

An adaptive error penalization method for training an efficient and generalized SVM

Yiqiang Zhan^{a, b, c, *}, Dinggang Shen^{a, c}

^aSection of Biomedical Image Analysis, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA

^bDepartment of Computer Science, The Johns Hopkins University, Baltimore, MD, USA

^cCenter for Computer-Integrated Surgical Systems and Technology, The Johns Hopkins University, Baltimore, MD, USA

Received 7 June 2005; received in revised form 14 September 2005

Abstract

A novel training method has been proposed for increasing efficiency and generalization of support vector machine (SVM). The efficiency of SVM in classification is directly determined by the number of the support vectors used, which is often huge in the complicated classification problem in order to represent a highly convoluted separation hypersurface for better nonlinear classification. However, the separation hypersurface of SVM might be unnecessarily over-convoluted around extreme outliers, as these outliers can easily dominate the objective function of SVM. This situation eventually affects the efficiency and generalization of SVM in classifying unseen testing samples. To avoid this problem, we propose a novel objective function for SVM, i.e., an adaptive penalty term is designed to suppress the effects of extreme outliers, thus simplifying the separation hypersurface and increasing the classification efficiency. Since maximization of the margin distance of hypersurface is no longer dominated by those extreme outliers, our generated SVM tends to have a wider margin, i.e., better generalization ability. Importantly, as our designed objective function can be reformulated as a dual problem, similar to that of standard SVM, any existing SVM training algorithm can be borrowed for the training of our proposed SVM. The performances of our method have been extensively tested on the UCI machine learning repository, as well as a real clinical problem, i.e., tissue classification in prostate ultrasound images. Experimental results show that our method is able to simultaneously increase the classification efficiency and the generalization ability of the SVM.

© 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Support vector machine; Training method; Computational efficiency; Generalization ability

1. Introduction

Support vector machine (SVM) is a new generation of learning systems based on statistical learning theory [1]. Considering a two-class classification problem with m labeled training samples, $\{(\vec{x}_i, y_i) | \vec{x}_i \in R^n, y_i \in \{-1, 1\}, i = 1 \dots m\}$, SVM aims to generate a hypersurface that has maximum margin to separate two classes. The classification of a testing sample is accomplished by calculating its distance

to the hypersurface:

$$d(\vec{x}) = \sum_{i=1}^m \alpha_i y_i K(\vec{x}_i, \vec{x}) + b, \quad (1)$$

where α_i and b are the parameters determined by SVM's learning algorithm, and $K(\vec{x}_i, \vec{x})$ is the kernel function. Samples \vec{x}_i with nonzero parameters α_i are called "support vectors".

Since its generation in 1995 [1], SVM has drawn considerable attentions in various research areas [3–8] due to its striking properties as described next. First, based on the idea of structural risk minimization [1], SVM can achieve high generalization ability by minimizing the Vapnik–Chervonenkis

* Corresponding author. Department of Computer Science, The Johns Hopkins University, Baltimore, MD, USA. Tel.: +1 215 662 7362.

E-mail addresses: yzhan@cs.jhu.edu (Y. Zhan), Dinggang.Shen@uphs.upenn.edu (D. Shen).

dimension. Second, by using the kernel trick [2], the samples are implicitly mapped to a higher dimensional space. Therefore, SVM can generate a convoluted hypersurface to nonlinearly separate different classes. Finally, the training procedure of SVM can be eventually formulated as a constraint quadratic optimization problem, which has a unique global minimum.

However, although SVM shows superior classification ability in pattern recognition problems, it usually needs a huge number of support vectors to parameterize the separation hypersurface, particularly when confronting large data classification problems. Since the calculation of the decision function with many nonzero parameters α_i in Eq. (1) is very time consuming, SVM exhibits substantially slower classification speed compared to the neural network [9]. This disadvantage unavoidably limits the capability of SVM in the applications that require a massive number of classifications [5] or real-time classification [10].

In this paper, we propose a novel training method to increase the efficiency as well as the generalization ability of SVM. We notice that the extreme outliers in the training set usually make the separation hypersurface unnecessarily over-convoluted, thus affecting both the efficiency and the generalization of SVM. This problem is actually resulted from the domination of the extreme outliers over the objective function of the standard soft-margin SVM [12]. To overcome this problem, we reformulate the objective function by designing an adaptive penalty term to suppress the effects of extreme outliers in objective function, thereby simplifying the separation surface and increasing the generalization ability of SVM. Importantly, we find that our designed objective function can be reformulated as a quadratic optimization problem with adaptive constraints, which is similar to the dual problem of the standard soft-margin SVM. Therefore, any existing SVM training method can be borrowed for training our proposed SVM.

The remainder of this paper is organized as following. In Section 2, we will first analyze the problem in details. Then, the reformulated objective function with an adaptive penalty term to outliers is presented. The training method for the reformulated SVM will also be provided. Section 3 will present the experimental results of our method on the UCI machine learning repository, as well as a real clinical problem, i.e., tissue classification in a set of prostate ultrasound images. This paper concludes in Section 4.

2. Methods

2.1. Problem description

As indicated in Eq. (1), the computational cost of SVM is determined by the number of the support vectors, i.e. training samples with nonzero parameters α_i . According to their relative positions to the separation hypersurface,

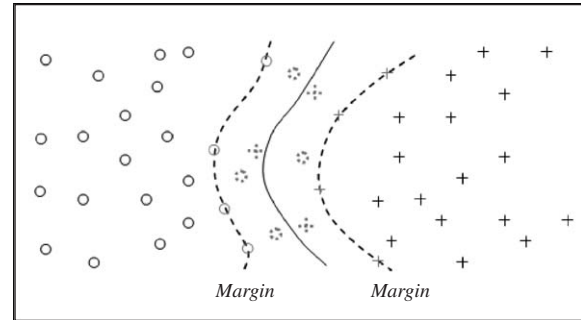


Fig. 1. Schematic explanation of the separation hypersurface (solid curves), margins (dashed curves), and support vectors of SVM (grey circles/crosses). The positive and the negative training samples are indicated by circles and crosses, respectively.

support vectors can be categorized into two types. The first type of support vectors are the training samples that exactly locate on the margins of the separation hypersurface, i.e., $d(\vec{x}_i) = \pm 1$, such as solid grey circles/crosses shown in Fig. 1. The second type of support vectors are the training samples that locate beyond their corresponding margins, i.e., $y_i d(\vec{x}_i) < 1$, such as dashed grey circles/crosses shown in Fig. 1. For a SVM, the second type of support vectors are regarded as misclassified samples, although some of them still locate at the correct side of the hypersurface.

SVM usually has a huge number of support vectors, when the distributions of the positive and the negative training samples from a large dataset highly overlap with each other. This unfavorable situation is resulted from two reasons: (1) a large number of the first-type support vectors are needed to construct a highly convoluted hypersurface, in order to separate two classes; (2) even the highly convoluted separation hypersurface has been constructed, a lot of confounding samples will be misclassified, thus selected as the second type of support vectors.

Some support vectors might be redundant to parameterize the separation hypersurface. Based on this hypothesis, researchers have proposed efficient SVM training methods [11,13]. Compared to Ref. [13], the method proposed by Osuna and Girosi in Ref. [11] is more feasible, and it offered a principle for controlling the accuracy of approximation. This method approximates the separation hypersurface with a subset of the support vectors by using a Support vector regression machine (SVRM). If the separation hypersurface is relatively simple, Osuan's method is quite effective to reduce the number of support vectors without system degradation. However, in many large dataset classification problems, SVM usually generates a locally over-convoluted separation hypersurface, which is difficult to be parameterized by a small number of support vectors as Osuan's method did. Therefore, in order to further decrease the number of support vectors, it is necessary to simplify the hypersurface without losing its classification ability.

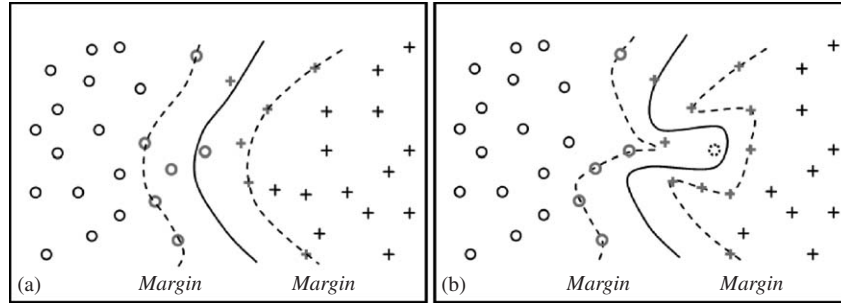


Fig. 2. Schematic explanation of the over-convoluted separation hypersurface incurred by an outlier. The solid and dashed curves denote the separation hypersurface and margins, respectively. Circles and crosses denote the positive and the negative training samples, respectively. The grey ones denote the support vectors. The dashed circle in (b) is an outlier that incurs the hypersurface over-convoluted. Notably, except the dashed circle in (b), other samples in (a) and (b) are identical.

It is widely accepted that the convoluted hypersurface of SVM is important for nonlinear separation of two classes that might overlap with each other in the original feature space. However, in certain cases, the hypersurface generated by SVM is unnecessarily over-convoluted in some local regions without increasing the classification ability of the SVM. Fig. 2 presents a toy problem to illustrate those cases. In Fig. 2(a), the separation hypersurface of the SVM is relatively simple and it has 12 support vectors (denoted by grey crosses or circles). The distribution of the training samples in Fig. 2(b) is almost the same as that of Fig. 2(a) except an additional positive sample (denoted by the dashed circle). However, the separation hypersurface in Fig. 2(b) became much more convoluted, in order to satisfy this additional sample, and the trained SVM has 16 support vectors. Notably, since this additional training sample locates in an isolated region that is far from samples of the same class, it might be an outlier produced by noise or error. Therefore, the over-convoluted hypersurface, used to satisfy this sample, will not increase the generalization ability of SVM but its computational cost. Obviously, this unfavorable situation should be avoided, in order to increase the classification efficiency and generalization of the SVM. In the next section, we will investigate this problem in detail, and finally prevent it by reformulating a new objective function for SVM.

2.2. Reformulation of objective function in SVM

It is necessary to briefly introduce the objective function of SVM, before investigating the reason of obtaining over-convoluted separation hypersurface in some classification cases. According to the statistical learning theory [1], SVM tries to generate a separation hyperplane, $\vec{w} \cdot \vec{x} + b = 0$, which has the maximum generalization ability. Here, \vec{w} is the normal of the hyperplane, and b is the distance from the hyperplane to the origin. Given m labeled training samples, i.e. $\{(\vec{x}_i, y_i) | \vec{x}_i \in R^n, y_i \in \{-1, 1\}, i = 1 \dots m\}$, the

training of SVM can be formulated as solving a quadratic optimal problem:

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

Here, $\|\cdot\|$ is the norm of a vector, and $\phi(\cdot)$ maps samples into a higher dimensional space and can be implicitly implemented by the kernel trick [2].

In the objective function of Eq. (2), the first term $\|\vec{w}\|$ measures the inverse of the margin distance that should be minimized to obtain the minimum structural risk [2]. The second term is a penalty term consisting of a number of nonnegative slack variables ξ_i , used to construct a soft margin hyperplane [12]. By using the relaxed separation constraints $y_i (\vec{w} \cdot \phi(\vec{x}_i) + b) \geq 1 - \xi_i$, some training samples are allowed to locate beyond their corresponding margins, i.e. $y_i (\vec{w} \cdot \phi(\vec{x}_i) + b) < 1$. The linear summation of all slack variables ξ_i is constrained as the second term in the objective function, in order to avoid the trivial solution that all ξ_i take large values.

According to Eq. (2), it is not difficult to understand the reason why the case described in Fig. 2 happens. The separation hypersurface in Fig. 2(b) has to be convoluted in order to satisfy that additional red positive sample; otherwise, the corresponding ξ_i of that additional sample will be very large, thus dominating over the objective function. However, as the additional sample locates in an isolated region far from samples of the same class, it is an outlier that might be generated by noise or error. Therefore, the hypersurface convoluted around the additional sample is unnecessary, and it will not increase the generalization of the trained SVM, but its computational cost.

To solve this problem, we introduce a nonlinear penalty term, instead of the linear penalty term in Eq. (2), which makes the effect of outliers overwhelming over the whole objective function. The objective function of SVM is

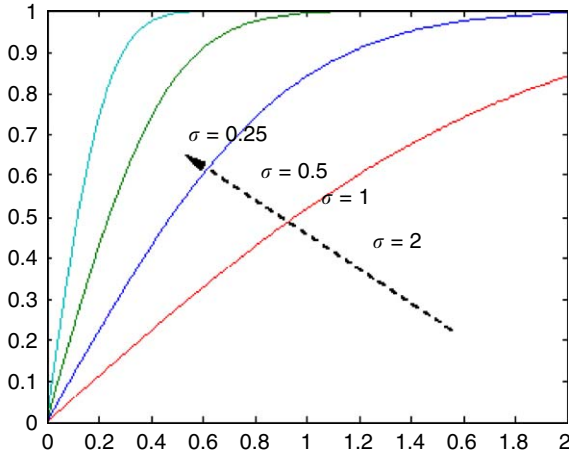


Fig. 3. A nonlinear error function for suppressing the slack variables with large value, thus adaptively penalizing outliers. The curves of different colors denote the error functions with respect to different parameter σ used. The dashed arrow indicates the decrease of σ with the progress of iterative training, i.e., transferring linear penalty to adaptive nonlinear penalty.

reformulated as following:

$$\begin{aligned} \min_{\vec{w}, b, \xi_i} \quad & \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^m \text{erf}(\xi_i; \sigma) \\ \text{s.t.} \quad & y_i (\vec{w} \cdot \phi(\vec{x}_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0, \end{aligned} \quad (3)$$

where $\text{erf}(\xi; \sigma)$ is a nonlinear error function, defined by $\text{erf}(\xi; \sigma) = (2/\sqrt{\pi}\sigma) \int_0^\xi e^{-z^2/\sigma^2} dz$, to adaptively penalize outliers.

As indicated by the function plot in Fig. 3, the error function $\text{erf}(\cdot)$ will suppress the slack variables when they are large. In this way, the objective function will be no longer dominated by the large slack variables, and thus the resulting separation hypersurface will not be over-convoluted around extreme outliers. On the other hand, if there are a considerable number of same-class samples clustering in an isolated region that is distant from other samples of this class, which means they are not outliers but the training samples reflecting the statistical characteristics of the problem under study, the generated hypersurface could still be convoluted to satisfy these samples, in order to decrease the total error penalty for these samples. In this way, the generalization ability of the SVM is not influenced.

Moreover, the first term of the objective function is related with the confidence interval between the empirical risk and the actual risk [12], i.e., the less the value of $\|\vec{w}\|$, the larger the generalization ability of SVM. Since $\|\vec{w}\|$ in the reformulated objective function can be minimized without being dominated by those extreme outliers in the second term, the confidence interval can be decreased and thus the generalization ability of the reformulated SVM is expected to increase in many cases.

2.3. Training of the reformulated SVM

In this section, we will discuss the training algorithm for the reformulated SVM. Similarly, the Lagrangian theory is employed here to solve the reformulated constrained quadratic problem. After introducing Lagrangian multipliers α_i and η_i , we obtain the primary Lagrangian of Eq. (3) as:

$$\begin{aligned} L(\vec{w}, b, \xi_i) &= \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^m \text{erf}(\xi_i) \\ &\quad - \sum_{i=1}^m [\alpha_i (y_i (\vec{w} \cdot \phi(\vec{x}_i) + b) - (1 - \xi_i)) + \eta_i \xi_i]. \end{aligned} \quad (4)$$

According to Kuhn–Tucker condition, we can express Eq. (4) as a dual problem given next, by differentiating Eq. (4) with respect to the primary variables \vec{w} and b , setting the derivatives as zero and finally resubmitting the relations obtained by these equations.

$$\begin{aligned} \min_{\alpha_i, \xi_i} \quad & \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j - \sum_{i=1}^m \alpha_i \\ & + C \sum_{i=1}^m [\xi_i \text{erf}'(\xi_i; \sigma) - \text{erf}(\xi_i; \sigma)] \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \cdot \text{erf}'(\xi_i; \sigma), \quad \xi_i \geq 0, \end{aligned} \quad (5)$$

where $\text{erf}'(\xi_i)$ is the derivative of the nonlinear error function, i.e., a Gaussian function with the standard deviation σ .

Notably, compared to the dual problem expressed for the standard SVM, the Lagrangian multiplier α_i in Eq. (5) is no longer constrained by a global constant C , but $C \cdot \text{erf}'(\xi_i; \sigma)$, which is adaptive to each sample according to its corresponding ξ_i . Since $\text{erf}'(\xi_i; \sigma)$ is actually a Gaussian function, the Lagrangian multiplier α_i of a training sample with large slack variable ξ_i is actually restricted by a very low upper bound. Therefore, this sample has very little contribution in constructing the separation hypersurface, even if it is selected as a support vector. Actually, the reformulated SVM offers a soft selection mechanism to adaptively determine the importance of different training samples, thus the effect of the outliers is suppressed. In another way, our method can be interpreted as an algorithm of adaptively and softly selecting samples to re-evaluate the objective function. In certain sense, our method can be considered as an extension from the method proposed by Lee and Mangasarian [14], which reformulate the objective function with a subset of randomly selected training samples, in order to speed up SVM.

Since the objective function in Eq. (5) has a very similar format with respect to the standard SVM, we can design an iterative framework to train the reformulated SVM, by borrowing any existing SVM training methods. First, α_i are optimized using a training method for the standard SVM,

except using the adaptive constraints $0 \leq \alpha_i \leq C \cdot \text{erf}(\xi_i; \sigma)$. Then, ξ_i can be calculated by the following equation,

$$\xi_i = \sum_{i=1}^m \alpha_i y_i K(\vec{x}_i, \vec{x}) + b - 1. \quad (6)$$

In the initial iterations, the parameter σ is set to be a large value, thus the error function performs as a linear slack term used in the standard SVM (c.f. Fig. 3). With progress of the training, the parameter σ becomes smaller and smaller, thereby the nonlinear error function starts to suppress the large slack variables more. After the training procedure converges, we generate an optimal separation hypersurface for the reformulated SVM. Finally, Osuna's method [11], which employs the SVRM to approximate the hypersurface by a subset of support vectors, is employed to further decrease the number of support vectors and thus increase the classification efficiency of the SVM.

The complete training method for our reformulated SVM can be summarized as follows:

Step 1: Initialization. Set the penalty factor to each training sample as $C_i = C$; and the variable σ in the nonlinear error function $\text{erf}(\xi; \sigma)$ as $\sigma = \sigma_0$. In our experiments, we select $C = 100$ and $\sigma_0 = 100$.

Step 2: Iterative training:

2a. Train a tentative SVM by a *SVM Torch* method [16], with the adaptive penalty factor C_i . (Note that any existing SVM training method can actually be borrowed to train the tentative SVM in step 2a, as long as the adaptive penalty factors can be embedded into the training procedure.)

2b. Calculate ξ_i using Eq. (6), with α_i and b determined by the training procedure for the tentative SVM in step 2a.

2c. Decrease the value of σ by $\sigma = \sigma/s$, where s is larger than 1.0, i.e., $s = 1.12$ used in our experiments. Then update C_i by $C_i = C \cdot \text{erf}(\xi_i; \sigma)$.

2d. If the value of σ is sufficiently small, go to step 3; otherwise, go to step 2a.

Step 3: Employ Osuna's method [11] to further decrease the number of support vectors.

3. Experiments

To validate the effectiveness of our method, we applied it to the UCI machine learning repository [15], as well as a real clinical problem, i.e., tissue classification in the prostate ultrasound images. The experimental results are presented next.

3.1. Experiments on UCI machine learning repository

UCI machine learning repository [15] contains a set of datasets that is used by the machine learning community for the empirical analysis of machine learning algorithms. Among these datasets, since the "Adult" dataset has a large size (45,222 samples, 14 features) and standard training and

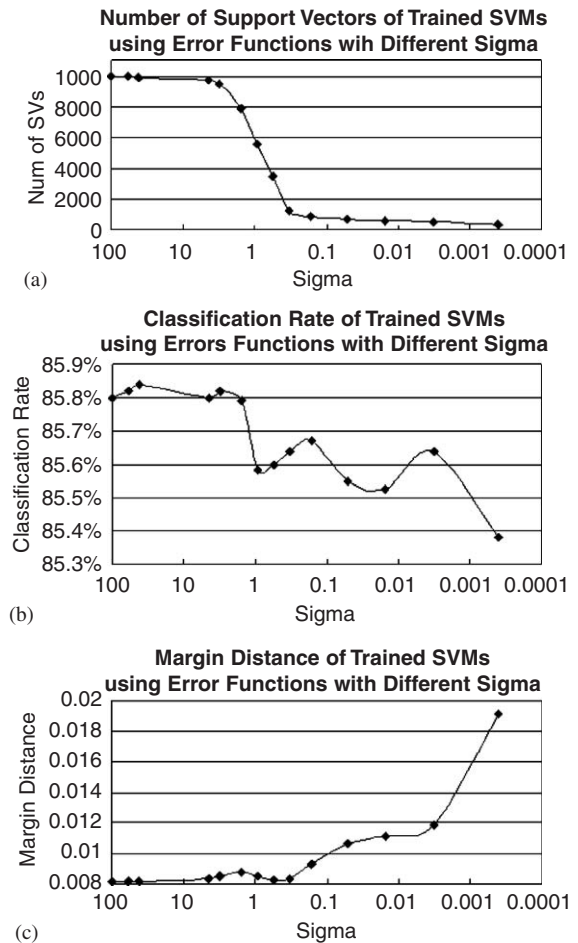


Fig. 4. Performance of the tentatively trained SVM with the decrease of σ in error function $\text{erf}(\xi_i; \sigma)$, during the training procedure. (a) The number of support vectors (SVs) used in tentatively trained SVM; (b) the classification rate by the tentatively trained SVM; (c) the margin distance of the tentatively trained SVM.

testing subsets (30,162 samples for training, 15,060 samples for testing), it is selected first to validate our method. Using the proposed method, we can obtain a SVM with 338 support vectors and 85.38% classification rate. Compared to the standard SVM, which has 9996 support vectors and 85.80% classification rate, the efficiency of SVM is dramatically increased while the loss of classification rate is small.

The performance of SVM is further investigated, with respect to the different parameters σ used in error function $\text{erf}(\xi_i; \sigma)$ during the training procedure. Notably, the parameter σ is decreased during the training procedure, in order to transfer the linear penalty into the adaptive nonlinear penalty in the objective function. Therefore, the performance of tentatively constructed SVM in different training stages can be compared, as shown in Fig. 4. According to Fig. 4(a), the number of support vectors is quickly decreased with the decrease of parameter σ . However, the classification rate of tentatively trained SVM does not degrade with the decrease of σ ; actually, it even increases in a certain range of σ . Fig. 4(c) further shows the trend of the margin distance of

Table 1

Comparison on the classification performances obtained by standard SVM, Osuna’s method, our method without step 3, and our complete method

Dataset	Standard SVM		Osuna’s method		Our method without step 3		Our method	
	# SV	Correct rate	# SV	Correct rate	# SV	Correct rate	# SV	Correct rate
Adult	9996	85.80%	1000	84.91%	650	85.58%	338	85.38%
Breast	50.8	96.46%	6.8	96.92%	9.2	97.15%	7.1	97.25%
Ionosphere	161.3	93.43%	160.4	93.43%	64.4	94.25%	37.5	94.28%
Monks	96.8	97.51%	53.9	97.50%	28.2	97.20%	25.9	97.00%

SV below denotes ‘support vector’.

tentatively trained SVM, i.e., increasing with the decrease of σ , implying that the confidence interval between empirical risk and actual risk [12] is reduced. Therefore, by stopping the training procedure at an appropriate parameter σ , we can increase the generalization ability as well as the efficiency of the finally trained SVM.

We also select “Breast Cancer” (687 samples, 10 features), “Ionosphere” (351 samples, 34 features), and “Monks” (432 samples, 6 features) to further test our method. For these three datasets, we randomly divided each dataset into several groups, with each group having 50 samples. In the training stage, one group is left out as testing samples, and other remaining groups are used as training samples. This leave one-group-out cross validation is repeated using standard SVM, Osuna’s method and our proposed method with/without step 3, i.e., employing Osuna’s method to further reduce the number of support vectors. For all these experiments, the SVMs use the Gaussian kernel with the standard deviations 100.0, 100.0, 1.5, 10.0 for “Adult”, “Breast”, “Ionosphere” and “Monks” datasets, respectively.

The averages on the correct classification rate and the number of support vectors from all tests are reported in Table 1 for comparison. Compared to the standard SVM, our method is able to dramatically reduce the number of support vectors while the generalization ability of the classifier is increased. Compared to Osuna’s method, our method is able to generate more efficient SVM, except for the “Breast Cancer” dataset that has a relatively simple distribution of samples. Also, the correct classification rates obtained by our method are slightly increased, except for “Monks” dataset.

3.2. Experiments on tissue classifications in prostate ultrasound images

In our study of 3D prostate segmentation from ultrasound images [5], SVM is used for texture-based tissue classification. The input of SVM is a set of texture features extracted by the Gabor filter bank [17], and the output is a soft label denoting the likelihood of the voxel belonging to the prostate. In this way, prostate tissues are differentiated from the surrounding tissues. In this study, the computational cost of SVM for tissue classification is a particularly critical problem to be concerned, as the tissue classification

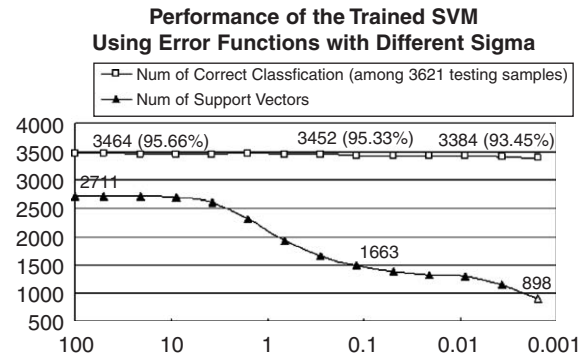


Fig. 5. Performance of tentatively trained SVM in prostate tissue classification, with respect to different σ used in error function $erf(\xi_i; \sigma)$.

is operated for lots of times (i.e., 10^6) in the segmentation stage and also the real-time segmentation is usually required for clinical applications. Therefore, the training method proposed in this paper is applied to speeding up the SVM for tissue classification.

In preparing the experimental dataset, we first randomly select prostate and nonprostate samples from six manually labeled ultrasound images, in which 3621 samples from one ultrasound image are used as testing samples and 18,105 samples from other five images are used as training samples. Each sample has 10 texture features, extracted by Gabor filters.

In validating our proposed method on the prostate dataset, we select a Gaussian Kernel with the standard deviation 100 for the SVMs. We also gradually decreased the parameter σ of the error function $erf(\xi_i; \sigma)$ and generated a number of tentatively trained SVMs. The number of support vectors and the classification rate of tentatively trained SVM are provided in Fig. 5. As shown in Fig. 5, the number of support vectors is reduced quickly with the decrease of σ , while the classification rate keeps similar. By stopping training at $\sigma = 0.0016$, we obtain a SVM with 898 support vectors, which is only 33.1% of those of the original SVM (2711); but its classification rate still reaches 93.45%. Compared to 95.66% classification rate achieved by the original SVM, the loss of classification rate is relatively small; thereby it will not affect the performance of our model-based segmentation algorithm [5]. Notably, if simultaneously requiring

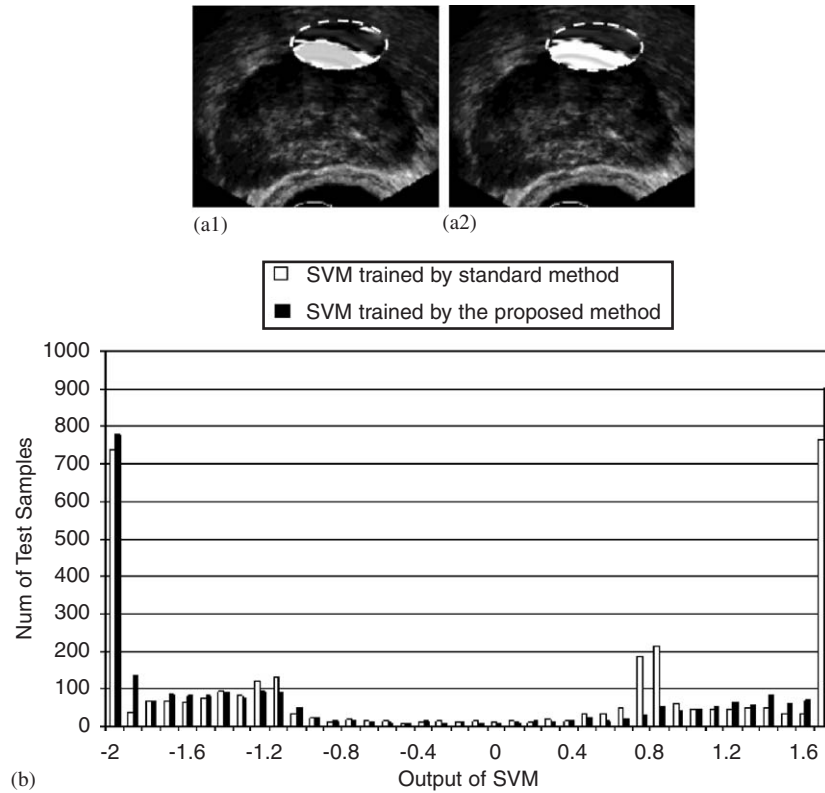


Fig. 6. Comparison on tissue classification results obtained by (a1) the standard SVM with 2711 support vectors, and (a2) our trained SVM with 898 support vectors. The tissue classification results are shown only in the regions surrounded by dashed ellipsoids. (b) Histograms of classification outputs on a testing dataset, with black bars representing the results obtained by our trained SVM and white bars representing the results obtained by the standard SVM.

classification efficiency and classification rate, we might stop our training procedure at a larger σ , i.e., $\sigma = 0.28$, in order to generate a SVM with 1663 support vectors for a more accurate classification, i.e., 95.33%

To further validate the performance of our trained SVM in tissue classification, the SVM with 898 support vectors (denoted by the white triangle in Fig. 5) is applied to a real ultrasound image for tissue classification. By comparing results in Fig. 6(a1) and (a2), our result in Fig. 6(a2) displays higher contrast between segmented prostate and nonprostate tissues, compared to that obtained by the original SVM with 2711 support vectors in Fig. 6(a1). This result can be further approved by the histograms of the classifications as shown in Fig. 6(b). The reason behind this result is that our trained SVM has larger margin distance (i.e., 0.027) than that of the standard SVM (i.e., 0.007).

We further compare the performances of SVMs generated by different training methods. Four methods are implemented for comparison: (1) a method of slackening the training criterion by decreasing the linear penalty factor [2]; (2) a heuristic method, which assumes the training samples distributing in a multi-variant Gaussian way, then excludes the “outliers” distant from the respective distribution centers, and finally trains a SVM only by the remaining samples; (3) Osuna’s method [11]; (4) our proposed method.

The performances of these four methods are evaluated in Fig. 7(a), by the number of support vectors used versus the number of correct classifications achieved. In these four methods, our proposed method is the most effective in reducing the number of support vectors.

The classification abilities of two SVMs, respectively trained by Osuna’s method and our proposed method, are further compared. The SVM trained by Osuna’s method, as denoted by the black triangle in Fig. 7(b), needs 901 support vectors and its classification rate is 92.81%. The SVM trained by our proposed method, as denoted by the black circle in Fig. 7(b), needs only 865 support vectors, while its classification rate is 93.10%, higher than that produced by Osuna’s method. Moreover, our trained SVM actually has much better generalization ability than the SVM trained by Osuna’s method, once checking the histograms of their classification outputs. As shown in Fig. 8, the classification outputs of Osuna’s SVM concentrate around 0, which means the margins between the positive and the negative samples are narrow. In contrast, most classification outputs of our trained SVM are either larger than 1.0 or smaller than -1.0 . This experiment further proves that our training method is better in achieving the generalization ability of the SVM.

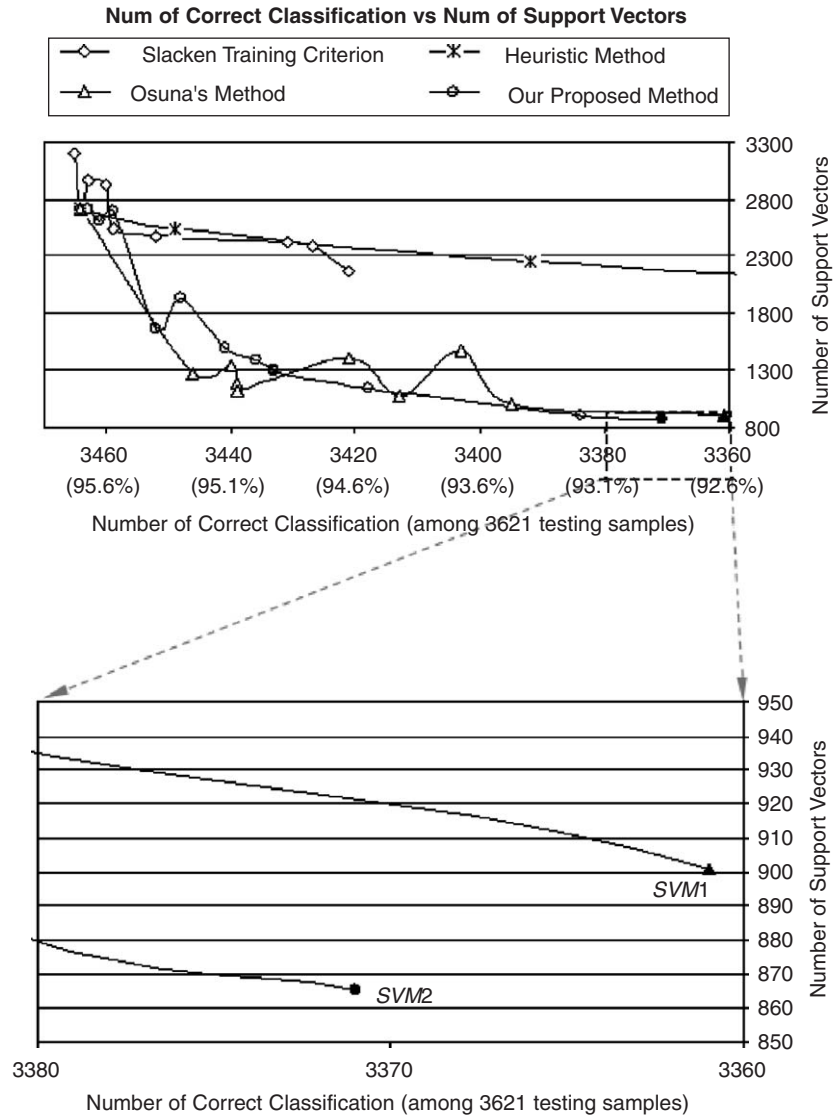


Fig. 7. (a) Comparison on the performances of four training methods in increasing the classification efficiency of SVM. (b) The zoomed version of the area surrounded by the dashed rectangle in (a), for clearly illustrating the classification rate and the number of support vectors obtained by the two SVMs under comparison.

4. Conclusion

In this paper, we proposed a novel training method to increase the classification efficiency as well as the generalization ability of the SVM. We noted that the optimal separation hypersurface generated by the standard training method might be unnecessarily over-convoluted around extreme outliers, thus requesting more computational cost and decreasing the generalization ability. This situation is actually resulted from the slack variables of outliers that dominate over the objective function, since all slack variables are linearly summed. To overcome this problem, we introduced a nonlinear mapping function to suppress the large slack variables of the outliers. Thus, the separation hypersurface can be simplified and the number of support vectors can be reduced. On the other hand, since the term in the

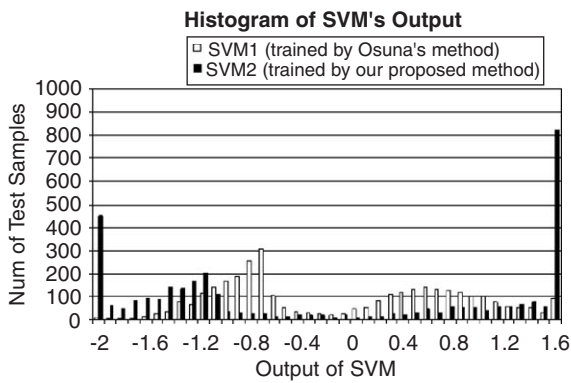


Fig. 8. Histograms of classification outputs on a testing dataset, respectively by our trained SVM (black bars) and Osuna's SVM (white bars).

objective function for measuring the margin distance of SVM is no longer dominated by extreme outliers, the reformulated SVM can obtain a larger margin distance and thus achieve better generalization ability. By using Lagrangian theory, the reformulated SVM can be transformed to a dual problem, similar to that of the standard SVM. Therefore, we design an iterative framework to train the reformulated SVM, by borrowing any existing SVM training algorithm.

Our method has been tested on the UCI machine learning repository, as well as a real clinical problem, i.e., tissue classification in prostate ultrasound images. Compared to the SVM trained by the standard training method and Osuna's method, our method is able to achieve a much more efficient SVM with higher generalization ability.

References

- [1] V.N. Vapnik, *The Natural of Statistical Learning Theory*, Springer, New York, 1995.
- [2] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Min. Knowl. Discovery* 2 (1998) 121–167.
- [3] E. Osuna, R. Freund, F. Giosi, Training support vector machines: an application to face detection, *Proc. IEEE. Conf. Comput. Vision Pattern Recognition* (1997) 130–136.
- [4] T. Joachims, A statistical learning model of text classification for support vector machines, in: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, 2001.
- [5] Y. Zhan, D. Shen, Automated segmentation of 3D US prostate images using statistical texture-based matching method, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, November 16–18, Canada, 2003.
- [6] V. Wan, S. Renals, Speaker verification using sequence discriminant support vector machines, *IEEE Trans. Speech Audio Process.* 13 (2) (2005) 203–210.
- [7] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr., D. Haussler, Knowledge-based analysis of microarray gene expression data by using support vector machines, *Genetics* 97 (1) (2004) 262–267.
- [8] Z. Lao, D. Shen, Z. Xue, B. Karacali, S.M. Resnick, C. Davatzikos, Morphological classification of brains via high-dimensional shape transformations and machine learning methods, *NeuroImage* 21 (1) (2004) 46–57.
- [9] Y. Lecun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drunker, I. Guyon, U. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of learning algorithms for handwritten digit recognition, *Int. Conf. Artif. Neural Networks* (1995) 53–60.
- [10] Y.-L. Tian, L. Brown, A. Hampapur, S. Pankanti, A. W. Senior, R.M. Bolle, Real world real-time automatic recognition of facial expressions, *IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, Graz, Austria, March 31, 2003.
- [11] E. Osuna, F. Giosi, Reducing the Run-time Complexity of Support Vector Machines, *ICPR*, Brisbane, Australia, 1998.
- [12] B. Scholkopf, A.J. Smola, *Learning with Kernels*, The MIT Press, Cambridge, 2002.
- [13] C.J.C. Burges, Simplified support vector decision rules, in: *Proceedings of the 13th International Conference on Machine Learning*, 1996, pp. 71–77.
- [14] Y.-J. Lee, O.L. Mangasarian, RSVM: Reduced support vector machines, in: *Proceedings of the First SIA International Conference on Data Mining*, 2001.
- [15] S. Hettich, C.L. Blake, C.J. Merz, Repository of machine learning databases, <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>, 1998.
- [16] R. Collobert, S. Bengio, SVM Torch: support vector machines for large-scale regression problems, *J. Mach. Learn. Res.* 1 (2001) 143–160.
- [17] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (1996) 837–842.