Scaling the Kernel Function to Improve Performance of the Support Vector Machine

Peter Williams, Sheng Li, Jianfeng Feng, and Si Wu

Department of Informatics, University of Sussex, UK

Abstract. The present study investigates a geometrical method for optimizing the kernel function of a support vector machine. The method is an improvement of the one proposed in [4, 5]. It consists of using prior knowledge obtained from conventional SVM training to conformally rescale the initial kernel function, so that the separation between two classes of data is effectively enlarged. It turns out that the new algorithm works efficiently, has few free parameters, consumes very low computational cost, and overcomes the susceptibility of the original method.

1 Introduction

The support vector machine (SVM) is a general method for pattern classification and regression proposed by Vapnik and co-authors [1]. The essential idea is to use a kernel function to map the original input data into a high-dimensional space so that two classes of data become, as far as possible, linearly separable [1, 2]. Thus, the kernel is the key that determines the performance of the SVM. From the viewpoint of regularization theory, the kernel implies a smoothness assumption on the structure of the discriminant function. In case we have some prior knowledge about the data, we may use it to construct a good kernel, otherwise, the kernel has to be optimized in a data-dependent way.

Amari and Wu [4, 5] have proposed a two-stage training process to optimize a kernel function. Their idea is based on the understanding of that the kernel mapping induces a Riemannian metric in the original input space [3, 4] and that a good kernel should enlarge the separation between the two classes. In their method, the first step of training involves using a primary kernel to find out where the separating boundary is roughly located. In the second step, the primary kernel is conformally scaled, which magnifies the Riemannian metric around the boundary and hence the separation between the two classes. In the original algorithm proposed in [4], the kernel is enlarged at the positions of support vectors (SVs), which takes into account the fact that SVs are in the vicinity of the boundary. This approach, however, is susceptible to the distribution of SVs, since the magnification tends to be biased towards the high density region of SVs, and the distribution of SVs is determined by the distribution of data points. Although a modified version was suggested in [5] to meet this difficulty, the algorithm still suffers a certain level of susceptibility. Also the modified algorithm is hard to apply in high dimensional cases.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3496, pp. 831–836, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In the present study we present a new way of scaling the kernel function. The new approach will enlarge the kernel by acting directly on the distance measure to the boundary, instead of the positions of SVs as used before. Experimental study shows that the new algorithm works robustly, and overcomes the susceptibility of the original method.

2 Scaling the Kernel Function

The SVM solution to a binary classification problem is given by a discriminant function of the form [1, 2]

$$f(\mathbf{x}) = \sum_{s \in SV} \alpha_s y_s K(\mathbf{x}_s, \mathbf{x}) + b \tag{1}$$

A new out-of-sample case is classified according to the sign of $f(\mathbf{x})$. The support vectors are, by definition, those $\mathbf{x}_i$ for which $\alpha_i > 0$. For separable problems each support vector $\mathbf{x}_s$ satisfies

$$f(\mathbf{x}_s) = y_s = \pm 1 \, .$$

In general, when the problem is not separable or is judged too costly to separate, a solution can always be found by bounding the multipliers α_i by the condition $\alpha_i \leq C$, for some (usually large) positive constant C.

2.1 Kernel Geometry

It has been observed that the kernel $K(\mathbf{x}, \mathbf{x}')$ induces a Riemannian metric in the input space S [3, 4]. The metric tensor induced by K at $\mathbf{x} \in S$ is

$$g_{ij}(\mathbf{x}) = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x'_j} K(\mathbf{x}, \mathbf{x}') \Big|_{\mathbf{x}'=\mathbf{x}} \, . \tag{2}$$

This arises by considering K to correspond to the inner product

$$K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}') \tag{3}$$

in some higher dimensional feature space H, where $\boldsymbol{\phi}$ is a mapping of S into H. The inner product metric in H then induces the Riemannian metric (2) in S via the mapping $\boldsymbol{\phi}$.

The volume element in S with respect to this metric is given by

$$dV = \sqrt{g(\mathbf{x})} \, dx_1 \cdots dx_n \tag{4}$$

where $g(\mathbf{x})$ is the determinant of the matrix whose (i, j)th element is $g_{ij}(\mathbf{x})$. The factor $\sqrt{g(\mathbf{x})}$, which we call the *magnification* factor, expresses how a local volume is expanded or contracted under the mapping $\boldsymbol{\phi}$. Amari and Wu [4] suggest that it may be beneficial to increase the separation between sample

points in S which are close to the separating boundary, by using a kernel $\tilde{K}$, whose corresponding mapping $\tilde{\phi}$ provides increased separation in H between such samples.

The problem is that the location of the boundary is initially unknown. Amari and Wu therefore suggest that the problem should first be solved in a standard way using some initial kernel K. It should then be solved a second time using a conformal transformation $\tilde{K}$ of the original kernel given by

$$\tilde{K}(\mathbf{x}, \mathbf{x}') = D(\mathbf{x})K(\mathbf{x}, \mathbf{x}')D(\mathbf{x}') \tag{5}$$

for a suitably chosen positive function $D(\mathbf{x})$. It is easy to check that $\tilde{K}$ satisfies the Mercer positivity condition. It follows from (2) and (5) that the metric $\tilde{g}_{ij}(\mathbf{x})$ induced by $\tilde{K}$ is related to the original $g_{ij}(\mathbf{x})$ by

$$\begin{aligned}\tilde{g}_{ij}(\mathbf{x}) = {} & D(\mathbf{x})^2 g_{ij}(\mathbf{x}) + D_i(\mathbf{x})K(\mathbf{x}, \mathbf{x})D_j(\mathbf{x}) \\ & + D(\mathbf{x})\Big\{K_i(\mathbf{x}, \mathbf{x})D_j(\mathbf{x}) + K_j(\mathbf{x}, \mathbf{x})D_i(\mathbf{x})\Big\}\end{aligned} \tag{6}$$

where $D_i(\mathbf{x}) = \partial D(\mathbf{x})/\partial x_i$ and $K_i(\mathbf{x}, \mathbf{x}) = \partial K(\mathbf{x}, \mathbf{x}')/\partial x_i \,|_{\mathbf{x}'=\mathbf{x}}$. If $g_{ij}(\mathbf{x})$ is to be enlarged in the region of the initial class boundary, $D(\mathbf{x})$ needs to be largest in that vicinity, and its gradient needs to be small far away. Amari and Wu consider the function

$$D(\mathbf{x}) = \sum_{i \in SV} e^{-\kappa \|\mathbf{x} - \mathbf{x}_i\|^2} \tag{7}$$

where κ is a positive constant. The idea is that support vectors should normally be found close to the boundary, so that a magnification in the vicinity of support vectors should implement a magnification around the boundary. A possible difficulty of (7) is that $D(\mathbf{x})$ can be rather sensitive to the distribution of SVs, consider magnification will tend to be larger at the high density region of SVs and lower otherwise. A modified version was proposed in [5] which consider different κ_i for different SVs. κ_i is chosen in a way to accommodate the local density of SVs, so that the sensitivity with respect to the distribution of SVs is diminished. By this the modified algorithm achieves some improvement, however, the cost it brings associated with fixing κ_i is huge. Also its performance in high dimensional cases is uncertain. Here, rather than attempt further refinement of the method embodied in (7), we shall describe a more direct way of achieving the desired magnification.

2.2 New Approach

The idea here is to choose D so that it decays directly with distance, suitably measured, from the boundary determined by the first-pass solution using K. Specifically we consider

$$D(\mathbf{x}) = e^{-\kappa f(\mathbf{x})^2} \tag{8}$$

where f is given by (1) and κ is a positive constant. This takes its maximum value on the separating surface where $f(\mathbf{x}) = 0$, and decays to $e^{-\kappa}$ at the margins of the separating region where $f(\mathbf{x}) = \pm 1$.

3 Geometry and Magnification

3.1 RBF Kernels

To proceed, we need to consider specific forms for the kernel K. Here, we consider the Gaussian radial basis function kernel

$$K(\mathbf{x}, \mathbf{x}') = e^{-\|\mathbf{x}-\mathbf{x}'\|^2/2\sigma^2} \ . \tag{9}$$

It is straightforward to show that the induced metric is Euclidean with

$$g_{ij}(\mathbf{x}) = \frac{1}{\sigma^2}\,\delta_{ij} \tag{10}$$

and the volume magnification is the constant

$$\sqrt{g(\mathbf{x})} = \frac{1}{\sigma^n} \ . \tag{11}$$

3.2 Conformal Kernel Transformations

For illustration, we consider a simple toy problem as shown in Fig.1(a), where 100 points have been selected at random in the square as a training set, and classified according to whether they fall above or below the curved boundary, which has been chosen as e^{-4x^2} up to a linear transform. Our approach requires a first-pass solution using conventional methods. Using a Gaussian radial basis kernel with width 0.5 and soft-margin parameter $C = 10$, we obtain the solution shown in Fig.1(b). This plots contours of the discriminant function f, which is of the form (1). For sufficiently large samples, the zero contour in Fig.1(a) should coincide with the curve in Fig.1(b).

To proceed with the second-pass we need to use the modified kernel given by (5) where K is given by (9) and D is given by (8). It is interesting first to calculate the general metric tensor $\tilde{g}_{ij}(\mathbf{x})$ when K is the Gaussian RBF kernel (9) and $\tilde{K}$ is derived from K by (5). Substituting in (6), and observing that in this case $K(\mathbf{x}, \mathbf{x}) = 1$ while $K_i(\mathbf{x}, \mathbf{x}) = K_j(\mathbf{x}, \mathbf{x}) = 0$, we obtain

$$\tilde{g}_{ij}(\mathbf{x}) = \frac{D(\mathbf{x})^2}{\sigma^2}\,\delta_{ij} + D_i(\mathbf{x})D_j(\mathbf{x}) \ . \tag{12}$$

Observing that $D_i(\mathbf{x})$ are the components of $\nabla D(\mathbf{x}) = D(\mathbf{x})\nabla \log D(\mathbf{x})$, it follows that the ratio of the new to the old magnification factors is given by

$$\sqrt{\frac{\tilde{g}(\mathbf{x})}{g(\mathbf{x})}} \ = \ D(\mathbf{x})^n \sqrt{1 + \sigma^2 \|\nabla \log D(\mathbf{x})\|^2} \ . \tag{13}$$

This is true for any positive scalar function $D(\mathbf{x})$. Let us now use the function given by (8) for which

$$\log D(\mathbf{x}) \ = \ -\kappa f(\mathbf{x})^2 \tag{14}$$

where f is the first-pass solution given by (1) and shown, for example, in Fig.1(b). This gives

$$\sqrt{\frac{\tilde{g}(\mathbf{x})}{g(\mathbf{x})}} \;=\; \exp\left\{-n\kappa f(\mathbf{x})^2\right\} \sqrt{1 + 4\kappa^2\sigma^2 f(\mathbf{x})^2 \|\nabla f(\mathbf{x})\|^2}\,. \tag{15}$$

This means that

1. *the magnification is constant on the separating surface* $f(\mathbf{x}) = 0$;
2. *along contours of constant* $f(\mathbf{x})$*, the magnification is greatest where the contours are closest.*

These two properties are illustrated in Fig.1(c).

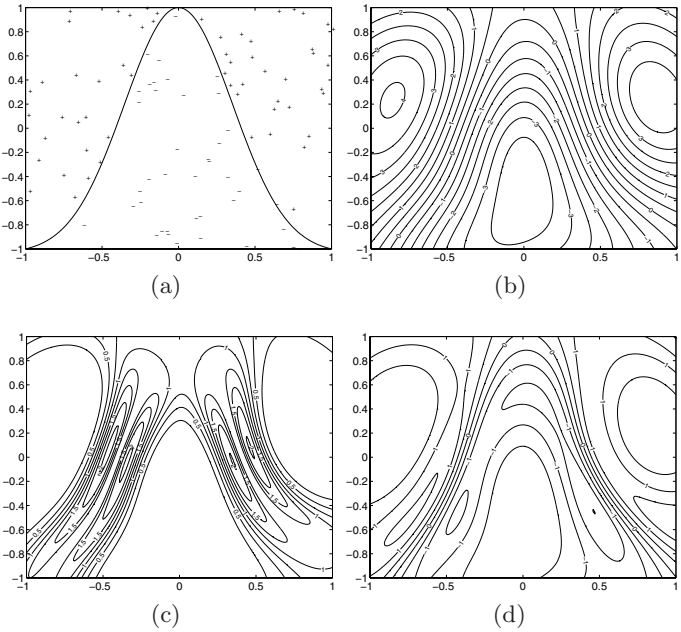

Fig. 1. (a) A training set of 100 random points classified according to whether they lie above (+) or below (−) the Gaussian boundary shown. (b) First-pass SVM solution to the problem in (a) using a Gaussian kernel. The contours show the level sets of the discriminant function f defined by (1). (c) Contours of the magnification factor (15) for the modified kernel using $D(\mathbf{x}) = \exp\{-\kappa f(\mathbf{x})^2\}$ with f defined by the solution of (b). (d) Second-pass solution using the modified kernel.

4 Simulation Results

The only free parameter in the new approach is κ. It is clear that κ is scale-invariant and independent of the input dimension. Through experimental study, we find that in most cases a suitable κ is approximately the reciprocal of $|f|_{max}$, the maximum of the absolute value of $f(\mathbf{x})$ in the first pass solution.

After applying the modified kernel $\tilde{K}$, we solve the classification problem in Fig.1(a) again, and obtain the solution in Fig.1(d). Comparing this with the first-pass solution of Fig.1(b), notice the steeper gradient in the vicinity of the boundary, and the relatively flat areas remote from the boundary. We have repeated the experiment 10000 times, with a different choice of 100 training sites and 1000 test sites on each occasion, and have found an average of 14.5% improvement in classification performance.

We also apply the new method for some real-world problems and obtain encouraging results. For instance, for the Mushroom dataset in the UCI Machine Learning Repository, we observe the improvement as shown in Table.1 (The misclassification rates are illustrated. The number of training and testing examples are 100 and 1000, respectively, which are both randomly chosen from the database. The results are calculated after averaging over 100 trials.).

	Before Modification	After Modification
$C = 10, \sigma = 0.6$	11.20%	7.05%
$C = 10, \sigma = 1.0$	4.02%	2.95%
$C = 50, \sigma = 0.6$	10.86%	7.46%
$C = 100, \sigma = 0.6$	11.97%	7.75%

5 Conclusion

The present study investigates a data-dependent way of optimizing the kernel functions in SVMs. The proposed algorithm is a modification of the one in [4, 5]. Compared with the original, the new algorithm achieves better performance in term of that it is more robust with respect to the data distribution. The new algorithm is also simple and has only one free parameter. It is therefore valuable as a general methodology for supplementing normal SVM training to enhance classification performance.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
2. Scholkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002)
3. Burges, C.: Geometry and Invariance in Kernel Based Method. In: Scholkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods, MIT Press (1999) 89–116
4. Amari, S., Wu, S.: Improving Support Vector Machine Classifiers by Modifying Kernel Functions. Neural Networks, **12** (1999) 783–789
5. Wu, S., Amari, S.: Conformal Transformation of Kernel Functions: A Data-Dependent Way to Improve Support Vector Machine Classifiers. Neural Processing Letters, **15** (2001) 59–67