Choosing Parameters for 2-norm Support Vector Machines

Xiaohuan Yang [a], Xiaoming Wang, Yong Tian, Xiao Zheng

School of Computer and Software Engineering, Xihua University, Chengdu 610039, China

[a]yangxh_09@aliyun.com

Abstract. The problem of automatically tuning multiple parameters for pattern recognition Support Vector Machines (SVM) is considered. This is done by minimizing some estimates of the generalization error of SVM using a gradient descent algorithm over the set of parameters. This error can be estimated via a bound given by theoretical analysis. Inspired by the relationship between the radius of the MEB and the trace of within-class scattering matrix, this paper incorporates the later into 2-norm Support Vector Machines(L_2 -SVM) to estimate the error, and automatically tune parameters of SVM by introducing the gradient descent algorithm. Detailed theoretical analysis is conducted to show how the resulting optimization is efficiently solved.

Keywords: support vector machines, radius-margin bound, gradient descent algorithm, within-class scattering matrix.

1. Introduction

As one of the most popular machine learning approaches, kernel methods have been widely used in many applications [1-3]. Support vector machines (SVM), as a kernel method, have been a promising tool for data classification [4]. For such tasks, the performance strongly depends on the choice of some parameters. These parameters include: the regularization parameter C, which determines the tradeoff between minimizing the training error and minimizing model complexity; and parameter σ of the kernel function that implicitly defines the nonlinear mapping from input space to some high-dimensional feature space. These higher level parameters are usually referred as hyperparameters.

The success of performance depends on the tuning of hyperparameters that affect the generalization error. Tuning these hyperparameters is usually done by minimizing the estimated generalization error such as the k-fold cross-validation error or the leave-one-out (LOO) error [5-8]. LOO is particularly of theoretical interest, because it makes use of the greatest possible data for training and does not involve random sampling. However, LOO exhibits serious limitations. In addition to the fact that it is computationally expensive, it has a larger variance than cross validation. Another type of bound incorporates the radius of the minimum enclosing ball (MEB) of training data into SVM formulation by considering that the generalization error of SVM is upper bounded by the ratio of the radius to the margin, so called the radius-margin bound. However, in [9], the radius of the MEB is incorporated by solving a more optimization method, which adds one extra level of quadratic optimization on top of the existing SVM framework, and the notorious sensitivity of the radius of the MEB to outliers or noisy samples can adversely affect the kernel learning performance of SVM.

From our point of view, the underlying cause for the aforementioned drawbacks lies at that the radius of the MEB is sensitive to outliers and increases the computation cost. To improve this situation, the paper proposes to incorporate the trace of within-class scattering matrix of training data into L_2 -SVM, which is inspired by its close relationship with the radius of the MEB. In particular, to well justify the incorporation of radius information, we strictly comply with the radius-margin bound and focus on the L_2 -SVM with a soft margin, which can be reformulated as the SVM with a hard margin using a slightly modified kernel, making the radius-margin bound still applicable. It has the following advantages.

(1) More robust to outliers and noisy samples. By uniformly weighting each sample, the trace of within-class scattering matrix is more robust than the radius of the MEB in characterizing the scattering of training points.

(2) More conducive to improve the classification performance.

(3) More computationally efficient. By substituting the radius with the trace of total scattering matrix, our method avoids the extra level of quadratic optimization needed to compute the radius, which well reduces the computational cost of each iteration.

In addition, through using the 2-norm soft-margin formulation of SVM, this work provides an efficient way for SVM to utilize the information of the radius of the MEB and uses the reduced gradient method [10] to tune the regularization parameter C and kernel parameter σ.

The rest of this paper is organized as follows. We review the related work in section 2; in section 3, we give the formulation of the proposed. Then, we prove that its optimization problem can be written into the common form of existing 1-norm SVM and show how it can be efficiently solved. After that, some preliminary discussion on the proposed criterion and radius-margin bound is conducted, and the conclusion is in section 4.

2. Related work

Let $\{x_i, y_i\}$ be a given set of training data, where x_i is the ith input vector and y_i is the target value, $y_i \in \{\pm 1\}$. $y_i = 1$ denotes that x_i is in class 1 and $y_i = -1$ denotes that x_i is in class 2.

2.1 L_2-SVM.

In this paper, we consider the support vector machine (SVM) problem formulation that uses L_2-norm soft-margin given by

$$\begin{aligned} &\min \ \frac{1}{2}\| w\|^2 + \frac{C}{2}\sum_i^l \xi_i^2 \\ &s.t \quad y_i(w\cdot\varphi(x_i)+b) \geq 1-\xi_i,\ i=1,2,\ldots,l \end{aligned} \tag{1}$$

This problem is computationally solved using the solution of its dual form:

$$\begin{aligned} &\max \ W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2}\sum_{i=1}^l\sum_{j=1}^l y_i y_j \alpha_i \alpha_j \tilde{K}(x_i, x_j) \\ &s.t \quad \sum_{i=1}^l y_i \alpha_i = 0,\ \alpha_i \geq 0,\ i=1,2,\ldots,l \end{aligned} \tag{2}$$

Where $\tilde{K}(x_i, x_j) = K(x_i, x_j) + \frac{1}{C}\delta_{ij}$, $K(x_i, x_j) = \varphi(x_i)\cdot\varphi(x_j)$ is a kernel function that satisfies the Mercer conditions (symmetric positive definite function), and δ_{ij} is the Kronecker δ, which is 1 when $i = j$, and 0 otherwise, C is the regularization parameter.

Different Kernel functions have been designed based on the kernel feature space, among which the following RBF kernel is commonly used:

$$K(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j\|^2}{2\sigma^2}\right) \tag{3}$$

Here, σ is kernel parameter that implicitly defines the nonlinear mapping from input space to some high-dimensional feature space.

2.2 The Radius Margin Bound.

The generalization ability of SVM depends on the margin of training points. For SVM with hard-margin formulation, it was shown by Chappelle et al. [9] that the following bound holds:

$$LOO_{err} \leq \frac{1}{4l} R^2 \| w\|^2 \tag{4}$$

Where w the weight is vector and $\|w\|^2$ is computed by (2), i.e. $\|w\|^2 = 2W(\alpha)$. R is the radius of the smallest sphere that contains all the training points in the feature space. The right-hand side of (4) is usually referred as the radius-margin (RM) bound. It has been shown that R^2 is the objective value of the following optimization problem.

$$\max_{\beta} R^2 = \sum_{i=1}^{l} \beta_i K(x_i, x_i) - \sum_{i=1}^{l}\sum_{j=1}^{l} \beta_i \beta_j K(x_i, x_j)$$
$$s.t. \quad \beta_i \geq 0, \sum_{i=1}^{l} \beta_i = 1 \tag{5}$$

In order to solve the RM bound, two optimization problems are considered, in the first step, $\|w\|^2$ is computed by solving the quadratic programming (QP) in (2), then, R^2 is calculated by another QP in (5). The aforementioned procedure is repeated until a stopping criterion is satisfied, resulting in the two QP are solved at each iteration. This can obviously increase the computation cost of SVM-based the RM bound, particularly when the size of the training points is large.

3. Proposed Method

In this section, we first discuss the close relationship between R^2 and $tr(S_w)$, then incorporate $tr(S_w)$ into RM, and present the optimization problem formulation of the proposed algorithm, employ the reduced gradient method to tune the hyperparameters C and σ.

3.1 Close relationship between R^2 and $tr(S_w)$.

Recall that $x_i (i = 1, \ldots, l)$ denotes the *ith* training sample. The within-class scattering matrix is defined as $S_w = \sum_{i=1}^{c} \left[\sum_{j=1}^{n_i} (x_j^{(i)} - m^{(i)})(x_j^{(i)} - m^{(i)})^T \right]$, where $m^{(i)} = (1/n_i) \sum_{j=1}^{n_i} x_j^{(i)}$ is the sample-based each class mean. Although each training sample is implicitly mapped onto a feature space via the kernel trick and $tr(S_w)$ in that space is inaccessible, its trace can be explicitly expressed by the kernel function as

$$tr(S_w^{\varphi}) = tr(K) - \sum_{s=1}^{c} \frac{Sum(K_s)}{n_s} = tr(K) - \sum_{s=1}^{c} \frac{e_s^T K_s e_s}{n_s} \tag{6}$$

Where K is the kernel matrix based on total sample, K_i denotes the kernel matrix of *ith* class sample. The close relationship between $tr(S_w)$ and the squared radius of the MEB R^2 has been revealed in the literature [11]. Both measure the scattering of samples in a kernel-induced feature space, and $tr(S_w)$ can be shown as an approximation of R^2. The detailed analysis on the relationship can be found in [11, Appendix]. In this paper, instead of incorporating the radius of the MEB directly, we incorporate $tr(S_w)$, and the advantages are threefold.

Based on the previous discussion, we substitute R^2 with $tr(S_w)$ in the following to incorporate the radius information into the L_2-SVM formulation.

3.2 Incorporating $tr(S_w)$ into L_2-SVM.

The generalization error of SVM with a hard margin can be estimated by leave-one-out (LOO) error. This error is upper bounded by the ratio of the radius of the MEB to the margin, called the radius-margin bound. A comparison of different methods for model selection is in Duan [12], which shows the radius margin bound for L_2-SVM performs quite well. However, optimizing (7) will incur extra computational cost to compute the radius at each iteration. More importantly, the notorious sensitivity of the radius to the outliers in training data will possibly adversely affect its performance in predicting the generalization error.

Following the idea proposed in this paper, R^2 is replaced with $tr(S_w)$, and this leads to the objective function

$$\min_{C,\sigma^2} \min_{w,b} \frac{1}{4l} tr(S_w) \| w \|^2 \tag{7}$$

Where $tr(S_w)$ is defined in (6), $\| w \|^2$ is calculated in (2). Note that, we use the reduced gradient method to tune the hyperparameters C and σ. Application of gradient calculations, we denote $f(C,\sigma^2) = \frac{1}{4l} tr(S_w) \| w \|^2$, yields the following expressions:

$$\begin{cases} \dfrac{\partial f(C,\sigma^2)}{\partial C} = \dfrac{1}{4l}\left(\| w \|^2 \cdot \dfrac{\partial tr(S_w)}{\partial C} + tr(S_w) \cdot \dfrac{\partial \| w \|^2}{\partial C} \right) \\ \dfrac{\partial f(C,\sigma^2)}{\partial \sigma^2} = \dfrac{1}{4l}\left(\| w \|^2 \cdot \dfrac{\partial tr(S_w)}{\partial \sigma^2} + tr(S_w) \cdot \dfrac{\partial \| w \|}{\partial \sigma^2} \right) \end{cases} \tag{8}$$

The derivatives of $\| w \|^2$ are given by

$$\frac{\partial \| w \|^2}{\partial C} = \frac{1}{C^2} \sum_{i=1}^{l} \alpha_i^2, \quad \frac{\partial \| w \|^2}{\partial \sigma^2} = -\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \frac{\partial \tilde{k}(x_i, x_j)}{\partial \sigma^2} \tag{9}$$

The derivatives of $tr(S_w)$ are given by

$$\begin{cases} \dfrac{\partial tr(S_w)}{\partial C} = \dfrac{1}{C^2}\left(-tr(E) + \sum_{s=1}^{c} \dfrac{e_s^T e_s}{n_s} \right) \\ \dfrac{\partial tr(S_w)}{\partial \sigma^2} = tr\left(\sum_{i,j=1}^{l} \dfrac{\partial \tilde{k}(x_i, x_j)}{\partial \sigma^2} \right) - \sum_{s=1}^{c} e_s^T \left(\sum_{i,j=1}^{n_s} \dfrac{\partial \tilde{k}(x_i^{(s)}, x_j^{(s)})}{\partial \sigma^2} \right) e_s \end{cases} \tag{10}$$

Also

$$\frac{\partial \tilde{k}(x_i, x_j)}{\partial \sigma^2} = \tilde{k}(x_i, x_j) \cdot \frac{\| x_i - x_j \|^2}{2\sigma^4} \tag{11}$$

Thus, gradient of f is cheaply computed once f has been computed, the hyperparameters C and σ are updated alternately until convergence. The algorithm is outlined in Algorithm 1.

Algorithm 1

Input:	Training points, initial regularization parameter C and kernel parameter σ, the step size of each iteration η.
Output:	Optimal parameters C and σ.
Step1:	Initialization. Assign an initial value to the parameters C, σ and η.
Step2:	$i \leftarrow 0$
Step3:	Repeat
Step4:	Obtain α_{i+1} by solving the quadratic programming problem (2) with C_i and σ_i, calculate $tr(S_w)$ by (6).
Step5:	Update parameters C_{i+1}, σ_{i+1} in terms of the gradients of f, and $C_{i+1} = C_i + \eta \frac{\partial f}{\partial C}$, $\sigma_{i+1} = \sigma_i + \eta \frac{\partial f}{\partial \sigma}$.
Step6:	$i \leftarrow i+1$
Step7:	Until Convergence

Therefore, after we obtain the optimal C^*, σ^* by Algorithm 1, we can calculate the L_2-SVM decision function as:

$$f(x) = \mathrm{sgn}\left(\sum_{i}^{l} \alpha_i^* y_i^* \tilde{K}(x, x_i^*) + b_0 \right) \tag{12}$$

And the bias term b_0 can be computed as follows:

$$b_0 = \frac{1}{L} \sum_{j=1}^{L} \left(y_j^* - \sum_{i=1}^{L} \alpha_i^* y_i^* \tilde{K}(x_j^*, x_i^*) \right) \tag{13}$$

Where α_i^* the support is vector and L is the number of support vectors.

3.3 Discussion.

It is shown in [11] that $tr(S_w)/n$ is a lower bound of R^2. As a result, our trace-margin criterion, i.e., $tr(S_w)\|w\|^2$, may not necessarily be an upper bound of LOO error like the radius-margin bound, i.e., $R^2\|w\|^2$. However, it is observed that the proposed criterion often provides more benefits in practice.

(1) Compared to the radius-margin formulation, the proposed trace-margin formulation can significantly shorten the kernel learning time by avoiding solving the QP problem required to compute the radius at each iteration. For the given parameters C and σ, each evaluation of the radius-margin bound needs to solve two quadratic optimization problems, which can considerably prolong the feature selection process. Comparatively, each evaluation of the proposed criterion has much less computational load, since it does not involve any optimization. It can significantly reduce the time cost, leading to faster obtain optimal C and σ.

(2) In the definition of $tr(S_w)$, the class relationships of the data points are taken into account when measure within-class scattering matrix, which reflects the global properties of the class distributions, it is available to estimate the generalization error of SVM and improve classfication accuracy.

(3) $tr(S_w)$ is less sensitive to an outlier that significantly deviates from the center of data cloud. The estimation of R^2 is prone to being affected by noisy samples. Comparatively, the proposed criterion is less sensitive to the scarcity of training samples and the presence of data noise, because it evaluates the average case of each class by computing within-class scattering matrix, reflects the global properties of the class distributions. So the proposed criterion may correlate well with the generalization performance.

4. Conclusions

In this paper, we propose a method that chooses parameters for 2-norm Support Vector Machines based on the new criterion. Different from L_2-SVM method based on the RM, the new bound approximates R by the within-class scattering matrix, it is a better approximation that can significantly short the computation time, improve classfication accuracy and enhance the robustness of classfication algorithm.

Acknowledgments

This work is supported by the graduate innovation fund of Xihua University (Grant No.ycjj2014032).

References

[1] B. Scholkopf, A. Smola, Learning With Kernels, MIT Press, Cambridge, MA, 2002.

[2] K.-R. Müller, S. Mika, G. Rätsch, and S. Tsuda, An Introduction to Kernel-Based Learning Algorithms, IEEE Transactions on Neural Networks, 2001, 12 (2): 181-202.

[3] G. Camps-Valls and L. Bruzzone, Kernel-based methods for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 6, pp. 1351–1362, June 2005.
[4] V. Vapnik, The nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[5] S. S. Keerthi, Efficient tuning of SVM hyper parameters using radius/margin bound and iterative algorithms, IEEE Trans. Neural Network, 13 (5) (2002), pp. 1225–1229.
[6] K. M. Chung, W. C. Kao, C. L. Sun, L. L. Wang, C. J. Lin, Radius margin bounds for support vector machines with the RBF kernel, Neural Compute., 15 (2003), pp. 2643–2681.
[7] M. M. Adankon, M. Cheriet, Optimizing resources in model selection for support vector machine, Patt. Recogn, 40 (3) (2007), pp. 953-963.
[8] N. E. Ayat, M. Cheriet, C. Y. Suen, Automatic model selection for the optimization of SVM kernels, Patt. Recogn, 38 (10) (2005), pp. 1733-1745.
[9] O. Chapelle, V. Vapnik, Choosing multiple parameters for support vector machines, Machine Learning, 46 (1–3) (2002), pp. 131–159.
[10] T. Glasmachers, C. Igel, Gradient-based adaptation of general Gaussian kernels, Neural Computation, 2005, 17(10), pp. 2099–2105.
[11] L. Wang, Feature selection with kernel class reparability, IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 9, pp. 1534–1546, Sep. 2008
[12] Duan, K., Keerthi, S. S., Poo, A. N., Evaluation of simple performance measures for tuning SVM hyper parameters, Neurocomputing, 2003, 51, 41-59.