0 and the range from 2 to a large number, influence the position of the SVMAC decision boundary greatly, in practical applications we need to adjust the scale parameter h to obtain the best classification result.

The support vectors obtained by traditional SVMs can be regarded as the training samples closing to noise. Therefore, we should assign them with confidence values less than 1. By training the proposed SVMACs on all the training samples with proper confidence values, we can obtain the decision boundaries shown in Fig. 2. From this figure, we can see that if the support vectors obtained by the traditional SVMs are assigned with appropriate confidence values, some of them may be turned into non-support vectors after training SVMACs. The decision boundaries obtained by SVMACs can be regarded as a fitting achieved by training a pattern classifier on the training sample set in which some noise samples are removed. As a result, the decision boundaries obtained by SVMACs are superior to those obtained by traditional SVMs. For example, since the training samples located in the lower left area in Fig. 2 are much denser and closer to the boundary formed by traditional SVMs than the training samples located in the upper right area, the movement of the decision boundary from the lower left corner to the upper right corner caused by the proposed SVMACs no doubt yields a better separation than that of traditional SVMs.

From the angle of the label confidence, the decision boundaries formed by our SVMACs as shown in Fig. 2 are superior to those generated by traditional SVMs. However, the decision boundaries produced by traditional SVMs and the proposed SVMACs are the same as shown in Fig. 3, where only the non-support vector training samples in $\mathcal{V}$ are assigned with confidence values less than 1 according to their distances to the original decision boundary γ and none of support vector training samples in $\mathcal{U}$ is assigned with confidence value. From Fig. 3, we see that non-support vector training samples in $\mathcal{V}$ with less than 1 confidence values don't affect the decision boundary. In other words, after the confidence values less than 1 are assigned to some non-support vector training samples by using the ALC algorithm, the whole generalization performance of the proposed SVMACs will not be decreased.



(a) $h = 0.1$

(b) $h = 1.0$

(c) $h = 1.5$

(d) $h = 2.0$

(e) $h = 5.1$

(f) $h = 10^8$

Figure 2: Comparison of the decision boundaries formed by the proposed SVMACs with different values of the scale parameter h.

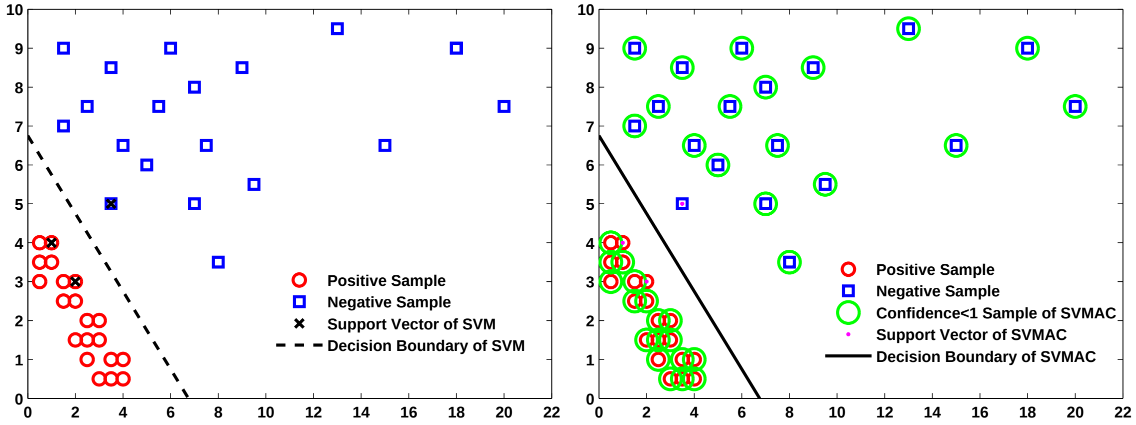


Figure 3: Comparison of the decision boundaries formed by traditional SVM (left) and our proposed SVMAC (rigfht), where we only assign the confidence values (less than 1) to non-support vector training samples in $\mathcal{V}$ for training SVMAC and do not consider any confidence values for support vector training samples in $\mathcal{U}$.

Table 1: Distribution of training and test data of the three benchmarking problems.

Data Set	Training		Test		No. of Input
	Positive	Negative	Positive	Negative	Dimensions
Arcene	44	56	44	56	2000
Dexter	150	150	150	150	4000
Gisette	3000	3000	500	500	4000

4. Benchmarking Problems and Application

To demonstrate the performance of the proposed SVMAC and compare it with traditional SVMs, we perform experiments on three benchmarking pattern classification problems and a challenging gender classification problem. The *k*NN algorithm is used as a baseline pattern classifier. Here, an optimal *k* is selected from the range of $[5, 50]$.

4.1. Benchmarking Problems

We select three benchmarking problems, namely Arcene, Dexter and Gisette, from UCI Machine Learning Repository[2]. Table 1 shows the distributions of these data sets.

[2] http://archive.ics.uci.edu/ml/

The Arcene's task is to distinguish cancer versus normal patterns from mass-spectrometric data, the Dexter data set is a text classification problem in a bag-of-word representation, and the Gisette task is a handwritten digit recognition problem of separating the highly confused digits '4' and '9'.

The experiment results on these benchmarking problems are shown in Table 2. From this table, we can see that our proposed SVMACs achieve the best classification accuracy among three pattern classifiers. It should be noted that the parameter C in Eqs. (1) and (14) is set to the same value.

Table 2: Comparison of classification accuracy of our SVMAC with that of *k*NN and traditional SVM. Here both SVMACs and traditional SVMs use a linear kernel.

	Data Set		
Method	Arcene	Dexter	Gisette
*k*NN	79.0	66.0	96.3
SVM	83.0	79.7	97.0
SVMAC	**84.0**	**82.3**	**97.2**

4.2. *Gender Classification*

In the last several years, various feature extraction and pattern classification methods have been developed for gender classification [12, 13, 14, 15]. Support vector machine is the most common used pattern classifier in gender classification [16, 13, 15]. To demonstrate the effectiveness our proposed SVMAC, we apply it to solving the gender classification problem. In this study, we use multi-view facial images from the CAS-PEAL face database [17] and frontal facial images from both the FERET[3] face database and the BCMI[4] face database.

4.2.1. *Experimental Setup*

A total of 10788 different-pose facial images from the CAS-PEAL, the FERET, and the BCMI face databases are organized into 11 groups. The distributions of these training and test data are shown in Table 3. A total of 8751 facial images are selected randomly

[3] http://www.frvt.org/FERET/default.htm

[4] BCMI face database is set up and packed up by the Center for Brain-Like Computing and Machine Intelligence, Shanghai Jiao Tong University, Shanghai, China.

Table 3: Description of training and test data from three face databases for gender classification.

Data Set	Description	Training		Test	
		Male	Female	Male	Female
CAS-PEAL (C)	PD00	311	311	284	134
	PD15	296	296	220	127
	PD30	296	296	220	127
	PM00	310	310	285	134
	PM15	295	295	221	127
	PM30	295	295	221	127
	PU00	311	311	284	134
	PU15	296	296	220	127
	PU30	296	296	220	127
FERET (F)	PM00	282	282	307	121
BCMI (B)	PM00	361	361	168	155

from the CAS-PEAL face database. These facial images belong to 9 different poses (See Fig. 4) including looking down pose, looking middle pose, and looking up pose with 0 degree, 15 degree, and 30 degree, respectively. For simplicity of description, the CAS-PEAL, the FERET, and the BCMI face databases are represented by “C”, “F”, and “B” in front of pose description names, respectively. For example, “C-PU30” denotes the facial images belonging to the group of looking up pose with 30 degree from the CAS-PEAL face database. In training phase, we performed 5-cross validation to find the best parameters for both SVMACs and traditional SVMs. The parameters h and D are selected from $\{h|h = \frac{n}{5}, 1 \leqslant n \leqslant 8\}$ and $\{D|D = 2^i, -10 \leqslant i \leqslant 10\}$, respectively. All experiments are performed on a Pentium fourfold CPU (2.83GHz) PC with 8GB RAM.

4.2.2. Feature Extraction

Before training, the original facial images are preprocessed by locating eye positions, geometric normalization, and cropping to obtain its face area. We use five feature selection methods, namely gray, Gabor, local binary pattern (LBP) [18] [19], multi-resolution local binary pattern (MLBP) [14], and local Gabor binary pattern (LGBP) [20]. The numbers of dimensions corresponding to these five kinds of features are $20m$, $40m$, $59m$, $3 \times 59m$, $40m$, respectively. Here, m is the number of subregions, into which each facial image is divided.

Table 4: Gender classification accuracy (%) in mean (odd row) and standard deviation (even row) achieved by *k*NN, SVMs, and SVMACs, where SVMs and SVMACs use a RBF kernel, the number of window blocks is set to $m = K \times K$, and K is ranged from 5 to 10.

Description	Method	LGBP-CCL	LGBP-LDA	MLBP	LBP	Gabor	Gray
C-PD00	KNN	93.4	99.7	93.3	92.4	73.6	76.9
		1.6	0.4	10.8	5.8	**12.8**	**3.5**
	SVM	96.7	99.6	95.5	94.2	90.6	92.5
		11.4	0.4	2.9	10.2	16.3	4.9
	SVMAC	**97.4**	**99.9**	**96.4**	**95.3**	**92.6**	**93.8**
		4.8	**0.2**	**1.1**	**4.4**	15.1	4.3
C-PD15	KNN	94.2	99.8	89.4	90.3	81.0	83.5
		4.1	0.1	19.3	2.6	**16.0**	**1.0**
	SVM	97.4	99.7	94.9	93.5	88.7	91.3
		7.5	0.1	1.1	3.0	26.0	1.1
	SVMAC	**97.7**	**99.8**	**95.6**	**94.6**	**90.5**	**92.6**
		7.3	**0.1**	**0.9**	**2.1**	17.2	2.0
C-PD30	KNN	91.3	99.7	89.8	88.1	75.6	76.6
		2.6	0.7	6.1	14.1	30.6	9.3
	SVM	96.4	99.8	92.7	92.2	89.9	89.4
		6.2	0.4	**2.0**	3.2	13.1	**1.5**
	SVMAC	**96.8**	**100.0**	**93.3**	**93.2**	**90.4**	**90.6**
		4.8	**0.0**	2.1	**1.9**	**8.4**	2.5
C-PM00	KNN	94.7	99.6	90.3	89.2	71.5	77.5
		9.3	0.6	6.9	15.6	29.1	4.7
	SVM	97.3	100.0	96.1	94.5	91.6	94.6
		10.6	0.0	**2.7**	10.4	26.3	6.0
	SVMAC	**97.5**	**100.0**	**96.6**	**95.5**	**92.7**	**95.4**
		10.8	**0.0**	3.3	**8.5**	**16.7**	**3.6**
C-PM15	KNN	95.0	99.8	91.4	91.7	81.3	85.3
		6.1	0.3	5.5	**10.5**	**21.6**	**4.9**
	SVM	97.1	99.7	95.1	94.2	92.7	94.5
		2.9	0.3	**0.9**	14.6	30.2	7.4
	SVMAC	**97.3**	**99.9**	**95.7**	**94.7**	**93.2**	**95.0**
		2.7	**0.2**	2.5	15.3	23.4	6.9

Table 4: (*Continued*) Gender classification accuracy (%) in mean (odd row) and standard deviation (even row) achieved by *k*NN, SVMs, and SVMACs, where SVMs and SVMACs use a RBF kernel, the number of window blocks is set to $m = K \times K$, and K is ranged from 5 to 10.

Description	Method	LGBP-CCL	LGBP-LDA	MLBP	LBP	Gabor	Gray
C-PM30	KNN	91.3	99.6	88.1	87.9	79.5	79.5
		5.0	0.9	28.1	**7.3**	24.1	8.7
	SVM	96.3	99.7	93.7	92.4	91.9	92.1
		6.8	0.5	9.1	15.3	17.4	**1.5**
	SVMAC	**96.9**	**100.0**	**95.1**	**93.2**	**93.1**	**92.6**
		2.8	**0.0**	**6.0**	11.3	**8.3**	2.0
C-PU00	KNN	92.7	99.8	90.1	88.7	69.9	74.4
		13.3	0.6	4.8	6.5	**13.9**	**12.2**
	SVM	96.7	99.9	95.4	94.5	90.2	89.0
		5.4	0.1	4.2	**4.7**	33.2	28.4
	SVMAC	**97.6**	**100.0**	**96.3**	**95.3**	**91.6**	**92.0**
		5.6	**0.0**	**2.4**	4.8	18.4	17.9
C-PU15	KNN	94.9	99.7	88.5	89.5	77.5	80.9
		13.7	0.5	38.2	5.3	38.4	**1.3**
	SVM	97.6	99.9	96.0	95.6	92.6	89.3
		1.3	0.1	4.2	**1.8**	8.9	16.5
	SVMAC	**98.3**	**100.0**	**96.6**	**96.0**	**93.1**	**90.2**
		2.5	**0.0**	**0.8**	2.5	**6.5**	18.8
C-PU30	KNN	93.9	99.8	88.3	87.4	76.9	81.5
		3.0	0.4	10.0	**2.5**	30.4	13.0
	SVM	96.9	99.9	94.0	92.8	89.0	88.2
		8.6	0.2	4.0	5.1	21.1	9.1
	SVMAC	**97.0**	**100.0**	**95.0**	**93.6**	**90.2**	**89.6**
		6.3	**0.0**	**1.2**	5.5	**13.0**	**5.9**
F-PM00	KNN	89.9	96.2	87.3	84.4	73.9	72.5
		4.8	3.2	9.8	9.4	**0.6**	**3.7**
	SVM	94.6	98.8	93.7	92.3	90.4	89.5
		11.6	2.3	1.3	9.0	9.8	3.7
	SVMAC	**95.1**	**99.1**	**93.8**	**93.1**	**91.6**	**91.2**
		8.8	**0.6**	**1.0**	**3.9**	15.1	4.6
B-PM00	KNN	91.5	98.9	91.0	90.4	86.0	89.8
		16.6	0.8	15.1	11.4	23.2	6.1
	SVM	97.3	99.4	96.3	96.2	93.4	95.4
		8.3	1.4	2.0	3.2	10.2	2.1
	SVMAC	**98.1**	**99.7**	**97.2**	**97.3**	**95.1**	**95.7**
		0.7	**0.2**	**0.9**	**0.9**	**6.1**	**1.2**

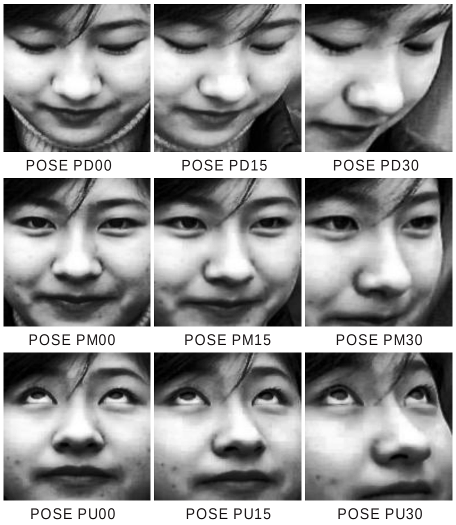


Figure 4: Nine different-pose facial images from the CAS-PEAL face database.

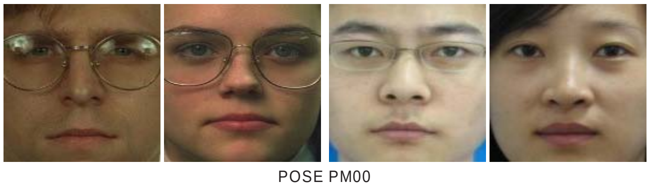
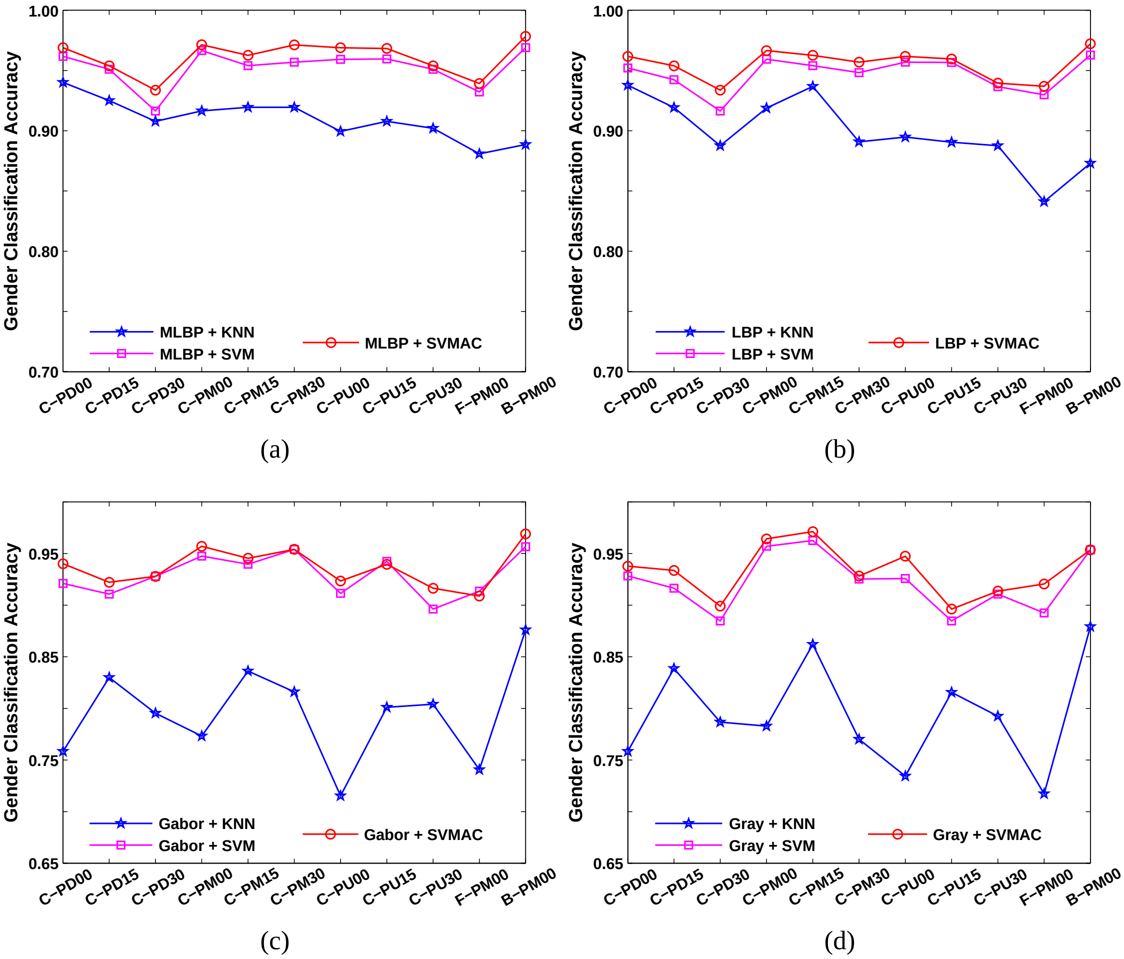


Figure 5: Examples from the FERET face database (left two) and the BCMI face database (right two).

4.2.3. Experimental Results

In this application study, the confidence value of each training sample is calculated automatically by using the ALC algorithm described in Section 3.1. For example, we see that the 2nd and the 10-th facial images shown in Fig. 7 are non-support vector training samples. Although their confidence values labeled manually are less than 1, this labeling results will not affect the whole classification accuracy according to the analysis described in Section 2. The 4-th sample is a support vector training sample, but its label confidence value is assigned to 1 manually. Thus, this indicates that we can not guarantee all the confidence values labeled manually because of subjectivity. The confidence values of other samples labeled manually are almost consistent with these calculated by the ALC algorithm. Consequently, the manual and automatic label confidence values are equivalent approximately in most situations.



(a)

(b)

(c)

(d)

Figure 6: Comparison of gender classification accuracy in *k*NN, SVMs, and SVMACs: (a) MLBP feature; (b) LBP feature; (c) Gabor feature; and (d) Gray feature. Here both SVMs and SVMACs use a RBF kernel, and $m = 9 \times 9$ blocks for each facial image are selected.

From Table 4, and Fig. 6, we conclude the following three observations: a) the average classification accuracy achieved by SVMACs is higher than that of traditional SVMs and *k*NN [21]; b) the corresponding standard deviation brought by SVMACs is lower than that of traditional SVMs and *k*NN in most cases; and c) the maximum improvement achieved by SVMACs on classification accuracy reaches 3.0%. We can also see that SVMACs achieve a largest improvement on classification accuracy when the Gabor and Gray features are used. This demonstrates that SVMACs can control more influence of noise data by introducing the label confidence value for each training sample. The reason is that there are more noises in the Gabor and Gray features than those in the LGBP, MLBP, and LBP features.

In addition, we observe that the classification accuracy is also dependent on the distributions of training samples. Generally speaking, there are two kinds of sample distributes as illustrated in Fig. 2. One is dense and the other is sparse. In this situation, If the confidence values less than 1 are set for the support vector samples, the decision boundary obtained by SVMACs will favor the sparse samples in comparison with traditional SVMs. Consequently, from both experimental results and theoretical analysis, we see that SVMCs separate the data samples more reasonably by modifying the confidence values of the support vector training samples. In a word, SVMACs can improve gender classification accuracy, regardless of the high-dimension features (LGBP, MLBP, and LBP) and low-dimension features (Gabor and Gray).

5. Conclusions and Future Work

We have proposed a novel support vector machine classifier with automatic confidence and introduced a simple algorithm for calculating the label confidence value of each training sample. We have derived the quadratic programming problem for this new SVM and discussed its generalization performance through an illustrative example. The most important advantage of the proposed SVMs over traditional SVMs is that some explicit human prior knowledge estimated by our confidence labeling algorithm on training samples can be incorporated into learning. By using a gender classification task based on facial images, we have shown that the manual confidence and automatic confidence are quite consistent in most cases. Experimental results on three benchmaring problems and a gender classification task indicate that the proposed SVMs can improve generalization performance, especially when the input features have much noise.

As future work, we would like to study the bound for the improvement on classification accuracy theoretically and apply SVMACs to other real-world pattern classification problems such as text classification and age estimation.

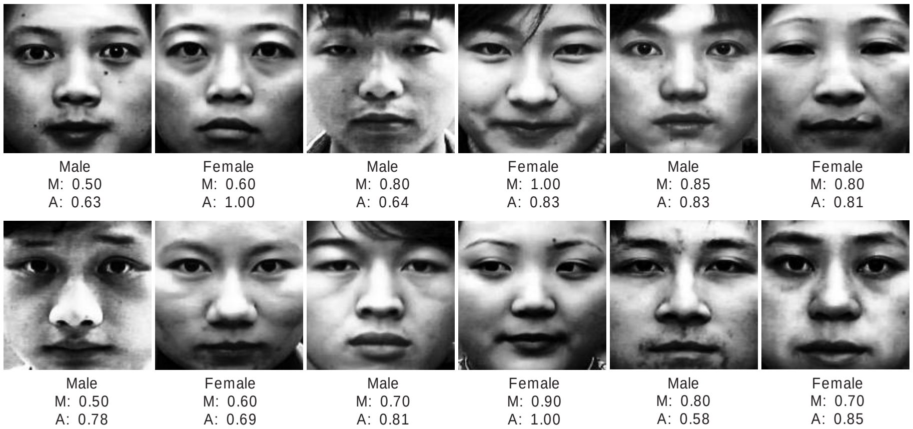


Figure 7: Examples of facial images and their confidence values. Here "M" and "A" denote the confidence values labeled manually and calculated automatically, respectively. These facial images are numbered from 1 to 12 from left to right and from upper to lower.

Acknowledgments

The authors would like to thank Dr. Si Wu for his useful discussions and suggestions on this paper. This work was partially supported by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), the National High-Tech Research Program of China (Grant No. 2008AA02Z315), and the Science and Technology Commission of Shanghai Municipality (Grant No. 09511502400).

References

[1] V. N. Vapnik, Statistical Learning Theory, New York:Wiley, 1998.

[2] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Publishing House of Electronics Industry, 2004.

[3] Z. Ji, W. Y. Yang, S. Wu, B. L. Lu, Encoding Human Knowledge for Visual Pattern Recognition, Submitted to Knowledge-Based & Intelligent Engineering Systems, 2009.

[4] B. B. Lu, X. L. Wang, M. Utiyama, Incorperating prior knowledge into learning by dividing training data, Frontiers of Computer Science 3 (1) (2009) 109–122.

[5] B. Jiang, X. G. Zhang, T. X. Cai, Estimating the confidence interval for prediction errors of support vector machine classifiers, Journal of Machine Learning Research 9 (2008) 521–540.

[6] C. L. Liu, Classifier combination based on confidence transformation, Pattern Recognition 38 (1) (2005) 11–28.

[7] J. Peng, D. R. Heisterkamp, H. K. Dai, Lda/svm driven nearest neighbor classification, IEEE Transactions on Neural Networks 14 (4) (2003) 158–163.

[8] A. Graf, F. Wichmann, H. Bülthoff, B. Schölkopf, Classification of faces in man and machine, Neural Computation 18 (1) (2005) 143–165.

[9] J. Feng, P. Williams, The generalization error of the symmetric and scaled support vector machine, IEEE Transactions on Neural Networks 12 (5) (2001) 1255–1260.

[10] Z. Ji, B. L. Lu, Gender classification based on support vector machine with automatic confidence, 16th International Conference on Neural Information Processing, Bangkok, Thailand, LNCS, Springer Berlin (2009) 685–692.

[11] Z. Ji, B. L. Lu, Web Link: Support Vector Machine with Automatic Confidence (SVMAC), http://bcmi.sjtu.edu.cn/ jizheng/, 2010.

[12] E. Mäkinen, R. Raisamo, An experimental comparison of gender classification methods, Pattern Recognition Letters 29 (10) (2008) 1544–1556.

[13] H. C. Lian, B. L. Lu, Multi-view gender classification using local binary patterns and support vector machines, Proceedings of the Third International Symposium on Neural Networks (2006) 202–209.

[14] H. C. Lian, B. L. Lu, Multi-view gender classification using multi-resolution local binary patterns and support vector machines, International Journal of Neural Systems 17 (6) (2007) 479–487.

[15] X. C. Lian, B. L. Lu, Gender classification by combining facial and hair information, 15th International Conference on Neural Information Processing, Auckland, New Zealand, LNCS, Springer Berlin (2008) 654–661.

[16] B. Moghaddam, M. H. Yang, An improved training algorithm for support vector machines, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 707–711.

[17] W. Gao, B. Cao, S. G. Shan, X. L. Chen, D. L. Zhou, X. H. Zhang, D. B. Zhao, The cas-peal large-scale chinese face database and baseline evaluations, IEEE Transactions on System, Man and Cybernetics Part A: Systems and Humans 38 (1) (2008) 149–161.

[18] T. Ojala, M. Pietikainen, T. Maeopaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[19] T. Ahonen, A. Hadid, M. Pietikainen, Face recognition with local binary patterns, Computer Vision Proceedings 3021/2004 (2004) 469–481.

[20] B. Xia, H. Sun, B. L. Lu, Multi-view gender classification based on local gabor binary mapping pattern and support vector machines, IEEE International Joint Conference on Neural Networks (2008) 3388–3395.

[21] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience, 2000.