PERGAMON

Neural Networks 12 (1999) 783–789

Neural Networks

www.elsevier.com/locate/neunet

Improving support vector machine classifiers by modifying kernel functions

S. Amari*, S. Wu

RIKEN Brain Science Institute, The Institute for Physical and Chemical Research, Hirosawa 2-1, Wako-shi, Saitama, Japan

Received 2 February 1999; received in revised form 19 February 1999; accepted 19 February 1999

Abstract

We propose a method of modifying a kernel function to improve the performance of a support vector machine classifier. This is based on the structure of the Riemannian geometry induced by the kernel function. The idea is to enlarge the spatial resolution around the separating boundary surface, by a conformal mapping, such that the separability between classes is increased. Examples are given specifically for modifying Gaussian Radial Basis Function kernels. Simulation results for both artificial and real data show remarkable improvement of generalization errors, supporting our idea. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Support vector machine; Pattern classification; Information geometry; Kernel function; Radial basis function; Riemannian geometry; Kernel Adatron; Nonlinear classification

1. Introduction

Support Vector Machine (SVM) is a new promising pattern classification technique proposed recently by Vapnik and co-workers (Boser, Guyon & Vapnik, 1992; Cortes & Vapnik, 1995, and Vapnik, 1995). Unlike traditional methods which minimize the empirical training error, SVM aims at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data. This can be regarded as an approximate implementation of the Structure Risk Minimization principle. What makes SVM attractive is the property of condensing information in the training data and providing a sparse representation by using a very small number of data points (SVs) (Girosi, 1998).

SVM is a linear classifier in the parameter space, but it is easily extended to a nonlinear classifier of the $\boldsymbol{\phi}$-machine type (Aizerman, Braverman & Rozonoer, 1964) by mapping the space $S = \{\boldsymbol{x}\}$ of the input data into a high-dimensional (possibly infinite-dimensional) feature space $F = \{\boldsymbol{\phi}(\boldsymbol{x})\}$. By choosing an adequate mapping $\boldsymbol{\phi}$, the data points become linearly separable or mostly linearly separable in the high-dimensional space, so that one can easily apply the structure risk minimization. We need not compute the mapped patterns $\boldsymbol{\phi}(\boldsymbol{x})$ explicitly, and instead we only need the dot products between mapped patterns. They are directly available from the kernel function which generates $\boldsymbol{\phi}(\boldsymbol{x})$. By choosing different kinds of kernels, SVM can realize Radial Basis Function (RBF), Polynomial and Multi-layer Perceptron classifiers. Compared with the traditional way of implementing them, SVM has an extra advantage of automatic model selection, in the sense that both the optimal number and locations of the basis functions are automatically obtained during training (Schölkopf et al., 1996).

The performance of SVM largely depends on the kernel. Smola, Schölkopf and Müller (1998) elucidated the relation between the SVM kernel method and the standard regularization theory (Girosi, Jones & Poggio, 1995). However, there are no theories concerning how to choose good kernel functions in a data-dependent way. The present paper is a first step to this important problem. We propose an information-geometric method of modifying a kernel to improve the performance. This is based on the structure of the Riemannian geometry induced in the input space by the kernel. A nonlinear function $\boldsymbol{\phi}$ embeds the input space $S = \{\boldsymbol{x}\}$ in a high-dimensional Euclidean or Hilbert feature space $F = \{\boldsymbol{\phi}\}$ as a curved submanifold. This embedding induces a Riemannian metric in the input space, which shows how a small volume element in the input space is enlarged or reduced in the feature space. The idea is as follows: in order to increase the margin or separability in the feature space without changing the volume of the entire space, it is efficient to enlarge volume elements locally in

* Corresponding author. Tel.: + 81-48-467-9669; fax: + 81-48-467-9693.

E-mail addresses: amari@brain.riken.go.jp (S. Amari); phwusi@brain.riken.go.jp (S. Wu)

PII: S0893-6080(99)00032-5

neighborhoods of support vectors which are located closely to the boundary surface. This makes it possible to enlarge the spatial resolution around the boundary so that the separability of classes is increased. To implement this idea, we use a conformal mapping of the input Riemannian space. This will be realized approximately by a conformal transformation of a kernel.

The practical training process consists of two steps: In the first step a primary kernel is used to obtain support vectors. The kernel is then modified conformally in a data dependent way by using the information of the support vectors. In the second step the modified kernel is used to obtain the final classifier. Examples are given specifically for modifying Gaussian RBF kernels. Simulation results for both artificial and real data support our method.

2. The method

2.1. Geometry of the SVM kernel

Consider a pattern classifier, which uses a hyperplane to separate two classes of patterns based on given examples $\{\mathbf{x}_i, y_i\}$ for $i = 1,\ldots,l$, where $\mathbf{x}_i$ is a vector in the input space $S = R^n$ and y_i denotes the class index taking a value $+1$ or -1. A nonlinear SVM maps the input data $\mathbf{x}$ into a high dimensional feature space $F = R^N$ (N may be infinite) by using a nonlinear mapping $\boldsymbol{\phi}$, $\mathbf{z} = \boldsymbol{\phi}(\mathbf{x})$. It then searches for a linear discriminant function

$$f(\mathbf{x}) = \mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}) + b \tag{1}$$

in the feature space. Patterns are classified by the sign of $f(\mathbf{x})$. Obviously, the classifier has a nonlinear boundary $f(\mathbf{x}) = 0$ in the input space. The SVM solution is obtained through maximizing the margin between the separating hyperplane and the data, where the margin is defined as $2/\|\mathbf{w}\|$. This is justified from the point of view of Structure Risk Minimization principle as minimizing an upper bound of the generalization error.

Let us consider a reproducing kernel function $K(\mathbf{x}, \mathbf{x}')$, and let its eigenfunctions $\varphi_\alpha(\mathbf{x})$ be

$$\int K(\mathbf{x}, \mathbf{x}')\varphi_\alpha(\mathbf{x}')\mathrm{d}\mathbf{x}' = \lambda_\alpha\varphi_\alpha(\mathbf{x}), \qquad \alpha = 1, 2, \ldots. \tag{2}$$

Then, the kernel is represented as

$$K(\mathbf{x}, \mathbf{x}') = \sum_\alpha \lambda_\alpha\varphi_\alpha(\mathbf{x})\varphi_\alpha(\mathbf{x}'). \tag{3}$$

We rescale functions φ_α as

$$\phi_\alpha(\mathbf{x}) = \sqrt{\lambda_\alpha}\varphi_\alpha(\mathbf{x}) \tag{4}$$

so that we have

$$K(\mathbf{x}, \mathbf{x}') = \sum_\alpha \phi_\alpha(\mathbf{x})\phi_\alpha(\mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})\cdot\boldsymbol{\phi}(\mathbf{x}'), \tag{5}$$

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_\alpha(\mathbf{x}))$.

The kernel SVM begins with a kernel $K(\mathbf{x}, \mathbf{x}')$ to obtain the nonlinear mapping $\boldsymbol{\phi}(\mathbf{x})$. The form of the SVM solution turns out to be

$$f(\mathbf{x}) = \sum_{i\in \mathrm{SV}} h_i y_i K(\mathbf{x}_i, \mathbf{x}) + b_0, \tag{6}$$

where summation runs over all the support vectors, and h_i is a positive number representing the contribution of the ith support vector (Boser et al., 1992, Cortes & Vapnik, 1995). The h_is are derived as dual variables to solve the problem of minimizing $1/\|\mathbf{w}\|^2$. The bias term b_0 can be found by using any support vector $\mathbf{x}_{\mathrm{sv}}$ as

$$b_0 = y_{sv} - \sum_{i\in SV} h_i y_i K(\mathbf{x}_i, \mathbf{x}_{sv}). \tag{7}$$

It is interesting to note that we need not compute the mapped pattern $\boldsymbol{\phi}(\mathbf{x})$ explicitly. We only need their dot products, which is available from the kernel function K. Actually, in the SVM study people work in the inverse way by starting from a kernel.

Let us analyze the geometrical structure induced in the input space by a kernel. The mapping $\boldsymbol{\phi}$ defines an embedding of S into F as a curved submanifold. When F is a Euclidean or Hilbert space, a Riemannian metric is thereby induced in the space S, where the length of a small line element $\mathrm{d}\mathbf{x}$ in S is defined by the length in the larger space F.

Denote by $\mathbf{z}$ the mapped pattern of $\mathbf{x}$ in the feature space, i.e., $\mathbf{z} = \boldsymbol{\phi}(\mathbf{x})$. A small vector $\mathrm{d}\mathbf{x}$ is mapped to

$$\mathrm{d}\mathbf{z} = \nabla\boldsymbol{\phi}\cdot\mathrm{d}\mathbf{x} = \sum_i \frac{\partial}{\partial x_i}\boldsymbol{\phi}(\mathbf{x})\mathrm{d}\mathbf{x}_i, \tag{8}$$

where

$$\nabla\boldsymbol{\phi} = \left(\frac{\partial}{\partial x_i}\boldsymbol{\phi}(\mathbf{x})\right). \tag{9}$$

The squared length of $\mathrm{d}\mathbf{z} = (\mathrm{d}z_\alpha)$ is written in the quadratic form as

$$|\mathrm{d}\mathbf{z}|^2 = \sum_\alpha (\mathrm{d}\mathbf{z}_\alpha)^2 = \sum_{i,j} g_{ij}(\mathbf{x})\,\mathrm{d}x_i\,\mathrm{d}x_j, \tag{10}$$

where

$$g_{ij}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i}\boldsymbol{\phi}(\mathbf{x})\right)\cdot\left(\frac{\partial}{\partial x_j}\boldsymbol{\phi}(\mathbf{x})\right), \tag{11}$$

the dot denoting the summation over index α of $\boldsymbol{\phi}$. The $n\times n$ positive-definite matrix $G(\mathbf{x}) = (g_{ij}(\mathbf{x}))$ is the Riemannian metric tensor induced in S. We show that the metric is directly derived from the kernel.

Theorem 1.

$$g_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_i}\frac{\partial}{\partial x'_j}K(\mathbf{x}, \mathbf{x}')|_{\mathbf{x}'=\mathbf{x}}. \tag{12}$$

Proof. From Eq. (5), we have

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial x'_j}K(\mathbf{x},\mathbf{x}') = \nabla\boldsymbol{\phi}(\mathbf{x})\cdot\nabla\boldsymbol{\phi}(\mathbf{x}'), \tag{13}$$

which proves Eq. (12) □.

There are some typical kernel functions. One is radial,

$$K(\mathbf{x},\mathbf{x}') = f(\frac{1}{2}\|\mathbf{x}-\mathbf{x}'\|^2), \tag{14}$$

which includes the Gaussian RBF kernel,

$$K(\mathbf{x},\mathbf{x}') = \mathrm{e}^{-\|\mathbf{x}-\mathbf{x}'\|^2/2\sigma^2}. \tag{15}$$

The other is functions of the inner product,

$$K(\mathbf{x},\mathbf{x}') = f(\mathbf{x}\cdot\mathbf{x}') \tag{16}$$

which includes the Polynomial kernel of degree d, $K(\mathbf{x},\mathbf{x}') = (1+\mathbf{x}\cdot\mathbf{x}')^d$, and the Multi-layer Perceptron kernel, $K(\mathbf{x},\mathbf{x}') = \tanh(\mathbf{x}\cdot\mathbf{x}'-\theta)$.

The Riemannian metric for the first case is given by

$$g_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_i}\frac{\partial}{\partial x'_j}K(\mathbf{x},\mathbf{x}')|_{\mathbf{x}'=\mathbf{x}} = -\delta_{ij}f'(\frac{1}{2}\|\mathbf{x}-\mathbf{x}'\|^2)|_{x'=x}$$

$$-f''(\frac{1}{2}\|\mathbf{x}-\mathbf{x}'\|^2)(x_i-x'_i)(x_j-x'_j)|_{\mathbf{x}'=\mathbf{x}} = -f'(0)\delta_{ij}. \tag{17}$$

In particular for the Gaussian RBF kernel, we have

$$g_{ij}(\mathbf{x}) = \frac{1}{\sigma^2}\delta_{ij}. \tag{18}$$

The induced metric is Euclidean, which is translational and rotational invariant. However, this does not imply that the image of $\boldsymbol{\phi}$ is linear. It may be curved in F, having non-zero Euler–Schouten embedding curvature but zero Riemannian–Christoffel curvature.

The metric for the inner product case can be calculated in a similar way, and is given by

$$g_{ij}(\mathbf{x}) = f'(0)\delta_{ij}+x_ix_jf''(0). \tag{19}$$

The metric is Riemannian and rotational invariant.

The volume form in a Riemannian space is defined as

$$\mathrm{d}V = \sqrt{g(\mathbf{x})}\,\mathrm{d}x_1\cdots\mathrm{d}x_n \tag{20}$$

where $g(\mathbf{x}) = \det|g_{ij}(\mathbf{x})|$. The factor $\sqrt{g(\mathbf{x})}$ represents how a local area is magnified in F under the mapping $\boldsymbol{\phi}$. Hereafter, we call it the magnification factor.

2.2. A data dependent way for modifying a kernel

Based on the Riemannian geometrical structure induced in the input space, we propose a method of modifying a kernel to improve the performance of a SVM classifier. The idea is rather straightforward. To increase the margin or separability of classes, we need to enlarge the spatial resolution around the boundary surface in F. Taking the Riemannian distance

$$\mathrm{d}s^2 = \sum_{i,j} g_{ij}\,\mathrm{d}x_i\,\mathrm{d}x_j \tag{21}$$

or the volume element $\sqrt{g(\mathbf{x})}$ into account, this leads us to increase the metric $g_{ij}(\mathbf{x})$ around the boundary of $f(\mathbf{x}) = 0$ and to reduce it around other points. More precisely, we modify the nonlinear mapping $\boldsymbol{\phi}$ (or the related kernel K) such that $\sqrt{g(\mathbf{x})}$ is enlarged around the boundary. In practice, the boundary is not known. By using the knowledge that support vectors are (mostly) located around the boundary, we solve the problem by increasing the metric in the neighborhood of the support vectors.

A conformal transformation

$$\tilde{g}_{ij}(\mathbf{x}) = \Omega(\mathbf{x})g_{ij}(\mathbf{x}), \tag{22}$$

gives a solution to this problem, because the metric is enlarged by a factor $\Omega(\mathbf{x})$ at point $\mathbf{x}$. See Okamoto, Amari and Takeuchi (1991) for another application of conformal transformation in information geometry. In our problem, $\Omega(\mathbf{x})$ should be chosen in the way that it has large values at the support vector positions. The advantage of a conformal transformation is that it keeps angles unchanged in the whole space and therefore won't affect much the spatial relationship between the data points. In practice, it is difficult to find a nonlinear mapping $\tilde{\boldsymbol{\phi}}$ which realizes the above conformal transformation. Therefore, we consider a quasi-conformal transformation obtained from modification of a kernel.

For a positive scalar function $c(\mathbf{x})$,

Definition.

$$\tilde{K}(\mathbf{x},\mathbf{x}') = c(\mathbf{x})c(\mathbf{x}')K(\mathbf{x},\mathbf{x}') \tag{23}$$

is called a conformal transformation of a kernel by factor $c(\mathbf{x})$.

Remark. Smola et al. (1998) gave a sufficient condition for a regularization operator P to give a solution equivalent to the SVM with kernel K. When a kernel K corresponds to a regularization operator P, its conformal transformation $\tilde{K}$ with $c(\mathbf{x})$ corresponds to the operator

$$\tilde{P} = c(\mathbf{x})P. \tag{24}$$

Theorem 2. The metric g_{ij} is changed into $\tilde{g}_{ij}$,

$$\tilde{g}_{ij}(\mathbf{x}) = c_i(\mathbf{x})c_j(\mathbf{x})+c(\mathbf{x})^2g_{ij}(\mathbf{x}) \tag{25}$$

by a conformal transformation of the Gaussian RBF kernel, where $c_i(\mathbf{x}) = \partial c(\mathbf{x})/\partial x_i$.

It should be remarked that $c_i(\mathbf{x}) = 0$ at positions on which $c(\mathbf{x})$ is maximal. In order to ensure that $c(\mathbf{x})$ have large

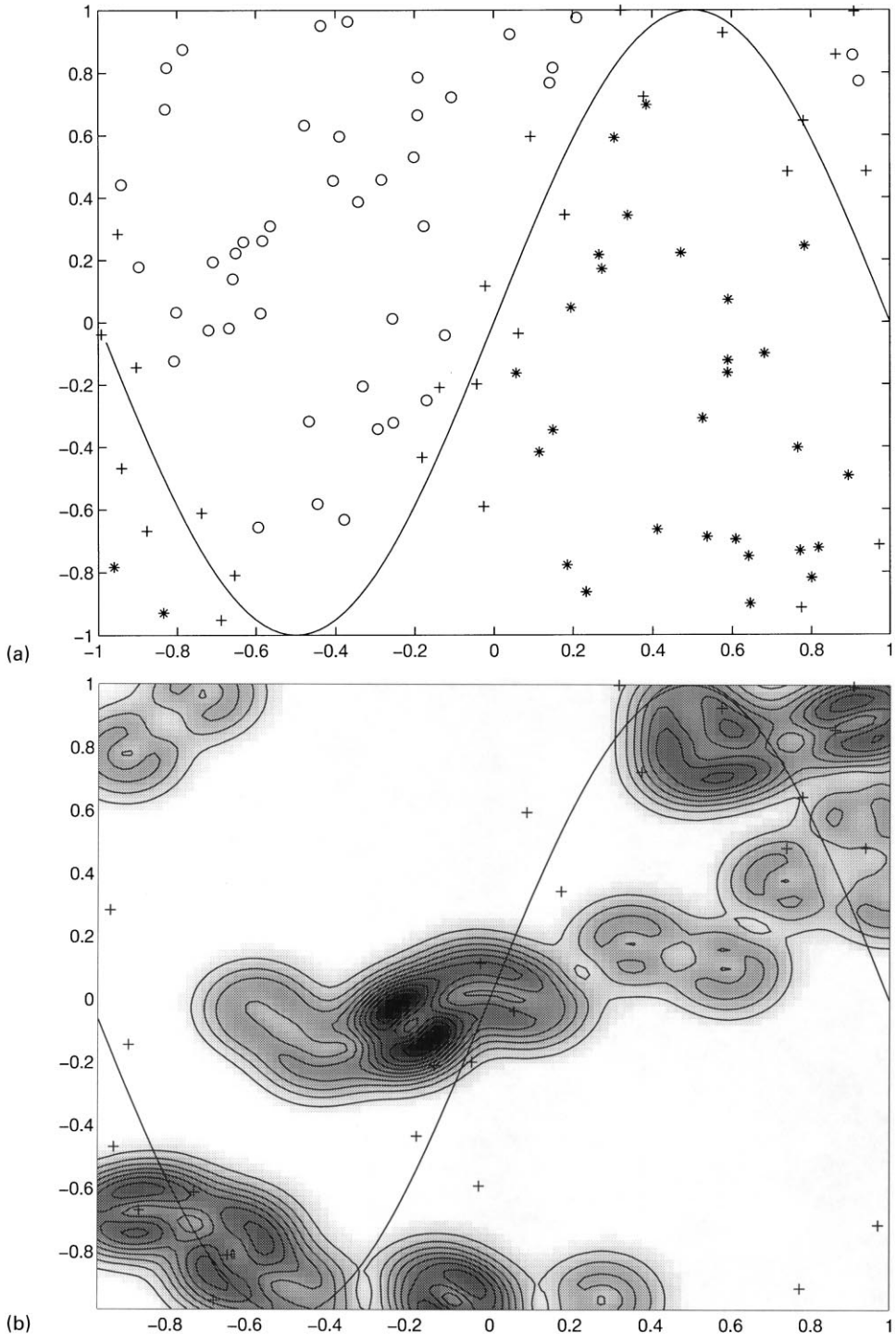

Fig. 1. (a) A two-dimensional artificial data set, where the two classes are denoted by '○' and '∗', respectively. The symbol '+' represents the support vectors of kernel K. The distribution of the magnification factor $\sqrt{\tilde{g}}$ for the three different values of τ: (b) 0.1; (c) 0.5; and (d) 0.25. The magnitude is represented by the gray level.

values at the support vector positions, we construct it in a data dependent way as

$$c(\boldsymbol{x}) = \sum_{i \in \mathrm{SV}} h_i \, \mathrm{e}^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2\tau^2} \tag{26}$$

where τ is a free parameter and summation runs over all the support vectors.

Now, let us study how the spatial resolution is enlarged by looking into the magnification factor $\sqrt{\tilde{g}}$ in the case of a Gaussian RBF kernel. Let us focus on the region of a

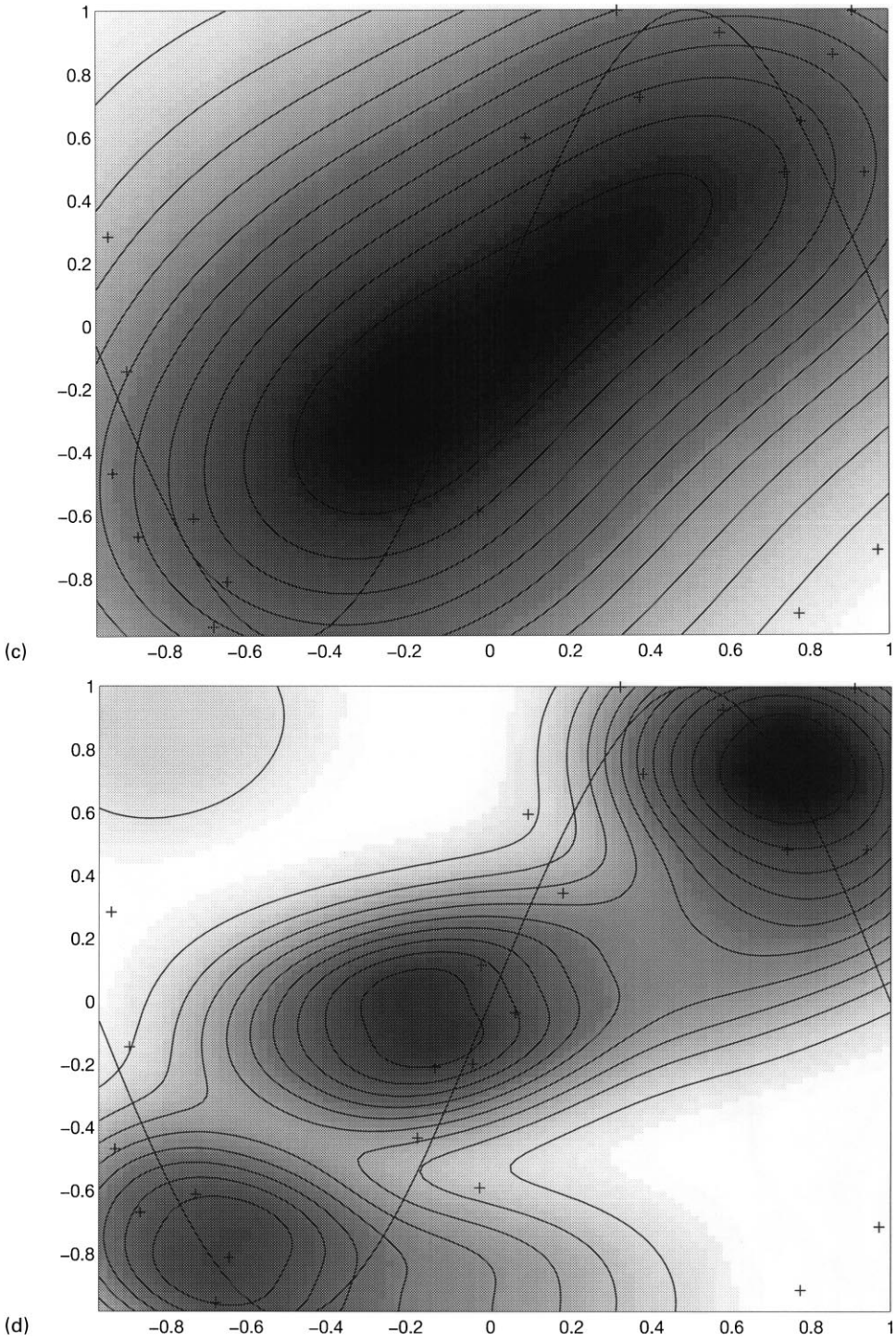

Fig. 1. (*continued*)

neighborhood of a support vector $\mathbf{x}_i$. We then have (see Appendix)

$$\sqrt{\tilde{g}(\mathbf{x})} = \frac{h_i^n}{\sigma^n}\, \mathrm{e}^{-nr^2/2\tau^2}\sqrt{1+\frac{\sigma^2}{\tau^4}r^2}. \tag{27}$$

where $r = \|\mathbf{x}-\mathbf{x}_i\|$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$.

Its derivative is given by

$$\frac{\mathrm{d}\sqrt{\tilde{g}(\mathbf{x})}}{\mathrm{d}r} = \frac{h_i^n}{2\sigma^n}\, \mathrm{e}^{-nr^2/2\tau^2}\left[\frac{2\sigma^2 r}{\tau^4} - \frac{2nr}{\tau^2}\left(1+\frac{\sigma^2}{\tau^4}r^2\right)\right] \Big/ \times\sqrt{1+\frac{\sigma^2}{\tau^4}r^2}. \tag{28}$$

Table 1
Comparing the training results for three different values of τ

(Train errors	*GE*	#SV	
Before modifying	0	0.080	24
$\tau = 0.1$	0	0.072	54
$\tau = 0.25$	0	0.042	17
$\tau = 0.5$	0	0.077	22

It is easy to check the following:

1. when $\tau < \sigma/\sqrt{n}, r = \tau\sqrt{1/n - \tau^2/\sigma^2} > 0$(obtained from $d\sqrt{\tilde{g}}/dr = 0$) is the place being mostly magnified;
2. when $\tau \geq \sigma/\sqrt{n}$, since $d\sqrt{\tilde{g}}/dr \leq 0$ for any value of r, $r = 0$ (the support vector position) is the place being mostly magnified.

In order to make sure the magnification is larger around the support vector, we need τ to be not much less than $\sigma/\sqrt{n}$. In contrast, in order to ensure the magnification is local (because a magnification over the whole space is meaningless), we need τ to be small. When the above two facts are considered, the optimal value for τ is around $\sigma/\sqrt{n}$, which roughly agrees with the simulation results.

In summary, the training process of the new method consists of the two steps:

1. Train SVM with a primary kernel K, which is then modified according to Eqs. (23) and (26):
2. Train SVM with the modified kernel $\tilde{K}$.

3. Simulation experiments

To evaluate the performance of our method, we did simulations on two classification problems, one artificial and one real. The primary kernel function is fixed to be a Gaussian RBF. The SVM solver we used is a gradient descent method. This method is an application of the Adatron method (Anlauf & Biehl, 1989) to the kernel SVM named the Kernel-Adatron algorithm (Friess, Cristianini & Campbell, 1998). We used a different version, developed independently, to look for an approximate solution by augmenting the input data with one more dimension (Vijayakumar & Wu, 1999). The method has proved to be simple, fast and is able to achieve high precision approximation.

3.1. An artificial nonlinear classification problem

Let us consider an artificial two-dimensional data set $\mathbf{x} = (x, y)$ uniformly distributed in the region $[-1, 1] \times [-1, 1]$, where the two classes are separated by a nonlinear boundary determined by $y = \sin(\pi x)$ (see Fig.1 (a)). A SVM classifier with a kernel K (or $\tilde{K}$) generates a new boundary f (or $\tilde{f}$), $f(x, y) = \sum_{i \in SV} h_i y_i K(\mathbf{x}_i, \mathbf{x}) + b_0 = 0$, which gives a boundary curve of the form $y = u(x)$(or $y = \tilde{u}(x)$). Misclassification happens when a pattern is in the region between the two boundaries, $y = sin(\pi x)$ and $y = u(x)$ (or $y = \tilde{u}(x)$). So the area of this region measures the generalization error (*GE*) of the classifier, which is given by

$$\mathrm{GE} = \frac{1}{4}\int_{-1}^{1} |u(x) - \sin(\pi x)|\, \mathrm{d}x, \tag{29}$$

where $|\cdot|$ denotes the absolute value.

In the simulation experiment, 100 training examples are randomly and uniformly generated. To exclude the possibility that improvement results from other factors, we set the value of σ to be equal to the optimal one, 0.4, in the sense that it achieves the best training result in the first step training. Simulation results for three different values of τ are compared in Table 1. It shows that when τ takes a proper value (0.25 in this example, which is a little smaller than the theoretically predicted value, $\sigma/\sqrt{n} = 0.283$), the performance of the classifier is remarkably improved. The generalization error probability is decreased to nearly a half, and interestingly, the number of support vectors decreases drastically.

To illustrate our method geometrically, we calculated the magnification factor $\sqrt{\tilde{g}}$ for the three cases. It shows that when τ is small (see Fig. 1(b)), magnification is isolated, not covering the boundary, and the performance is bad. When τ is large (see Fig. 1(c)), magnification occurs over the whole space and the performance is also bad. When τ takes a proper value (see Fig. 1(d)), magnification occurs roughly around the boundary and the performance is significantly improved. This observation agrees with the geometrical picture underlying our method.

We should note that, since the data set is randomly generated, results in the generalization error and the optimal values of σ and τ may change in different trials. However, the essential character of the above observation which supports our method always holds.

3.2. Wisconsin breast cancer data classification

In this section a benchmark problem for Wisconsin breast cancer data classification is tested (Ster & Dobnikar, 1996). The data consists of 10 medical attributes (one of them is the id number which we don't use for the classification task), which are used to make a binary decision on the medical condition: whether the cancer is malignant or benign. The data set consists of 699 instances including missing values. We used a random selection of 200 training data and 200 testing data, excluding the instances with missing values.

The parameter σ is set to be 0.6 to achieve the best result in the primary training. Simulation results for three different values of τ are compared in Table 2. It shows that when τ takes a proper value (0.3 in this example, which is a little greater than the theoretically predicted value, $\sigma/\sqrt{n} = 0.2$), the performance of the classifier is improved remarkably, and the number of support vectors decreases from 42 to 34.

Table 2
Comparing the training results for three different values of τ

# Train errors	# Test errors	# SV	
Before modifying	1	9	42
$\tau = 0.2$	0	11	59
$\tau = 0.3$	0	7	34
$\tau = 0.4$	0	8	38

4. Conclusion

In this paper we presented a new method of modifying a kernel to improve the performance of a SVM classifier. It is based on information-geometric consideration of the structure of the Riemannian geometry induced by the kernel. The idea is to enlarge the spatial resolution around the boundary by a conformal transformation so that the separability of classes is increased. This geometrical picture is confirmed by simulations. Examples are given specifically for modifying a Gaussian RBF kernel. Simulation results for both artificial and real data support our idea. In a future work we will analyze in more detail the theoretical underpinning of the method and extend it to different kernel cases. The present paper is expected to open a new study on the optimal choice of kernel functions.

Appendix A. Calculating the magnification factor $\sqrt{\tilde{g}}$ for Gaussian RBF kernel

From the Eq. (25), we have

$$\tilde{g}_{ij}(\boldsymbol{x}) = \frac{c^2(\boldsymbol{x})}{\sigma^2} A_{ij} \tag{30}$$

where

$$A_{ij} = a_i a_j + \delta_{ij}, \tag{31}$$

and

$$a_i = \frac{\sigma}{c} c_i(\mathbf{x}). \tag{32}$$

The matrix $\mathbf{A} = \{A_{ij}\}$ can be written as

$$\mathbf{A} = \mathbf{I} + a^2 \mathbf{e}\mathbf{e}^T, \tag{33}$$

where $\mathbf{I}$ is the identity matrix, $a^2 = \sum_i^n a_i^2$, $\mathbf{e}$ is a $n \times 1$ unit vector with elements $\{\mathbf{e}_{i1} = a_i/a\}$, $\mathbf{e}^T$ is the transposition of $\mathbf{e}$.

By using an adequate orthogonal transformation, we can easily get

$$\det|\mathbf{A}| = 1 + a^2 \tag{34}$$

and

$$\tilde{g}(\mathbf{x}) = \det|\tilde{g}_{ij}| = \frac{c^{2n}(\mathbf{x})}{\sigma^{2n}} \left(1 + \sum_{i=1}^{n} a_i^2\right). \tag{35}$$

Thus, the magnification factor is given by

$$\sqrt{\tilde{g}(\mathbf{x})} = \frac{c^n(\mathbf{x})}{\sigma^n} \left(1 + \sum_{i=1}^{n} a_i^2\right)^{1/2}. \tag{36}$$

Let us now focus on a neighborhood of a support vector $\boldsymbol{x}_i$, where

$$c(\mathbf{x}) \cong h_i \, \mathrm{e}^{-r^2/2\tau^2} \tag{37}$$

in which $r \equiv |\boldsymbol{x} - \boldsymbol{x}_i|$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$. Substituting Eq. (37) in Eq. (36), we have

$$\sqrt{\tilde{g}(\mathbf{x})} = \frac{h_i^n}{\sigma^n} \, \mathrm{e}^{-nr^2/2\tau^2} \sqrt{1 + \frac{\sigma^2}{\tau^4} r^2}. \tag{38}$$

Notes added in proof: The authors found that Burges (1999) defined the same Riemannian metric induced by a kernel as ours.

References

Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, *25*, 821–837.

Anlauf, J. K., & Biehl, M. (1989). The Adatron: an adaptive perceptron algorithm. *Europhysics Letters*, *10*, 687–692.

Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Fifth Annual Workshop on Computational Learning Theory*, New York: ACM Press.

Burges, C. J. C. (1999). Geometry and invariance in kernel based methods. In B. Schölkopf et al. (Eds.), *Advances in Kernel methods: support vector learning* (pp. 89–116). MIT Press.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, *20*, 273–297.

Friess, T. T., Cristianini, N., & Campbell, C. (1998). The Kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines. *Proceedings of the 15th International Conference on Machine Learning, Madison*, Los Altos, CA: Morgan-Kaufman.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation*, *20*, 1455–1480.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural network architectures. *Neural Computation*, *7*, 219–269.

Okamoto, I., Amari, S., & Takeuchi, K. (1991). Asymptotic theory of sequential estimation: differential geometrical approach. *Annals of Statistics*, *19*, 961–981.

Schölkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., Vapnik, V. (1996). Comparing support vector machines with gaussian kernels to radial basis function classifiers. A.I.Memo 1599, M.I.T. AI Labs.

Smola, A. J., Schölkopf, B., & Müller, K. R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, *11*, 637–649.

Ster, B., Dobnikar, A. (1996). Neural networks in medical diagnosis: comparison with other methods. In A. Bulsari et al. (Eds.), Proceedings of the International Conference EANN'96, p. 427–430.

Vapnik, V. (1995). *The nature of statistical learning theory*, Springer.

Vijayakumar, S., & Wu, S. (1999). Sequential support vector classifiers and regression. *Soft Computing*, in press.