Feature Selection by Block Addition and Block Deletion

Takashi Nagatani and Shigeo Abe

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

Abstract. In our previous work, we have developed methods for selecting input variables for function approximation based on block addition and block deletion. In this paper, we extend these methods to feature selection. To avoid random tie breaking for a small sample size problem with a large number of features, we introduce the weighted sum of the recognition error rate and the average of margin errors as the feature selection and feature ranking criteria. In our methods, starting from the empty set of features, we add several features at a time until a stopping condition is satisfied. Then we search deletable features by block deletion. To further speedup feature selection, we use a linear programming support vector machine (LP SVM) as a preselector. By computer experiments using benchmark data sets we show that the addition of the average of margin errors is effective for small sample size problems with large numbers of features in realizing high generalization ability.

Keywords: Backward feature selection, feature ranking, forward feature selection, pattern classification, support vector machines.

1 Introduction

Feature selection aims at selecting the set of features, with the minimum number, that realizes a classifier with high generalization ability. In its original form, in backward selection, a feature is deleted sequentially from the full set of initial features or in forward selection, a feature is added sequentially to the initial empty set. If the recognition rate or more generally the generalization ability is used as the selection criterion, the selection method is called a wrapper method, and otherwise the filter method.

The wrapper method gives good generalization ability but its computational burden is high. In [1–3] the recognition rate for the cross-validation data set is used as a selection criterion. To speed up feature selection in such a situation, in [3], input variables for function approximation are added or deleted not one by one but in a block. In [4], a filter method and a wrapper method are combined to alleviate the computational burden of the wrapper method. In the filter stage, features with low class separability and high correlation with other features are

N. Mana, F. Schwenker, and E. Trentin (Eds.): ANNPR 2012, LNAI 7477, pp. 48–59, 2012.

eliminated. In the wrapper stage, a feature is added one by one by training the support vector machine (SVM) several times and selecting the feature with the maximum objective function value of the SVM.

Many filter methods have been developed and selection criteria based on mutual information are often used [5, 6]. As for SVM-based feature selection, the margin and the weights are widely used as a selection criterion. In [7], the SVM-RFE (SVM-Recursive Feature Elimination) is proposed in which backward feature selection is used with the minimum absolute weight in the separating hyperplane as a selection criterion.

As for sequential selection, a combination of forward selection with backward selection is proposed. In [8], sequential floating selection is proposed, in which after each sequential forward selection, backward selection is repeated so long as the selection criterion is satisfied.

With the introduction of SVMs, imbedded methods have been proposed, in which feature selection and training are done simultaneously. In [9], L0 norm, namely, the number of nonzero elements in the coefficient vector is included in the objective function. This term works to suppress irrelevant features. However, the generalization ability is usually inferior to regular SVMs. Therefore, in [10], like regular SVMs, the quadratic term is included in the objective function, in addition to the L0 or L1 norm term. This approach is extended to nonlinear cases.

To speed up wrapper methods, some feature selection methods use filter methods as a preselector such as LP SVMs with linear kernels [11, 3]. After training an LP SVM, input variables with small absolute values of weights are deleted.

In this paper, we extend the variable selection methods for function approximation [3] to pattern classification. The direct extension would replace, as a selection criterion, the approximation error with the recognition error rate. But because the recognition error rate is discrete, random tie breaking of feature ranking will occur for a large number of features and a small number of training data. To avoid this, we introduce the average of margin errors in addition to the recognition error rate.

The procedure for feature selection is almost the same as that in [3] excluding some alterations for improvement. We select features by block forward addition followed by block backward deletion. If the number of features is very large we use a filter method as a preselector.

Unlike feature ranking methods, the proposed feature selection method terminates when the selection criterion is no longer improved. Initially, we set the threshold value for the selection criterion using all the features. Then during feature selection we update the threshold value if the selection criterion better than the current threshold value is obtained. This guarantees to find the feature set with the selection criterion better than that of the initial set of features.

In block addition and block deletion, we simultaneously add or delete multiple features so long as the selection criterion is not worsened. Assuming that the number of features is large and also many irrelevant or redundant features are

included, we fist do block addition then do block deletion because the former is more efficient than the latter is.

In Section 2, we discuss the selection criteria and the stopping conditions of feature selection. Then in Section 3 we discuss the proposed methods based on block addition and block deletion, and in Section 4, we show the simulation results using two-class benchmark data sets.

2 Selection Criteria and Stopping Conditions

In feature selection, we want to obtain a minimum feature set that realizes the generalization ability comparable to or better than that using the original feature set. From this point of view, the generalization ability estimated by cross-validation, namely, the recognition error rate for the validation data set that is hold out during cross-validation, is a good choice.

Let the decision function for a two class problem be trained so that

$$y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) \geq 1 - \xi_i \qquad \text{for} \quad i = 1, \ldots, M, \tag{1}$$

where, $\mathbf{x}_i$ and y_i are the ith $(i = 1, \ldots, M)$ training input and output, respectively, $\mathbf{w}$ is the coefficient vector of the separating hyperplane in the feature space, $\boldsymbol{\phi}(\mathbf{x})$ is the mapping function that maps $\mathbf{x}$ into the feature space, b is the bias term, and $\xi_i \, (\geq 0)$ is the slack variable associated with $\mathbf{x}_i$.

Then the recognition error rate E_{C} of the training data is given by

$$E_{\mathrm{C}} = \frac{1}{M} \sum_{i=1}^{M} e_i, \tag{2}$$

where

$$e_i = \begin{cases} 0 & \text{for} \quad y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) \geq 0, \\ 1 & \text{for} \quad y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) < 0. \end{cases} \tag{3}$$

The recognition error rate for the validation data set is calculated for the training data that are hold out in cross-validation.

Unlike the approximation error for function approximation, the recognition error rate is discrete. Therefore, if the number of features is large and the number of training data is small, the same recognition error rate will be obtained for different subsets of features. And random tie breaking during feature selection may not give a good selection result.

As a continuous criterion we consider using the average of margin errors:

$$E_{\mathrm{M}} = \frac{1}{M} \sum_{i=1}^{M} \xi_i, \tag{4}$$

where

$$\xi_i = \begin{cases} 0 & \text{for} \quad y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) \geq 1, \\ 1 - y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) & \text{for} \quad y_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b \right) < 1. \end{cases} \tag{5}$$

Because we only want to use (4) to break ties, we consider the margin error-based criterion $E_{\mathrm{M_C}}$ as follows:

$$E_{\mathrm{M_C}} = E_{\mathrm{C}} + r\, E_{\mathrm{M}}, \tag{6}$$

where r is a positive parameter. Because the minimum positive change of E_{C} is $1/M$, assuming $E_{\mathrm{M}} \leq 1$ and $r \leq 1/M$, the ranking list of features by $E_{\mathrm{M_C}}$ and that by E_{C} are the same except for the feature subsets with the same E_{C}. Therefore we set $r = 1/M$.

To evaluate $E_{\mathrm{M_C}}$, we consider the following three criteria for feature selection and feature ranking:

1. MM criterion: $E_{\mathrm{M_C}}$ for both feature selection and feature ranking;
2. CM criterion: E_{C} for feature selection and $E_{\mathrm{M_C}}$ for feature ranking;
3. CC criterion: E_{C} for both feature selection and feature ranking.

In the following, if there is no confusion, we simply say E instead of $E_{\mathrm{M_C}}$ or E_{C} and E is usually used for the selection criterion, and if it is used for the ranking criterion we denote it as *ranking* E.

At the start of feature selection, we set the threshold value for the selection criterion using all the features. Let the threshold be T. Then T is determined by $T = E^m$, where m is the number of initial features and E^m is the selection criterion evaluated by cross-validation. We update the threshold value when we obtain the selection criterion smaller than the threshold value as follows. Let the current selection criterion with j features be E^j. Then if

$$E^j < T, \tag{7}$$

we consider E^j as a new threshold value and set

$$T \leftarrow E^j \tag{8}$$

and continue feature selection. To obtain the smallest subset of features, we add features so long as $E < T$, namely we do not add features if $E = T$. And we delete features so long as $E \leq T$.

Now consider the difference among the three criteria for a small sample problem with a large number of features, where the problem is linearly separable. Suppose we obtain a subset of features with $E_{\mathrm{C}} = 0$ adding features using the CM or CC criterion. Because of the discussion above, we stop feature addition. But even if we obtain a subset of features with $E_{\mathrm{C}} = 0$ by feature deletion, we proceed feature deletion so long as $E_{\mathrm{C}} = 0$.

Now by the MM criterion, we add features so long as $E_{\mathrm{C}} = 0$ and $E_{\mathrm{M_C}}$ decreases. And we stop feature deletion, if $E_{\mathrm{M_C}}$ increases although $E_{\mathrm{C}} = 0$. Therefore, the MM criterion tends to select more features that the CM or CC criterion does.

To control that the MM criterion does not select much more features than the CM or CC criterion does, we consider two conditions to stop feature addition:

$$\Delta T < \varepsilon_{\mathrm{M}} \quad \text{or} \quad E_{\mathrm{C}} = 0, \tag{9}$$

where $\Delta T = T - E^j$ when (7) is satisfied and ε_{M} is a positive value.

3 Block Addition and Block Deletion

From the standpoint of quality of the selected feature set, backward selection, which deletes irrelevant or redundant features from the feature set, is more stable than forward selection, which selects features that are important only for the selected features. But if we need to select a small number of features from a large number of features, backward selection is slower than forward selection is. And this is usually the case for a large number of features.

To speed up feature selection in such a situation, in the following we discuss the method called feature selection by block addition and block deletion (BABD). In BABD, to speed up backward selection, we use forward selection as a pre-selector and afterwards, for the set of selected features we perform backward selection. To speedup forward and backward selection processes, we delete or add multiple features at a time and repeat addition or deletion until the stopping condition is satisfied.

3.1 Block Addition

First, we calculate the selection criterion E^m from the initial set of features $I^m = \{1, \ldots, m\}$ and set the threshold value of the stopping condition $T = E^m$. We start from the empty set of selected features. Assume that we have selected j features with the set of features I^j. Then we add the ith feature in set $I^m - I^j$ temporarily to I^j, calculate $E^j_{i_{\text{add}}}$, where i_{add} indicates that the ith feature is added to the feature set, and calculate the selection criterion for the validation data set. Then we rank the features in $I^m - I^j$ in the ascending order of the ranking criteria. We call this ranking feature ranking V^j.

We add k ($k = 1, 2^1, \ldots, 2^A$) features from the top of V^j to the feature set temporarily, where $2^A \leq m$ and A is a user-defined parameter, which determines the number of added candidates. We compare E^{j+k} with the value of threshold T. If

$$E^{j+k} < T, \tag{10}$$

we update the threshold, add the features to the feature set permanently, and continue feature addition unless (9) is not satisfied for the MM criterion.

If (10) is not satisfied for $k = 1, 2^1, \ldots, 2^A$, we check if for some k the selection criterion is decreased by adding k features to I^j:

$$E^{j+k} < E^j. \tag{11}$$

Here we assume that $E^0 = \infty$. If it is satisfied let

$$k = \arg \min_{i=1,2^1,\ldots,2^A} E^{j+i} \tag{12}$$

and we add to I^j the first k features in the feature ranking permanently and continue feature addition.

If (10) and (11) are not satisfied, but $E^j \leq T$, we stop feature addition. Otherwise we add to I^j the first feature in the feature ranking and continue feature addition to guarantee obtaining a feature set.

3.2 Block Deletion

Le the set of features obtained after block addition be I^j. Now by block deletion we delete redundant features from I^j. The reason for block deletion is as follows. In block addition, we evaluate feature ranking by temporarily adding one feature and we add multiple, high-ranked features. Thus, redundant features may be added by block addition.

We delete the ith feature in I^j temporarily from I^j and calculate $E^j_{i_{\rm del}}$, where $E^j_{i_{\rm del}}$ is the recognition error rate when we delete the ith feature from I^j. Then we consider the features that satisfy $E^j_{i_{\rm del}} \leq T$, as candidates of deletion and generate the set of features that are candidates for deletion by

$$S^j = \{i \,|\, E^j_{i_{\rm del}} \leq T, i \in I^j\}. \tag{13}$$

We generate V^j, ranking the candidates in the ascending order of *ranking* $E^j_{i_{\rm del}}$ and delete all the candidates from I^j temporarily. We compare $E^{j'}$ with the threshold T, where j' is the number of features after the deletion. If

$$E^{j'} \leq T, \tag{14}$$

block deletion has succeeded and we delete the candidate features permanently from I^j and update the threshold. If block deletion has failed, we backtrack and delete the features in the upper half of V^j. We iterate the procedure until block deletion succeeds.

We iterate the above procedure until no features are deleted.

3.3 Preselection by Filter Methods

If the number of features is very large, even block addition or block deletion may be inefficient. To overcome this problem we combine BABD with a filter method. We call this method BABD-FL. If the value of E for the subset of features obtained by the filter method is smaller than, or equal to, the threshold T, we update the threshold and delete features by block deletion from the set. If larger, we add features by block addition to the set of features obtained by preselection.

At first, we set the threshold of the stopping condition $T = E^m$ from the initial set of features I^m. By the filter method, we calculate the subset of features and set $j_{\rm FL}$ as the number of features after preselection. Then we compare the current selection criterion $E^{j_{\rm FL}}$ with the threshold T. If

$$E^{j_{\rm FL}} \leq T, \tag{15}$$

we update the threshold and search more deletable features by block deletion. If (15) is not satisfied, we add features to the current feature set by block addition until the selection criterion is below or equal to T. After block addition is finished we delete features by block deletion.

3.4 Algorithm of BABD-FL

In the following we show the algorithm of BABD-FL. If preselection is not used, we start from Step 4.

Preselection

Step 1 Calculate E^m for I^m. Set $T = E^m$.

Step 2 Calculate the subset of features by the filter method. Let the resulting subset of features obtained by preselection be $I^{j_{\rm FL}}$.

Step 3 Calculate $E^{j_{\rm FL}}$ for $I^{j_{\rm FL}}$ and set $j \leftarrow j_{\rm FL}$ and $E^j \leftarrow E^{j_{\rm FL}}$. If $E^{j_{\rm FL}} > T$, go to Step 5. Otherwise, go to Step 7.

Block Addition

Step 4 Calculate E^m for I^m. Set $T = E^m$, $j = 0$, and $E^0 = \infty$.

Step 5 Add the ith feature in $I^m - I^j$ temporarily to I^j, calculate *ranking* $E^j_{i_{\rm add}}$, and generate V^j. Set $k = 1$.

Step 6 Calculate E^{j+k} $(k = 1, 2^1, \ldots, 2^A)$. If (10) is satisfied, set $j \leftarrow j+k, T \leftarrow E^j$. And if (9) is satisfied for the MM criterion or selection is by BABD-FL go to Step 7; if not, go to Step 5. Otherwise, if (11) is satisfied, set $j \leftarrow j+k$ and go to Step 5. Otherwise, if both (10) and (11) are not satisfied but $E^j \leq T$, go to Step 7; otherwise set $j \leftarrow j+1, T \leftarrow E^j$ and go to Step 5.

Block Deletion

Step 7 Delete temporarily the ith feature in I^j and calculate *ranking* $E^j_{i_{\rm del}}$.

Step 8 Calculate S^j. If S^j is empty, stop feature selection. If only one feature is included in S^j, set $I^{j-1} = I^j - S^j$, $j \leftarrow j-1$ and go to Step 7. If S^j has more than two features, generate V^j and go to Step 9.

Step 9 Delete all the features in V^j from I^j: $I^{j'} = I^j - V^j$, where $j' = j - |V^j|$ and $|V^j|$ denotes the number of elements in V^j. Then, calculate $E^{j'}$ and if $E^{j'} > T$, go to Step 10. Otherwise, update j with j', $T \leftarrow E^{j'}$, and go to Step 7.

Step 10 Let $V^{'j}$ include the upper half elements of V^j. Set $I^{j'} = I^j - \{V^{'j}\}$, where $\{V^{'j}\}$ is the set that includes all the features in $V^{'j}$ and $j' = j - |\{V^{'j}\}|$. Then, if $E^{j'} \leq T$, delete features in $V^{'j}$ and go to Step 7 updating j with j' and T with $E^{j'}$. Otherwise, update V^j with $V^{'j}$ and iterate Step 10 until (14) is satisfied.

4 Performance Evaluation

In this section, we compare the three feature selection and feature ranking criteria for several two class problems. We also compare the proposed methods with the wrapper methods [2, 4] and the embedded method [10].

4.1 Evaluation Conditions

As a classifier we used a least squares SVM (LS SVM) whose primal problem is: minimize $1/2\mathbf{w}^\top\mathbf{w} + C/2\sum_{i=1}^M \xi_i^2$ subject to $y_i\,(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) = 1 - \xi_i$ for $i = 1, \ldots, M$, where C is the margin parameter. In training the LS SVM, we solved

the set of linear equations that is derived by transforming the primal problem into the dual problem. As a kernel function, we used RBF kernels: $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x})\,\boldsymbol{\phi}(\mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$, where γ is a positive parameter.

For BABD-FL, similar to the method in [3], we used the linear programming SVM (LP SVM) as a preselector and called the method BABD-LP. We selected the margin parameter in the LP SVM from $\{10, 10000\}$.

For BABD and BABD-LP, we set $A = 5$ and $\varepsilon_{\mathrm{M}} = 10^{-5}$.

We determine the γ and C values by fivefold cross-validation. We selected the γ value from $\{0.001, 0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 50, 100\}$, and the C value from $\{1, 10, 50, 100, 500, 1000, 2000\}$. We trained the LS SVM for all pairs of parameter values and selected the values that realized the minimum value of the feature selection criterion for the validation data set.

For the MM criterion, the initial solution may be different from that for the CM and CC criteria. To make comparison easy, we set the same initial solution for the MM criterion as that for the CM and CC criteria.

To reduce computational cost of feature selection, first we determined the stopping threshold using all the features optimizing the γ and C values and then during feature selection, we fixed these parameter values to the determined values. This sped up feature selection without much deterioration of the generalization ability.

4.2 Comparison of Feature Selection Methods

We compared BABD-LP, BABD, and BD with the MM, CM, and CC criteria for two microarray problems, each of which consisted of 100 pairs of training and test data sets. Because these problems were linearly separable, we used linear kernels with $C = 1$. We calculated the average recognition rate for the test data set (validation data set), the average number of features selected by each method and its standard deviation, and measured the average computation time per data set using a personal computer with 3GHz CPU and 2GB memory.

Table 1 shows the results. In the "Data (Tr/Te/In)" column, we show the data set name followed by the numbers of training data, test data, and inputs in parentheses; and the recognition rates of the test data sets and, in parentheses, of validation data sets using all the features. In the "Method" column, LP, BA, and BD denote the BABD-LP, BABD, and BD, respectively. And BA* denotes that block addition was terminated when $E_{\mathrm{C}} = 0$. In the "Recognition rate" column, the best recognition rate among the three criteria is shown in boldface. In the "LP," "BA," and "BD" columns, we show the numbers of features selected by the LP SVM, BA, and BD, respectively, and the smallest average number of selected features in boldface. In the "Time (s)" column, we list the CPU time for feature selection per data set and the minimum time in bold face.

From the table, the MM criterion showed the best recognition rates for the six cases tested, although it required more features than the other two did, and thus, for BABD and BD, it required more computation time. If feature selection was terminated at $E_{\mathrm{C}} = 0$, feature selection was sped up but with the sacrifice

Table 1. Comparison of selection methods for microarray data sets

Data (Tr/Te/In)	Method	Recognition rate	LP	BA	BD	Time (s)
B. cancer	LP MM	**77.88**±10.44 (100±0)	**10.7**±1.0	0	8.4±1.5	**0.33**
(14/8/3226)	LP CM	70.50±13.41 (100±0)	**10.7**±1.0	0	**2.0**±0.4	**0.33**
73.88±11.47	LP CC	71.75±12.45 (100±0)	**10.7**±1.0	0	2.2±0.6	**0.33**
(76.50±7.09)	BA MM	**80.50**±11.36 (100±0)	—	46.0±11.3	40.5±11.9	6.98
	BA* MM	78.12±11.23 (100±0)	—	15.6±2.0	9.7±1.9	**0.42**
	BA CM	70.38±11.41 (100±0)	—	**2.4**±2.1	**1.7**±0.6	0.82
	BA CC	69.75±12.14 (99.93±0.71)	—	3.8±3.5	1.9±0.8	0.83
	BD MM	**78.88**±11.55 (100±0)	—	—	48.7±28.7	48.8
	BD CM	70.62±12.04 (89.14±13.87)	—	—	**1.6**±0.6	**48.0**
	BD CC	66.25±11.92 (85.93±12.22)	—	—	1.9±1.2	52.9
Leukemia	LP MM	**93.38**±3.98 (100±0)	**24.8**±2.3	0	16.1±2.5	**7.48**
(38/34/7129)	LP CM	87.97±5.97 (100±0)	**24.8**±2.3	0	**3.6**±0.9	7.56
94.44±4.70	LP CC	86.15±7.28 (100±0)	**24.8**±2.3	0	4.0±1.1	7.52
(92.45±3.32)	BA MM	**94.38**±3.88 (100±0)	—	57.9±12.6	47.9±12.2	68.0
	BA* MM	92.82±5.15 (100±0)	—	26.6±12.7	16.7±7.9	**13.8**
	BA CM	87.68±6.60 (99.79±0.71)	—	8.6±6.1	**3.4**±1.1	18.1
	BA CC	86.71±6.99 (99.21±1.26)	—	**8.3**±5.3	3.6±1.4	16.3
	BD MM	**94.68**±3.83 (99.97±0.26)	—	—	96.1±52.7	1862
	BD CM	87.82±6.99 (99.92±0.58)	—	—	**3.8**±0.9	**952**
	BD CC	81.85±7.45 (98.45±2.47)	—	—	6.2±2.9	1132

of the recognition rate (see BA* MM rows for the breast cancer and leukemia problems).

For BABD-LP, the average numbers of selected features in "BA" column are zero for the breast cancer and leukemia problems. This means that feature selection by LP SVM improved the selection criterion and no addition of features was necessary. In addition, the recognition rates for the validation data sets are all 100%. Therefore, preselection by LP SVM worked well.

By BA MM, the recognition rates for the validation data sets were all 100% while those for the CM and CC criteria were not. Even if $E_{\mathrm{C}} = 0$ is reached, $E_{\mathrm{M_C}}$ may be positive for $E_{\mathrm{C}} = 0$ and thus, by the MM criterion, feature addition may be continued after $E_{\mathrm{C}} = 0$ so long as $E_{\mathrm{M_C}}$ is improved. Then at the block deletion stage, features will not be deleted if $E_{\mathrm{C}} > 0$. But for the CM and CC criteria, this sort of thing is not satisfied.

By BD MM for the leukemia problem, the recognition rate for the validation data set is not 100%. This is because features were deleted before E_{C} reached 0.

The CM criterion showed better recognition rates than the CC criterion did except for LP CM for the breast cancer data set. Therefore, feature ranking by $E_{\mathrm{M_C}}$ worked better than by E_{C} for these data sets.

The generalization abilities of BABD and BD are comparable, but the feature selection time of BD is slower due to longer time for feature ranking.

Next, we compared BABD with other methods using the four data sets in [13]. As shown in [4], we randomly divided each data set into training and test data sets with the ratio of 80% and 20% and generated 20 pairs of training and

Table 2. Comparison of selection methods

Data (Tr/Te/In)	Method	Recognition rate	BA	BD
Bupa liver (276/69/6)	MM	**71.74**±5.79 (**73.13**±2.01)	**5.8**±0.5	5.7±0.6
72.68±6.14 (72.92±2.08)	CM, CC	71.67±5.71 (**73.13**±2.01)	**5.8**±0.6	**5.6**±0.7
	[4]	70.2	—	4.6
66.7±0.8	[10]	67.5±0.8	—	3.2
Ionosphere (281/70/34)	MM	**93.93**±2.59 (**97.10**±0.80)	25.1±4.6	**15.2**±5.0
94.21±1.89 (95.57±0.67)	CM	**93.93**±2.99 (96.83±0.64)	22.1±5.2	15.3±5.0
	CC	93.79±2.61 (96.74±0.72)	**21.6**±5.3	15.9±5.9
	[4]	92.0	—	10
92.9±0.2	[10]	92.3±0.3	—	6.6
Pima Indians (614/154/8)	MM	**75.39**±2.41 (**78.37**±0.55)	**6.5**±1.3	**6.0**±1.3
75.81±2.52 (77.81±0.82)	CM, CC	75.36±2.47 (78.33±0.56)	6.7±1.3	6.2±1.3
	[4]	74.5	—	4.2
76.6±0.2	[10]	73.0±0.2	—	1.4
WDBC (455/114/30)	MM	97.11±1.15 (**98.41**±0.33)	**23.4**±4.9	**16.6**±4.4
97.41±0.98 (98.09±0.34)	CM	**97.15**±1.07 (98.32±0.31)	24.1±6.0	21.4±7.0
	CC	97.11±1.30 (98.35±0.32)	24.4±5.9	22.4±6.6
	[4]	93.0	—	15
98.25±2.0	[2]	97.69±0.9	—	12

test data sets. But because the generated data sets, the number of data sets in some cases, and the classifiers used are different, exact numerical comparison is meaningless.

Table 2 shows the results. In [4], forward feature selection was done by the combination of filter and wrapper methods using the L1 SVM. In [10], the embedded method is used for feature selection. And in [2], sequential backward selection is used. The difference of the method with BD is that BD uses block deletion and threshold updating.

From the table, performance of BABD is better than or comparable to that of these methods.

Finally, using the two class data sets [12] with the numbers of features smaller than or equal to 60, we compared the feature selection criteria. We used BABD. The data sets consisted of 100 or 20 pairs of training and test data sets. Table 3 shows the results. The asterisk in the "Recognition rate" column means that the recognition rate is lower than that with all the features. The last to the third last rows of the table show the summary. For example, 6/1/5 in the MM row of the "Recognition rate" column means that the MM criterion performed best six times, the second best once, and the worst five times.

From the summary, it is interesting to note that the MM criterion showed the best recognition performance for the validation data sets but comparable with other two for the test data sets. But comparing the recognition rates, the difference among three criteria is very small for these data sets.

Although the feature selection methods guarantee the improvement of recognition rates for the validation data sets, it does not always lead to improvement in the recognition rates for the test data sets. Out of 12 problems, they deteriorated for 7 problems.

Table 3. Comparison of three selection criteria for two-class data sets

Data (Tr/Te/Im)	Method	Recognition rate	BA	BD
Cancer (200/77/9)	MM	73.14±4.35 (**77.90**±1.79)	5.8±2.6	4.4±1.8
	CM	73.22±4.45 (77.73±1.77)	**5.3**±2.7	**4.2**±2.0
73.05±4.61 (75.75±2.03)	CC	**73.35**±4.29 (77.74±1.77)	5.5±2.8	**4.2**±2.0
Diabetes (468/300/8)	MM	**76.08***±1.89 (78.64±1.08)	6.5±1.4	6.0±1.4
	CM	**76.08***±1.97 (**78.65**±1.08)	**6.4**±1.5	**5.9**±1.4
76.49±1.93 (78.00±1.18)	CC	**76.08***±1.92 (78.64±1.08)	**6.4**±1.5	**5.9**±1.5
Solar (666/400/9)	MM	**66.70**±2.07 (67.73±1.16)	5.9±2.5	4.7±1.9
	CM	**66.70**±2.06 (**67.74**±1.16)	**5.3**±3.0	**4.5**±2.5
66.29±1.95 (67.36±1.22)	CC	66.68±2.07 (67.73±1.14)	5.8±2.9	4.6±2.5
German (700/300/20)	MM	**75.46***±2.34 (**77.64**±0.96)	17.4±3.9	**13.6**±3.4
	CM	75.42*±2.21 (77.60±0.98)	**17.1**±4.1	13.9±3.9
75.93±1.89 (76.55±1.11)	CC	75.43*±2.00 (77.62±1.01)	18.9±2.5	13.9±3.5
Heart (170/100/13)	MM	82.05*±3.94 (85.81±2.04)	9.5±2.5	**8.2**±2.5
	CM	82.34*±3.78 (85.83±2.04)	**9.3**±2.8	8.3±2.7
82.48±3.61 (84.58±2.23)	CC	**82.45***±3.67 (**85.84**±2.03)	9.4±2.8	**8.2**±2.7
Image (1300/1010/18)	MM	97.86±0.34 (97.93±0.28)	**16.8**±0.6	**9.9**±3.0
	CM	**97.91**±0.38 (**97.97**±0.31)	17.2±0.7	10.9±2.1
97.69±0.56 (97.37±0.30)	CC	**97.91**±0.35 (97.94±0.30)	17.0±0.8	11.3±2.4
Ringnorm (400/7000/20)	MM	**97.55***±0.58 (**98.83**±0.51)	19.4±1.0	18.0±1.5
	CM	97.17*±0.79 (98.79±0.32)	**19.0**±1.3	**17.0**±2.0
98.11±0.27 (98.38±0.60)	CC	97.17*±0.79 (98.78±0.54)	**19.0**±1.3	**17.0**±2.1
Splice (1000/2175/60)	MM	92.50±1.12 (93.45±0.83)	9.4±4.4	8.6±4.3
	CM	92.46±1.06 (93.45±0.83)	9.3±4.5	8.5±4.3
89.05±0.81 (88.87±0.99)	CC	**92.56**±1.04 (**93.49**±0.81)	**9.2**±4.5	**8.3**±4.4
Thyroid (140/75/5)	MM	95.20*±2.45 (**97.40**±1.02)	**4.7**±0.5	**4.3**±0.8
	CM	95.25*±2.35 (97.38±1.01)	**4.7**±0.5	4.4±0.8
95.36±2.37 (97.14±1.03)	CC	**95.27***±2.37 (97.37±1.00)	**4.7**±0.5	4.4±0.8
Titanic (150/2051/3)	MM	**77.49**±0.67 (**79.41**±3.52)	**2.1**±0.9	**2.1**±0.9
	CM	**77.49**±0.67 (**79.41**±3.52)	2.4±0.8	2.4±0.8
77.43±0.77 (79.31±3.54)	CC	**77.49**±0.67 (**79.41**±3.52)	2.4±0.8	2.4±0.8
Twonorm (400/7000/20)	MM	**96.82***±0.73 (**98.29**±0.50)	18.7±1.7	18.3±1.6
	CM	96.48*±0.86 (98.24±0.58)	**18.3**±1.8	**17.5**±2.0
97.42±0.27 (98.00±0.58)	CC	96.52*±0.85 (98.28±0.53)	**18.3**±1.9	17.7±2.0
Waveform (400/4600/21)	MM	89.37*±1.17 (**92.30**±1.36)	19.0±1.6	**14.9**±2.8
	CM	**89.50***±1.13 (92.17±1.39)	18.8±1.7	15.5±2.8
90.09±0.58 (91.02±1.52)	CC	89.45*±1.16 (92.14±1.43)	**18.6**±2.0	15.6±2.7
Summary	MM	6/1/5 (7/3/2)	3/1/8	6/0/6
	CM	5/4/3 (4/5/3)	8/3/1	5/6/1
	CC	7/4/1 (3/6/3)	6/5/1	5/5/2

5 Conclusions

In this paper, we proposed a wrapper-based feature selection method by block addition and block deletion of features. Because feature selection and feature ranking by the recognition rate may cause random tie breaking especially for

a large number of features and a small number of samples, we proposed using the weighted sum of the recognition error rate and the average of margin errors. The weight is determined so that the averages of margin errors do not change orders when there are no ties in the recognition rates. We select features first by adding several features at a time while the selection criterion is larger than the threshold value. Then, we delete several features at a time while the selection criteria is smaller than, or equal to, the threshold value. Initially, the threshold value is determined by using all the features. Then during feature selection, it is updated when the selection criterion lower than the threshold is obtained.

The computer experiments for two class data sets showed that the proposed selection criterion is better than the recognition error rate especially for the microarray data sets with a large number of features and a small number of samples.

References

1. Abe, S.: Modified backward feature selection by cross validation. In: Proc. ESANN 2005, pp. 163–168 (2005)
2. Maldonado, S., Weber, R.: A wrapper method for feature selection using support vector machines. Information Sciences 179(13), 2208–2217 (2009)
3. Nagatani, T., Ozawa, S., Abe, S.: Fast variable selection by block addition and block deletion. J. Intelligent Learning Systems & Applications 2(4), 200–211 (2010)
4. Liu, Y., Zheng, Y.F.: FS_SFS: A novel feature selection method for support vector machines. Pattern Recognition 39(7), 1333–1345 (2006)
5. Peng, H., Long, F., Dingam, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
6. Herrera, L.J., Pomares, H., Rojas, I., Verleysen, M., Guilén, A.: Effective Input Variable Selection for Function Approximation. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006, Part I. LNCS, vol. 4131, pp. 41–50. Springer, Heidelberg (2006)
7. 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)
8. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15(11), 1119–1125 (1994)
9. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proc. ICML 1998, pp. 82–90 (1998)
10. Neumann, J., Schnörr, C., Steidl, G.: Combined SVM-based feature selection and classification. Machine Learning 61(1-3), 129–150 (2005)
11. Bi, J., Bennett, K.P., Embrechts, M., Breneman, C.M., Song, M.: Dimensionality reduction via sparse support vector machines. J. Machine Learning Research 3, 1229–1243 (2003)
12. IDA Benchmark Repository, `http://www.fml.tuebingen.mpg.de/members/raetsch/benchmark`
13. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`