Multiple Parameter Selection for LS-SVM Using Smooth Leave-One-Out Error

Liefeng Bo, Ling Wang, and Licheng Jiao

Institute of Intelligent Information Processing and National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an 710071, China
{blf0218,wliiip}@163.com

Abstract. In least squares support vector (LS-SVM), the key challenge lies in the selection of free parameters such as kernel parameters and tradeoff parameter. However, when a large number of free parameters are involved in LS-SVM, the commonly used grid search method for model selection is intractable. In this paper, SLOO-MPS is proposed for tuning multiple parameters for LS-SVM to overcome this problem. This method is based on optimizing the smooth leave-one-out error via a gradient descent algorithm and feasible to compute. Extensive empirical comparisons confirm the feasibility and validation of the SLOO-MPS.

1 Introduction

In classification learning, we are given a set of samples of input vector along with corresponding output, and the task is to find a deterministic function that best represents the relation between the input-output pairs. The presence of noise (including input noise and output noise) implies that the key challenge is to avoid over-fitting on the training samples.

A very successful approach for classification is Support Vector Machines (SVMs) [1-2] that attempt to minimize empirical risk while simultaneously maximize the margin between two classes. This is a highly effective mechanism for avoiding over-fitting, which leads to good generalization ability. At present, SVMs have been widely used in pattern recognition, regression estimation, probabilistic density estimation and time series prediction. In this paper, we focus on least squares support vector machine (LS-SVM) [3-4], where one uses equality constraints instead of inequality constraints and a least squares error term in order to obtain a linear set of equations in the dual space. This expression is close related to regularization networks.

In LS-SVM, the key challenge lies in the selection of free parameters, i.e. kernel parameters and tradeoff parameter. A popular approach to solve this problem is grid search [5] where free parameters are firstly discretized in an appropriate interval, and then model selection criterion is performed on every parameters vector. The computational complexity of this approach increases exponentially with the number of free parameters. As a result it becomes intractable when a large number of free parameters are involved.

Motivated from that leave-one-out error of LS-SVM can be expressed as closed form, we propose an algorithm, named SLOO-MPS for tuning multiple parameters for LS-SVM. SLOO-MPS is constructed by two steps, i.e. replacing step function in leave-one-out error with sigmoid function and optimizing the resulting smooth

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3496, pp. 851–856, 2005.

leave-one-out error via a gradient descent algorithm. Extensive empirical comparisons confirm the feasibility and validation of the SLOO-MPS.

2 Least Squares Support Vector Machine

In this section, we briefly introduce least squares support vector machine. For more details, the interested reader can refer to [6]. In the feature space, LS-SVM models take the form

$$y = \mathbf{w}^T \varphi(\mathbf{x}) \tag{1}$$

where the nonlinear mapping $\varphi(\mathbf{x})$ maps the input data into a higher dimensional feature space whose dimensionality can be infinite. Note that the bias is ignored in our formulation. In LS-SVM, the following optimization problem is formulated

$$\begin{aligned} &\min\left(\frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{2}e_i^2\right) \\ &s.t. \quad y_i = \mathbf{w}^T\varphi(\mathbf{x}_i) + e_i \quad i = 1, \cdots l. \end{aligned} \tag{2}$$

Wolfe dual of (2) is

$$\min\left(\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) + \frac{1}{2}\sum_{i=1}^{l}\frac{\alpha_i^2}{C} - \sum_{i=1}^{l}\alpha_i y_i\right). \tag{3}$$

For computational convenience, the form $\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ in (3) is often replaced with a so-called kernel function $\mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$. Then (3) is translated into (4)

$$\min\left(\frac{1}{2}\boldsymbol{\alpha}^T\left(\mathbf{K} + \frac{1}{C}\mathbf{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{y}\right). \tag{4}$$

where $\mathbf{I}$ denotes a unit matrix. According to KKT condition, the equality

$$\left(\mathbf{K} + \frac{1}{C}\mathbf{I}\right)\boldsymbol{\alpha} = \mathbf{y} \tag{5}$$

holds true.

Any kernel function that satisfies the Mercer's theorem can be expressed as the inner product of two vectors in some feature space and therefore can be used in LS-SVM.

3 Smooth Leave-One-Error for LS-SVM

Cross-validation is a method for estimating generalization error based on re-sampling. The resulting estimate of generalization error is often used for choosing free parameters. In k-fold cross-validation, the available data are divided into k subsets of (approximately) equal size. Models are trained k times, each time leaving out one of the subsets from training. The k-fold cross-validation estimate of generalization error is mean of the testing errors of k models on the removed subset. If k equals the samples size, it is called leave-one-out cross-validation that has been widely studied

due to its mathematical simplicity. Let f_k be the residual error for the k^{th} training samples during the k^{th} iteration of the leave-one-out cross validation procedure. Then the following theorem holds true.

Theorem 1. [7]: $f_k = \dfrac{\left(\mathbf{H}^{-1}\mathbf{y}\right)_k}{\mathbf{H}_{kk}^{-1}}$, where $\mathbf{H} = (\mathbf{K} + \dfrac{1}{C}\mathbf{I})$, $\mathbf{H}_{kk}^{-1}$ denotes the k^{th} diagonal element of $\mathbf{H}^{-1}$. Define

$$\mathbf{f} = \left(\mathbf{H}^{-1}\mathbf{y}\right) \odot \mathrm{D}\left(\mathbf{H}^{-1}\right), \tag{6}$$

where $\odot$ denotes elementwise division. According to Theorem 1, leave-one-out error of LS-SVM is given by

$$loo(\boldsymbol{\theta}) = \frac{1}{l}\sum_{k=1}^{l}\left(\frac{1 - y_k \operatorname{sgn}\left(y_k - f_k\right)}{2}\right), \tag{7}$$

where θ denotes free parameters of kernel function and $\operatorname{sgn}(x)$ is 1, if $x \geq 0$, otherwise $\operatorname{sgn}(x)$ is -1. There exists a step function $\operatorname{sgn}(\bullet)$ in leave-one-out error $loo(\boldsymbol{\theta})$; thereby, it is not differentiable. In order to use a gradient descent approach to minimize this estimate, we approximate the step function by a sigmoid function

$$\tanh(\gamma t) = \frac{\exp(\gamma t) - \exp(-\gamma t)}{\exp(\gamma t) + \exp(-\gamma t)}, \tag{8}$$

where we set $\gamma = 10$. Then smooth leave-one-out error can be expressed as

$$loo(\boldsymbol{\theta}) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1 - y_i \tanh\left(\gamma\left(y_i - f_i\right)\right)}{2}\right). \tag{9}$$

According to the chain rule, the derivative of $loo(\boldsymbol{\theta})$ is formulated as

$$\frac{\partial\left(loo(\boldsymbol{\theta})\right)}{\partial\theta_k} = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{\partial\left(loo(\boldsymbol{\theta})\right)}{\partial f_i}\frac{\partial f_i}{\partial\theta_k}\right). \tag{10}$$

Thus we need to calculate $\dfrac{\partial\left(loo(\boldsymbol{\theta})\right)}{\partial f_i}$ and $\dfrac{\partial f_i}{\partial\theta_k}$, respectively. Together with $\dfrac{\partial\left(\tanh(t)\right)}{\partial t} = \operatorname{sech}^2(t)$, we have

$$\frac{\partial\left(loo(\boldsymbol{\theta})\right)}{\partial f_i} = \frac{\gamma y_i \operatorname{sech}^2\left(\gamma\left(y_i - f_i\right)\right)}{2}. \tag{11}$$

In terms of (11), (10) is translated into

$$\frac{\partial\left(loo(\boldsymbol{\theta})\right)}{\partial\theta_k} = \frac{1}{l}\left(\frac{\gamma\mathbf{y} \otimes \operatorname{sech}^2\left(\gamma(\mathbf{y} - \mathbf{f})\right)}{2}\right)^T\left(\frac{\partial\mathbf{f}}{\partial\theta_k}\right), \tag{12}$$

where $\otimes$ denotes elementwise multiplication The major difficulty to calculate $\left(\frac{\partial \mathbf{f}}{\partial \theta_k}\right)$ lies in obtaining the derivative of $\mathbf{H}^{-1}$. A good solution is based on the equality: $\mathbf{H}^{-1}\mathbf{H} = \mathbf{I}$. Differentiating that with respect to θ_k, we have

$$\frac{\partial \mathbf{H}^{-1}}{\partial \theta_k} = -\mathbf{H}^{-1}\frac{\partial \mathbf{H}}{\partial \theta_k}\mathbf{H}^{-1}. \tag{13}$$

Thus $\frac{\partial \mathbf{f}}{\partial \theta_k}$ is given by

$$\begin{aligned}\frac{\partial \mathbf{f}}{\partial \theta_k} &= \left(\frac{\partial\left(\mathbf{H}^{-1}\mathbf{y}\right)}{\partial \theta_k}\right)\odot \mathrm{D}\left(\mathbf{H}^{-1}\right) - \left(\mathbf{H}^{-1}\mathbf{y}\right)\odot\left(\mathrm{D}\left(\mathbf{H}^{-1}\right)\right)\odot\left(\mathrm{D}\left(\mathbf{H}^{-1}\right)\right)\otimes\left(\frac{\partial\left(\mathrm{D}\left(\mathbf{H}^{-1}\right)\right)}{\partial \theta_k}\right) \\ &= -\left(\mathbf{H}^{-1}\frac{\partial \mathbf{H}}{\partial \theta_k}\mathbf{H}^{-1}\mathbf{y}\right)\odot \mathrm{D}\left(\mathbf{H}^{-1}\right) + \left(\mathbf{H}^{-1}\mathbf{y}\right)\odot\left(\mathrm{D}\left(\mathbf{H}^{-1}\right)\right)\odot\left(\mathrm{D}\left(\mathbf{H}^{-1}\right)\right)\otimes \mathrm{D}\left(\mathbf{H}^{-1}\frac{\partial \mathbf{H}}{\partial \theta_k}\mathbf{H}^{-1}\right)\end{aligned} \tag{14}$$

where $\mathrm{D}(\mathbf{A})$ denotes diagonal elements of matrix $\mathbf{A}$. Combining (12) and (14), we can compute the derivative of smooth leave-one-out error with respect to θ_k.

4 Empirical Study

In this section, we will employ SLOO-MPS to tune the weights of the linear mixture kernel

$$\mathbf{H} = \sum_{i=1}^{m}\theta_i^2\mathbf{K}_i + \mathbf{I}, \tag{15}$$

where the mixing weights are positive to assure the positive semidefiniteness of $\mathbf{H}$. Since the weights of mixing kernel can be adjusted, it is reasonable to fix C to 1. Conjugate gradient algorithm is used to optimize the smooth leave-one-out error (10).

Kernel matrices are constructed by Gaussian and Laplace kernel

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\beta\sum_{m=1}^{d}\left(\mathbf{x}_i^m - \mathbf{x}_j^m\right)^2\right), \tag{16}$$

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\beta\sum_{m=1}^{d}\left|\mathbf{x}_i^m - \mathbf{x}_j^m\right|\right). \tag{17}$$

Three kinds of mixing schemes are evaluated. The first scheme is to mix 5 Gaussian kernels with $\beta \in \{0.01, 0.1, 1, 10, 100\}$. The second scheme is to mix 5 Laplace kernels with $\beta \in \{0.01, 0.1, 1, 10, 100\}$. The third scheme is to mix 5 Gaussian kernels and 5 Laplace kernel with $\beta \in \{0.01, 0.1, 1, 10, 100\}$. Thus all the free parameters of LS-SVM can be selected by SLOO-MPS, hence our algorithm is very automatic.

In order to how well SLOO-MPS woks, we test it on the ten benchmark data sets from UCI Machine learning Repository [8]. These data sets have been extensively used in testing the performance of diversified kinds of learning algorithms. One-against-one method is used to extend LS-SVM to multi-class classifiers. Ten-fold cross validation errors on benchmark data sets are summary in Table 1.

Table 1. Ten-fold cross validation errors on benchmark data sets.

Problems	Size	Dim	Class	Gaussian	Laplace	Mix
Australian Credit	690	15	2	15.22	14.20	14.15
Breast Cancer	277	9	2	23.74	26.24	23.89
German	1000	20	2	23.50	23.50	23.50
Glass	214	9	6	29.87	21.00	21.23
Heart	270	13	2	17.41	14.07	14.41
Ionosphere	351	34	2	4.86	5.99	4.14
Liver disorders	345	6	2	29.89	25.82	25.97
Vehicle	846	18	4	17.38	20.45	17.50
Vowel	528	10	11	0.95	1.57	0.76
Wine	178	13	3	1.11	1.70	1.11
Mean	/	/	/	16.39	15.45	**14.67**

From Table 1, we can see that mix kernel is better than either of Gaussian or Laplace kernel. This suggests that Gaussian and Laplace kernels indeed provide complementary information for the classification decision and SLOO-MPS approach is able to find a combination that exploits this complementarity.

We also test SLOO-MPS on the Olivetti Research Lab (ORL) face data set in Cambridge (http://www.cam-orl.co.uk/facedatabase.html). The ORL data set contains 40 distinct subjects, with each containing 10 different images taken at different time, with the lighting varying slightly. The experiment is similar to that done by Yang [9]. The leave-one-out errors for different method are summarized in Table 2. We can see that our method obtain the best performance on this data set.

Table 2. Performance on benchmark data sets.

Method	Reduced Space	Misclassification Rate
Eigenface	40	2.50
Fisherface	39	1.50
ICA	80	6.25
SVM, d=4	N/A	3.00
LLE # neighbor=70	70	2.25
ISOMAP, $\varepsilon = 10$	30	1.75
Kernel Eigenface, d=2	40	2.50
Kernel Eigenface, d=3	40	2.00
Kernel Fisherface (P)	39	1.25
Kernel Fisherface (G)	39	1.25
SLOO-MPS(Mix)	N/A	**0.75**

5 Conclusion

SLOO-MPS is presented for tuning the multiple parameters for LS-SVM. Empirical comparisons show that SLOO-MPS works well for the various data sets.

References

1. Boser, B., Guyon, I. and Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers. In Haussler, D. Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (1992) 144-152
2. Cotes, C. and Vapnik, V.: Support Vector Networks. Machine Learning, **20** (1995) 273-279
3. Suykens, J.A.K. and Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters, **9** (1999) 293-300
4. Van Gestel, T., Suykens, J., Lanckriet, G., Lambrechts, A., De Moor, B. and Vandewalle, J.: Bayesian Framework for Least Squares Support Vector Machine Classifiers. Gaussian Processes and Kernel Fisher Discriminant Analysis. Neural Computation, **15** (2002) 1115-1148
5. Hsu, C.W. and Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks, **13** (2002) 415-425
6. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience Publication. New York (1998)
7. Sundararajan, S. and Keerthi, S.S.: Predictive Approaches for Choosing Hyper- parameters in Gaussian Processes. Neural Computation, **13** (2001) 1103-1118
8. Blake, C.L. and Merz, C.J.: UCI Repository of Machine Learning Databases (1998). Available at: http://www.ics.uci.edu/~mlearn/MLRepository.html
9. Yang, M.H. Kernel: Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Method. Proceedings of the Fifth International Conference on Automatic Face and Gesture Recognition (2002)