Incremental Feature Selection by Block Addition and Block Deletion Using Least Squares SVRs

Shigeo Abe

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

Abstract. For a small sample problem with a large number of features, feature selection by cross-validation frequently goes into random tie breaking because of the discrete recognition rate. This leads to inferior feature selection results. To solve this problem, we propose using a least squares support vector regressor (LS SVR), instead of an LS support vector machine (LS SVM). We consider the labels (1/-1) as the targets of the LS SVR and the mean absolute error by cross-validation as the selection criterion. By the use of the LS SVR, the selection and ranking criteria become continuous and thus tie breaking becomes rare. For evaluation, we use incremental block addition and block deletion of features that is developed for function approximation. By computer experiments, we show that performance of the proposed method is comparable with that with the criterion based on the weighted sum of the recognition error rate and the average margin error.

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

1 Introduction

To realize a classifier with high generalization ability, feature selection, which eliminates redundant and irrelevant features, is especially important for a small sample problem with a large number of features (SSPLF). In such a problem, to avoid deleting important features for classification, wrapper methods [1–3], which use recognition rate-based criteria, are preferable to filter methods, which use more simpler criteria [4–6].

For kernel-based classifiers, imbedded methods, in which feature selection and training are done simultaneously are also used [7, 8].

For wrapper methods, forward selection and backward selection are often used. In forward selection, a feature is sequentially added to an initially empty set, and in backward selection, a feature is sequentially deleted from the set initialized with all the features. Because forward selection is faster than backward selection

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 35–46, 2014.
© Springer International Publishing Switzerland 2014

if the number of selected features is small, but less stable, the combination of forward selection and backward selection is also used [3, 9–11].

There are several approaches to speed up wrapper methods: some feature selection methods combine filter methods and wrapper methods and use filter methods as a preselector [12–14]. In [3], instead of sequential forward selection and backward selection, block addition (BA) of features followed by block deletion (BD) of features is proposed.

Incremental selection has also been proposed [15–19] to speed up feature selection. In [19], BABD for input variable selection is extended to incremental selection and speedup was shown for the small sample problems with a large number of input variables (SSPLV).

In applying a wrapper method to an SSPLF, frequently we need to break ties in feature selection and feature ranking, because the feature selection/ranking criterion is discrete. In addition, the number of selected features is very small because the 100% recognition rate is easily obtained for the validation data set. This worsens the generalization ability. To avoid this, we used the weighted sum of the recognition error rate and the average margin error [3]. This led to more stable feature selection for microarray data sets.

In this paper, instead of the weighted sum of error rate and the average margin error used in [3], we propose using the mean absolute error by the least squares support vector regressor (LS SVR), assuming the labels $(1/-1)$ as the targets of regression. Because, unlike the regular SVM, for the LS SVM, classifiers and regressors have the same form, training for the LS SVM and that for the LS SVR are the same. The only difference is whether the recognition error is calculated or the mean absolute error is calculated. Thus, a classification problem is easily converted into the associated regression problem, whose absolute error is continuous. Therefore, unlike the LS SVM, tie breaking rarely happens for the LS SVR.

The procedure for feature selection is based on incremental block addition and block deletion [3, 19]. Starting from the empty set, we repeat adding multiple features at a time to the set. We stop addition when the generalization ability of the set is no longer improved. Then from the set of selected features, we delete multiple features at a time until the generalization ability is not improved.

In Section 2, we discuss the idea of feature selection and selection criteria. Then in Section 3 we discuss the proposed methods based on incremental block addition and block deletion, and in Section 4, we show the results of computer experiments using two-class benchmark data sets including microarray data sets.

2 Idea of Feature Selection and Selection Criteria

For an SSPLF such as microarray data sets, the optimal set of features that realizes the generalization ability comparable to that of the original set of features is usually not so large. In such a situation, forward selection is faster than backward selection. Therefore, by forward selection we select a set of features whose generalization ability is comparable to that of the original set of features.

But because an added feature may become redundant after another feature is added, we perform backward selection for the set of features selected by forward selection.

To speedup feature selection, we use multiple feature addition (block addition) and multiple feature deletion (block deletion) and combine BABD with incremental feature selection.

To avoid frequent tie breaking in feature selection and feature ranking, we use a continuous selection criterion.

Let the decision function for a two class problem be

$$z = f(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b, \tag{1}$$

where $\mathbf{x}$ and z are the feature vector and the decision 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, and b is the bias term.

For M training input-output pairs $\{\mathbf{x}_i,\, y_i\}$ $(i = 1, \ldots, M)$, the LS SVM is given by

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \tag{2}$$

$$\text{subject to} \quad y_i\, f(\mathbf{x}_i) = 1 - \xi_i \qquad \text{for} \quad i = 1, \ldots, M, \tag{3}$$

where C is the margin parameter, $y_i = 1$ for Class 1 and -1 for Class 2, and ξ_i is the slack variable associated with $\mathbf{x}_i$.

Multiplying y_i to both sides of (3) and replacing $y_i\,\xi_i$ with ξ_i, we obtain

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \tag{4}$$

$$\text{subject to} \quad f(\mathbf{x}_i) = y_i - \xi_i \text{ for } i = 1, \ldots, M. \tag{5}$$

The above LS SVM is the same as the LS SVR.

In a wrapper method, we use the recognition error rate E_{C}. For the training data set it is given by

$$E_{\mathrm{C}} = \frac{1}{M}\sum_{i=1}^{M} e_i \quad \text{for} \quad e_i = \begin{cases} 0 & \text{for} \quad y_i\, f(\mathbf{x}_i) \geq 0, \\ 1 & \text{for} \quad y_i\, f(\mathbf{x}_i) < 0. \end{cases} \tag{6}$$

Because the recognition error rate is discrete, for an SSPLF, frequent tie breaking occurs for feature selection and feature ranking. Therefore, in [3] we proposed the following MM criterion:

$$E_{\mathrm{M_C}} = E_{\mathrm{C}} + r\, E_{\mathrm{M}}, \tag{7}$$

where r is a positive parameter and $r = 1/M$, and E_{M} is the mean margin error given by

$$E_{\mathrm{M}} = \frac{1}{M}\sum_{i=1}^{M} \xi_i \quad \text{where} \quad \xi_i = \begin{cases} 0 & \text{for} \quad y_i\, f(\mathbf{x}_i) \geq 1, \\ 1 - y_i\, f(\mathbf{x}_i) & \text{for} \quad y_i\, f(\mathbf{x}_i) < 1. \end{cases} \tag{8}$$

Because the LS SVM can also be used as a regressor, we consider the classification problem as a function approximation problem: we assume that the class labels (1/−1) are target values of a function approximation problem. Then, training the LS SVM is equivalent to training the associated LS SVR.

Thus, instead of (7), we consider using the mean absolute error:

$$E_{\mathrm{MAE}} = \frac{1}{M} \sum_{i=1}^{M} |y_i - f(\mathbf{x}_i)|. \tag{9}$$

Because $y_i = 1$ or -1, minimization of (9) leads to minimization of the recognition error. But model selection by cross-validation using (9) does not necessarily lead to the same model obtained by cross-validation using the recognition error or the MM criterion given by (7).

3 Feature Selection by Incremental Block Addition and Block Deletion

We use incremental BABD for function approximation discussed in [19]. The algorithm for pattern classification is essentially the same. In the following we explain incremental BABD.

In incremental BABD, initially we select a subset from the set of initial features and select features from the subset by BABD. Then we add features that are not yet processed to the set of selected features and repeat BABD until all the features are processed. This procedure is called one-pass incremental BABD.

By this method, important features may be discarded before the new features are added. To prevent this, we repeat one-pass BABD until no further improvement in the selection criterion is obtained. This procedure is called multi-pass incremental BABD.

Now we explain incremental BABD more in detail referencing the corresponding steps in Algorithm 1, which is an extension of iterative BABD discussed in [20].

Let $I^m = \{1, \ldots, m\}$ be the set of the original m features. Initially, we select the set of m' features, $I^{m'}$, from I^m as the initial set of features (Step 1), and calculate the MAE for $I^{m'}$, $E^{m'}$, by cross-validation. This is used as the threshold of feature selection for $I^{m'}$, $T^{m'}$ (Step 2):

$$T^{m'} = E^{m'}. \tag{10}$$

By BA, we iterate feature ranking and feature addition until

$$E^j \leq T^{m'} \leq E^j + \varepsilon_{\mathrm{M}} \tag{11}$$

is satisfied, where ε_{M} is a positive value, I^j is the set of selected j features, $j \leq m'$, and $I^j \subseteq I^{m'}$. The right-hand side inequality is to control the number of selected features, and as the value of ε_{M} is decreased, the number of selected features is increased. Then if $E^j < T^{m'}$, we update the threshold by

$$T^{m'} = E^j. \tag{12}$$

In the feature ranking we rank features in $I^{m'}$ in the ascending order of MAEs, which are evaluated by temporarily adding a feature to the set of selected features. Then we add, to the set of selected features, from the top ranked to the 2^kth ranked features, where $k = 1, \ldots, 2^A$ and A is a user defined parameter, and evaluate the MAE by cross-validation (Step 3). If the minimum MAE for $k \in \{1, \ldots, 2^A\}$ is smaller than or equal to $T^{m'}$, we permanently add the associated features, and update the threshold. If the right-hand side inequality in (11) is satisfied, finish BA. If not, repeat BA. Otherwise, if the minimum MAE is smaller than that at the previous BA step, we permanently add the associated features, update the threshold, and repeat BA. Otherwise, we add the top ranked feature and repeat BA (Step 4).

Because redundant features may be added by BA, we delete these features by BD repeating feature ranking and deletion of features.

For each feature in I^j we evaluate the MAE by cross-validation temporarily deleting the feature (Step 5).

We generate set S^j that includes features whose MAE is not larger than $T^{m'}$. If S^j is empty we terminate BD. If only one element is in S^j, delete this feature and iterate BD (Step 6). Otherwise, we temporarily delete all the features in S^j and evaluate the MAE by cross-validation. If it is not larger than $T^{m'}$, we permanently delete these features and update j, and repeat BD (Step 7). If not, we rank features in S^j and temporarily delete the top half and evaluate the MAE by cross-validation. We repeat this until feature deletion is succeeded (Step 8).

After BD is succeeded, E^j for the resulting set of features I^j satisfies

$$E^j \leq T^{m'}. \tag{13}$$

Then we update the threshold by $T^{m'} = E^j$ and repeat BD.

The above procedure guarantees that the MAE for the selected features is not larger than that for $I^{m'}$, i.e., $E^j \leq E^{m'}$.

Let i_{Inc} be the number of features that are added at the incremental step. We add i_{Inc} features from $I^m - I^{m'}$ to I^j,

Let the resulting set of features be $I^{j+i_{\text{Inc}}}$. Then the MAE for $I^{j+i_{\text{Inc}}}$ is $E^{j+i_{\text{Inc}}}$. We set the threshold $T^{m'+i_{\text{Inc}}}$ by $T^{m'+i_{\text{Inc}}} = E^{j+i_{\text{Inc}}}$. Here, we must notice that

$$T^{m'+i_{\text{Inc}}} \leq T^{m'}. \tag{14}$$

is not always satisfied.

We iterate the above BABD for $I^{j+i_{\text{Inc}}}$. Let the resulting set of features be I^o, where $o \leq j + i_{\text{Inc}}$. Then

$$E^o \leq T^{m'+i_{\text{Inc}}} \tag{15}$$

is satisfied. If (14) is satisfied,

$$E^o \leq T^{m'} \tag{16}$$

is also satisfied. But otherwise, there is no guarantee that the above inequality is satisfied.

If (16) is satisfied, we repeat BABD adding the variables not processed. Otherwise, we consider that the BABD for this step failed and undo the feature selection at this step; namely, we restart BABD with threshold $T^{m'}$ and I^j, and add remaining features to I^j.

In one-pass incremental BABD, we repeat the BABD until all the variables are processed. In multi-pass incremental BABD, to reduce the absolute error further, we repeat the above procedure until the selection criterion does not change (Step 9).

Algorithm 1 (Incremental BABD).

Initialization

Step 1 Set $I^{m'}(\subseteq I^m)$, $j = 0$, and $E^j = \infty$.

Block Addition

Step 2 Calculate $E^{m'}$ for $I^{m'}$. Set $T^{m'} = E^{m'}$.

Step 3 Add feature i in $I^{m'} - I^j$ temporarily to I^j, calculate $E^j_{i_{\text{add}}}$, where i_{add} denotes that feature i is temporarily added, and generate feature ranking list V^j. Set $k = 1$.

Step 4 Calculate E^{j+k} $(k = 1, 2^1, \ldots, 2^A)$. If $E^{j+k} < T^{m'}$, set $j \leftarrow j+k, T^{m'} \leftarrow E^j$. And if $T^{m'} \leq E^j + \varepsilon_{\text{M}}$, go to Step 5; if not, go to Step 3. Otherwise, if $E^{j+k} < E^j$ is satisfied, set $j \leftarrow j + k$ and go to Step 3. Otherwise, if $E^j \leq T^{m'}$, go to Step 5; otherwise, set $j \leftarrow j + 1, T^{m'} \leftarrow E^j$ and go to Step 3.

Block Deletion

Step 5 Delete temporarily feature i in I^j and calculate $E^j_{i_{\text{del}}}$, where i_{del} denotes that feature i is temporarily deleted.

Step 6 Calculate S^j. If S^j is empty, $I^o = I^j$ and go to Step 9. If only one feature is included in S^j, set $I^{j-1} = I^j - S^j$, set $j \leftarrow j - 1$ and go to Step 5. If S^j has more than two features, generate V^j and go to Step 7.

Step 7 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 $E^{j'} > T^{m'}$, go to Step 8. Otherwise, update j with j', $T^{m'} \leftarrow E^{j'}$, and go to Step 5.

Step 8 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 $E^{j'} \leq T^{m'}$, delete features in $V^{'j}$ and go to Step 5 updating j with j' and $T^{m'}$ with $E^{j'}$. Otherwise, update V^j with $V^{'j}$ and iterate Step 8 until $E^{j'} < T^{m'}$ is satisfied.

Step 9 If E^o is larger than $T^{m'}$ in the previous step, undo current BABD. If some features in I^m are not added, $I^{m'} = I^o \cup I^{i_{\text{Inc}}}$, $m' \leftarrow o + i_{\text{Inc}}$, $j = 0$, $E^j = \infty$, and go to Step 2. Otherwise, if one-pass, terminate feature selection; otherwise if $T^{m'}$ decreases from previous $T^{m'}$, go to Step 1. If not, stop feature selection.

4 Performance Evaluation

Because feature selection based on the E_{C} criterion performed poorly for a large number of features [3], in this section, we compare the MAE criterion with the

MM criterion and incremental BABD with batch BABD using two kinds of data sets: data sets with small numbers of features and microarray data sets with large numbers of features. We set $A = 5$ and $\varepsilon_{\mathrm{M}} = 10^{-5}$ as in [3]. In incremental feature selection, we set $m' = i_{\mathrm{Inc}}$ and add features from the first to the last.

4.1 Data Sets with Small Numbers of Features

We used the ionosphere and WDBC data sets [21]. We divided each data set randomly into training and test data sets and generated 20 pairs.

For these data sets, in [3] we showed that the recognition rates of the test data sets and the numbers of selected features by batch BABD were comparable to those shown in [2, 8, 13]. Therefore, here, we only compare the proposed method with batch BABD.

We used the RBF kernels: $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^{\top}(\mathbf{x})\, \boldsymbol{\phi}(\mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2/m)$, where γ is a positive parameter. Using all the features we determined the γ and C values by fivefold cross-validation changing $\gamma = \{0.001, 0.01, 0.5, 1.0, 5.0, 10, 15, 20, 50, 100\}$ and $C = \{1, 10, 50, 100, 500, 1000, 2000\}$. During and after feature selection we fixed the γ and C values to the determined values.

We measured the average feature selection time per data set using a personal computer with 3GHz CPU and 2GB memory.

Table 1 shows the results for the ionosphere and WDBC data sets. The upper part for each data set shows the result for the MM criterion and the lower part, the MAE criterion. In the “Data (Tr/Te/In)” column, the first row of each data set shows the name of the data set followed by the numbers of training data, test data, and features. The first column also includes performance with the standard deviation using all the features: the recognition rates for the test data sets and those for the validation data sets in the parentheses. For the MAE criterion, MAEs are shown in the parentheses.

In the second column, MM denotes batch BABD with the MM criterion and MAE, that with the MAE criterion. And for instance “20” denotes the one-pass incremental BABD with 20 features added, and “m” in 20m denotes the multi-pass incremental BABD. The third column shows the recognition (approximation) performance after feature selection. And the fourth and the fifth columns show the number of selected features and the feature selection time, respectively.

For each performance measure, the best performance is shown in bold face.

From the table, except for two cases by one-pass incremental BABD, the recognition rates (MAEs) by cross-validation were improved by feature selection, but for the test data sets, the recognition rates were decreased. This was caused by overfitting.

Now compare the MM and MAE criteria. Using all the features, the recognition rates of the test data sets by the MAE criterion were better for both data sets. This means that different γ and C values were selected by cross-validation. But the differences including those after feature selection were small.

As for the effect of incremental BABD, although multi-pass incremental BABD improved the recognition rates (MAEs) by cross-validation, in some cases

Table 1. Comparison of selection methods

Data (Tr/Te/In)	Method	Test Rate (CV Rate/MAE)	Selected	Time [s]
Ionosphere (281/70/34)	MM	**93.93**±2.59(97.10±0.80)	15.20±5.0	**14.70**±2.12
94.21±1.89(95.57±0.67)	20	92.64±3.07(96.57±0.84)	13.9±3.3	15.55±2.82
	20m	92.79±2.73(97.12±0.55)	13.1±3.7	37.80±13.06
	10	91.86±3.50(96.51±0.86)	11.3±3.1	17.70±1.31
	10m	92.29±3.04(**97.17**±0.71)	11.4±3.3	45.15±17.36
	1	91.29±2.78(95.14±1.51)	**5.7**±1.7	35.90±5.84
	1m	91.71±3.55(96.05±1.63)	7.2±2.3	142.9±66.60
95.29±2.31(0.2640±1.16)	MAE	**94.21**±2.57(0.2278±0.0120)	13.5±2.4	**13.75**±1.41
	20	93.14±3.25(0.2315±0.0134)	10.7±2.7	14.15±1.42
	20m	93.43±3.33(**0.2267**±0.0127)	10.9±2.6	33.95±9.86
	10	92.21±2.77(0.2321±0.0145)	7.8±2.9	15.80±1.29
	10m	92.14±3.11(0.2274±0.0145)	8.6±3.2	34.45±9.46
	1	91.50±3.85(0.2406±0.0140)	**4.8**±0.7	32.90±3.99
	1m	91.36±3.51(0.2344±0.0112)	5.4±1.2	91.90±41.29
WDBC(455/114/30)	MM	**97.11**±1.15(98.41±0.33)	16.6±4.4	**40.45**±8.99
97.41±0.98(98.09±0.34)	20	97.02±1.09(98.32±0.38)	14.4±2.7	41.50±6.34
	20m	96.93±1.13(**98.57**±0.24)	14.7±3.6	100.5±27.00
	10	97.06±1.01(98.26±0.35)	13.2±3.4	42.90±4.38
	10m	96.71±1.24(98.56±0.34)	12.7±3.4	126.5±25.51
	1	96.14±1.02(98.01±0.38)	**6.6**±1.4	114.3±9.81
	1m	95.96±1.16(98.33±0.34)	7.5±2.0	381.7±134.0
97.72±1.22(0.2335±0.0067)	MAE	96.14±1.40(0.1622±0.0058)	5.3±1.0	**30.45**±2.42
	20	96.10±1.43(0.1622±0.0058)	5.2±1.0	34.20±2.27
	20m	95.92±1.62(0.1619±0.0058)	5.0±1.2	62.10±9.72
	10	96.10±1.43(0.1622±0.0058)	5.2±1.0	35.55±2.31
	10m	95.92±1.62(0.1619±0.0058)	5.0±1.2	71.25±10.50
	1	**96.19**±1.60(0.1617±0,0051)	**4.2**±1.0	86.05±8.23
	1m	96.05±1.58(**0.1616**±0.0052)	4.4±1.2	366.7±810.1

one-pass incremental BABD showed better recognition rates for the test data sets. Except for the WDBC data set with the MAE criterion, the recognition rates for the test data sets decreased as $i_{\rm Inc}$ was decreased.

The numbers of selected features decreased as $i_{\rm Inc}$ was decreased and they were minimum when $i_{\rm Inc} = 1$ both for one- and multi-pass feature selection.

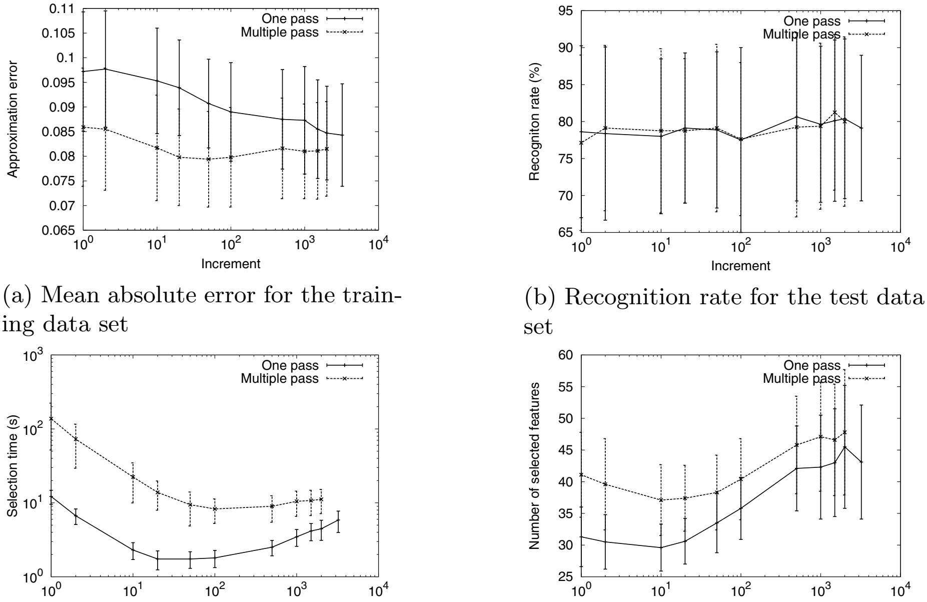
Feature selection time by batch BABD was shortest for all four cases. This means that because the numbers of features were not so large, incremental feature selection did not contribute in speeding up feature selection.

4.2 Microarray Data Sets

We compared BABD with the MM criterion and BABD with the MAE criterion for microarray data sets (see [22] for details of data sets), each of which consisted of 100 pairs of training and test data sets. Because microarray data sets have a small number of samples and a large number of features, they are linearly separable and overfitting occurs easily. Therefore, we used linear kernels: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$ and fixed $C = 1$.

To measure feature selection time, we used a personal computer with 3.4GHz CPU and 16GB memory.

To determine the number of added features (i_{Inc}), we carried out incremental BABD with the MAE criterion for the breast cancer data set (1) changing i_{Inc}. Figure 1 shows the result for one- and multi-pass BABD. As shown in Fig. (a), the MAE for the training data by multi-pass BABD was better than that by one-pass BABD. But there was not much difference in the recognition rates of the test data by one- and multi-pass BABD (Fig. (b)), although by one-pass BABD the feature selection time was shorter and the number of selected features was smaller. From Figs. (b) and (c), we set $i_{\mathrm{Inc}} = 500$ in the following experiments.



(a) Mean absolute error for the training data set

(b) Recognition rate for the test data set

(c) Feature selection time

(d) Number of selected features

Fig. 1. Feature selection for the breast cancer (1) data set

Table 2 shows the results. In the table if 100% recognition rates were obtained, they are not shown. The triplet in the "Summary" row shows from the left the numbers that the best/second best/third performance were obtained. In the "Selected" and "Time" columns, the average value with the asterisk shows that it is statistically significant between the values for the MM criterion and one-pass incremental method with $i_{\mathrm{inc}} = 500$ by the Welch t-test with the confidence interval of 95%.

Comparing the results for the MM and MAE criteria, there is not much difference of the recognition rates of the test data sets (statistically comparable). In some cases (e.g., the breast cancer (3) and hepatocellular carcinoma data sets),

Table 2. Performance comparison of incremental BABD and batch BABD

Data (Tr/Te/In)	Method	Test Rate (CV Rate/MAE)	Selected	Time [s]
B. cancer (1) (14/8/3226)	MM	**80.50**±11.36	**40.5***±11.9	4.04±2.08
73.87±11.47 (76.50±7.09)	500	78.25±10.55	44.7±8.3	**2.20***±0.57
	500m	79.25±10.93	47.7±7.3	6.71±2.58
73.87±11.47(0.6215±0.0709)	MAE	79.12±9.85(0.0843±0.0104j	43.1±9.0	5.85±1.88
	500	**80.63**±11.37(0.0875±0.0101)	**42.1**±6.7	**2.51***±0.59
	500m	79.25±12.15(**0.0816**±0.0102)	45.8±7.7	8.98±3.52
B. cancer (2) (14/8/3226)	MM	**83.38**±13.12	**43.9***±12.4	4.34±2.08
91.88±10.21(83.50±7.93)	500	82.63±13.45	50.1±7.9	**2.39***±0.63
	500m	82.13±13.02	55.0±9.7	7.45±2.60
91.88±10.21(0.6356±0.0729)	MAE	82.00±11.64(0.0950±0.0138)	49.9±12.8	7.16±2.35
	500	83.50±12.10(0.0982±0.0117)	**47.9**±9.3	**3.21***±0.65
	500m	**83.87**±12.03(**0.0904**±0.0117)	52.9±10.8	10.21±3.69
B. Cancer (3) (78/19/24188)	MM	63.37±9.93	**70.7***±15.5	847.0±358.0
67.32±9.42(66.96±4.58)	500	62.95±9.29	82.6±8.5	**555.4***±74.74
	500m	**64.58**±10.28	84.6±8.1	1999±673.8
67.32±9.42(0.8167±0.0474)	MAE	**63.47**±10.39(**0.1547**±0.0122)	115.8±15.8	3557±915.2
	500	62.05±8.82(0.1701±0.0095)	**94.3***±12.0	**1463***±107.4
	500m	62.79±11.25(0.1576±0.0117)	97.7±11.8	5595±1724
B. cancer (s) (14/8/3226)	MM	67.00±13.17	**39.5***±12.4	3.85±2.09
69.12±10.82(72.79±9.30)	500	**68.87**±11.92	46.5±7.9	**2.31***±0.69
	500m	68.75±12.69	50.9±7.5	7.47±2.61
69.13±10.82(0.7248±0.0816)	MAE	67.37±13.33(0.1051±0.0149)	46.0±10.5	6.56±2.39
	500	67.50±12.75(0.1110±0.0139)	**43.9**±7.8	**3.03***±0.67
	500m	**69.13**±13.16(**0.1012**±0.0121)	49.3±8.8	10.25±4.30
C. cancer (40/20/2000)	MM	**81.05**±6.68(99.53±1.10)	91.8±35.7	47.22±43.42
79.64±6.54(79.67±6.21)	500	80.82±7.06(99.70±0.89)	**84.1**±23.4	**30.17***±11.83
	500m	80.86±7.05(**99.95**±0.35)	87.0±16.2	73.55±45.44
79.64±6.54(0.6819±0.0880)	MAE	**81.82**±6.49(0.2423±0.0319)	**66.1***±19.9	28.17±10.93
	500	81.50±6.40(0.2357±0.0268)	71.1±15.6	**22.68***±4.68
	500m	81.23±6.88(**0.2223**±0.0275)	76.9±13.7	72.78±26.70
H. Carcinoma (33/27/7129)	MM	64.63±7.45	**53.0***±14.4	42.21±20.45
67.96±7.00(66.21±7.34)	500	64.70±7.81	61.0±8.8	**26.52***±3.84
	500m	**64.74**±7.80	66.4±8.2	84.05±28.07
67.96±7.00(0.8263±0.0708)	MAE	63.56±8.14(0.1538±0.0196)	65.3±13.9	101.5±34.46
	500	**65.04**±8.24(0.1601±0.0192)	**63.5**±9.6	**45.64***±5.34
	500m	64.78±7.99(**0.1480**±0.0176)	66.4±9.0	153.3±56.05
H. glioma (21/29/12625)	MM	70.07±8.46	**49.6***±13.6	78.66±36.30
75.59±7.58(72.71±10.23)	500	70.38±8.39	61.6±9.7	**22.66***±3.03
	500m	**70.52**±8.58	66.3±9.4	79.46±27.45
75.59±7.58(0.7718±0.0124)	MAE	**71.17**±8.63(0.1364±0.0232)	**52.5**±12.9	131.1±38.38
	500	70.10±8.60(0.1409±0.0217)	54.6±10.2	**30.74***±4.26
	500m	70.41±9.13(**0.1286**±0.0192)	58.9±9.2	110.6±40.56
Leukemia (38/34/7129)	MM	94.38±3.88	**47.9***±12.2	43.76±20.11
94.44±4.70(92.45±3.32)	500	**94.41**±3.87	56.6±8.6	**25.93***±4.94
	500m	94.29±3.90	62.3±7.3	74.31±24.85
94.44±4.70(0.4866±0.0392)	MAE	94.06±3.58(0.0883±0.0129)	66.3±14.5	126.3±42.17
	500	94.32±3.80(0.0896±0.0110)	**62.1***±10.0	**66.78***±7.83
	500m	**94.59**±3.83(**0.0829**±0.0095)	64.9±10.5	196.6±59.68
P. cancer (102/34/12600)	MM	**84.65**±6.08(99.18±1.64)	350.5±243.8	33970±34855
87.03±4.56(88.52±2.27)	500	83.74±6.75(99.77±0.44)	**251.3***±134.1	**9625***±5279
	500m	84.29±6.54(99.88±0.32)	288.2±115.6	39593±36943
87.03±4.56(0.8757±0.0429)	MAE	80.68±6.30(0.4039±0.0385)	**105.4***±26.4	2490±1037
	500	82.62±6.18(0.3988±0.0235)	135.3±21.8	**1974***±176.6
	500m	**83.38**±6.45(**0.3662**±0.0213)	153.7±20.3	9157±3141
	MM	7/2/9	10/4/4	0/17/1
Summary	500	4/8/6	8/10/0	18/0/0
	500m	7/8/3	0/4/14	0/1/17

the MAE criterion selected more features and thus feature selection time was longer. But for the colon cancer data sets, the opposite was true. The above results confirm that the MAE criterion is comparable to the MM criterion.

From the "Summary" rows, we found that multi-pass incremental BABD showed the best recognition rates for the test data sets, but the numbers of selected features were the largest and also feature selection was slowest. The recognition rates by one-pass incremental BABD were comparable with those by batch BABD and feature selection was the fastest, but the numbers of selected features were the second to batch BABD. Therefore, one-pass BABD can be an alternative to the batch BABD.

The reason why one-pass BABD performed well for the microarray data sets although it was not for the ionosphere and WDBC data sets is as follows: because the numbers of features are very large and the number of training samples are very small, there exist many alternative subsets of features that realize best generalization performance. In addition, because the number of added features was usually much larger than the number of selected features, during incremental BABD, optimal features were not deleted, or even if deleted, alternative features remained.

5 Conclusions

In this paper, we proposed using the MAE (mean absolute error) criterion in selecting features of small sample problems with a large number of features. Setting class labels $(1/-1)$ as the targets of regression, we train the least squares SVM and calculate the MAE. Because the MAE is continuous, tie breaking, which is a problem for a discrete criterion, does not occur frequently. Therefore, feature selection is stabilized.

We evaluate the MAE criterion by incremental block addition and block deletion (BABD) using the microarray data sets. The results show that the MAE criterion is comparable with the MM criterion, which is the weighted sum of the recognition error rate and the average margin error, and that the one-pass incremental BABD is comparable in generalization abilities to batch BABD with faster feature selection.

Acknowledgment. This work was supported by JSPS KAKENHI Grant Number 25420438.

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., Abe, S.: Feature selection by block addition and block deletion. In: Mana, N., Schwenker, F., Trentin, E. (eds.) ANNPR 2012. LNCS, vol. 7477, pp. 48–59. Springer, Heidelberg (2012)

4. 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)
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. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proc. ICML 1998, pp. 82–90 (1998)
8. Neumann, J., Schnörr, C., Steidl, G.: Combined SVM-based feature selection and classification. Machine Learning 61(1-3), 129–150 (2005)
9. Stearns, S.D.: On selecting features for pattern classifiers. In: Proc. ICPR, pp. 71–75 (1976)
10. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15(11), 1119–1125 (1994)
11. Zhang, T.: Adaptive forward-backward greedy algorithm for sparse learning with linear models. In: NIPS 21, pp. 1921–1928 (2009)
12. 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)
13. Liu, Y., Zheng, Y.F.: FS_SFS: A novel feature selection method for support vector machines. Pattern Recognition 39(7), 1333–1345 (2006)
14. 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)
15. Liu, H., Setiono, R.: Incremental feature selection. Applied Intelligence 9(3), 217–230 (1998)
16. Perkins, S., Lacker, K., Theiler, J.: Grafting: Fast, incremental feature selection by gradient descent in function space. J. Machine Learning Research 3, 1333–1356 (2003)
17. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition 39(12), 2383–2392 (2006)
18. Bermejo, P., Gamez, J.A., Puerta, J.M.: Speeding up incremental wrapper feature subset selection with naive Bayes classifier. Knowledge-Based Systems 55, 140–147 (2014)
19. Abe, S.: Incremental input variable selection by block addition and block deletion. In: ICANN 2014 (accepted, 2014)
20. Abe, S.: Feature selection by iterative block addition and block deletion. In: Proc. SMC 2013, pp. 2677–2682 (2013)
21. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
22. Abe, S.: Support Vector Machines for Pattern Classification, 2nd edn. (2010)