Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

PERGAMON

Pattern Recognition 38 (2005) 157–161

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Rapid and brief communication

Design efficient support vector machine for fast classification

Yiqiang Zhan[a,b,c,*], Dinggang Shen[b,c]

[a]Department of Computer Science, The Johns Hopkins University, Baltimore, MD, USA
[b]Center for Computer-Integrated Surgical Systems and Technology, The Johns Hopkins University, Baltimore, MD, USA
[c]Section of Biomedical Image Analysis, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA

Received 25 May 2004; accepted 1 June 2004

Abstract

This paper presents a four-step training method for increasing the efficiency of support vector machine (SVM). First, a SVM is initially trained by all the training samples, thereby producing a number of support vectors. Second, the support vectors, which make the hypersurface highly convoluted, are excluded from the training set. Third, the SVM is re-trained only by the remaining samples in the training set. Finally, the complexity of the trained SVM is further reduced by approximating the separation hypersurface with a subset of the support vectors. Compared to the initially trained SVM by all samples, the efficiency of the finally-trained SVM is highly improved, without system degradation.

Keywords: Support vector machine; Training method; Computational efficiency

1. Introduction

Support vector machine (SVM) is a statistical classification method proposed by Vapnik in 1995 [1]. Given m labeled training samples, $\{(\vec{x}_i, y_i) | \vec{x}_i \in R^n, y_i \in \{-1, 1\}, i = 1 \cdots m\}$, SVM is able to generate a separation hypersurface that has maximum generalization ability. Mathematically, the decision function can be formulated as

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

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 which implicitly maps the samples to a higher dimensional space. Those samples $\vec{x}_i$ with nonzero parameters α_i are called "support vectors" (SVs).

SVM usually needs a huge number of support vectors to generate a highly convoluted separation hypersurface, in order to well address a complicated nonlinear separation problem. This unavoidably increases the computational burden of SVM in classifying new samples, since it is computationally expensive to calculate the decision function with many nonzero parameters α_i in Eq. (1).

In this paper, a novel training method is proposed to improve the efficiency of SVM classifier, by selecting appropriate training samples. The basic idea of our training method is to exclude the samples that incur the separation hypersurface highly convoluted, such that a few number of support vectors are enough to describe a less convoluted hypersurface for separating two classes.

2. Methods

2.1. Problem analysis

Support vectors in SVM 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

* Corresponding author. Tel.: +1-215-662-7362; fax: +1-215-614-0266.
E-mail address: yzhan@cs.jhu.edu (Y. Zhan).

0031-3203/$30.00
doi:10.1016/j.patcog.2004.06.001

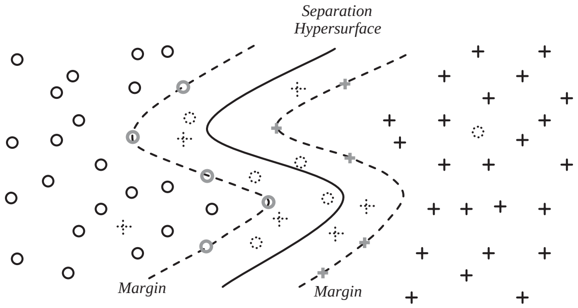


Fig. 1. Schematic explanation of the separation hypersurface (solid curves), margins (dashed curves) and support vectors of SVM (gray circles/crosses). The positive and the negative training samples are indicated by circles and crosses, respectively.

hypersurface, i.e., $d(\vec{x}_i) = \pm 1$, as the gray circles/crosses shown in Fig. 1. As these samples exactly locate on the margins of the separation hypersurface, their number is directly related to the shape of the separation hypersurface. The second type of support vectors are the training samples that locate beyond their corresponding margin, i.e., $y_i d(\vec{x}_i) < 1$, as the dashed circles/crosses shown in Fig. 1. For SVM, these training samples are regarded as misclassified samples even though 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 highly overlap with each other. This is because, (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, and thus selected as the second type of support vectors.

Reducing the computational cost of the SVM is equivalent to decreasing the number of the support vectors, i.e. the number of training samples $\vec{x}_i$ with nonzero α_i in Eq. (1). Osuna and Girosi have proposed an effective method to reduce the number of support vectors of the trained SVM without system degradation [2]. Its basic idea is to approximate the separation hypersurface with a subset of the support vectors by using support vector regression machine (SVRM). However, in many real applications, SVM usually generates a highly convoluted separation hypersurface in the high-dimensional feature space. In this case, Osuna's method still needs a large number of support vectors to approximate the hypersurface. Obviously, an efficient way to further decrease the number of the support vectors is to simplify the shape of the separation hypersurface, by sacrificing a very limited classification rate.

An intuitive method to simplify the shape of the hypersurface is to exclude some training samples, thereby it is possible to separate the remaining samples by a less convoluted hypersurface. To minimize the loss of the classification rate, only the training samples that have largest contributions to the convolution of the hypersurface are preferred to be excluded from the training set. Since the support vectors determine the shape of the separation hypersurface, they are the best candidates to be excluded from the training set, in order to simplify the shape of the separation hypersurface.

Excluding different sets of support vectors from the training set will lead to different simplifications of the separation hypersurface. Fig. 2 presents a schematic example in the two-dimensional feature space, where we assume support vectors exactly locating on the margins. As shown in Fig. 2(a), SVM trained by all the samples has 10 support vectors, and the separation hypersurface is convoluted. Respective exclusion of two different support vectors, SV_1 and SV_2, denoted as gray crosses in Fig. 2(a), will lead to two different separation hypersurfaces as shown in Figs. 2(b) and (c), respectively. SVM in Fig. 2(b) has only 7 support vectors, and its hypersurface is less convoluted, after re-training SVM with all samples except SV_1, which was previously selected as a support vector in Fig. 2(a). Importantly, two additional samples, denoted as dashed circle/cross, were previously selected as support vectors in Fig. 2(a), but they are no longer selected as support vectors in Fig. 2(b). In contrast, SVM in Fig. 2(c) still has 9 support vectors, and the hypersurface is very similar to that in Fig. 2(a), even SV_2, which was previously selected as a support vector in Fig. 2(a), has been excluded from the training set. Obviously, the computational cost of SVM in Fig. 2(b) is less than that in Fig. 2(c), while the correct classification rates are the same.

It is usually more effective to simplify the shape of the hypersurface by excluding the support vectors, like SV_1, which contribute more to the convolution of the hypersurface. For each support vector, its contribution to the convolution of hypersurface can be approximately defined as the generalized curvature of its projection point on the hypersurface. The projection point on the hypersurface can be located by projecting each support vector to the hypersurface along the gradient of the decision function. For example, for two support vectors SV_1 and SV_2 in Fig. 2(a), their projection points on the hypersurface are P_1 and P_2. Obviously, the curvature of the hypersurface at point P_1 is much larger than that at point P_2, which means the support vector SV_1 has more contribution to make the hypersurface convoluted. Therefore, it is more effective to "flatten" the separation hypersurface by excluding the support vectors, like SV_1, with their projection points having the larger curvatures on the hypersurface.

2.2. *Our training algorithm*

By the analysis given above, we design a four-step training algorithm for SVM as detailed below:

Step 1: Use all the training samples to train an initial SVM [3], resulting in l_1 support vectors $\{SV_i^{\mathrm{In}}, i = 1, 2, \ldots, l_1\}$ and the corresponding decision function $d_1(\vec{x})$.

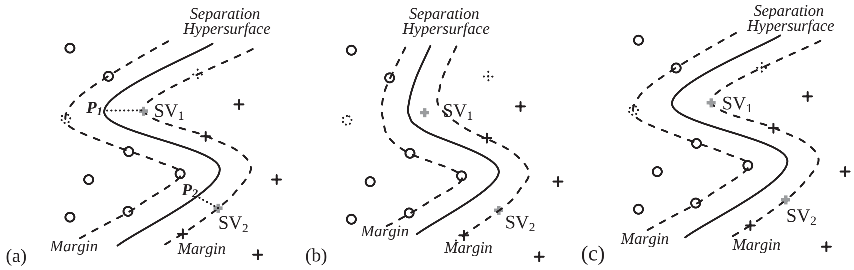


Fig. 2. Schematic explanation of how to selectively exclude the support vectors from the training set, in order to effectively simplify the separation hypersurface. The circles and the crosses denote the positive and the negative training samples, which are identical in (a)–(c). The training samples locating on the margins are the support vectors.

Step 2: Exclude from the training set the support vectors, whose projections on the hypersurface have the largest curvatures:

2a: For each support vector SV_i^{In}, find its projection on the hypersurface, $p(SV_i^{\mathrm{In}})$, along the gradient of decision function $d_1(\vec{x})$.
2b: For each support vector SV_i^{In}, calculate the generalized curvature of $p(SV_i^{\mathrm{In}})$ on the hypersurface, $c(SV_i^{\mathrm{In}})$.
2c: Sort SV_i^{In} in the decrease order of $c(SV_i^{\mathrm{In}})$, and exclude the top *n* percentage of support vectors from the training set.

Step 3: Use the remaining samples to re-train the SVM, resulting in l_2 support vectors $\{SV_i^{\mathrm{Re}}, i = 1, 2, \ldots, l_2\}$ and the corresponding decision function $d_2(\vec{x})$. Notably, l_2 is usually less than l_1.

Step 4: Use the l_2 pairs of data points $\{SV_i^{\mathrm{Re}}, d_2(SV_i^{\mathrm{Re}})\}$ to finally train the SVRM, resulting in l_3 support vectors $\{SV_i^{\mathrm{Fl}}, i{=}1, 2, \ldots, l_3\}$ and the corresponding decision function $d_3(\vec{x})$. Notably, l_3 is usually less than l_2.

3. Experiment

In our study of 3D prostate segmentation from ultrasound images [4], SVM is used for texture-based tissue classification. The input of SVM is texture features extracted from the neighborhood of each voxel under study, and the output is a soft label denoting the likelihood of that voxel belonging to the prostate tissue. In this way, prostate tissues can be differentiated from the surrounding tissues. As the tissue classification is performed for many times (i.e. 10^6) during the segmentation stage and the real-time segmentation is usually required in clinical applications, our proposed training method is very critical for speeding up the SVM in tissue classification. The experimental dataset consists of 18105 training samples collected from five ultrasound images and 3621 testing samples collected from a new ultrasound image. Each sample has 10 texture features, extracted by a Gabor filter bank [4].

In the first experiment, we use our method to train a series of SVMs by excluding different percentages of support vectors in Step 2c. The performances of these SVMs are measured by the number of support vectors finally used, and the number of correct classifications among 3621 testing samples. As shown in Fig. 3(a), after excluding 50% of initially selected support vectors, the finally-trained SVM has 1330 support vectors, which is only 48% of the support vectors (2748) initially selected in the original SVM; but its classification rate still reaches 95.39%. Compared to 96.02% classification rate achieved by original SVM with 2748 support vectors, the loss of classification rate is relatively trivial. If we want to further reduce the computational cost, we can exclude 90% of initially selected support vectors from the training set. Our finally-trained SVM has only 825 support vectors, which means the speed is triple, and it still has 93.62% classification rate. To further validate the effect of our trained SVM in prostate segmentation, the SVM with 825 support vectors (denoted by the white triangle in Fig. 3(a)) is applied to a real ultrasound image for tissue classification. As shown in Fig. 3(b), the result of our trained SVM is not inferior to that of the original SVM with 2748 support vectors (denoted by the white square in Fig. 3(a)), in terms of differentiating prostate tissues from the surrounding ones.

In the second experiment, we compare the performances of different training methods in reducing the computational cost of the finally trained SVM and also in correctly classifying the testing samples. The five methods are compared; they are (1) a method of slackening the training criterion by decreasing the penalty factor to errors [3]; (2) a heuristic method, which assumes the training samples distributing in a multi-variant Gaussian way, then excludes the “abnormal” training samples distant from the respective distribution centers, and finally trains a SVM only by the remaining samples; (3) a method of directly excluding the initial support vectors from the training set and then training a SVM only by the remaining samples, i.e., our proposed method without using Step 4; (4) Osuna’s method [2]; (5) our proposed

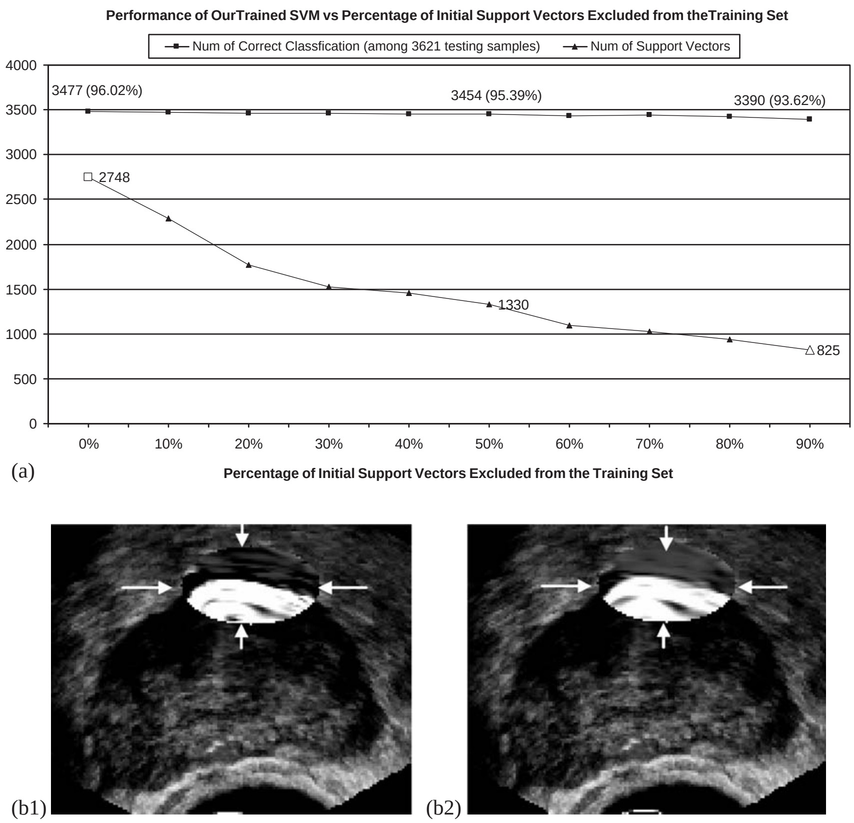
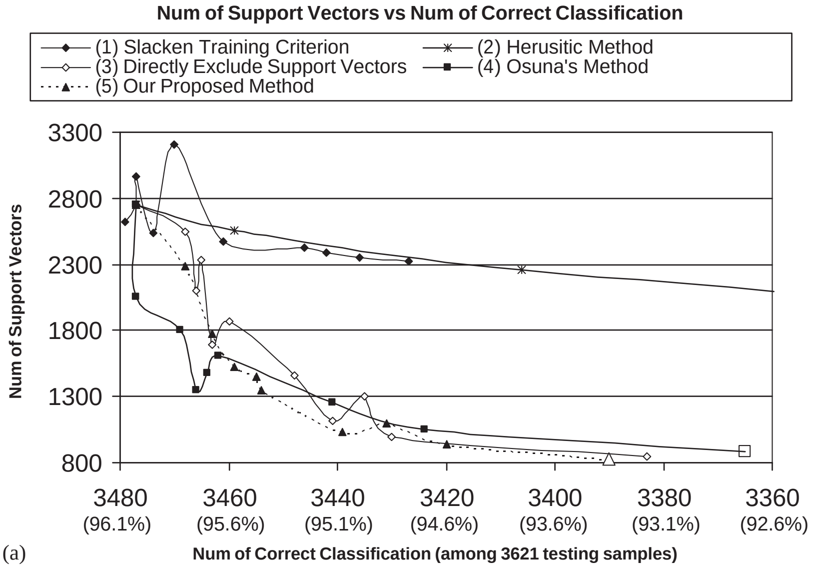


Fig. 3. (a) The performance of the finally trained SVM changes with the percentages of initial support vectors excluded from the training set. (b) Comparisons of tissue classification results using (b1) the original SVM with 2748 support vectors and (b2) our trained SVM with 825 support vectors. The tissue classification results are shown only in an ellipsoidal region.

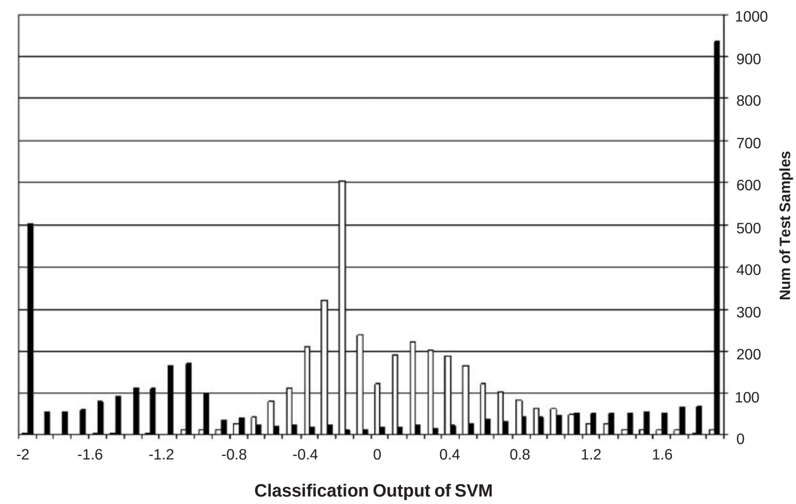
method. The performances of these five methods are evaluated in Fig. 4(a), by the number of support vectors used vs the number of correct classifications achieved. As shown in Fig. 4(a), the methods 3–5 are obviously more effective in reducing the number of the support vectors. In particular, by checking the beginning curves of methods 3–5, Osuna's method is the most effective in initially reducing the number of support vectors. However, to further reduce the support vectors with limited sacrifice of classification rate, our method has better performance than Osuna's method, i.e., less support vectors required for the similar classification rate. The classification abilities of two SVMs, respectively trained by Osuna's method and our method, are further compared here. The SVM trained by Osuna's method, as denoted by the white square in Fig. 4(a), needs 884 support vectors and its classification rate is 92.93%. The SVM trained by our method, as denoted by the white triangle in Fig. 4(a), needs only 825 support vectors, while its classification rate is 93.62%, higher than 92.93% produced by Osuna's method, which uses more support vectors (884). Moreover, our trained SVM actually has much better classification ability than the SVM trained by Osuna's method, once checking the histograms of their classification outputs on the same testing dataset. As shown in Fig. 4(b), the classification outputs of Osuna's SVM concentrate around 0, which means the classification results of the positive and the negative samples are not widely separated. 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 keeping the classification ability of the finally trained SVM, after reducing a considerable number of initially selected support vectors.

4. Conclusion

We have presented a training method to increase the efficiency of SVM for fast classification, without system degradation. This is achieved by excluding from the training set the initial support vectors that incur the separation hypersurface highly convoluted. By combining with Osuna's method, which uses SVRM to efficiently approximate the hypersurface, our training method can highly increase the classification speed of the SVM, with very limited loss of classification ability. Experimental results on real prostate



(a)



(b)

Fig. 4. (a) Comparing the performances of five training methods in increasing the efficiency of SVM. (b) Histograms of classification outputs on a testing dataset respectively from our trained SVM (black bars) and Osuna's SVM (white bars).

ultrasound images show good performance of our training method in discriminating the prostate tissues from other tissues. Compared to other four training methods, our proposed training method is able to generate more efficient SVMs, with better classification abilities.

References

[1] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[2] E. Osuna, F. Girosi, Reducing the run-time complexity of support vector machines, ICPR, Brisbane, Australia, 1998.
[3] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowledge Discovery 2 (2) (1998) 121–167.
[4] Y. Zhan, D. Shen, Automated segmentation of 3D US prostate images using statistical texture-based matching method, MICCAI, Montreal, Canada, 2003.