Support Vector Regression Using Mahalanobis Kernels

Yuya Kamada and Shigeo Abe

Graduate School of Science and Technology
Kobe University
Rokkodai, Nada, Kobe, Japan
abe@eedept.kobe-u.ac.jp
http://www2.eedept.kobe-u.ac.jp/~abe

Abstract. In our previous work we have shown that Mahalanobis kernels are useful for support vector classifiers both from generalization ability and model selection speed. In this paper we propose using Mahalanobis kernels for function approximation. We determine the covariance matrix for the Mahalanobis kernel using all the training data. Model selection is done by line search. Namely, first the margin parameter and the error threshold are optimized and then the kernel parameter is optimized. According to the computer experiments for four benchmark problems, estimation performance of a Mahalanobis kernel with a diagonal covariance matrix optimized by line search is comparable to or better than that of an RBF kernel optimized by grid search.

1 Introduction

Support vector regressors (SVRs) have been used for various applications as a powerful function approximation tool. One of the problems of SVRs is that model selection, in which the values of the margin parameter, the error threshold, and the kernel parameter are optimized, is time consuming. There are several approaches to ease model selection but the most reliable method is cross-validation [1].

In most cases radial basis function network (RBF) kernels are used for SVRs. But Mahalanobis kernels [2,3], which are an extension of RBF kernels, and which exploit the data distribution information more than RBF kernels do, are used to ease model selection for pattern classification problems [4].

In this paper, based on [4] we propose model selection of SVRs using Mahalanobis kernels. Namely, using all the training data, we calculate the covariance matrix for the Mahalanobis kernel. We then optimize the margin parameter, the error threshold, and the kernel parameter that scales the Mahalanobis distance by line search: after optimizing the margin parameter and the error threshold by cross-validation, we optimize the kernel parameter by cross-validation. We show the usefulness of Mahalanobis kernels over RBF kernels using benchmark data sets.

F. Schwenker and S. Marinai (Eds.): ANNPR 2006, LNAI 4087, pp. 144–152, 2006.

In Section 2, we summarizes the SVRs, and in Section 3, we discuss Mahalanobis kernels. Then, in Section 4 we discuss model selection using Mahalanobis kernels. Finally in Section 5, we compare performance of Mahalanobis kernels with RBF kernels using some benchmark data sets.

2 Support Vector Regressors

In this section we briefly summarize the architecture of support vector regressors.

Let the M input-output pairs be $(\mathbf{x}_i, y_i)\,(i = 1, \ldots, M)$ and the mapping function be $\mathbf{g}(\mathbf{x})$, in which the input vector $\mathbf{x}$ is mapped into the l-dimensional feature space. Then the approximation function $f(\mathbf{x})$ is given by

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{g}(\mathbf{x}) + b, \tag{1}$$

where $\mathbf{w}$ is the l-dimensional vector and b is the bias term.

For the loss function:

$$E(y, f(\mathbf{x})) = \begin{cases} 0 & \text{for } |y - f(\mathbf{x})| \le \varepsilon, \\ |y - f(\mathbf{x})| - \varepsilon & \text{otherwise,} \end{cases} \tag{2}$$

where ε is a user-defined error threshold, the dual problem of the SVR is given by

$$\begin{aligned} \text{maximize} \quad & -\frac{1}{2} \sum_{i,j=1}^{M} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) H(\mathbf{x}_i, \mathbf{x}_j) \\ & -\varepsilon \sum_{i=1}^{M} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{M} y_i (\alpha_i - \alpha_i^*) \qquad (3) \\ \text{subject to} \quad & \sum_{i=1}^{M} (\alpha_i - \alpha_i^*) = 0, \qquad (4) \\ & 0 \le \alpha_i,\ \alpha_i^* \le C, \qquad (5) \end{aligned}$$

where $H(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{g}^T(\mathbf{x})\,\mathbf{g}(\mathbf{x})$ is a kernel, and α_i and α_i^* are Lagrange multipliers associated with $\mathbf{x}_i$.

The obtained approximation function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{M} (\alpha_i - \alpha_i^*) H(\mathbf{x}_i, \mathbf{x}) + b. \tag{6}$$

3 Mahalanobis Kernels

In function approximation we consider that all the training data belong to one cluster. For the cluster we define the Mahalanobis distance between a datum $\mathbf{x}$ and the center vector of the cluster:

$$d(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{c})^T Q^{-1} (\mathbf{x} - \mathbf{c})}, \tag{7}$$

where the center vector and the covariance matrix of the data are given, respectively, by

$$\mathbf{c} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i, \tag{8}$$

$$Q = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \mathbf{c}) (\mathbf{x}_i - \mathbf{c})^T. \tag{9}$$

The Mahalanobis distance is linear translation invariant [5]. Thus we need not worry about the scales of input variables.

Another interesting characteristic is that the average of the square of Mahalanobis distances is m [5]:

$$\frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \mathbf{c})^T Q^{-1} (\mathbf{x}_i - \mathbf{c}) = m. \tag{10}$$

Then, we define the Mahalanobis kernel by

$$H(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\delta}{m} (\mathbf{x} - \mathbf{x}')^T Q^{-1} (\mathbf{x} - \mathbf{x}') \right), \tag{11}$$

where $\delta\,(> 0)$ is the scaling factor to control the Mahalanobis distance. Here, the Mahalanobis distance is calculated between $\mathbf{x}$ and $\mathbf{x}'$, not between $\mathbf{x}$ and $\mathbf{c}$. The Mahalanobis kernel is an extension of the RBF kernel. Namely, by replacing $\delta Q^{-1}/m$ by $\gamma\, I$, where $\gamma(> 0)$ is a parameter for slope control and I is the $m \times m$ unit matrix, we obtain the RBF kernel:

$$\exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2). \tag{12}$$

From (10), by dividing the square of the Mahalanobis distance by m, it is normalized to 1 irrespective of the number of input variables. Although (11) is an approximation of the Mahalanobis kernel, this may enable to limit the search of the optimal δ value in a small range.

If we use the full covariance matrix, it will be time-consuming for a large number of input variables. Thus we consider two cases: Mahalanobis kernels with diagonal covariance matrices and Mahalanobis kernels with full covariance matrices. Hereafter we call the former diagonal Mahalanobis kernels and the latter non-diagonal Mahalanobis kernels.

4 Model Selection

Model selection is to optimize kernels and parameters to obtain the high generalization ability of SVRs. In this section, we discuss model selection for RBF kernels and Mahalanobis kernels by cross-validation.

4.1 RBF Kernels

For RBF kernels, we need to determine the values of ε, γ, and C by grid search. To set the proper search range of γ, it is better to normalize the input ranges into $[0, 1]$. Thus, because the maximum value of $\|\mathbf{x} - \mathbf{x}'\|^2$ is m, we use the following RBF kernels instead of (12) [6]:

$$\exp\left(-\frac{\gamma}{m}\|\mathbf{x} - \mathbf{x}'\|^2\right). \tag{13}$$

Because RBF kernels are not scale invariant, rescaling of the range into $[0, 1]$ is not always optimal.

4.2 Mahalanobis Kernels

For Mahalanobis kernels, we need to determine the values of ε, δ, and C. But because Mahalanobis kernels given by (11) are determined according to the data distribution and normalized by m, the initial value of $\delta = 1$ is a good selection. Thus, we can carry out model selection by line search not by grid search. Namely, the model selection is done as follows:

1. Set $\delta = 1$ and determine the values of C and ε by cross-validation for the values of C and ε on grid points. We call this the first stage.
2. Setting the values of C and ε as those determined by the first stage, determine the value of δ by cross-validation. We call this the second stage.

Because $\delta = 1$ is a good initial value, we may search the optimal value around 1. In addition, because Mahalanobis kernels are normalized by the covariance matrix, it is scale invariant. Therefore, unlike RBF kernels, the scale transformation of input variables does not affect the approximation error of SVRs.

5 Performance Evaluation

In this section, we evaluate the proposed model selection method. For this purpose, we performed model selection using Mahalanobis kernels and RBF kernels by grid search and line search and investigated whether the Mahalanobis kernel by line search performs well both from the approximation ability and model selection speed.

5.1 Evaluation Conditions

We used the benchmark data sets listed in Table 1. The water purification data set [7,8] is to estimate coagulant to be added to purify water. The Mackey-Glass data set [8,9] is a time series data set with chaotic behaviors. The Boston 5 and 14 data sets are from the Boston data set [10,11]. The Boston 5 data set predicts the nitrous oxide level, which is the 5th variable in the Boston data and Boston

Table 1. Parameter setting

Data	Inputs	Train.	Test
Water Purification	10	241	237
Mackey-Glass	4	500	500
Boston 5	13	506	—
Boston 14	13	506	—

14 data set predicts the median value of a home price, which is the 14th variable in the Boston data set. Since the Boston data set is not divided into training and test data sets, we randomly divided the set into two with almost equal sizes.

We ran c programs on a Xeon 2.8G personal computer with Linux operating systems. We trained the SVRs using the primal-dual interior-point method without using the decomposition technique. We used 5-fold cross-validation both for grid search and line search in determining the kernel parameter γ or δ, ε, and C.

5.2 Water Purification Data

We performed 5-fold cross-validation changing $C = \{1, 5, 10, 50, 100, 500, 1000, 3000, 5000, 10000, 50000, 100000\}$ and $\varepsilon = \{0.001, 0.005, 0.01, 0.05, 0.1\}$ for both kernels, $\gamma = \{0.1, 0.5, 1.0, 5.0, 10, 15\}$ for RBF kernels, and $\delta = \{0.1, 0.2, \ldots, 1.0, \ldots, 1.9, 2.0\}$ for Mahalanobis kernels.

Table 2 shows the results. "G" and "L" denote that the grid search and line search are performed for model selection and "Diag" and "Non-Diag" denote that the diagonal and non-diagonal covariance matrices are used for Mahalanobis kernels, respectively. The "Optimal" columns list the parameter values selected by model selection. The "Time" column lists the time for model selection by cross-validation. Approximation errors were evaluated by the average error and the maximum approximation error.

From the table, by grid search, the model selection time using Mahalanobis kernels is about three times longer than that using RBF kernels. But by line search they are almost comparable. The average estimation errors for the test data using a kernel by line search are worse than those using the same kernel by grid search but the maximum errors are smaller. Although the results are different for different kernels and different model selection methods, the difference is small.

5.3 Mackey-Glass Data

We performed 5-fold cross-validation changing C=$\{1, 10, 100, 500, 1000, 3000, 5000, 8000, 10000, 50000, 100000\}$ and $\varepsilon = \{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 0.001, 0.01\}$ for both kernels, $\gamma = \{0.1, 0.5, 1.0, 5.0, 10, 15\}$ for RBF kernels, and $\delta = \{0.1, 0.2, \ldots, 1.0, \ldots, 1.9, 2.0\}$ for Mahalanobis kernels.

Table 2. Performance comparison for water purification data

	Optimal			Time	Train. Error		Test Error	
Kernel	C	ε	γ/δ	[s]	Ave	Max	Ave	Max
RBF (G)	5	0.05	10.	4384	0.779	16.1	0.892	6.22
RBF (L)	100	0.001	1.0	827	0.852	17.9	0.954	6.04
Diag (G)	5	0.05	0.4	13050	0.806	16.4	0.919	6.27
Diag (L)	1	0.05	1.1	808	0.844	15.4	0.936	5.88
Non-Diag (G)	5	0.1	0.4	12611	0.706	14.6	0.942	5.57
Non-Diag (L)	1	0.001	0.8	770	0.817	15.1	0.965	4.40

Table 3. Performance comparison for Mackey-Glass data

	Optimal			Time	NRMSE	
Kernel	C	ε	γ/δ	[s]	Train.	Test
RBF (G)	10^5	10^{-5}	15.0	105173	0.00172	0.00215
RBF (L)	10^5	10^{-4}	15.0	14796	0.00191	0.00213
Diag (G)	10^5	10^{-7}	2.0	176293	0.00027	0.00272
Diag (L)	10^5	10^{-5}	2.0	13166	0.00025	0.00280
Non-Diag (G)	500	10^{-4}	1.2	169645	0.00284	0.00231
Non-Diag (L)	500	10^{-7}	0.9	11367	0.00390	0.00313

We evaluated the estimation performance of the Mackey-Glass data set by the Normalized Root Mean Square Error (NRMSE), i.e. the root-mean-square error divided by the standard deviation of the time series data.

Table 3 shows the results for the Mackey-Glass data set. The optimal values of ε are very small because the data set does not include noise. Both for RBF and diagonal Mahalanobis kernels, grid search and line search do not give much difference in estimation error but the estimation errors for the diagonal Mahalanobis kernels are a little worse. Non-diagonal Mahalanobis kernels by line search show worst estimation error.

Model selection by grid search for Mahalanobis kernels is slower than that for RBF kernels, but model selection time by line search is comparable for three kernels.

5.4 Boston Data

Since the Boston data set is not divided into training and test data sets, we randomly divided the set into 20 training and test data sets with almost equal

sizes. And we determined the optimal parameter values by 5-fold cross-validation using training data sets and evaluated the average errors and their standard deviation for the training and test data sets. In cross-validation we changed $C = \{1, 10, 100, 1000, 5000, 10000, 50000, 100000\}$ and $\varepsilon = \{0.001, 0.01, 0.1\}$ for both kernels, and $\gamma = \{0.1, 0.5, 1.0, 5.0, 10, 15\}$ for RBF kernels, and $\delta = \{0.1, 0.2, \ldots, 1.0, \ldots, 1.9, 2.0\}$ for Mahalanobis kernels.

Table 4 lists the results for the Boston 5 data set. The optimal values of the parameters show the most frequently selected values among 20 trials. "Time" column lists the average time for model selection for 20 trials. We evaluated the estimation performance by average errors and their standard deviations. The boldface numbers show the best performance group, in which there are no statistical difference in both averages and variances among the members of the group. And the italic numerals show the second performance group, in which the averages are statistically different from the best group although there is no statistical difference in the variances. This means that estimation performance of RBF kernels by line search is inferior to that of RBF kernels by grid search. But estimation performance of Mahalanobis kernels shows the best irrespective of a diagonal or non-diagonal covariance matrix or line search or grid search. In addition since model selection is speeded up by line search, the line search strategy is suitable for Mahalanobis kernels for this data set.

Table 4. Performance comparison for Boston 5 data

	Optimal			Time	Error & Stand. Dev.	
Kernel	C	ε	γ/δ	[s]	Train.	Test
RBF (G)	10^5	0.001	5.0	3473	0.0273 ± 0.0060	*0.0371 ± 0.0024*
RBF (L)	10^5	0.001	1.0	609	0.0369 ± 0.0229	0.0469 ± 0.0219
Diag (G)	1	0.001	0.9	6330	0.0154 ± 0.0052	**0.0287 ± 0.0022**
Diag (L)	1	0.001	0.8	504	0.0130 ± 0.0036	**0.0280 ± 0.0021**
Non-Diag (G)	1	0.001	0.4	7035	0.0123 ± 0.0037	**0.0287 ± 0.0021**
Non-Diag (L)	1	0.001	0.5	586	0.0119 ± 0.0034	**0.0286 ± 0.0020**

Table 5 shows the results for the Boston 14 data set. Since the optimal values show the most frequently selected values, although they are the same for the RBF kernels with grid search and line search, the average errors and the standard deviations are different. The boldface numerals show the best estimation performance group and the italic numerals show the second best group. Therefore, for this data set, estimation performance of Mahalanobis kernels is statistically better than that of RBF kernels. And the diagonal Mahalanobis kernel by line search is a good choice from the standpoint of estimation performance and model selection speed.

Table 5. Performance comparison for Boston 14 data

	Optimal			Time	Error & Stand. Dev.	
Kernel	C	ε	γ/δ	[s]	Train.	Test
RBF (G)	10^5	0.1	15.	2870	2.20 ± 0.173	2.84 ± 0.206
RBF (L)	10^5	0.1	15.	639	2.19 ± 0.151	2.83 ± 0.196
Diag (G)	100	0.1	0.3	7020	1.46 ± 0.295	$\mathbf{2.40 \pm 0.153}$
Diag (L)	10	0.1	0.7	639	1.57 ± 0.362	$\mathbf{2.48 \pm 0.187}$
Non-Diag (G)	100	0.1	0.2	8061	1.20 ± 0.250	$\mathbf{2.50 \pm 0.151}$
Non-Diag (L)	10	0.1	0.6	842	1.48 ± 0.326	*2.63 ± 0.187*

5.5 Discussions

In cross-validation, we used 6 parameter values for RBF kernels and 20 for Mahalanobis kernels, which is more than three times larger. But according to the experiments, for the Mahalanobis kernels model selection by line search was 10 to 16 times faster. In addition, model selection for Mahalanobis kernels by line search was three to five times faster than that for RBF kernels by grid search, although model selection for Mahalanobis kernels by grid search was slower than that for RBF kernels by grid search.

For the 4 benchmark data sets, diagonal Mahalanobis kernels by line search showed stable estimation performance and especially for Boston data sets Mahalanobis kernels by line search belonged to the best estimation group in a statistical sense. But RBF kernels and non-diagonal Mahalanobis kernels by line search showed inferior estimation performance in some cases.

Therefore, from estimation performance and model selection speed, the Mahalanobis kernels by line search can be alternative kernels for RBF kernels. In addition, the diagonal Mahalanobis kernels are enough for this purpose.

6 Conclusions

We discussed model selection using the Mahalanobis kernels for function approximation. We calculate the covariance matrix using the training data and determine the optimum values of the margin parameter, the error threshold, and the kernel parameter by line search. Namely, first we determine the margin parameter and the error threshold by grid search fixing the value of the kernel parameter, and then we determine the value of the kernel parameter. The computer experiments showed that the performance of the Mahalanobis kernels by line search was comparable to, or better than that of RBF kernels by grid search.

References

1. K. Duan, S. S. Keerthi, and A. N. Poo. An Empirical Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters. *Proc. ICONIP-2001*, Paper ID# 159, 2001.
2. R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, MA, 2002.
3. F. Friedrichs and C. Igel. Evolutionary Tuning of Multiple SVM Parameters. *Proc. ESANN 2004*, pp. 519–524, 2004.
4. S. Abe. Training of Support Vector Machines with Mahalanobis Kernels. *Proc. ICANN 2005*, pp. 571-576, 2005.
5. S. Abe. *Pattern Classification: Neuro-Fuzzy Methods and Their Comparison*. Springer-Verlag, London, 2001.
6. S. Abe. *Support Vector Machines for Pattern Classification*. Springer-Verlag, London, 2005.
7. K. Baba, I. Enbutu, and M. Yoda. Explicit Representation of Knowledge Acquired from Plant Historical Data Using Neural Network. *Proc. IJCNN 1990*, Vol. 3, pp. 155–160, 1990.
8. S. Abe. *Neural Networks and Fuzzy Systems: Theory and Applications*. Kluwer, Boston, MA, 1997.
9. R. S. Crowder. Predicting the Mackey-Glass Time Series with Cascade-Correlation Learning. *Proc. 1990 Connectionist Models Summer School*, pp. 117–123, Carnegie Mellon University, 1990.
10. D. Harrison and D. L. Rubinfeld. Hedonic Prices and the Demand for Clean Air. *Journal of Environmental Economics and Management*, Vol. 5, pp. 81-102, 1978.
11. http://www.cs.toronto.edu/~delve/data/datasets.html