Evaluation of Feature Selection by Multiclass Kernel Discriminant Analysis

Tsuneyoshi Ishii and Shigeo Abe

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

Abstract. In this paper, we propose and evaluate the feature selection criterion based on kernel discriminant analysis (KDA) for multiclass problems, which finds the number of classes minus one eigenvectors. The selection criterion is the sum of the objective function of KDA, namely the sum of eigenvalues associated with the eigenvectors. In addition to the KDA criterion, we propose a new selection criterion that replaces the between-class scatter in KDA with the sum of square distances between all pairs of classes. To speed up backward feature selection, we introduce block deletion, which deletes many features at a timeC and to enhance generalization ability of the selected features we use cross-validation as a stopping condition.

By computer experiments using benchmark datasets, we show that the KDA criterion has performance comparable with that of the selection criterion based on the SVM-based recognition rate with cross-validation and can reduce computational cost. We also show that the KDA criterion can terminate feature selection stably using cross-validation as a stopping condition.

1 Introduction

Feature selection is to select from the original set of features the minimum subset of features that realizes the maximum generalization ability. To realize this, during the process of feature selection, the generalization ability of a subset of features needs to be estimated. This type of feature selection is called a wrapper method. Instead of estimating the generalization ability, some selection criterion, which is considered to well reflect the generalization ability, is used. This method is called a filter method.

The forward or backward selection method using a selection criterion is widely used. In backward selection, we start from all the features and delete one feature at a time, which deteriorates the selection criterion the least. We delete features until the selection criterion reaches a specified value. In forward selection, we start from an empty set of features and add one feature at a time, which improves the selection criterion the most. We iterate this procedure until the selection

F. Schwenker and N. El Gayar (Eds.): ANNPR 2010, LNAI 5998, pp. 13–24, 2010.

criterion reaches a specified value. Because forward or backward selection is slow, we may add or delete more than one feature at a time based on feature ranking, or we may combine backward and forward selection [1].

Because these selection methods are local optimization techniques, global optimality of feature selection is not guaranteed. Usually, backward selection is slower but is more stable in selecting optimal features than forward selection [2]. If a selection criterion is monotonic for deletion or addition of a feature, we can terminate feature selection when the selection criterion violates a predefined value [3].

By the introduction of support vector machines (SVMs), various selection methods suitable for support vector machines have been developed. The selection criterion for filter methods used in the literature is, except for some cases [4,5,6,7,8], the margin [9,10,11,12]. In addition, in most cases, a linear support vector machine is used.

In [4,6], the objective function of kernel discriminant analysis called the KDA criterion, namely the ratio of the between-class scatter and within-class scatter, is proved to be monotonic for the deletion of features for two-class problems, and feature selection based on the KDA criterion was shown to be robust for benchmark data sets.

As a wrapper method, in [13,14], block deletion of features in backward feature selection is proposed using the generalization ability by cross-validation as the selection criterion.

In addition to filter and wrapper methods, the embedded methods combine training and feature selection; because training of support vector machines results in solving a quadratic optimization problem, feature selection can be done by modifying the objective function [15,16,17].

In this paper we discuss backward feature selection based on KDA proposed in [18]. For an n-class problem, KDA gives $n-1$ projection axes. We use as the selection criterion the sum of the objective function values associated with the eigenvectors, which is equivalent to the sum of eigenvalues. To speedup feature selection we use block deletion of features used in [13,14], which deletes features at the same time that give the larger KDA criterion than the threshold value if each is deleted. To stabilize stopping of feature selection, we use cross-validation for the selected sequence by block deletion. Further, to improve the separability measure of KDA, as the between-class scatter, we propose using the scatter between all the class pairs.

We compare the proposed KDA criterion with the SVM-based criterion with cross-validation and the between-class and within-class ratio and demonstrate usefulness of the proposed criterion from the standpoint of selected features and the computation time.

In Sections 2, we summarize KDA and in Section 3, we discuss selection criteria. In Section 4, we explain feature selection methods. In Section 5 we demonstrate the validity of the proposed methods by computer experiments.

2 Kernel Discriminant Analysis for Multiclass Problems

In this section, we explain kernel discriminant analysis for multiclass problems based on [19].

We assume that the center of mapped training data in the feature space is zero. Then the total scatter matrix $Q_{\rm T}$ and the between-class scatter matrix $Q_{\rm B}$ are given, respectively, by

$$Q_{\rm T} = \frac{1}{M}\sum_{k=1}^{n}\sum_{j=1}^{M_k}\boldsymbol{\phi}(\mathbf{x}_{kj})\boldsymbol{\phi}^\top(\mathbf{x}_{kj}), \tag{1}$$

$$Q_{\rm B} = \frac{1}{M}\sum_{k=1}^{n} M_k \mathbf{c}_k \mathbf{c}_k^\top, \tag{2}$$

where $\mathbf{x}_{kj}$ is the jth training data for class k, n is the number of classes, M_k is the number of training data for class k, $M = M_1 + \cdots + M_n$, $\mathbf{c}_k$ is the center of class k, and $\boldsymbol{\phi}(\mathbf{x})$ is the mapping function that maps the input space to the high-dimensional feature space.

For n class problems, we obtain $n-1$ projection axes. Let them be $\mathbf{w}_i$ ($i = 1, \ldots, n-1$). Then the total scatter and the between-class scatter on this axis are given, respectively, by

$$\frac{1}{M}\sum_{k=1}^{n}\sum_{j=1}^{M_k}(\mathbf{w}_i^\top\boldsymbol{\phi}(\mathbf{x}_{kj}))^2 = \mathbf{w}_i^\top Q_{\rm T}\mathbf{w}_i, \tag{3}$$

$$\frac{1}{M}\sum_{k=1}^{n} M_k(\mathbf{w}_i^\top\mathbf{c}_k)^2 = \mathbf{w}_i^\top Q_{\rm B}\mathbf{w}_i. \tag{4}$$

We seek the projection axis $\mathbf{w}_i$ that maximizes the between-class scatter and minimizes the total scatter. Namely,

$$\text{maximize} \quad J(\mathbf{w}_i) = \frac{\mathbf{w}_i^\top Q_{\rm B}\mathbf{w}_i}{\mathbf{w}_i^\top Q_{\rm T}\mathbf{w}_i}. \tag{5}$$

Here, $\mathbf{w}_i$ can be expressed by the linear combination of the mapped training data:

$$\mathbf{w}_i = \sum_{k=1}^{n}\sum_{j=1}^{M_k} a_i^{kj}\boldsymbol{\phi}(\mathbf{x}_{kj}), \tag{6}$$

where a_i^{kj} are constants.

Substituting (6) into (5), we obtain

$$J(\mathbf{a}_i) = \frac{\mathbf{a}_i^\top KWK\mathbf{a}_i}{\mathbf{a}_i^\top KK\mathbf{a}_i}, \tag{7}$$

where $\mathbf{a}_i = \{a_i^{kj}\}$ $(i = 1, \ldots, n-1, k = 1, \ldots, n, j = 1, \ldots, M_k)$, K is the kernel matrix, and $W = \{W_{ij}\}$ is the block diagonal matrix given by

$$W_{ij} = \begin{cases} \dfrac{1}{M_k} & \mathbf{x}_i, \mathbf{x}_j \in \text{class } k, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Taking the partial derivative of (7) with respect to $\mathbf{w}_i$, and the resulting equation to 0, we obtain the following generalized eigenvalue problem:

$$KWK\mathbf{a}_i = \lambda_i KK\mathbf{a}_i, \tag{9}$$

where λ_i are eigenvalues.

Let singular value decomposition of K be $K = P\mathit{\Gamma} P^\top$, where $\mathit{\Gamma}$ is the diagonal matrix with nonzero eigenvalues and $P^\top P = I$. Substituting $K = P\mathit{\Gamma} P^\top$ into (7) and replacing $\mathit{\Gamma} P^\top \mathbf{a}_i$ with $\boldsymbol{\beta}_i$, we obtain

$$J(\boldsymbol{\beta}_i) = \frac{\boldsymbol{\beta}_i^\top P^\top W P \boldsymbol{\beta}_i}{\boldsymbol{\beta}_i^\top P^\top P \boldsymbol{\beta}_i} = \frac{\boldsymbol{\beta}_i^\top P^\top W P \boldsymbol{\beta}_i}{\boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i}. \tag{10}$$

Therefore, the resulting eigenvalue problem is

$$P^\top W P \boldsymbol{\beta}_i = \lambda_i \boldsymbol{\beta}_i. \tag{11}$$

Solving (11) for $\boldsymbol{\beta}_i$ we obtain $\mathbf{a}_i$ from $\mathbf{a}_i = P\mathit{\Gamma}^{-1}\boldsymbol{\beta}_i$.

3 Selection Criteria

3.1 KDA Criterion for Multiclass Problems

The feature selection method based on KDA for two-class problems [6] can be extended to multiclass problems but will be architecture dependent. Therefore, we extend the method to multiclass problems using the multiclass KDA. In multiclass KDA, $n-1$ projection axes are obtained for an n-class problem. We propose using the sum of the objective function values associated with the $n-1$ projection axes. We can easily show that

$$J(\mathbf{w}_i) = \lambda_i. \tag{12}$$

Therefore the selection criterion is

$$\sum_{i=1}^{n-1} J(\mathbf{w}_i) = \sum_{i=1}^{n-1} \lambda_i, \tag{13}$$

where λ_i are given by (11). Because the sum of eigenvalues is the trace of the associated matrix, (13) becomes

$$\sum_{i=1}^{n-1} J(\mathbf{w}_i) = \mathrm{trace}\{P^\top W\, P\}, \tag{14}$$

which leads to speeding up the calculation of the selection criterion. We call this the KDA criterion.

3.2 New Between-Class Scatter

The between-class scatter for multiclass problems is calculated by the square sum of distances between the center of the mapped training data and class centers $\mathbf{c}_k$. Namely, the between-class scatter does not consider the overlap between classes. Suppose for a multiclass problem in which data of different classes do not overlap, we rotate all the data of some classes around the center of the mapped training data until different classes overlap under the constraint that the center of the mapped data does not move. Then, the between-class scatters of the initial and the rotated problems are the same. But this is unfavorable from the standpoint of class separability.

This problem can be avoided if we use the following between-class scatter:

$$Q_{\mathrm{B}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\mathbf{c}_i - \mathbf{c}_j)(\mathbf{c}_i - \mathbf{c}_j)^{\top}. \tag{15}$$

The eigenvalue problem becomes

$$KAK\mathbf{a} = \lambda KK\mathbf{a}, \tag{16}$$

where

$$A_{ij} = \begin{cases} \dfrac{n-1}{M_k^2} & \mathbf{x}_i, \mathbf{x}_j \ \in \text{class } k, \\ -\dfrac{1}{M_p M_q} & \mathbf{x}_i \in \text{class } p, \mathbf{x}_j \in \text{class } q, p \neq q. \end{cases} \tag{17}$$

The difference with the previous method is that in (9), W is replaced with A. Thus the calculation time will not change very much.

4 Feature Selection Methods

4.1 Backward Feature Selection

We select features by backward feature selection. In sequential backward feature selection, first we calculate the value of the selection criterion using all the features. Then starting from the initial set of features we temporarily delete each feature, calculate the value of the selection criterion, and delete the feature with the largest value of the selection criterion from the set. We iterate feature deletion so long as the value of the selection criterion is larger than the prescribed threshold.

The KDA criterion for two-class problems is nonincreasing for the deletion of features. We assume that this hold for the KDA criterion for multiclass problems. Then to determine the threshold we normalize the selection criterion by that evaluated using all the features. Then we set the threshold smaller than 1. It is difficult to set a proper value but in the following study based on some preliminary experiment we set $\delta = 0.95$ for multiclass problems.

Let the initial number of features be m and F^k and F_j^k denote the set of k features and the set of k features with the jth element temporarily deleted from the set, respectively. And let the selection criterion for F_j^k be $T(F_j^k)$. Then the normalized selection criterion $c(F_j^k)$ is

$$c(F_j^k) = \frac{T(F_j^k)}{T(F^m)}. \tag{18}$$

The procedure of backward feature selection is as follows:

1. Set the initial set of features as $F^m = \{1, \ldots, m\}$, and evaluate the selection criterion $T(F^m)$. Set $k = m$ and go to Step 2.
2. Delete the ith $(i = 1, \ldots, k)$ feature temporarily from F^k and calculate the normalized selection criterion $c(F_i^k)$. For the KDA criterion, if

   $$c(F_j^k) > \delta \quad \text{for} \quad j = \arg\max_{i \in F^k} c(F_i^k), \tag{19}$$

   where δ is the threshold for the KDA criterion, go to Step 3. Otherwise stop feature selection.
3. Permanently delete j from F^k:

   $$F^{k-1} = F^k - \{j\}. \tag{20}$$

   Then $k \leftarrow k - 1$ and go to Step 2.

4.2 Block Deletion

To speed up sequential backward selection, backward selection with block deletion is proposed [13,14]. To speed up variable selection, we use this method.

In block deletion, we reorder the candidate features with $c(F_j^k) > \delta$ in the descending order of $c(F_j^k)$ and delete the features simultaneously. If the selection criterion after deletion is larger than or equal to the threshold value we continue backward deletion. If not, we delete the lower half of the candidate features and repeat the above procedure until the deletion succeeds. Because one feature can be deleted, the block deletion does not fail. The algorithm is as follows.

1. Calculate $T(F^m)$ and set $k = m$.
2. Calculate $c(F_i^k)$ $(i \in F^k)$ and if

   $$c(F_i^k) > \delta \qquad \text{for } i \in F^k,$$

   include i in the candidate set V^k, which is ordered in descending order of $c(F_i^k)$ and go to Step 3. If there is no i, terminate the algorithm.
3. If $|V^k| = 1$, where $|V^k|$ is the number of elements in V^k, delete that element from F^k, $k \leftarrow k - 1$, and go to Step 2. If $|V^k| > 1$ and $c(F^k - V^k) > \delta$, set $F^k \leftarrow F^k - V^k$, $k \leftarrow k - |V^k|$, and go to Step 2. Otherwise, go to Step 4.
4. Delete the lower half of V^k, $k \leftarrow |V^k|$, and go to Step 3.

4.3 Cross-Validation as a Stopping Condition

Usually it is difficult to set a proper value to the threshold δ. To solve this problem, we use the recognition rate of the validation set by cross-validation of the SVM as a stopping condition. Namely, at Step 1 of block deletion in Section 4.2 we calculate the recognition rate, r, of the validation data set in cross-validation of the SVM. And at the end of Step 3, by sequentially deleting features according to the order of V^k, we repeat evaluating the recognition rate of the validation data set so long as it is equal to, or higher than, r. And we stop feature selection if it is lower.

In the following we show the selection algorithm for the sequence of deleted features obtained by block deletion in Section 4.2.

1. Generate the sequence of features $[f_1, \ldots, f_k]$, where f_i is a feature deleted by KDA and k is the number of deleted features.
2. Calculate the recognition rate of the validation data set with all the features, r.
3. Calculate r_i, where r_i is the recognition rate of the validation data set with i features $f_1, \ldots, f_i$ deleted.
4. Find maximum r_i that satisfies $r_i \geq r$ and delete $[f_1, \ldots, f_i]$.

5 Performance Evaluation

5.1 Data Sets and Evaluation Conditions

We performed feature selection normalizing the input range into $[0, 1]$ and using polynomial kernels: $(\mathbf{x}^\top \mathbf{x}' + 1)^d$ or RBF kernels: $\exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$, where d is the polynomial degree and γ is the width of the radius. We selected the parameter value from $d = [1, 2, 3, 4]$ for polynomial kernels and $\gamma = [0.1, 1, 10]$ for RBF kernels by fivefold cross-validation using the SVM.

In evaluating the classification performance we used the SVM with the same kernel parameter values used for feature selection and the margin parameter value selected from $C = [1, 10, 50, 100, 500, 1000, 2000, 3000, 5000, 8000, 10000, 50000, 100000]$ by fivefold cross-validation.

Table 1 lists the data sets used in the study. It also includes the kernel parameter value determined by cross-validation. In eigenvalue analysis we used the QR algorithm with the error limit for the off-diagonal elements being 10^{-6} and with the maximum iteration number of 100. We used Athlon 64×2 4800+ personal computer running on Linux.

The threshold value for the proposed method was set to $\delta = 0.95$ and we compared the following four selection methods:

1. Sequential backward selection using the proposed criterion with $\delta = 0.95$ (abbreviated as KDA),
2. Block deletion using the proposed criterion with $\delta = 0.95$ (KDA+B),
3. Block deletion using the proposed criterion with cross-validation (KDA+BC),

Table 1. Data sets

Data	Inputs	Classes	Train. Data	Test Data	kernel
Iris	4	3	75	75	$\gamma = 0.1$
Numeral	12	10	810	820	$d = 3$
Thyroid	21	3	3772	3428	$d = 1$
Blood cell	13	12	3097	3100	$\gamma = 10$
Hiragana-13	13	38	8375	8375	$\gamma = 10$
Hiragana-50	50	39	4610	4610	$\gamma = 10$
Satimage [20]	36	6	4435	2000	$\gamma = 10$

4. Block deletion using the SVM with cross-validation (SVM+BC),
5. Kernel class separability (KCS).

SVM+BC and KCS were used for comparing the proposed methods and SVM+BC used the same selection procedure as that of KDA+BC. The only difference is the selection criterion.

Kernel class separability, which is a simplified version of KDA, is a well-used measure and is defined by [21,22]

$$\frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ||\mathbf{c}_i - \mathbf{c}_j||^2}{\sum_{i=1}^{n} \frac{1}{M_i} \sum_{j=1}^{M_i} ||\boldsymbol{\phi}(\mathbf{x}_{ij}) - \mathbf{c}_i||^2}. \tag{21}$$

We selected the same number of features as that by KDA+BC.

5.2 Experimental Results

Table 2 shows the results. In the table, the "Deleted (Remaining) Features" column lists the deleted features in the order of deletion and the sequence of features in parentheses is that of remaining features. The first row of each data set shows the results using all the features and for each data set, the best recognition rate of the test data is shown in boldface. The "C" column lists the value of C selected by cross-validation of the SVM for the selected features. And "Train." and "Test" columns show the recognition rates of training data and test data, respectively.

First we compare the results for the proposed methods. For hiragana-50 and satimage data sets we could not obtain the selected features by KDA because of slow sequential backward selection. Now compare KDA and KDA+B. If different features were selected, they are shown in boldface. From the table both methods selected the same or similar sets of features. And except for the thyroid data set, both methods gave the similar recognition rates for the test data. Comparing the recognition rates of the test data with those without deleting variables, for all the data sets except for the iris data set, the recognition rates of the both methods were inferior. This means that too many features were deleted because of the improper selection of the threshold value. For example, for the hiragana-50 data by KDA+B, only five features remained and the recognition rate was

Table 2. Performance comparison of feature selection methods

Data	Method	Deleted (Remaining) Features	C	Train.	Test
Iris	—	None	100	100	**97.33**
	KDA	2	3000	97.33	**97.33**
	KDA+B	2	3000	97.33	**97.33**
	KDA+BC	2	3000	97.33	**97.33**
	SVM+BC	1	50	100	**97.33**
	KCS	1	—	100	**97.33**
Numeral	—	None	10^5	100	**99.76**
	KDA	3, 12, 7, 4, 10, 2, 5	10^5	100	99.51
	KDA+B	3, 10, 12, 5, 4, 7, 2	10^5	100	99.51
	KDA+BC	3, 10, 12	500	100	**99.76**
	SVM+BC	3, 7, 12, 10	10^5	100	99.51
	KCS	4, 5, 9	—	99.26	98.29
Thyroid	—	None	10^5	98.83	97.64
	KDA	(10, 17, 19, 20)	10^5	95.20	95.01
	KDA+B	(2, 18, 3, 10, 17, 19, 20)	10^5	97.64	96.79
	KDA+BC	(8, 2, 18, 3, 10, 17, 19, 20)	10^5	98.73	**97.90**
	SVM+BC	(3, 8, 17, 19, 20)	10^5	98.59	97.81
	KCS	(7, 12, 13, 15, 18, 19, 20, 21)	—	94.03	93.47
Blood	—	None	50	97.22	**93.55**
	KDA	1, 8, 13, 10, 11, 6	500	96.03	92.32
	KDA+B	1, 8, 13, 10, 11, **4**	50	96.03	92.32
	KDA+BC	1, 8	50	97.13	93.16
	SVM+BC	9, 8, 1, 6	50	96.96	92.41
	KCS	5, 6	—	96.19	92.00
H-13	—	None	500	100	**99.76**
	KDA	13, 11, 10	10^5	100	99.62
	KDA+B	13, **3**, **12**, 11	50	99.64	99.53
	KDA+BC	13	1000	100	**99.76**
	SVM+BC	13	1000	100	99.72
	KCS	1	—	99.99	99.55
H-50	—	None	50	100	99.07
	KDA+B	(14, 18, 28, 30, 33)	500	99.74	90.65
	KDA+BC	43, 15, 37	50	100	99.05
	SVM+BC	6, 8, 9, 12, 13, 18, 20, 23, 25, 26, 32, 35, 37, 38, 42, 43, 44, 47, 49	100	100	98.52
	KCS	43, 46, 5	—	100	**99.08**
Satimage	—	None	1000	97.34	89.20
	KDA+B	(1, 3, 5, 7, 9, 11, 13, 19, 21, 23, 25, 27, 30, 31, 33,35)	1000	95.78	87.70
	KDA+BC	24, 20, 16, 4, 32, 8	1000	97.47	88.95
	SVM+BC	(1, 2, 3, 5, 10, 11, 18, 20, 23, 25, 26, 30, 36)	1000	95.65	89.15
	KCS	3, 27, 26, 19, 35, 10	—	96.82	**89.60**

90.65%, which was much lower than 99.07% with all the features. This happened as follows. At the initial stage of feature deletion, deletion of any feature did not decrease the selection criterion. Therefore, we needed to delete features randomly until deletion of a feature led to a decrease of the selection criterion. In such a situation, we needed to use an alternative selection criterion.

By replacing the stopping condition of the threshold value in KDA+B with cross-validation, the selection became much more stable. Out of seven data sets, KDA+BC performed best four times and for the remaining data sets: the blood cell, hiragana-50, and satimage data sets, the differences from the best values were small. From the standpoint of recognition rate of the test data, KDA+BC was better than SVM+BC except for the satimage data set.

For the iris, hiragana-50 and satimage data sets, KCS showed better recognition rates than KDA+BC but for other four data sets, KDA+BC showed better recognition rates. Because KCS is not monotonic for the deletion of features selection was not stable.

In the above evaluation, we used (2) as the between-class scatter. Instead of (2), we used (15) and compared the difference of the selected features for KDA+BC, but there were not much difference between the two. For the thyroid data set, the obtained sequences were different, but the selected features for $\delta = 0.95$ were the same.

Table 3 shows the feature selection time for the four methods. In each problem the shortest time is shown in boldface. For the thyroid data set, we measured the feature selection time confining the value of C in $C = [10000, 50000, 100000]$. By introducing block deletion two to five times speedup was realized. Except for the blood cell and hiragana-13 data sets, KDA+BC was faster than SVM+BC. But for the hiragana-13 data set, SVM+BC was faster because only one feature was deleted.

For two-class problems the KDA criterion is proved to be monotonic for the deletion of features. But for the KDA criterion for multiclass problems, it is an open problem whether the KDA criterion is monotonic. Figure 1 shows the change of KDA criteria for the deletion of features for five data sets. We set $\gamma = 10$, which gave the maximum class separability. Except for the thyroid data set, the KDA criterion monotonically decreased as the features were deleted. For the thyroid data set, until six features were deleted, the KDA criterion monotonically increased. And afterwards, it decreased monotonically. For the KCS criterion, this sort of monotonicity was not observed.

Table 3. Comparison of feature selection time in seconds

Data	KDA	KDA+B	KDA+BC	SVM+BC
Iris	1	1	1	1
Numeral	313	**122**	391	773
Thyroid	87,303	**25,686**	31,822	116,026
Blood cell	20,430	**11,240**	12,371	5,432
Hiragana-13	835,648	165,408	174,703	**51,357**
Hiragana-50	–	**165,408**	271,001	404,445
Satimage	–	82,737	**12,664**	166,182

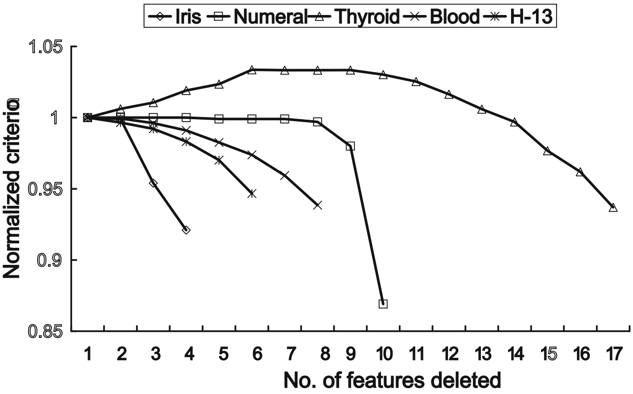

Fig. 1. Monotonicity of KDA Criterion

6 Conclusions

We proposed using the sum of objective function values associated with the eigenvectors of KDA as the selection criterion. This criterion reduces to the sum of eigenvalues of the KDA. To speed up feature selection by backward selection, we proposed to use block deletion of features at a time, and to improve the generalization ability of the selected features we proposed to use cross-validation. We also proposed calculating the between-class scatter using all the class pair distances.

By the computer experiment, we showed that the KDA criterion with block deletion performed better than the recognition rate of the SVM calculated by cross-validation if cross-validation is used to stop feature selection. But feature selection by the proposed between-class scatter did not give much difference from the conventional between-class scatter calculated based on the distances between the class centers and the center of the mapped training data.

References

1. Somol, P., Pudil, P., Novovicŏvá, J., Paclík, P.: Adaptive floating search method in feature selection. Pattern Recognition Letters 20(11-13), 1157–1163 (1999)
2. Abe, S.: Pattern Classification: Neuro-Fuzzy Methods and Their Comparison. Springer, London (2001)
3. Thawonmas, R., Abe, S.: A novel approach to feature selection based on analysis of class regions. IEEE Transactions on Systems, Man, and Cybernetics—Part B 27(2), 196–207 (1997)
4. Ashihara, M., Abe, S.: Feature selection based on kernel discriminant analysis. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 282–291. Springer, Heidelberg (2006)
5. Louw, N., Steel, S.J.: Variable selection in kernel Fisher discriminant analysis by means of recursive feature elimination. Computational Statistics & Data Analysis 51(3), 2043–2055 (2006)

6. Ishii, T., Ashihara, M., Abe, S.: Kernel discriminant analysis based feature selection. Neurocomputing 71(13-15), 2544–2552 (2008)
7. Evgeniou, T., Pontil, M., Papageorgiou, C., Poggio, T.: Image representations for object detection using kernel classifiers. In: Proc. ACCV 2000, pp. 687–692 (2000)
8. Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J.P., Poggio, T.: Support vector machine classification of microarray data, Technical Report AI Memo 1677, Massachusetts Institute of Technology (1999)
9. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)
10. Perkins, S., Lacker, K., Theiler, J.: Grafting: Fast, incremental feature selection by gradient descent in function space. Journal of Machine Learning Research 3, 1333–1356 (2003)
11. Liu, Y., Zheng, Y.F.: FS_SFS: A novel feature selection method for support vector machines. Pattern Recognition 39(7), 1333–1345 (2006)
12. Wang, L.: Feature selection with kernel class separability. Pattern Analysis and Machine Intelligence 30(9), 1534–1546 (2008)
13. Abe, S.: Modified backward feature selection by cross validation. In: Proc. ESANN 2005, pp. 163–168 (2005)
14. Nagatani, T., Abe, S.: Backward variable selection of support vector regressors by block deletion. In: Proc. IJCNN 2007, pp. 2117–2122 (2007)
15. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proc. ICML 1998, pp. 82–90 (1998)
16. Brown, M.: Exploring the set of sparse, optimal classifiers. In: Proc. ANNPR 2003, pp. 178–184 (2003)
17. Bo, L., Wang, L., Jiao, L.: Sparse Gaussian processes using backward elimination. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006, Part 1. LNCS, vol. 3971, pp. 1083–1088. Springer, Heidelberg (2006)
18. Ishii, T., Abe, S.: Feature selection based on kernel discriminant analysis for multiclass problems. In: Proc. IJCNN 2008, pp. 2456–2461 (2008)
19. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10), 2385–2404 (2000)
20. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
21. Cantú-Paz, E.: Feature subset selection, class separability, and genetic algorithms. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 959–970. Springer, Heidelberg (2004)
22. Wang, L., Chan, K.L.: Learning kernel parameters by using class separability measure. In: Sixth Kernel Machines Workshop, In conjunction with Neural Information Processing Systems, NIPS (2002)