Incremental Input Variable Selection by Block Addition and Block Deletion

Abe, Shigeo

Incremental Input Variable Selection by Block Addition and Block Deletion

Shigeo Abe

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

Abstract. In selecting input variables by block addition and block deletion (BABD), multiple input variables are added and then deleted, keeping the cross-validation error below that using all the input variables. The major problem of this method is that selection time becomes large as the number of input variables increases. To alleviate this problem, in this paper, we propose incremental block addition and block deletion of input variables. In this method, for an initial subset of input variables we select input variables by BABD. Then in the incremental step, we add some input variables that are not added before to the current selected input variables and iterate BABD. To guarantee that the cross-validation error decreases monotonically by incremental BABD, we undo incremental BABD if the obtained cross-validation error rate is worse than that at the previous incremental step. We evaluate incremental BABD using some benchmark data sets and show that by incremental BABD, input variable selection is speeded up with the approximation error comparable to that by batch BABD.

1 Introduction

Input variable selection for regression is to select a set of input variables deleting irrelevant or redundant input variables from an original set of input variables. This is an important step in realizing a regressor with high generalization ability. In the following, we simply say variables instead of input variables, if there is no confusion. Because variable selection methods are usually applicable to feature selection in pattern recognition, variables and features are used interchangeably.

According to the selection criterion, the variable selection methods are classified into wrapper methods, which use an approximation error by regressors and filter methods, which use other selection criteria. Since the introduction of support vector machines (SVMs) [1–3], imbedded methods [4] are proposed, in which the variable selection criterion is included in the objective function of SVMs.

In wrapper or filter methods, variables are selected by forward selection, in which informative variables are added step by step, or by backward selection, in which unnecessary variables are deleted step by step. As their variant, forward selection and backward selection are combined [5–7].

To speed up variable selection, incremental selection has been proposed [8–11]. In [8], for the randomly selected set of training samples, feature selection is performed. Then if the inconsistency occurs in the remaining training data, in that the features of samples of different classes match, these samples are added to the randomly selected samples and repeat feature selection until no inconsistency is found. In [9], the L_0 feature selection criterion is added to the objective function. Starting from a small set of selected features, at each iteration, the feature that is estimated to improve the objective function value most is added to the selected feature set, and the objective function excluding the feature selection criterion is improved by the steepest descent method. In [10], initially all the features are ranked, and then sequential forward selection is performed using the ranked features. In [12], to speed-up wrapper methods, multiple variables are added by forward selection (block addition), then multiple variables are deleted by backward selection (block deletion).

To speed up BABD, in this paper, we propose incremental BABD. Initially, we calculate the approximation error by cross-validation using the subset of the initial variable set and set it as the threshold of variable selection. Then, we select variables from the subset by BABD. If the approximation error lower than the threshold is obtained, we update the threshold. We add subset of variables to the set of selected variables and do variable selection by BABD. But if the obtained threshold is worse than that at the previous step, we undo the variable selection. We iterate the above procedure, until all the variables are processed. We evaluate this incremental BABD using some benchmark data sets.

In Section 2, we discuss the idea of incremental BABD and its algorithm and in Section 3, we show the results of computer experiments using benchmark data sets.

2 Incremental BABD

2.1 Idea

In selecting a set of variables from a large number of variables, forward selection is more efficient than backward selection. But variables are selected only considering the relation among selected variables and the candidate variable. While by backward selection, the variable that does not deteriorate the selection criterion the least among the remaining variables is deleted. Therefore, forward selection is less stable than backward selection. To alleviate such a problem, we have proposed BABD. In BA, multiple variables are added according to the ranked variables until the selected set realizes the approximation error smaller than or equal to that for the set of original variables. Then by BD, multiple variables are deleted that do not increase the approximation error.

In BA, variables are ranked according to the approximation errors, which are calculated by temporarily adding one variable to the selected set of variables. Therefore, if the number of variables is large, the ranking procedure takes time. To alleviate the computation of ranking, we consider incremental variable selection. Initially, we start with a subset of original variables, and select variables

by BABD for the subset. Then we add remaining variables to the set and iterate the BABD until all the variables are processed. If we replace BABD with BD in the above procedure, incremental BD is also possible.

2.2 Algorithm

Now we explain incremental BABD more in detail. (Please see [12] for details of BABD.)

Let $I^m = \{1, \ldots, m\}$ be the set of variables, where m is the number of variables. We select the subset of I^m, I^j, as the initial set of variables, where j is the number of initial variables. We calculate the approximation error for I^j, E^j, by cross-validation and set the threshold of variable selection, T^j:

$$T^j = E^j. \tag{1}$$

By BA, we first rank variables whose indices are in I^j in the ascending order of approximation errors, which are evaluated by adding a variable to the set of selected variables temporarily, and add multiple variables to the selected set from the top ranked variables that decrease the approximation error most. For the variables in I^j that are not selected, we iterate the above procedure until

$$E^{j'} \leq T^j. \tag{2}$$

is satisfied, where $I^{j'}$ is the selected set of variable indices, j' is the number of selected variables, $j' \leq j$, and $I^{j'} \subseteq I^j$. Then we set the threshold by

$$T^{j'} = E^{j'}. \tag{3}$$

Further by BD first we rank variables whose indices are in $I^{j'}$, according to the approximation errors, which are calculated by deleting a variable temporarily. And we delete multiple variables that decrease the approximation error the most. We iterate the above variable ranking and deletion until no further variables are deleted. Let the resulting set of variable indices be I^k, where k is the number of selected variables. The approximation error E^k for I^k satisfies

$$E^k \leq T^{j'}. \tag{4}$$

Then we update the threshold by $T^k = E^k$.

According to the above procedure, the approximation error for the selected variables is not larger than that for I^j, i.e., $E^k \leq E^j$.

Now we add i_{Inc} indices from $I^m - I^j$ to I^k, where i_{Inc} is the number of variables that are added at the incremental step. The resulting set of indices be $I^{k+i_{\mathrm{Inc}}}$. The approximation error for $I^{k+i_{\mathrm{Inc}}}$ is $E^{k+i_{\mathrm{Inc}}}$ and we set the threshold $T^{k+i_{\mathrm{Inc}}}$ by $T^{k+i_{\mathrm{Inc}}} = E^{k+i_{\mathrm{Inc}}}$. We must notice that

$$T^{k+i_{\mathrm{Inc}}} \leq T^k. \tag{5}$$

is not always satisfied.

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

$$E^o \leq T^{k+i_{\text{Inc}}} \tag{6}$$

is satisfied. But there is no guarantee that the following inequality is satisfied:

$$E^o \leq T^k \tag{7}$$

If (7) is satisfied, we repeat BABD adding the variables not processed. If it is not satisfied, we consider that the BABD for this step failed and undo the variable selection at this step; namely, we restart BABD with threshold T^k and I^k, and add remaining indices of variables to I^k.

We repeat the BABD until all the variables are processed. This is a one-pass incremental variable selection. To reduce the approximation error further, we may repeat the above procedure until the selected variable set does not change. But it will increase the computation time. Therefore, in the following we only consider one-pass incremental BABD.

3 Performance Evaluation

Because BABD has been compared with other methods in [12, 13], and shown to be comparable to or better than other methods, in this section, we compare incremental BABD with batch BABD.

3.1 Evaluation Conditions

In performance evaluation we used the mean absolute error (MAE) for the validation data set evaluated by cross-validation using least squares support vector regressors (LS SVRs). We used a personal computer (3GHz, 2GB memory, Windows XP operating system) in measuring variable selection time.

The primal problem of the LS SVR is given by

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{C}{2}\sum_{i=1}^{M} \xi_i^2 \tag{8}$$

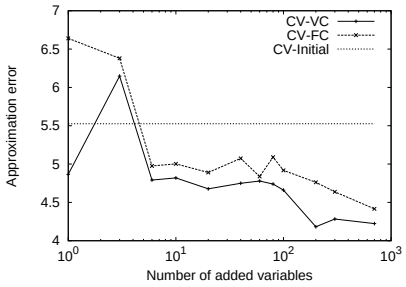
$$\text{subject to} \quad y_i = \mathbf{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \xi_i \quad \text{for} \quad i = 1, \ldots, M, \tag{9}$$

where $\mathbf{w}$ is the coefficient vector of the hyperplane, C is the margin parameter, $\boldsymbol{\phi}(\mathbf{x})$ is the mapping function that maps $\mathbf{x}$ into the feature space, and M is the number of training data. In training the LS SVR, we solve the set of linear equations that is derived by transforming the primal problem into the dual problem. As a kernel function, we use linear kernels: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$ or RBF kernels: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2/m)$, where $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^T(\mathbf{x})\,\boldsymbol{\phi}(\mathbf{x}')$, γ is a parameter for determining the spread of the radius, and m is the number of variables.

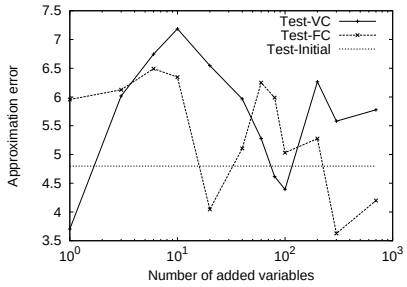
We determined the initial MAE by fivefold cross-validation changing $\gamma = \{0.001, 0.01, 0.5, 1.0, 5.0, 10, 15, 20, 50, 100\}$ and $C = \{1, 10, 50, 100, 500, 1000, 2000\}$. To reduce the computational cost of training the LS SVR during variable selection, fixing the kernel parameter value, we optimize the margin parameter value by cross-validation. To reduce the computation cost further, we can fix the margin parameter value.

To determine whether we should change the C value during variable selection, we carried out variable selection for the orange juice data [16] using RBF kernels. Figure 1 shows the result. For the validation data set, the MAEs by the fixed C value (FC) was higher than those by the variable C value (VC) for the change of number of added variables. In some cases, the MAEs were larger than initial MAE using all the variables (see Fig. (a)).

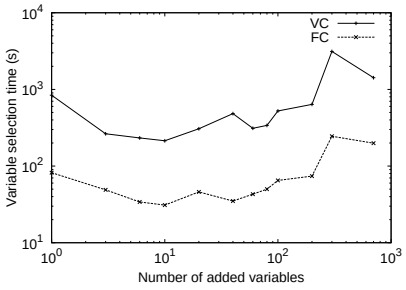
For the test data set, depending on the number of added variables, MAEs by VC were not always lower than those by FC (see Fig. (b)) but variable selection time by VC was much longer than by FC (Fig. (c)). The numbers of selected variables did not vary much between the two but as the number of added variables was increased, the number of selected variables was also increased (see Fig. (d)). According to the above results, because VC did not always give better results than FC, we used FC in the following study.



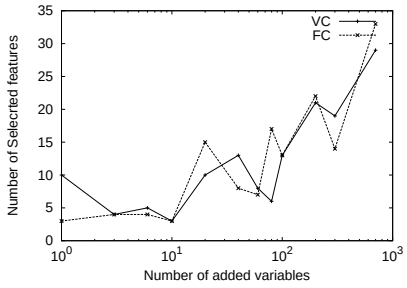
(a) Approximation error by cross-validation



(b) Approximation error for the test data set



(c) Variable selection time



(d) Number of selected features

Fig. 1. Variable selection for the orange juice data set

3.2 Experimental Results

We evaluated Incremental BABD using the six benchmark data sets listed in Table 1. The first column shows the benchmark data sets with the numbers of variables, training data, and test data. If the data set was not divided into training and test data sets, the corresponding number of test data is shown in "−." For the first three data sets, we randomly divided the set into training and test data sets with the ratio of 3 to 2 and generated 20 files for triazines and pyrimidines data sets and 40 files for phenetylamines data set. For the orange juice, breast cancer, and leukemia data sets, we combined the training and test data sets, and randomly generated 100 training and test data sets, each with the numbers of training and test data equal to the original numbers. The breast cancer and leukemia data sets are classification problems but we treated them as function approximation problems considering the $+1/-1$ labels as target values. For these data sets we used linear kernels with $C = 1$ because overfitting occurs for the C value larger than 1. For the other data sets, we used RBF kernels.

We calculated MAEs for the original variables, those after variable selection, and measured the feature selection time. In the first column of the table, the MAEs with the standard deviations for the test data sets using the original variables are shown. In the parentheses those for the validation data sets are shown.

In the "Method" column, BABD and BD are original variable selection methods without incremental variable selection (i.e., batch BABD/BD). The number below BABD/BD shows i_{Inc}, i.e., the number of variables that were added in incremental variable selection.

The third to fifth columns list the results. The smallest values among BABD/BD and their incremental versions are shown in bold.

For cases where the MAEs after the variable selection were larger than the associated initial MAEs, we performed the Welch t-test with a 5% significance level. If the MAEs and the standard deviations after variable selection are statistically inferior, we add asterisk to the associated values. For the validation data set, inferior MAEs and the standard deviations occurred only for the orange juice data set by incremental BABD/BD with $i_{\text{Inc}} = 1$. As discussed in the previous section, improvement of MAEs is guaranteed for batch BABD/BD but not for incremental BABD/BD. But the experimental results show that in most cases one-pass is enough to reduce the MAE by incremental variable selection.

For the orange juice and leukemia test data sets, MAEs by incremental variable selection with $i_{\text{Inc}} = 1$ were statistically inferior to the initial MAEs. This happened for the phenetylamines data set by BD with $i_{\text{Inc}} = 100$. Except for these cases, incremental BABD/BD performed better than or comparable to batch BABD/BD.

The fourth column shows the average number of selected variables. The numbers of selected variables were decreased by adopting incremental variable selection and usually the minimum number of variables was obtained for $i_{\text{Inc}} = 1$.

The final column shows the variable selection time. For the first three data sets, there is not much difference in selection time because the number of features

and/or the number of data is small. But for the remaining three data sets, incremental training was faster except for some cases with $i_{\mathrm{Inc}} = 1$.

Table 1. Performance comparison of incremental BABD and batch BABD

Data	Method	Average Error	Selected	Time [s]
Triazines (60/186/—) [14]	BABD	0.0036±0.0039(0.0019±0.0011)	4.5±2.3	**2.30**±0.84
0.0052±0.0036(0.0070±0.0023)	10	**0.0032**±0.0030(0.0018±0.0011)	3.9±1.6	3.30±0.64
	1	**0.0032**±0.0024(**0.0015**±0.0010)	**3.0**±1.0	6.60±0.86
	BD	0.0034±0.0033(**0.0016**±0.0010)	5.7±3.2	**2.05**±0.59
	10	**0.0025**±0.0020(0.0018±0.0010)	4.2±2.0	2.35±0.57
	1	0.0033±0.0020(**0.0016**±0.0013)	**3.1**±0.9	4.25±0.77
Pyrimidines (27/74/—) [14]	BABD	0.0191± 0.0112(**0.0112**±0.0089)	2.3±1.3	0.25±0.43
0.0309±0.0093(0.037±0.012)	10	0.0193±0.0110(0.0120±0.0091)	2.2±1.2	**0.20**±0.40
	1	**0.0183**±0.0110(0.0124±0.0097)	**2.0**±0.9	0.35±0.48
	BD	0.0228±0.0124(0.0130±0.0100)	3.3±3.6	0.20±0.40
	10	0.0195±0.0104(**0.0123**±0.0088)	3.0±2.1	**0.15**±0.36
	1	**0.0177**±0.0114(0.0124±0.0095)	**2.1**±1.0	0.20±0.40
Phenetylamines (628/22/—) [15]	BABD	0.2589*±0.1206*(0.0520±0.0289)	12.1±4.6	0.85±0.42
0.2092±0.0586(0.1875±0.0502)	100	**0.2227**±0.0861*(**0.0435**±0.0240)	10.7±5.9	**0.60**±0.49
	1	0.2488±0.1390*(0.0562±0.0300)	**6.3**±2.1	1.70±0.87
	BD	**0.2282**±0.0659(**0.0428**±0.0268)	21.2±11.0	1.12±0.51
	100	0.2718*±0.1652*(0.0452±0.0245)	8.6±3.4	**0.37**±0.48
	1	0.2593±0.1710*(0.0623±0.0263)	**5.9**±1.6	0.90±0.58
Orange Juice (700/150/68) [16]	BABD	5.6395±0.9222(**4.2132**±0.7461)	26.0±16.3	173.21±320.08
5.8184±0.8649(5.3375±0.6187)	200	**5.6357**±1.0307(4.2676±0.6672)	15.4±8.1	**59.02**±46.72
	1	6.8030*±1.0408(5.8087*±0.8223*)	**4.3**±2.1	91.99±29.51
	BD	5.4824±0.8409(**4.1034**±0.6046)	33.7±35.6	108.06±89.13
	200	**5.4578**±0.9275(4.2464±0.6297)	17.5±13.3	**46.51**±27.20
	1	6.5802*±0.9712(5.8136*±0.8226*)	**5.3**±2.5	56.19±22.33
B. Cancer (3226/14/8) [17]	BABD	**0.2891**±0.0585(**0.0424**±0.0055)	42.4±9.5	7.13±2.41
0.4076±0.1544(0.3107±0.0355)	200	0.3036±0.1156(0.0443±0.0051)	37.8±6.0	**3.00**±0.55
	1	0.3305±0.1954(0.0486±0.0058)	**31.2**±4.3	17.56±3.58
	BD	0.3260±0.1349(**0.0477**±0.0125)	105.0±86.1	28.64±10.22
	200	0.3260±0.1453(0.0562±0.0078)	34.8±10.4	**2.40**±0.63
	1	**0.3155**±0.1013(0.0511±0.0077)	**31.9**±4.1	13.60±2.63
Leukemia (7129/38/34) [18]	BABD	**0.1984**±0.0431(**0.0447**±0.0065)	62.2±15.9	134.00±53.09
0.2220±0.0208(0.2433±0.0196)	200	0.2072±0.0313(0.0452±0.0061)	57.4±8.8	**71.18**±6.86
	1	0.2414*±0.0344*(0.0601±0.0089)	**48.8**±6.0	536.08±110.09
	BD	**0.2037**±0.0250(**0.0388**±0.0106)	271.9±205.7	1350.17±672.78
	200	0.2278±0.0309*(0.0598±0.0075)	**40.0**±6.7	**34.62**±2.88
	1	0.2497*±0.0383*(0.0634±0.0094)	51.4±5.5	438.99±79.46

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 25420438.

4 Conclusions

In this paper we extended batch block addition and block deletion (BABD) to incremental BABD. For a given subset of variables we select variables by BABD

and add remaining variables to the selected variable set. We iterate BABD for the augmented set and if the obtained approximation error is smaller than that of the previous step, we iterate the above procedure adding remaining variables to the set of selected variables. If not, we undo the variable selection of the current step and iterate the above procedure. By computer experiments using six benchmark data sets, the approximation errors of the incremental BABD were comparable to or better than batch BABD and the selection time of incremental BABD was shortened for large numbers of variables when an appropriate number of data were added.

References

1. V. N. Vapnik, Statistical Learning Theory, John Wiley & Sons, 1998.
2. J. A. K. Suykens, Least squares support vector machines for classification and nonlinear modeling, Neural Network World, 10(1–2), pp. 29–47, 2000.
3. S. Abe, Support Vector Machines for Pattern Classification, Springer 2005.
4. G. M. Fung and O. L. Mangasarian, A feature selection Newton method for support vector machine classification, Computational Optimization and Applications, 28(2), pp. 185–202, 2004.
5. S. D. Stearns, On selecting features for pattern classifiers, Proc. Third International Conference on Pattern Recognition, pp. 71–75, 1976.
6. P. Pudil, J. Novovičová, and J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters, 15(11), pp. 1119–1125, 1994.
7. T. Zhang, Adaptive forward-backward greedy algorithm for sparse learning with linear models, Proc. NIPS 21, pp. 1921–1928. MIT Press, 2009.
8. H. Liu and R. Setiono, Incremental feature selection, Applied Intelligence, 9(3), pp. 217–230, 1998.
9. S. Perkins, K. Lacker, and J. Theiler, Grafting: Fast, incremental feature selection by gradient descent in function space, Journal of Machine Learning Research, 3, pp. 1333–1356, 2003.
10. R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recognition, 39(12), pp. 2383–2392, 2006.
11. P. Bermejo, J. A. Gamez, and J. M. Puerta, Speeding up incremental wrapper feature subset selection with naive Bayes classifier, Knowledge-Based Systems, 55, pp. 140–147, 2014.
12. T. Nagatani, S. Ozawa, and S. Abe, Fast variable selection by block addition and block deletion, Journal of Intelligent Learning Systems and Applications, 2(4), pp. 200–211, 2010.
13. T. Nagatani and S. Abe, Feature selection by block addition and block deletion, Proc. ANNPR 2012, pp. 48-59, 2012.
14. A. Asuncion and D. J. Newman, UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.
15. Milano Chemometrics and QSAR Research Group, http://michem.disat.unimib.it /chm/download/download.htm.
16. UCL Machine Learning Group, http://www.ucl.ac.be/mlg/index.php?page=home.
17. I. Hedenfalk et al., Gene-expression profiles in hereditary breast cancer, The New England Journal of Medicine, 344(8), pp. 539–548, 2001.
18. T. R. Golub et al., Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science, 286, pp. 531–537, 1999.