Training Mahalanobis Kernels by Linear Programming

Abe, Shigeo

Training Mahalanobis Kernels by Linear Programming

Shigeo Abe

Kobe University
Rokkodai, Nada, Kobe, Japan
abe@kobe-u.ac.jp
http://www2.kobe-u.ac.jp/~abe

Abstract. The covariance matrix in the Mahalanobis distance can be trained by semi-definite programming, but training for a large size data set is inefficient. In this paper, we constrain the covariance matrix to be diagonal and train Mahalanobis kernels by linear programming (LP). Training can be formulated by ν-LP SVMs (support vector machines) or regular LP SVMs. We clarify the dependence of the solutions on the margin parameter. If a problem is not separable, a zero-margin solution, which does not appear in the LP SVM, appears in the ν-LP SVM. Therefore, we use the LP SVM for kernel training. Using the benchmark data sets we show that the proposed method gives better generalization ability than RBF (radial basis function) kernels and Mahalanobis kernels calculated using the training data and has a good capability of selecting input variables especially for a large number of input variables.

1 Introduction

Regular support vector machines do not assume a priori data distributions and determine the decision boundary using only support vectors that are near the boundary. However, if data of one class have a large variance and those of the other class have a small variance, it may not be good to place the hyperplane in the middle of the unbounded support vectors. In such a situation, instead of the Euclidean distance, the Mahalanobis distance is sometimes effective [1–5].

There are two ways to incorporate the Mahalanobis distance into support vector machines: one is to reformulate support vector machines so that the margin is measured by the Mahalanobis distance [1, 2], and the other is to use Mahalanobis kernels [3–5], which calculate the kernel value according to the Mahalanobis distance between the associated two argument vectors.

Radial basis function (RBF) kernels are widely used because they usually give good performance for most applications. To improve the generalization ability of RBF kernels, generalized RBF kernels are proposed, in which each input variable has a weight in calculating the kernel value. Mahalanobis kernels are an extension of generalized RBF kernels and if the covariance matrix is restricted to a diagonal matrix, the Mahalanobis kernels reduce to generalized RBF kernels [4].

In [3], training of a support vector machine is reformulated so that a generalized RBF kernel is trained simultaneously. The formulation, however, is no longer quadratic. In [4], the covariance matrix for Mahalanobis kernels is calculated using the training data.

There are several discussions to obtain Mahalanobis metric by training, which results in semi-definite programming [6]. Usually, semi-definite programming is difficult to solve for large size problems and its speedup has been discussed. However if we confine the covariance matrix to a diagonal matrix, training results in linear programming.

In this paper, we train Mahalanobis kernels with the diagonal covariance matrix by linear programming (LP). Restricting the covariance matrix to a diagonal matrix in the formulation of [6], we obtain a linear programming program with the explicit margin similar to a ν-LP SVM [7]. We analyze the dependence of the solution of the ν-LP SVM for the margin parameter and show that if a training problem is not linearly separable, the zero-margin solution is obtained for the margin parameter value larger than some value. We show that this does not happen for the LP SVM, which is equivalent to the ν-LP SVM with the positive margin and objective function values. We also derive the lower bound of the margin parameter value of the LP SVM, in which the nonzero solution is obtained. We, therefore, use the LP SVM for kernel training. We compare the proposed method with the RBF kernels and Mahalanobis kernels whose diagonal elements are calculated using training data.

In Section 2, we formulate kernel training by linear programming. In Section 3, we clarify the characteristics of ν-LP SVMs and LP SVMs. In section 4, we demonstrate the effectiveness of the proposed method using some benchmark data sets.

2 Formulation of Kernel Training

In [4], the Mahalanobis kernel for m-dimensional inputs $\mathbf{x}$ and $\mathbf{x}'$ is defined by

$$K(\mathbf{x}, \mathbf{x}') = \exp(-(\delta/m)\,(\mathbf{x} - \mathbf{x}')^\top Q^{-1}\,(\mathbf{x} - \mathbf{x}')), \tag{1}$$

where $\delta\,(> 0)$ is a parameter, $\top$ and -1 denote the matrix transpose and inverse, respectively, and Q is the covariance matrix with the M data $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$:

$$Q = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \mathbf{c})\,(\mathbf{x}_i - \mathbf{c})^\top, \quad \mathbf{c} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i. \tag{2}$$

According to the computer experiment [4], the diagonal covariance matrix is sufficient for high generalization ability. Therefore, in the following we consider only diagonal covariance matrix for (1).

In [6], the inverse of Q is trained using the training data. We reformulate the method discussed in [6] for the diagonal Q. We let $R = Q^{-1}$.

Let set P be a set of training triplets:

$$P = \{P_r\} = \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)\} \qquad \text{for} \quad r = 1, \ldots, |P|, i, j, k \in \{1, \ldots, M\}, \tag{3}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class but $\mathbf{x}_k$ and $\mathbf{x}_i$ (and thus $\mathbf{x}_j$) belong to different classes, and $|P|$ denotes the number of elements in P. In [6], the details of how to generate triplets are not described. Here, we consider generating the set of triplets, P. For a training sample $\mathbf{x}_r$ ($r = 1, \ldots, M$) belonging to a class, we find the nearest $\mathbf{x}_j$ belonging to the same class and the nearest $\mathbf{x}_k$ belonging to the other class and make $(\mathbf{x}_r, \mathbf{x}_j, \mathbf{x}_k)$ the triplet. In this way, we obtain P with M triplets.

A Mahalanobis distance between two data belonging to the same class needs to be shorter than that between data of different classes. Thus, we define a margin ρ_r for the rth triplet as follows:

$$\rho_r = (\mathbf{x}_r - \mathbf{x}_k)^\top R\,(\mathbf{x}_r - \mathbf{x}_k) - (\mathbf{x}_r - \mathbf{x}_j)^\top R\,(\mathbf{x}_i - \mathbf{x}_j) = \sum_{i=1}^{m} a_{ir}\,R_{ii}, \quad (4)$$

where $a_{ir} = (x_{ir} - x_{ik})^2 - (x_{ir} - x_{ij})^2 \quad \text{for} \quad i = 1, \ldots, m, \quad r = 1, \ldots, M$, x_{ir} is the ith element of $\mathbf{x}_r$, and R_{ii} is the ith diagonal element of R.

We want ρ_r as large as possible but similar to support vector machines, for some triplets we allow negative margins. Then, we formulate the following optimization problem:

$$\begin{aligned}
\text{maximize} \quad & J_\rho(\rho, R, \boldsymbol{\xi}) = \rho - C_\rho \sum_{r=1}^{M} \xi_r && (5)\\
\text{subject to} \quad & \sum_{i=1}^{m} R_{ii} = 1 && (6)\\
& R_{ii} \geq 0 \quad \text{for} \quad i = 1, \ldots, m, && (7)\\
& \sum_{i=1}^{m} a_{ir}\,R_{ii} \geq \rho - \xi_r \quad \text{for} \quad r = 1, \ldots, M, && (8)\\
& \xi_r \geq 0 \quad \text{for} \quad r = 1, \ldots, M, \quad \rho > 0, && (9)
\end{aligned}$$

where ρ is the margin, ξ_r are slack variables to allow negative margins, C_ρ is a margin parameter, and (6) is to make the left-hand side of (8) be unique.

The above formulation is similar to the ν-LP SVM discussed in [7]. Because the ν-LP SVM treats a two-class problem, $a_{ir}\,R_{ii}$ in (8) is multiplied by y_r which takes 1 or -1. But this is a trivial difference. Thus, we call the above support vector machine primal ν-LP SVM or ν-LP SVM for short.

According to [7], for positive ρ, the ν-LP SVM is equivalent to the following formulation:

$$\begin{aligned}
\text{minimize} \quad & J(R, \boldsymbol{\xi}) = \sum_{i=1}^{m} R_{ii} + C_{\mathrm{M}} \sum_{r=1}^{M} \xi_r && (10)\\
\text{subject to} \quad & R_{ii} \geq 0 \quad \text{for} \quad i = 1, \ldots, m, && (11)\\
& \sum_{i=1}^{m} a_{ir}\,R_{ii} \geq 1 - \xi_r \quad \text{for} \quad r = 1, \ldots, M, && (12)\\
& \xi_r \geq 0 \quad \text{for} \quad r = 1, \ldots, M, && (13)
\end{aligned}$$

where C_{M} is a margin parameter.

The above formulation is similar to standard LP SVMs with linear kernels. The differences are that the coefficients of the hyperplane, R_{ii}, are non-negative and no bias term is included. In the following, we call the above support vector machine regular LP SVM or LP SVM for short.

In the ν-LP SVM, the margin is $\rho/\sum_{i=1}^{m} R_{ii} = \rho$, whereas in the LP SVM, the margin is $1/\sum_{i=1}^{m} R_{ii}$. By negating (5) and replacing ρ with $-\sum_{i=1}^{m} R_{ii}$, the objective function (10) is obtained. Because the equality constraint (6) makes the optimization meaningless it is deleted, and ρ in (8) is replaced with 1 in the LP SVM.

Let the optimal solution of the ν-LP SVM be $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$, that of the LP SVM be $(\hat{R}, \hat{\boldsymbol{\xi}})$, $\bar{J}_\rho = J_\rho(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$, $\hat{J} = J(\hat{R}, \hat{\boldsymbol{\xi}})$, and $\bar{R} \neq 0$ denote that at least one diagonal element of R is nonzero. In [7], the equivalence of the ν-LP SVM and LP SVM is shown as follows:

Theorem 1. *If the ν-LP SVM has the optimal solution $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ with $\bar{\rho} > 0$ and $\bar{J}_\rho > 0$ for C_ρ, $(\hat{R}, \hat{\boldsymbol{\xi}}) = (\bar{R}/\bar{\rho}, \bar{\boldsymbol{\xi}}/\bar{\rho})$ is the optimal solution of the LP SVM with $C_{\mathrm{M}} = C_\rho/\bar{J}_\rho$. Conversely, if the LP SVM with C_{M} has the optimal solution $(\hat{R}, \hat{\boldsymbol{\xi}})$ with $\hat{R} \neq 0$, $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}}) = (1/\sum_{i=1}^{m} \hat{R}_{ii}, \bar{\rho}\,\hat{R}, \bar{\rho}\,\hat{\boldsymbol{\xi}})$ is the optimal solution of the ν-LP SVM with $C_\rho = C_{\mathrm{M}}\,\hat{J}$.*

We add the condition $\bar{J}_\rho > 0$ at the first part of Theorem 1 to guarantee equivalence.

In [7], it is recommended to use the ν-LP SVM rather than the LP SVM for the ease of parameter value selection for C_ρ: $C_\rho = \frac{1}{\nu M}$ for $1/M < \nu < 1$. Therefore, $1/M < C_\rho < 1$.

3 Properties of ν-LP SVMs and LP SVMs

In this section we clarify the properties of ν-LP SVMs and LP SVMs and then compare them. Because of the space limitation, we omit proofs of the theorems.

3.1 ν-LP SVMs

We investigate the dependence of the ν-LP SVM on the C_ρ value. To do so, we derive the dual form of the ν-LP SVM given by (5) to (9) as follows:

$$\text{minimize} \quad z \tag{14}$$

$$\text{subject to} \quad \sum_{r=1}^{M} \delta_r \geq 1 \tag{15}$$

$$\sum_{r=1}^{M} a_{ir}\,\delta_r \leq z \quad \text{for} \quad i = 1, \ldots, m, \tag{16}$$

$$C_\rho \geq \delta_r \geq 0 \quad \text{for} \quad r = 1, \ldots, M, \tag{17}$$

where z is the dual variable associated with the constraint (6) and δ_r are dual variables associated with (8). We call the above formulation dual ν-LP SVM. In [7], the equality is used for (15) but that is valid only for $\rho > 0$.

From (15) and (17), the dual ν-LP SVM has a feasible solution for $C_\rho \geq 1/M$, but does not have a feasible solution for $1/M > C_\rho > 0$. For the ν-LP SVM, because of the slack variables ξ_i, a feasible solution always exists for $C_\rho > 0$. But for $1/M > C_\rho > 0$, the optimal solution of the ν-LP SVM is unbounded:

Theorem 2. *For $0 < C_\rho < 1/M$, the solution of the ν-LP SVM is unbounded.*

But for $C_\rho \geq 1$, R_{ii} $(i = 1, \ldots, m)$ are the same:

Theorem 3. *For $C_\rho \geq 1$ the optimal R_{ii} $(i = 1, \ldots, m)$ are the same for different values of C_ρ.*

From Theorem 3, it is clear that for $1 \geq C_\rho \geq 1/M$, the bounded optimal solutions exist for the primal and dual ν-LP SVMs. From (15) and (17), for the optimal solution of the ν-LP SVM with $1/(j-1) > C_\rho \geq 1/j$ $(j = 2, \ldots, M)$, at least j inequality constraints in (8) are active. Theorem 3 is further refined as follows:

Theorem 4. *If for $C_\rho = C_0\,(\geq 1/M)$, the solution of the ν-LP SVM satisfies $\rho > 0$ and $\boldsymbol{\xi} = \mathbf{0}$, or $\rho = 0$, the same solution is obtained for $C_\rho = t\,C_0\,(t > 1)$.*

Assume that for $C_\rho = 1/j\,(j = 2, \ldots, 1/M)$ the solutions of the primal and dual ν-LP SVMs are $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ and $(\bar{z}, \bar{\boldsymbol{\delta}})$, respectively. Then what can we say about the solutions for $C_\rho = t/j\,(j/j - 1 > t > 1)$? From Theorem 4, if $\bar{\rho} > 0$ and $\bar{\boldsymbol{\xi}} = \mathbf{0}$, or $\bar{\rho} = 0$, R does not change for $C_\rho = t/j$. Then, for $\bar{\rho} > 0$ and $\bar{\boldsymbol{\xi}} \neq \mathbf{0}$, can $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ and $(\bar{z}, \bar{\boldsymbol{\delta}})$ or $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ and $(t\,\bar{z}, t\,\bar{\boldsymbol{\delta}})$ be the solutions for $C_\rho = t/j$? Neither can. The solutions $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ and $(\bar{z}, \bar{\boldsymbol{\delta}})$ do not satisfy the complementarity condition for $\bar{\xi}_r > 0$. And the solutions $(\bar{\rho}, \bar{R}, \bar{\boldsymbol{\xi}})$ and $(t\,\bar{z}, t\,\bar{\boldsymbol{\delta}})$ do not satisfy the complementarity condition. This means that the optimal R may change for $1/(j-1) > C_\rho > 1/j$.

3.2 LP SVMs

Now investigate the dependence of the solution of the LP SVM on the C_{M} value.

Because of the slack variables, the LP SVM has a feasible solution. In addition, because the objective function given by (10) is restricted to be non-negative, the optimal solution of the minimization problem always exists.

For a small C_{M} value, the solution $R_{ii} = 0$ $(i = 1, \ldots, m)$ is obtained as the following theorem shows:

Theorem 5. *Define*

$$C_{\min} = \min_{\substack{i=1,\ldots,m \\ \sum_{r=1}^{M} a_{ir} > 0}} 1/\sum_{r=1}^{M} a_{ir}. \tag{18}$$

Then, if $C_{\min}$ exists, for $C_{\min} > C_{\mathrm{M}} > 0$ the solution of the LP SVM is

$$R_{ii} = 0 \quad \textit{for} \quad i = 1, \ldots, m. \tag{19}$$

Otherwise, (19) is satisfied for $C_{\mathrm{M}} > 0$.

This is a degenerate solution but because tradeoff between minimization of the margin and the maximization of the classification accuracy is controlled by the margin parameter C_{M}, the degenerate solution is not an anomaly solution.

The relation $\sum_{r=1}^{M} a_{ir} \leq 0$ may hold when the nearest training pairs, measured by the ith input variable, with the opposite classes are dominant.

Now consider a general case where all the constraints (12) are not active for the optimal solution and let S be the index set for the active constraints. Then, (10) for the optimal solution becomes

$$\text{minimize} \quad J(R, \boldsymbol{\xi}) = |S|\, C_{\mathrm{M}} + \sum_{i=1}^{m} (1 - C_{\mathrm{M}} \sum_{r \in S} a_{ir}) R_{ii}. \tag{20}$$

The set of $\{\mathbf{x}_r \,|\, r \in S\}$ is the set of support vectors. Thus, for $1 - C_{\mathrm{M}} \sum_{r \in S} a_{ir} > 1$, $R_{ii} = 0$. This means that by training the LP SVM, input variable selection is simultaneously performed.

Similar to the ν-LP SVM, for a large value of C_{M}, the optimal solution does not change as the following theorem shows:

Theorem 6. *There exists a positive C_0 such that for $C_{\mathrm{M}} \geq C_0$ the optimal solution $(R, \boldsymbol{\xi})$ of the LP SVM does not change.*

3.3 Comparison of ν-LP SVMs and LP SVMs

As Theorem 1 shows, the ν-LP SVM and LP SVM are equivalent when ρ is positive and the objective function value for the ν-LP SVM is positive. To obtain a solution with positive ρ and the positive objective function, C_ρ needs to be selected in $[1/M, 1/j]$, where $j \in \{1, \ldots, M-1\}$. But, there is no way of selecting the value of j.

For the optimal solution with $C_\rho = 1/j$, at least j constraints are active. Therefore, controlling the number of active constraints is easy but again the optimal j is not known in advance.

For the LP SVM, the lower bound of C_{M} that gives nonzero R is $C_{\min}$ given by (18). But unlike the ν-LP SVM, there is no upper bound of C_{M}. This is because a zero-margin solution (i.e., $R_{ii} \to \infty$) is not obtained.

Using either the ν-LP SVM or LP SVM, we need to optimize the value of C_ρ or C_{M} by e.g., cross-validation. And because by the LP SVM, we do not worry about the upper bound of C_{M}, in the following we use the LP SVM for Mahalanobis kernel training.

4 Computer Experiment

We compared performance of the proposed method with that of Mahalanobis kernels given by (1) and (2), RBF kernels, and the Mahalanobis distance proposed in [6]. The RBF kernel is given by $\exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2/m)$, where γ is a positive constant. We used one-against-all fuzzy L1 SVMs [8].

We determined the parameter values by fivefold cross-validation. For the margin parameter C of the L1 SVM, we selected the value from $\{1, 10, 50, 100, 500, 1,000, 2,000\}$. For the proposed Mahalanobis kernel and the RBF kernel, we selected the value of δ and γ from $\{0.1, 0.5, 1, 5, 10, 20, 50, 100, 200\}$ and for the Mahalanobis kernel given by (1) and (2), we selected the value of δ from $\{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 20\}$.

For the proposed Mahalanobis kernel, to speed up model selection, we determined the value of C_{M} in two stages for single training and test data sets: First, we trained the LP SVM with $C_{\mathrm{M}} = 2,000$, and determined the values of C and δ for the SVM by cross-validation. Then, we determined the values of C_{M} and C by cross-validation, fixing δ value with the value selected by cross-validation. For multiple training and test data sets, we set $C_{\mathrm{M}} = 1$ or 2,000, and determined the C and δ values by cross-validation. And we selected the C_{M} value with the higher recognition rate for the cross-validation data set.

Table 1 shows the results. In the table, the In/Cl/Tr/Te column shows the numbers of inputs, classes, training data, and test data, the following four columns show the recognition rates for the test data sets: the proposed method; the Mahalanobis kernel calculated by (1) and (2); RBF kernels; and the Mahalanobis distance in [6]. The "Final" column shows the average number of inputs per decision function selected by the proposed method.

The results of the balance and Pima Indians data sets were obtained by randomly dividing files 10 times. Among the first three methods, the best recognition rate is shown in boldface, the second in Roman, and the third in italic. The last row of the table shows the summary: e.g., the first numeral in the three numerals shows the number that gives the best recognition rate. From the table, the proposed kernel shows the best and the Mahalanobis and RBF kernels are comparable. For the last three data sets, all the three methods showed better recognition rates than the Mahalanobis distance by [6] did.

In the "Final" column, the reduction rates for the hiragana-50, hiragana-105, and USPS data sets are high because they are gray scale images.

5 Conclusions

We discussed training Mahalanobis kernels with a diagonal covariance matrix by linear programming support vector machines (LP SVMs). Training can be formulated either by ν-LP SVMs or regular LP SVMs. We clarified the dependency of ν-LP SVMs and regular LP SVMs on the margin parameter and proved that the ν-LP SVM may give a zero-margin solution for the margin parameter value close to 1. But this kind of solution does not appear in the regular LP SVM.

Table 1. Comparison of recognition rates and the number of selected inputs

Data	In/Cl/Tr/Te	Proposed	Mahalanobis	RBF	[6]	Final
Thyroid [9]	21/3/3772/3428	**97.93**	97.49	*96.47*	—	17.7
Blood [8]	13/12/3097/3100	**93.77**	*93.32*	93.52	—	12.3
Hiragana-50 [8]	50/39/4610/4610	*98.85*	**99.31**	99.26	—	24.0
Hiragana-13 [8]	13/38/8375/8356	**99.86**	*99.80*	*99.77*	—	10.4
Hiragana-105 [8]	105/38/8375/8356	**100**	**100**	**100**	—	39.2
Abalone [9]	8/3/3133/1044	66.38	**67.63**	*65.04*	—	7.0
Satimage [9]	36/6/4435/2000	91.20	*91.00*	**91.90**	—	31.8
USPS [10]	256/10/7291/2007	95.32	*94.77*	**95.47**	—	116
Letter [9]	16/26/16000/4000	**98.23**	*97.58*	97.83	96.54	15.8
Balance [9]	4/3/532/93	**99.03**±1.12	97.96±2.01	**99.03**±1.22	90.22±3.17	4
Pima Indians [9]	8/2/653/115	*74.96*±3.92	**75.31**±3.98	75.13±4.14	72.36±3.71	8
Ranking	—	6:3:2	4:2:5	4:4:3	—	

We also derive the lower bound of the margin parameter for the LP SVM that gives the nonzero solution. Because of the zero-margin problem, we used regular LP SVMs. Using several benchmark data sets we showed that the generalization ability of the proposed method is better than that of RBF kernels.

References

1. Lanckriet, G. R. G., Ghaoui, L. El., Bhattacharyya, C., Jordan, M. I.: A robust minimax approach to classification, Journal of Machine Learning Research, 3:555–582 (2002)
2. Xue, H., Chen, S., Yang, Q.: Structural regularized support vector machine: A framework for structural large margin classifier, IEEE Trans. Neural Networks, 22(4):573–587 (2011)
3. Grandvalet, Y., Canu, S.: Adaptive scaling for feature selection in SVMs, Advances in Neural Information Processing Systems 15, 569–576. MIT Press (2003)
4. Abe, S.: Training of support vector machines with Mahalanobis kernels, Proc. ICANN 2005, 571–576 (2005)
5. Wang, D., Yeung, D. S., Tsang, E. C. C.: Weighted Mahalanobis distance kernels for support vector machines, IEEE Trans. Neural Networks, 18(5):1453-1462 (2007)
6. Shen, C., Kim, J., Wang, L.: Scalable large-margin Mahalanobis distance metric learning, IEEE Trans. Neural Networks, 21(9):1524–1530 (2010)
7. Demiriz, A., Bennett, K. P., Shawe-Taylor, J., Linear programming boosting via column generation, Machine Learning, 46(1-3):225–254 (2002)
8. Abe, S.: Support Vector Machines for Pattern Classification, Springer, Heidelberg (2010)
9. Asuncion, A., Newman, D. J.: UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, (2007)
10. USPS Dataset, http://www-i6.informatik.rwth-aachen.de/˜keysers/usps.html.