Scaling Gaussian RBF kernel width to improve SVM classification

Qun Chang Qingcai Chen Xiaolong Wang
ITNLP Lab, School of Computer Science and Technology
Harbin Institute of Technology, Harbin, 150001, China
E-mail: qchang@insun.hit.edu.cn
qcchen@insun.hit.edu.cn
wangxl@insun.hit.edu.cn

***Abstract*—Support vector classification with Gaussian RBF kernel is sensitive to the kernel width. Small kernel width may cause over-fitting, and large one under-fitting. The so-called optimal kernel width is merely selected based on the tradeoff between under-fitting loss and over-fitting loss. So, there exists urgent need to further reduce the tradeoff loss. To circumvent this, we scale the kernel width in a distribution-dependent way. Experiments validate the feasibility of this method. Existing problems are also discussed.**

I. INTRODUCTION

Support vector machines (SVMs, SVM) are a class of learning algorithms which are based on the principle of structural risk minimization (SRM) [1] [2]. SVM has been used in many machine learning fields, such as classification, regression estimation, and kernel PCA, for its good generalization ability. Each kind of classifier needs a metric to measure the similarity or distance between patterns. SVM classifier uses inner product as metric. If there are dependent relationships among pattern's attributes, such information will be accommodated through additional dimensions, and this can be realized by a mapping [3] $\Phi: X \to H$; $\mathbf{x} \to \Phi(\mathbf{x})$, where H is called feature space. And herein inner product similarity is computed through $<\Phi(\mathbf{x}), \Phi(\mathbf{y})>$ for pattern $\mathbf{x}$ and $\mathbf{y}$. In SVM literature, the above course is realized through kernel function

$$k(\mathbf{x},\mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \tag{1}$$

Kernels can be regarded as generalized dot products [3]. Among various kernels, Gaussian RBF kernel is suggested to use [2] [3] [4]. Keerthi and Lin show that the linear kernel is a special case of RBF [5]. The sigmoid kernel behaves like RBF for certain parameters [6]. A Gaussian RBF kernel is formulated as

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right) \tag{2}$$

Throughout this paper, we only discuss the support vector classification with Gaussian RBF kernel. In feature space patterns are rarely evenly distributed. In the viewpoint of metric induced by RBF kernel, each training pattern shapes a covering sphere centered by this pattern self. All these spheres have the same radius which is subject to the width σ . The radius is understood in such a way: Where the distance from a pattern exceeds this radius, any kernel value centered by the pattern is below some low threshold value and the pattern's effects almost vanish. The feature space is spanned by these spheres, which overlap each other. If the overlapping topology is severely unbalanced, the kernel-induced metric will not adapt well to the SVM learning strategy. The unevenness of feature space is out of the reach of the strength of const kernel width, as the SVM learning strategy has a close relationship with nearest neighbors (NN) rule [9] [10]. If σ is very small, the patterns will tend to be dissimilar and so over-fitting will happen. If σ is very large, the patterns will tend to be very similar and under-fitting will arise. The uniform σ tends to make the dense areas over-fitting and the sparse areas under-fitting. The so-called optimal σ is merely the tradeoff between the over-fitting loss in dense areas and the under-fitting loss in sparse areas. So, there exists urgent need to further reduce the tradeoff loss. To aim at this, we scale the kernel width in a distribution-dependent way, and experiments validate the feasibility of this method. This idea is illustrated in Fig.1.

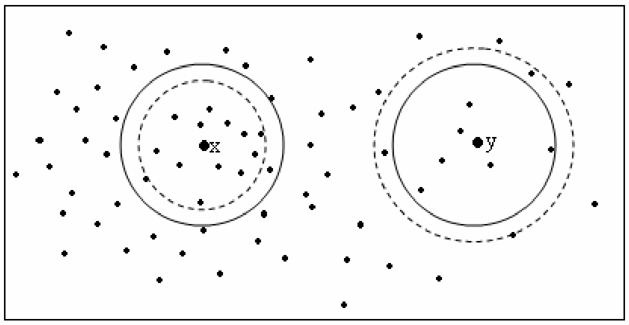

Fig. 1. Two covering spheres (solid line) shaped by point **x** and **y** have the same radius which is subject to the width σ. The area where **x** lies is dense and sparse where **y** lies on the contrary. The width σ can be scaled to adapt to the distribution. The spheres with dotted lines are ones shaped by the metric with scaled width.

0-7803-9422-4/05/$20.00 ©2005 IEEE

The rest of the paper is organized as follows. Brief introductions on SVM and related work are in Section Ⅱ. In Section Ⅲ the kernel width-scaling scheme is proposed, and then follow the experiments and related discussions in Section Ⅳ. At last, we make a brief conclusion in Section Ⅴ.

II. SVM AND RELATED WORK

A. SVM Classification

Given a dataset $D = \{(\mathbf{x}_i, y_i)\}^l_{i=1}$ of labeled patterns, where $y_i \in \{-1, 1\}$, SVM classification [1] [2] is to construct a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ to classify the patterns into two classes. This hyperplane is found by maximizing the margin between two classes, i.e., optimizes such a primal problem:

$$\min \quad J(\mathbf{w}, b, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{3}$$

$$\text{s.t.} \quad y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i \tag{4}$$

$$\xi_i \geq 0, \quad i = 1, \cdots, l \tag{5}$$

If the two classes are non-separable, SVM introduces slack variables ξ to get the tradeoff between margin and misclassification errors. The solution of primal problem is solved through its dual problem which is in the form of

$$\max \ W(\mathbf{a}) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \tag{6}$$

$$\text{s.t.} \ 0 \leq \alpha_i \leq C, \ i = 1, \ldots, l \tag{7}$$

$$\sum_{i=1}^{l} a_i y_i = 0 \tag{8}$$

The coefficients a_i are the solution of the dual problem, and then the decision function is formulated by

$$f(\mathbf{x}) = sign\left(\sum\nolimits_{i=1}^{l}\alpha_i y_i \left\langle \mathbf{x}, \mathbf{x}_i \right\rangle + b\right) \tag{9}$$

The weight vector **w** is expressed with support vectors by setting the derivatives of *J* with respect to primal variables.

$$\mathbf{w} = \sum_{i=1}^{l} a_i y_i \mathbf{x}_i \tag{10}$$

By adopting kernel substituting tricks, cf. (1), (9), we get the nonlinear decision function:

$$f(\mathbf{x}) = sign\left(\sum\nolimits_{i=1}^{l}\alpha_i y_i k\left(\mathbf{x}, \mathbf{x}_i\right) + b\right) \tag{11}$$

Detailed discussions on SVM refer to [1] [2] [3].

B. Related work

How to select the width σ of RBF kernel in SVM literature has been discussed in [11] [12] [13] [14] [15]. Our interest is to vary the width σ to eliminate the inconsistency of the coexisting over-fitting and under-fitting in SVM learning. Varying the window size (i.e. width σ in Gaussian kernel) is the strategy of *k*th nearest neighbor estimate [16] and the adaptive kernel estimate [17] [18] [19]. *k*th nearest neighbor estimate is to estimate density by modifying the window size to let every local volume accommodate sufficient patterns. The adaptive kernel estimate treats the window size as being data-driven, that is, using broader kernel bandwidth (i.e. width) where data is sparse and narrower kernels bandwidth where data is dense, to estimate density. Studies on regression estimation with variable width mainly concentrate on the polynomial and not the Gaussian kernels [20] [21], and therefore such work will be not mentioned here. Paper [22] proposed a method of dynamic adjustment of width of Gaussian kernel in General Regression Neural Networks (GRNN) [23]. This method chooses kernel width according to the distance between input vectors and training samples, that is, $\sigma = [(\mathbf{x}\text{-}\mathbf{x}_i)^T (\mathbf{x}\text{-}\mathbf{x}_i)]^\alpha$, where $\alpha \in (0, 1)$, $\mathbf{x}_i$: training samples, **x**: input vector. Modifying the width of RBF kernel approximately has the same pursuit as does modifying the kernel value. Amari and Wu [24] [25] modified a kernel with a conformal mapping. If $c(\mathbf{x})$ is a positive real valued function of **x**, then a new kernel is created by

$$\tilde{k}\left(\mathbf{x}, \mathbf{x}'\right) = c\left(\mathbf{x}\right)c\left(\mathbf{x}'\right)k\left(\mathbf{x}, \mathbf{x}'\right) \tag{12}$$

$$c\left(\mathbf{x}\right) = \sum_{i \in SV}\alpha_i \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{x}_i\right\|}{2\tau^2}\right) \tag{13}$$

where *SV* are support vectors, α_i the positive number representing the contributions of $\mathbf{x}_i$, τ a free parameter. Since the support vectors lie on the boundary of the margin, this method enhances the spatial resolution on the boundary surface. The width τ of (13) is computed through the distances, in input space, between support vectors, so it is dynamic but not well adaptive to spatial distribution in Feature Space. Paper [26] also introduces a conformal transformation on kernel function, but width τ is calculated through the distances in feature space, that is,

$$\begin{aligned}\tau_k &= average\left(\left\|\Phi\left(\mathbf{x}_i\right) - \Phi\left(\mathbf{x}_k\right)\right\|\right), \\ & i \in \left\{\left\|\Phi\left(\mathbf{x}_i\right) - \Phi\left(\mathbf{x}_k\right)\right\|^2 < M, y_i \neq y_k\right\}\end{aligned} \tag{14}$$

where the *average* in (14) comprises all the support vectors in $\Phi(\mathbf{x}_k)$'s neighborhood within the radius of *M* but having a different class label. Setting τ_k in this way takes into consideration the spatial distribution of the support vectors in feature space and so τ_k is adaptive [26].

III. WEIGHTING KERNEL WIDTH

To reduce the coexisting over-fitting and under-fitting loss in support vector classification with the Gaussian RBF kernel, the kernel width must adapt, to some extent, the feature space distribution. The scaling principle is that: in dense areas the width will become narrow through some weights less than 1, and in sparse areas the width will be enlarged through some weights more than 1. Firstly, we introduce several relationships.

(a) The relationship of σ and λ

We rewrite the (2) in the new form of

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\lambda\|\mathbf{x}-\mathbf{y}\|^2\right) \tag{15}$$

(b) The relationship of similarity and distance

In feature space, in view of (1) and (15), the square distance of pattern **x** and **y** is

$$d^2 = \|\Phi(\mathbf{x})-\Phi(\mathbf{y})\|^2 == 2-2k(\mathbf{x},\mathbf{y}) \tag{16}$$

From (16), we see, the value of $k(\mathbf{x}, \mathbf{y})$ is inverse to d.

(c) Dense vs sparse in feature space

If a pattern **x** drops in dense area its neighbors are close to it, and on the contrary in sparse area. We consider the **x**'s k-NNs (k nearest neighbors) and get such a value

$$sim\_knn(\mathbf{x}) = \frac{1}{k}\sum_i k(\mathbf{x},\mathbf{x}_i),\ \mathbf{x}_i \in k\text{ - NNs} \tag{17}$$

From (17) and the relationship *(b)*, we see, *sim_knn*(**x**) is an index of density of **x**'s neighborhood. A large *sim_knn*(**x**) means a dense area where **x** lies. Selecting the **x**'s k-NNs is to select the k patterns which have kernel value $k(\mathbf{x}, \cdot)$ not less than these of the rest part in training set.

(d) Scaling scheme

Based on the weighting principle and the relationship *(a)* above, we have such a weighting scheme. Each pattern **x** has a *λ_weight*(**x**) which modifies the λ in (16). If the *sim_knn*(**x**) is large (i.e. dense area), a large *λ_weight*(**x**) will increase the λ (i.e. decrease the width σ in (2)). And on the contrary a small *sim_knn*(**x**) means that a small *λ_weight*(**x**) is used. We confine all weights to fluctuate in a very small range around 1 to reduce distorting the metric induced by RBF kernel as possible. The modified Gaussian RBF kernel is formulated as

$$\tilde{k}(\mathbf{x},\mathbf{y}) = \exp\left(-\lambda\_weight(\mathbf{x})\times\lambda\_weight(\mathbf{y}) \times \lambda\times\|\mathbf{x}-\mathbf{y}\|^2\right) \tag{18}$$

Now, we describe how to compute the *λ_weight*(**x**). Assume we have a training set S with l patterns, and the steps are as follows.

1) Compute sim_knn($\mathbf{x}_i$), $\mathbf{x}_i \in$ S, according to (17)
2) Compute the mean of sim_knn($\mathbf{x}_i$) by

$$mean\_sim\_knn = \frac{1}{l}\sum_i sim\_knn(\mathbf{x}_i),\ \mathbf{x}_i \in S \tag{19}$$

3) Compute λ_weight(**x**) by

$$\lambda\_weight(\mathbf{x}) = 1+\eta[sim\_knn(\mathbf{x}) - mean\_sim\_knn] \tag{20}$$

where $\eta \in [0, 1]$ is a factor of weighting intensity. Thus, *λ_weight*(**x**) will fluctuate within a very small range around 1. Note, in prediction, the future pattern finds its *λ_weight*( ) by (20) in training set and not in testing set. Let *neighbor_rate* denote the ratio of k (cf. (17)) to l, and we get k by

$$k = floor(l \times rate) \tag{21}$$

where *floor*(x) returns the largest integer not larger than x, e.g., *floor*(8.6)=8. In the following experiment, we will use *rate* instead of k.

IV. EXPERIMENTS

Stalog heart disease data [27] contains 270 observations related to heart disease (i.e. two classes) and has 13 attributes. We rescale linearly all the attributes within the interval [-1, 1] to avoid dominating phenomenon of attributes with large numeric ranges and numerical difficulties during the calculation. This advice is from [28]. The dataset is divided into two parts, one with 150 patterns as training set, and the other with 120 for testing. In experiment, fixed-width kernel and weighted-width kernel adopt the forms of (15) and (18), respectively. The parameters C, λ, *rate*, η come from Eq. (3), (15) or (18), (21), (20), respectively. The experimental results are listed in TABLE Ⅰ.

TABLE I

RESEULTS OF STALOG HEART DISEASE CLASSIFICATION

width λ	*C*	Fixed-width kernel Accuracy	Width-scaled kernel		
			rate	η	Accuracy
1	10	73.33%	0.03	1	75.83%
			0.03	0.5	75%
0.8	10	75.83%	0.03	1	76.67%
			0.03	0.5	78.33%
0.6	10	76.67%	0.03	1	76.67%
			0.03	0.5	76.67%
0.4	10	75%	0.03	1	76.67%
			0.03	0.5	76.67%
0.1	10	78.33%	0.03	1	72.5% **
			0.02	1	78.33%
			0.03	0.5	79.17%

Note: each unfilled position adopts the same value as that just above.

In each row from TABLE Ⅰ, we see, apart from the row with **, the performance with weighted-width Gaussian RBF kernel behaves relatively better than that with const-width kernel. Since SVM performance is globally optimal, its little improvement, nevertheless, will be positively evaluated. Through changing the parameter *rate* and η, the performance of weighted width kernel (the row with **) improved (the last two row). However, there exists a problem that the kernel with weighted-width kernel may be indefinite [8]. In practice, the indefinite kernel has many successful applications in SVM literature [6] [7] [30]. Generally, we can use indefinite kernel in SVM as usual, although theoretical foundation is missing [8]. Some theoretical studies and practical methods on indefinite kernel in SVM literature are proposed in [6] [8] [29] and limited space precludes the further discussions on the aspect.

V. CONCLUSION

The previous studies on SVM learning didn't take much assumption on the patterns' spatial distribution, however, which affects the learning quality. The coexisting under-fitting and over-fitting phenomenon shows the insufficiency of fixed-width kernels, and to overcome this shortage, we may let the kernel, to some extent, carry such distribution information into learning process. Our scheme of scaling the kernel width can, to some extent, take use of the spatial distribution information to improve the learning. The experiment validates the feasibility of our method, which is simple and practical, yet to be further improved, we think. The weighted-width kernel also has possible side-effect, i.e. being indefinite, which nevertheless isn't a serious problem practically [8]. The SVM with indefinite kernels tends to be a hot topic in SVM literature.

ACKNOWLEDGMENT

The authors would like to thank Chih-Chung Chang and Chih-Jen Lin for their LIBSVM on which our many algorithms are based, whether in this work or in daily studies. This work is supported by the key program project (No. 60435020) of National Natural Science Foundation of China (NSFC).

REFERENCES

[1] Vladimir N. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998.
[2] Vladimir N. Vapnik. *The Nature* of *Statistical Learning Theory*, New York: Springer-Verlag, 1995
[3] Bernhard Schölkopf and Alex Smola, *Learning with kernels*. MIT Press, Cambridge, MA, 2002.
[4] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," In Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, vol. 5, pp. 147-155, Morgan Kaufmann, San Mateo, CA, 1993.
[5] S. S. Keerthi, C. -J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation* 15(7), pp. 1667-1689, 2003.
[6] H. T. Lin, C. J. Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," *Technical report*, Dept. of Computer Science and Information Engineering, National Taiwan Univ., 2003.
[7] B. Haasdonk, D. Keysers, "Tangent distance kernels for support vector machines," *In Proc. of ICPR' 02*, vol. 2, pp. 864-868, 2002.
[8] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE Trans. Pattern Analysis and Machine intelligence*, vol. 27, pp. 482-492, Apr. 2005.
[9] B. Karaçali, H. Krim, "Fast minimization of structural risk by nearest neighbor rule," *IEEE Trans. Neural Network* 14 (1), pp. 127-137, 2003.
[10] B. Karaçali, R. Ramanath, W. E. Snyder, "A comparative analysis of structural risk minimization by support vector machines and nearest neighbor rule," *Pattern Recognition Letters* 25(1), pp. 63-71, 2004.
[11] N. Cristianini, J. Shawe-Tayor, C. Campbell, "Dynamically adapting kernels in support vector machines," In M. Kearns, S. Solla, D. Cohn, eds, *Adances in Neural Information Processing Systems*, 11, Denver, CO, MIT Press, 1998.
[12] G. Rätsch, T. Onoda, K. R. Müller, "Soft margins for AdaBoost," *Machine Learning* 42, pp. 287-320, 2001.
[13] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning* 46, pp. 131-160, 2002.
[14] R. Burbidge, "Adaptive kernels for support vector classification," *Lecture Notes in Statistics*, Issu. 171, pp. 345-356, Springer Verlag, NY, 2003.
[15] S. S. Keerthi, C. –J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation* 15(7), pp. 1667-1689, 2003.
[16] D. Loftsgaarden, C. Quesenberry, "A nonparametric estimate of a multivariate density function," *Ann. Math. Stat.* 36, pp. 1049–1051, 1965.
[17] L. Breiman, W. Meisel, E. Purcell, "Variable kernel estimates of multivariate densities," *Technometrics* 19, pp. 135-144. 1977.
[18] L. Devroye, C. S. Penrod, "The consistency of automatic kernel density estimates," *Ann. Statist.* 12, pp1231-1249, 1984.
[19] V. Katkovnik, I. Shmulevich, "Kernel density estimation with varying window size," *Pattern Recognition Letters*, Vol. 23, pp. 1641-1648, 2002.
[20] C. G. Atkeson, A. W. Moore, S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11(1-5), pp. 11-73, 1997.
[21] V. Katkovnik, "A new method for varying adaptive bandwidth selection," *IEEE Trans. Signal Processing,* vol. 47, pp. 2567-2571, Sept. 1999.
[22] Z. –W. Li, J. –Z. Sun, J. –W. Zhang, Z. Wei, "Computing models based on GRNNS," *In IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1853-1856, 2003.
[23] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol.2, pp. 568-576, Nov. 1991.
[24] S. Amari, S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks* 12(6), pp. 783-789, 1999.
[25] S. Wu, S. Amari, "Conformal transformation of kernel functions: A data-dependent way to improve the performance of support vector machine classifiers," *Neural Processing Letters*, vol. 15, pp. 59-67, Feburary, 2002.
[26] G. Wu, E. Y. Chang, "Adaptive feature-space conformal transformation for imbalanced-data learning," *In Twentieth International Conference on Machine Learning*, Washington, DC, pp. 816-823, Aug. 2003.
[27] UCI Machine Learning Repository: Statlog Project Databases: Heart data. Available: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog
[28] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
[29] D. DeCoste, Bernhard Schölkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, pp. 161–190, 2002.