Supplementary materials for this article are available at http://pubs.amstat.org/toc/jcgs/18/3.

Variable Selection in Linear Regression With Many Predictors

Airong CAI, Ruey S. TSAY, and Rong CHEN

With advanced capability in data collection, applications of linear regression analysis now often involve a large number of predictors. Variable selection thus has become an increasingly important issue in building a linear regression model. For a given selection criterion, variable selection is essentially an optimization problem that seeks the optimal solution over 2^m possible linear regression models, where m is the total number of candidate predictors. When m is large, exhaustive search becomes practically impossible. Simple suboptimal procedures such as forward addition, backward elimination, and backward-forward stepwise procedure are fast but can easily be trapped in a local solution. In this article we propose a relatively simple algorithm for selecting explanatory variables in a linear regression for a given variable selection criterion. Although the algorithm is still a suboptimal algorithm, it has been shown to perform well in extensive empirical study. The main idea of the procedure is to partition the candidate predictors into a small number of groups. Working with various combinations of the groups and iterating the search through random regrouping, the search space is substantially reduced, hence increasing the probability of finding the global optimum. By identifying and collecting "important" variables throughout the iterations, the algorithm finds increasingly better models until convergence. The proposed algorithm performs well in simulation studies with 60 to 300 predictors. As a by-product of the proposed procedure, we are able to study the behavior of variable selection criteria when the number of predictors is large. Such a study has not been possible with traditional search algorithms.

This article has supplementary material online.

Key Words: Best subset; BIC; Grouping.

1. INTRODUCTION

Ever since its first rigorous treatment by Pearson (1896), linear regression has become one of the most commonly used statistical techniques and its inferences have been ex-

Airong Cai is Senior Statistician, DemandTec Inc., 1 Circle Star Way, Suite 200, San Carlos, CA 94070. Ruey S. Tsay is H.G.B. Professor of Econometrics and Statistics, Booth School of Business, University of Chicago, 5807 S. Woodlawn Ave, Chicago, IL 60637. Rong Chen is Professor, Department of Statistics and Biostatistics, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854 and Department of Business Statistics and Econometrics, Peking University, Beijing 100871, China (E-mail: *rongchen@stat.rutgers.edu*).

Journal of Computational and Graphical Statistics, Volume 18, Number 3, Pages 573–591
DOI: 10.1198/jcgs.2009.06164

tensively studied. In building a linear regression model, variable selection, which is also known as subset selection or model selection, is a crucial issue. Irrelevant predictors, if included in the model, may lead to erroneous inference and an increase in prediction errors. Interpretation of the fitted model may also become difficult because of the presence of "junk" variables. Variable selection or certain dimension reduction methods become a necessity in the extreme case in which the number of observations N is less than the number of candidate variables m. Due to recent advances in data collection and computing capability, modern statistical applications often involve a large number of candidate variables, making variable selection even more important for the purpose of building an effective regression model.

All variable selection procedures require a selection criterion to start with. Well-known variable selection criteria in linear regression analysis include *adjusted-R^2*, AIC (Akaike 1973, 1974), Schwarz's BIC (Schwarz 1978), Mallows's C_p (Mallows 1973), and PRESS (Allen 1971). In addition, considerable efforts have been made by many researchers to obtain better criteria, by either modifying the existing ones or creating innovative new measures. Examples include Tibshirani and Knight (1999); Pauler (1998); Zheng and Loh (1997); Ronchetti and Staudte (1994); Rao and Wu (1989), and Shen and Ye (2002).

Given a selection criterion, simple variable selection procedures such as forward selection, backward elimination, and stepwise procedure (Draper and Smith 1998) are available for linear regression analysis in many standard statistical software packages. Miller (1990) provided details of these procedures. By examining only a small portion of all possible subset models, these procedures are easy and fast. But the selected model is often far from the optimal one. On the other hand, exhaustive subset search can always find the optimal model under a given criterion, but the complexity involved quickly becomes insurmountable as the number of explanatory variables increases. The "leaps and bounds" algorithm (Furnival and Wilson 1974) makes use of a regression-tree structure to facilitate faster search for the global optimum. Recently the regression-tree approach has also been implemented in the "branch-and-bound" algorithm in Gatu and Kontoghiorghes (2006) for best subset search. In both cases, the algorithms are still not fast enough when m is large, say $m > 50$. This difficulty leads to the development of a variety of advanced suboptimal search algorithms. For example, George and McCulloch (1993) proposed a Bayesian variable selection procedure via Gibbs sampling. Chatterjee, Laudato, and Lynch (1996) described Genetic Algorithms and discussed their applications for variable selection. A recent development of variable selection using rate distortion theory can be found in Jornsten (2007). Other examples of studying variable selection procedures include Miller and Ribic (1995); Mendieta, Boneh, and Walsh (1994); Aeberhard, de Vel, and Coomans (1993); Johnsson (1992); Ferri and Piccioni (1992); Mitchell and Beauchamp (1988), and Antoch (1986).

In this article we propose a simple algorithm for searching the optimal subset of variables. The main idea of the algorithm is to use a grouping scheme to reduce computational complexity of the search. Exhaustive search among all possible combinations at the group level is used to identify the important variables. The remaining "less important" variables are then divided randomly into groups and the search process is iterated until convergence. The proposed algorithm is easy to implement and proved to be more effective than some existing search methods in simulation studies and real data analysis. The proposed algorithm is ad hoc by nature and does not guarantee to find the optimal solution. Our goal is to develop a simple algorithm that can find the "optimal" solution more frequently and, when it fails, find a better suboptimal solution than the existing methods available in the literature, for example, the Genetic Algorithm and simple forward-and-backward search.

It is worth mentioning that we endeavor to develop a new search procedure under a given model selection criterion. Therefore, we focus on comparing search capability of the algorithms instead of comparing model selection criteria. For illustrative purpose, Schwarz's BIC is used in this article to demonstrate the proposed search procedure. However, no specific property of BIC is used in the proposed algorithm. Other selection criterion, such as AIC or C_p, can be used. We also note that severe collinearity among the predictors often results in competing models with nearly equal BIC values. In this article we restrict ourselves to the problem of finding the "best" model under a selection criterion (e.g., BIC) and do not venture into model averaging and other related issues.

Information criteria such as AIC and BIC penalize the number of variables in a model. Recently, penalized least squares approaches have attracted much attention. Examples include the bridge regression (Frank and Friedman 1993), LASSO (Tibshirani 1996; Zou 2006), LARS (Efron et al. 2004), SCAD (Antoniadis and Fan 2001), and Dantzig selector (Candes and Tao 2007). Some of these approaches enjoy attractive computational efficiency, sparseness solutions as well as oracle properties. More references can be found in Donoho (2000) and Fan and Li (2006). Comparison between the information criteria and the penalized least squares approaches is beyond the scope of this article, though we will demonstrate with an example that such a comparison becomes possible using the proposed algorithm.

The article is organized as follows. In Section 2 we briefly discuss the main idea of the proposed algorithm and show that it outperforms the Genetic Algorithm in a simulation study. We then introduce the basic operations used in the proposed algorithm and give details of the algorithm. Section 3 conducts a simulation study and provides further comparisons of the proposed algorithm with several other procedures. Three nontrivial settings are used in the simulation. Section 4 uses the proposed algorithm to investigate performance of the BIC criterion in linear regression analysis when the number of predictors is large. Such a study was not possible before. A brief comparison between the BIC criterion and LASSO is presented in Section 5. It is not a systematic comparison. Our goal is simply to demonstrate that the proposed algorithm makes such a comparison feasible when the number of candidate variables is large.

2. A SEARCH ALGORITHM BASED ON GROUPING

For ease in describing the proposed algorithm, we start with its basic concept and a comparison with the Genetic Algorithm. We then introduce two basic operations used in the proposed algorithm. Section 2.3 gives details of the proposed algorithm.

2.1 PRELIMINARIES

2.1.1 The Basic Concept

As mentioned earlier, it is impossible to carry out the naive exhaustive search among all possible regressions when the number of candidate predictors is large. To overcome this difficulty, the proposed algorithm divides the predictors into $k + 1$ nonoverlapping groups and performs exhaustive search on the group level. More specifically, given a grouping scheme of all explanatory variables, namely Group 0, Group $1, \ldots,$ Group k, there exist 2^k possible combinations among Group 1 to Group k at the group level. Each combination of the groups is then combined with Group 0 to form a subset of the predictors. This results in 2^k different regression models with Group 0 being the intersection of all entertained models. To each of these 2^k models, we apply some simple and fast procedure to obtain an improved suboptimal model and its corresponding BIC value. For instance, backward elimination can be used to remove one variable at a time until there is no improvement on the BIC value. We then order ascendingly the resulting 2^k improved suboptimal models according to their BIC values and select the best one as the new Group 0. The variables not in the new Group 0 are then randomly divided into k new groups (new Group $1, \ldots,$ new Group k), and the search procedure is iterated. Obviously, Group 0 plays an important role in the search algorithm as it contains variables tentatively identified as the *important ones* in the regression. The initial Group 0 can be constructed by using either prior information or random selection. It is worth mentioning that a variable may leave or reenter Group 0 throughout the iterations. Whenever a better subset is found during the iterative procedure, it is saved as a new Group 0 for the next iteration. The search procedure stops when Group 0 does not change for a certain number of consecutive iterations. The final Group 0 is the selected model.

Note that the number of groups k should be properly chosen to balance between the number of groups and the size of each group. In particular, k should be kept relatively small so that it remains feasible to conduct an exhaustive search over the 2^k possible combinations of groups. The total number of the available candidate explanatory variables does not directly affect the exhaustive search even though it influences the sizes of the groups. It is desirable to have a small group size. By working at the group level, efficient calculations for each group combination enable the proposed algorithm to search quickly over a large number of subset models and find the better ones (e.g., with lower BIC values) for further analysis.

2.1.2 A Refinement

To further improve the performance of the proposed algorithm, we keep q models with the smallest BIC values in each iteration to form q candidates of Group 0, instead of simply taking the one with the best BIC value. In other words, in each iteration of the search procedure, we entertain q grouping schemes. Each grouping scheme has its own Group 0 and 2^k possible subset configurations. In the next iteration a set of q candidates of Group 0 is again selected, corresponding to the q suboptimal models with smallest BIC values among all $q \times 2^k$ entertained models based on the q grouping schemes. This refinement substantially expands the search space of the procedure. The effects of q and k on the performance of the algorithm are investigated and demonstrated in subsequent sections.

The proposed algorithm is an iterative procedure of grouping variables and applying a simple and fast selection procedure to search for an improved suboptimal model given a combination of groups. It can have different variants depending on the simple selection procedure used. For example, if the forward-backward stepwise procedure is used, then we have an algorithm consisting of grouping and backward-forward selection. We denote such an algorithm by "GBF." Similarly, we use GBK and GFW throughout this article to denote "grouping and backward elimination" and "grouping and forward selection," respectively.

2.1.3 A Simple Comparison

Before formally introducing the proposed algorithm, we compare its performance with the well-known Genetic Algorithm (GA) in a simulation study. Scenario B of Section 3 is used to generate 100 datasets, each having 60 explanatory variables and 1000 observations. Both the proposed algorithm (with backward elimination, i.e., GBK) and the Genetic Algorithm (denoted as GA with various settings) are applied to each dataset searching for the optimal model that minimizes the BIC value. Based on the minimum BIC values obtained by the procedures for each dataset, we classify an algorithm as a "success" if its BIC value is no greater than those of all other algorithms. Finally the total number of successes based on 100 datasets is recorded. The algorithm with a higher frequency of successes is considered to have higher searching capability (or better performance). In addition, we also report the CPU time of each algorithm, observed on a Linux machine equipped with Intel(R) Xeon(TM) dual-processors (CPU: 1500 MHz; RAM: 1 GB).

Table 1 shows the result of comparison. For the Genetic Algorithm, we employ different configurations of population size (pop) and number of generations (gen). For reference purposes, Table 1 also shows the result of the standard forward-and-backward stepwise procedure, denoted by "BF." It is seen from the table that the GBK procedure has the largest number of successes and outperforms all GA methods considered. The performance of GA improves as the population size increases, but the associated computing time also increases. Furthermore, for all of the 100 simulated datasets, GA methods never find a smaller BIC value than the proposed GBK algorithm. Surprisingly, the simple BF procedure is very fast with comparable performance to the GA methods.

Table 1. Performance comparison between GBK and GA. The results are based on 100 datasets generated under Scenario B in Section 3. The sample size is $N = 1000$ and the number of predictors is $m = 60$. In the table, GA(k_1, k_2) indicates a GA procedure with k_1 as the population size and k_2 as number of generations, and GBK(k, q) denotes GBK with k groups and q grouping schemes.

Procedure	GBK(10, 6)	GA(300, 200)	GA(500, 200)	GA(1,000, 400)	GA(2,000, 400)	BF
Number of successes	100	11	22	64	77	53
CPU (seconds)	636	719	1,237	2,988	5,519	2

2.2 THE BASIC OPERATIONS

In this subsection, we introduce two basic operations that are repeatedly used in the proposed algorithm. These operations focus on computational efficiency.

(1) *Simple Variable Selection (SVS)*: For a set of explanatory variables G and the dependent variable Y, the operation of Simple Variable Selection, $SVS(G)$, is defined as applying a simple and efficient variable selection procedure to G to select a linear regression model. The selection procedure considered includes backward-forward (BF), backward elimination (BK), and forward addition (FW). This operation produces two outcomes: a selected subset of G and its corresponding BIC value. We call the selected subset an "improved set" of G and denote it by *ISET*. For ease in referencing, the corresponding BIC value is denoted by *IBIC*.

Some properties of *SVS* are given below:

1. Given a set G of explanatory variables, different search procedures may result in different *ISET* and *IBIC*. For instance, for a given G, backward elimination tends to select a larger subset than forward addition does.

2. Two different sets G_1 and G_2 of explanatory variables may produce the same improved set *ISET* and thus have the same *IBIC* value.

3. $SVS(G)$ does not necessarily produce the optimal subset which has the minimum BIC value among all possible subsets of G. The goal of *SVS* is to perform a quick and efficient selection, not to focus on the optimal selection. The improvement of model selection is achieved through iterations.

4. Given a set G and a deterministic search procedure, $SVS(G)$ produces unique *ISET* and *IBIC*. If a stochastic SVS is used, the unique result is not guaranteed. However, this should not have a significant impact on the performance of the proposed procedure.

(2) *Regrouping*: Given a positive integer k, regrouping of the predictors in G, denoted by $Rg(G)$, produces k nonoverlapping and nearly equal sized subgroups of variables $\{G_1, \dots, G_k\}$ such that $\bigcup_{i=1}^{k} G_i = G$. The subgroups are obtained by either a random process or a set of specific rules. Our experience shows that random grouping works well.

2.3 THE PROPOSED ALGORITHM

We now describe the proposed algorithm. Suppose that at the jth iteration, we have q different grouping schemes:

$$\{G_0^{(j,i)}, G_1^{(j,i)}, \dots, G_k^{(j,i)}\}, \qquad i = 1, \dots, q,$$

where q is a prespecified positive integer and the superscript i denotes the ith grouping scheme of the m explanatory variables. As mentioned earlier, for each i, group $G_0^{(j,i)}$ contains variables that are likely to be important in the regression.

We perform the following steps to complete the jth iteration.

1. For each i $(i = 1, \ldots, q)$, consider all possible subsets $g_s \subset \{1, \ldots, k\}$ for $s = 1, \ldots, 2^k$ and apply operation SVS to obtain the improved set
$$ISET(i,s) = SVS\Bigg(G_0^{(j,i)} \bigcup_{t \in g_s} G_t^{(j,i)}\Bigg)$$
and the corresponding improved BIC value $IBIC(i,s)$.

   The total $q \times 2^k$ possible combinations of (i,s) produce at most $q \times 2^k$ possible improved sets $ISET$ and their corresponding $IBIC$ values. We identify the q distinct improved sets that have the smallest $IBIC$ values and denote them as $ISET_1^*, \ldots, ISET_q^*$. Use these selected q improved sets as the new Group 0 for the next iteration, namely,
$$G_0^{(j+1,i)} = ISET_i^*, \qquad i = 1, \ldots, q.$$
2. For each $i = 1, \ldots, q$, obtain the complement set of $G_0^{(j+1,i)}$ and perform the regrouping operation (Rg) on the complement set to form k random groups, that is,
$$\big\{G_1^{(j+1,i)}, \ldots, G_k^{(j+1,i)}\big\} = Rg\big(\mathbf{X} \setminus G_0^{(j+1,i)}\big),$$
where $\mathbf{X}$ denotes the set of all explanatory variables.

At the completion of each iteration, we check whether there is any improvement on the smallest $IBIC$ value from the previous iteration. If an improvement is found, we continue the iteration. The procedure is stopped if the smallest BIC value found does not improve in v consecutive iterations. The extra iterations are used to reduce the possibility of early stopping due to random regrouping. In our experience, $v = 10$ is sufficient in most of the cases.

Note that the exhaustive search over all possible subsets of $\mathbf{X}$ is a special case of the proposed algorithm when $k = m$, $n_0 = 0$, and $n_i = 1$ $(i = 1, \ldots, k)$, where n_i denotes the number of variables in group G_i. Moreover, when $n_0 = 0$, $k = 1$, and $n_1 = m$, the proposed algorithm is equivalent to performing a one-step selection using the specified simple variable selection procedure, that is, the SVS method.

Regarding the stability of the selected model, we note that stepwise procedures are deterministic and, hence, always produce the same answer. However, the procedures are sensitive to addition or deletion of observations and variables. The proposed procedure should be more stable under the latter situation. On the other hand, in the case of limited sample size and severe collinearity among the predictors, the BIC criterion may not be stable, because there may exist multiple competing models with similar BIC values.

3. SIMULATION STUDY

In this section we investigate the performance of the proposed algorithm using simulation and provide further comparison with other methods. We also study the effects of design parameters on the performance of the proposed algorithm, including the number of groups k and the number of grouping schemes q.

Three scenarios with different data generating models are used in our simulation to represent various degrees of difficulty in variable selection. They are chosen to provide information concerning the performance of the proposed algorithm under general working environment. In applying the proposed algorithm, we use three procedures in simple variable selection, namely, the backward-forward stepwise regression, backward elimination, and forward selection, which are denoted by GBF, GBK, and GFW, respectively. We also include simple backward-forward (BF), backward elimination (BK), and forward addition (FW) procedures as references. Finally, the "leaps-and-bounds" algorithm is included in one of the scenarios when it is computationally feasible. For GBF, GBK, and GFW, we use $q = 10$ and $k = 5$ unless specified otherwise.

For each of the three scenarios, 100 datasets are generated and various search procedures are compared. For a given dataset, a procedure is marked as a "success" if it finds a model whose BIC value is the minimum among those obtained by all procedures. The number of successes based over the 100 datasets is then recorded for each procedure. A procedure with a higher number of successes is considered to have higher searching capability and better performance. For each procedure, the associated computing time for processing the 100 datasets is also reported.

Scenario A: In this setting, the sample size is $N = 150$, and each data point is generated as follows. Let $X_j^* \sim N(0, 2)$ for $j = 1, \ldots, 60$, $e_0 \sim N(0, 4)$, and $e_i \sim N(0, 2)$ for $i = 1, \ldots, 6$ and X_j^*, e_0, and e_i are all mutually independent. The explanatory variables are then constructed as

$$X_j = X_j^* + e_0 + e_i \quad \text{for } j = 1, \ldots, 60 \text{ and } i = \text{floor}((j-1)/10) + 1,$$

where floor(x) is the largest integer less than or equal to x. Thus, there are six clusters among the explanatory variables. The within- and cross-cluster correlation coefficients of the predictors are 0.75 and 0.5, respectively. The response variable Y is generated by

$$Y = 1 + \sum_{i=1}^{4} X_i + \sum_{i=11}^{13} X_i + X_{21} + X_{22} + \epsilon,$$

where $\epsilon \sim N(0, 100)$ and is independent of other predictors. Thus, Y depends on 9 of the 60 predictors.

Table 2 summarizes the simulation results for Scenario A. In this example we also applied the "leaps-and-bounds" algorithm denoted by "LB." The "leaps-and-bounds" is an optimal algorithm that performs a branch-and-bound search and guarantees to find the

Table 2. Performance comparison of various algorithms based on 100 datasets under Scenario A. The sample size is $N = 150$ and the number of candidate predictors is $m = 60$.

Procedures	LB	GBF	GBK	GFW	BF	BK	FW
Number of successes	99	100	100	78	31	26	40
Time used (seconds)	45,200	2,855	1,002	47	4.25	1.36	0.39

global optimum. However, it requires extensive computation when the number of candidate variables is large. In this particular instance, because the true model only involves nine predictors, we restrained the LB method to search among the subsets of size less than 13 to make the procedure computationally feasible. From the table, we make the following observations. First, although the LB algorithm is capable of performing the exhaustive search, it requires extensive computation. Even with the model size restriction that dramatically reduces the search space, the LB algorithm is still nearly 16 times more computationally expensive than the GBF procedure. In addition, LB fails to find the optimal BIC in one of the 100 datasets because the optimal BIC for that particular dataset is achieved by a model with more than 12 predictors, exceeding the maximum subset size we set for the exhaustive search. Second, as expected, the proposed algorithms substantially outperform their simple counterparts by having much higher frequencies of successes, even though the simple procedures are extremely fast. Third, the proposed GBF and GBK algorithms perform equally well with the LB algorithm whereas the GFW algorithm fails in about 25% of the datasets. Finally, the GBK algorithm works the best as it achieves high performance with reasonable computing time.

Scenario B: Here each data point is generated as follows. Let $X_j^* \sim N(0,1)$ for $j = 1, \dots, 60$ and define

$$\begin{aligned}
X_j &= X_j^* \quad \text{for } j = 1, \dots, 30,\\
X_j &= X_j^* + 0.3X_{j-30} + 0.5X_{j-29} - 0.7X_{j-28} + 0.9X_{j-27} + 1.1X_{j-26}\\
&\quad \text{for } j = 31, \dots, 40,\\
X_j &= X_j^* + 0.3X_{j-30} + 0.5X_{j-29} + 0.7X_{j-28} - 0.9X_{j-27} + 1.1X_{j-26}\\
&\quad \text{for } j = 41, \dots, 50,\\
X_j &= X_j^* + 0.3X_{j-30} - 0.5X_{j-29} + 0.7X_{j-28} + 0.9X_{j-27} + 1.1X_{j-26}\\
&\quad \text{for } j = 51, \dots, 60.
\end{aligned}$$

The response variable Y is given by

$$Y = 1 + \sum_{i=31}^{60} X_i + \epsilon,$$

where $\epsilon \sim N(0, 100)$ and is independent of all other explanatory variables. This setting is an extended version of that used in Fernandez, Ley, and Steel (2001).

Each of the 100 datasets generated under this scenario has 300 observations. Because the true model involves 30 predictors, the LB algorithm is no longer practical and is omitted. Table 3 gives the simulation results, from which we obtain essentially the same conclusions as those of Scenario A. Both the GBF and GBK algorithms have the highest frequency of successes but the GBK algorithm uses less computing time. On the other hand, the performance of GFW is disappointing. All three newly proposed algorithms outperform their simple counterparts. Recall that Scenario B is also employed to obtain Table 1

Table 3. Performance comparison for various algorithms based on 100 datasets under Scenario B. The sample size is 300 and the number of candidate predictors is 60.

Procedures	GBF	GBK	GFW	BF	BK	FW
Number of successes	100	100	19	24	18	2
Time used (seconds)	3,098	824	309	4.56	1.17	0.39

which shows that GBK outperforms the Genetic Algorithm in both accuracy and computing efficiency.

THE EFFECT OF INITIAL GROUP 0

Based on prior discussions, Group 0 plays an important role in the proposed algorithm. It can be formed initially by using either prior information or random selection. Here we use Scenario B to study the performance of the GBK algorithm under various initializations of Group 0. More specifically, for each of the 100 datasets generated from Scenario B, the GBK algorithm with $(k, q) = (6, 30)$ is applied with initial Group 0 being (a) randomly selected, (b) an empty set, or (c) the set of variables selected by a Genetic Algorithm. Furthermore, to accommodate the stochastic nature of the proposed algorithm, five independent runs of the GBK algorithm are carried out for each initialization of Group 0. An individual run of the GBK algorithm is considered to achieve convergence when there is no BIC improvement in 10 consecutive iterations. Table 4 contains the results.

The table also gives the empirical distributions of the number of iterations for the 500 runs of each initialization. Note that the number of iterations needed is greater than 10 because of the stopping criterion used. From the table, we see first that different initializations of Group 0 have little impact on the performance of the GBK algorithm. Second, the GBK algorithm converges faster if the initial Group 0 consists of the variables selected by the GA. However, there are several datasets that require many more iterations. This result indicates that a more informed Group 0 can indeed speed up the convergence, but it does not necessarily have any significant impact on the final selection result. If the algorithm starts at a local mode, it may converge slowly because it needs to climb out of the local mode first. It is assuring that the GBK algorithm is capable of getting out of the local mode

Table 4. Performance of the GBK algorithm under three different schemes of initialization for Group 0. The results are based on 100 datasets generated from Scenario B. The sample size is 300 and the number of predictors is 60. "GA" denotes using the variables selected by a Genetic Algorithm as the initial Group 0.

Initialization	Number of successes					Iterations					
of Group 0	Run 1	Run 2	Run 3	Run 4	Run 5	12	13	14	15	16	≥17
Random	100	100	100	100	100	216	260	21	1	1	1
Empty	100	100	100	100	100	158	321	21	0	0	0
GA	100	99	100	99	100	420	43	23	4	5	5

that has trapped the GA algorithm. Finally, using the empty set as Group 0 seems to be a reasonable choice for the GBK algorithm. Overall, the simulation results indicate that the initialization of Group 0 does not have any significant impact on the performance of the algorithm, though it has some effect on the number of iterations needed.

Scenario C: In this setting, data are generated as follows. Let $\{X_j^*\}_{j=1}^{30} \sim N(0,1)$. The first 30 explanatory variables, $X_1, \ldots, X_{30}$, are generated by $X_j = X_j^* + e$ where $e \sim N(0,1)$ and $j = 1, \ldots, 30$. Hence the correlation among the predictors X_j's is 0.5 for $j = 1, \ldots, 30$. Variables X_{31} to X_{64} are generated by

$$X_{30+j} = 0.3X_j + 0.5X_{j+1} - 0.7X_{j+2} + 0.9X_{j+3} + 1.1X_{j+4} + e_{30+j}, \qquad k = 1, \ldots, 34,$$

where e_{30+j} are independent $N(0,1)$. The response variable Y follows a linear regression model with X_{31} to X_{60} as the predictors, that is,

$$Y = 1 + \sum_{i=31}^{60} X_i + \epsilon,$$

where $\epsilon \sim N(0, 30^2)$ and is independent of the explanatory variables.

Furthermore, we generate some additional "junk" variables to be included in the set of predictors. We simulate 30 sets of coefficients ϕ_{ts} ($t = 1, \ldots, 30$; $s = 1, \ldots, 5$) from a uniform distribution $U(0.6, 1.2)$ and randomly assign a positive or negative sign to the coefficients. The coefficients are then renormalized so that $\sum_{s=1}^{5} \phi_{ts}^2 = 4$ for $t = 1, \ldots, 30$. We then generate eight groups of junk variables by

$$X_j^{(i)} = \phi_{t1}X_{30+j} + \phi_{t2}X_{31+j} + \phi_{t3}X_{32+j} + \phi_{t4}X_{33+j} + \phi_{t5}X_{34+j} + \varepsilon_j^{(i)},$$

where $\varepsilon_j^{(i)}$ are independent $N(0, 0.5^2)$ for $i = 1, \ldots, 8$; $j = 1, \ldots, 30$ and $t = (i - 2 + j)(\text{mod}\, 30) + 1$. If necessary, additional groups of junk variables can be obtained in the same manner.

This is a difficult setting for variable selection. The variables used to generate Y, referred to as the "important" variables, are linear combinations of some correlated variables, and the "junk" variables are linear combinations of the "important" variables in a rotated manner so that they have high collinearity.

First, we let the set of predictors be $\{X_{31}, \ldots, X_{60}, X_1^{(1)}, \ldots, X_{30}^{(1)}\}$ and use sample size $N = 10{,}000$. Table 5 shows the comparison results. Based on results of Scenarios A and B, we focus on comparison between the GBK algorithm and the Genetic Algorithm (GA). Furthermore, for simplicity, we only employ the BF procedure as the reference point. From Table 5, the GBK algorithm continues to outperform the GA. The results also show that the performance of the GA improves with increasing population size and number of generations. But, similarly to that of Table 1, the improvement is at a slow rate. The one-step BF procedure is fast and has a comparable performance with GA (pop = 300, gen = 200).

THE EFFECTS OF q AND k

In the prior simulations, we fixed the number of random grouping schemes q and the number of groups k. These two design parameters may affect the performance of the pro-

Table 5. Performance comparison between the GBK and GA under Scenario C. The sample size is $N = 10{,}000$ and the results are based on 100 datasets.

Procedures	Number of successes	Time used (seconds)
GBK($q = 10, k = 6$)	100	515
GA(pop = 300, gen = 200)	18	700
GA(pop = 500, gen = 200)	21	1,218
GA(pop = 1,000, gen = 400)	34	2,838
GA(pop = 2,000, gen = 400)	59	5,343
BF	17	2

posed algorithm. To investigate their potential effects, we employ the GBK algorithm and datasets generated from Scenario C for further study.

First we generated 100 datasets, each with sample size $N = 10{,}000$, from Scenario C with the predictors $\{X_1, \dots, X_{60}, X_1^{(1)}, \dots, X_{30}^{(1)}\}$. Thus, there are 90 predictors. We let the number of random groups k be in $\{4, 5, 6, 7, 8\}$ and the initial number of random grouping schemes q be in $\{5, 10, 15, 20, 25, 30\}$. Two methods are used to govern the choice of q for each given k. In the first method, $q^{(j)} = q$ is fixed throughout the process. In the second method, $q^{(j)}$ decreases gradually over the selection iterations, for example, $q^{(j)} = 0.9^j q$. The reason for using a decreasing $q^{(j)}$ is that a large initial q would allow the proposed algorithm to entertain more subsets of the predictors at the beginning of the search, yet a smaller $q^{(j)}$ could reduce the computation burden as the search becomes more advanced.

Table 6 shows the results. For each q, there are two rows, one for fixed q and the other for decreasing q, starting at the stated values. From the table, we make several observa-

Table 6. Performance and efficiency of the GBK algorithm with different q and k under Scenario C. The sample size is 10,000 and the results are based on 100 datasets. For each q, there are two rows, one for fixed q value and the other for decreasing q value, starting at the stated value and decreasing by 10% for each iteration until reaching 1. The entries of the table show the number of successes out of 100 datasets, with total CPU time in seconds shown in parentheses.

		k				
q		4	5	6	7	8
5	fixed	52 (263)	57 (451)	67 (846)	74 (1,488)	78 (2,721)
	decreasing	33 (114)	43 (299)	45 (557)	65 (877)	72 (1,414)
10	fixed	65 (513)	72 (893)	86 (1,555)	90 (2,812)	91 (4,475)
	decreasing	58 (309)	70 (600)	75 (1,023)	81 (1,930)	90 (3,582)
15	fixed	73 (751)	89 (1,287)	91 (2,291)	97 (4,733)	97 (7,631)
	decreasing	68 (628)	81 (939)	88 (1,440)	90 (2,700)	94 (5,120)
20	fixed	83 (1,039)	91 (1,733)	91 (3,055)	98 (5,405)	100 (9,893)
	decreasing	74 (898)	91 (1,126)	91 (2,049)	98 (3,800)	99 (7,229)
25	fixed	85 (1,217)	94 (2,677)	95 (3,735)	95 (6,647)	99 (13,223)
	decreasing	79 (949)	90 (1,620)	90 (3,013)	95 (6,057)	96 (10,037)
30	fixed	85 (1,404)	96 (2,481)	98 (4,566)	98 (8,610)	99 (15,976)
	decreasing	83 (973)	95 (1,831)	97 (3,916)	98 (6,963)	99 (12,757)

tions. First, as expected, each row of the table shows that the performance of the GBK algorithm improves as the number of groups k increases, though at the expense of longer CPU time. Similarly, each column of the table also shows the same performance pattern as the number of grouping schemes q increases. Second, a large k only needs a moderate q for the algorithm to work well. On the other hand, a smaller k requires a much larger q for the algorithm to be effective. More importantly, there is no need to have large k and large q simultaneously. The proposed algorithm does not work well when both k and q are small. Third, in the case of fixed q, the combination of a moderate k and large q seems to be the best choice, for example, $(k, q) = (5, 30)$. Such a combination performs well in variable selection and requires less computing time. Fourth, as expected, decreasing q can save computing time. But one must start with a relatively large q. For this particular example, a moderate k with a large initial q, which decays gradually, works well; see the last row of Table 6.

In summary, the results shown in Table 6 indicate that proper choices of q and k enable the proposed algorithm to achieve outstanding performance with reasonable efficiency. The computation time increases exponentially as k increases, but increases linearly with q. To improve the performance of the GBK algorithm, it seems more effective to increase k than q, though more computationally costly. In general, the values of k and q should be selected carefully to achieve a trade-off between effectiveness in variable selection and computational demand. For the two examples considered, $(k, q) = (6, 30)$ seems to be a good combination. In addition, decreasing q values gradually helps to improve the efficiency of the algorithm without sacrificing much of the performance. This is particularly so when both k and q are large. Obviously, the values of k and q should be adjusted based on the number of predictors and sample size. Our limited experience indicates that $(k, q) = (8, 30)$ performs reasonably well in cases that involve 300 predictors.

THE EFFECT OF THE STOPPING CRITERION

Let v be the number of consecutive iterations of no improvement needed to stop the algorithm. Here we investigate the effect of v on the performance of the proposed algorithm. We randomly generate 100 datasets under Scenario C, each having 1,000 observations and 90 predictors given by $\mathbf{X} = \{X_1, \ldots, X_{60}, X_1^{(1)}, \ldots, X_{30}^{(1)}\}$. The GBK algorithm is then applied to each dataset with $k = 8$ and $q = 30$. For completeness, we use both a random subset and an empty set as the initial Group 0. The stopping criterion is $v \in \{4, 6, 8, 10\}$. Table 7 summarizes the simulation results. From the table, it is seen that the performance of the GBK algorithm is not sensitive to the choice of v. As expected, an increase in v tends to increase the computation time. The table further shows that the initialization of Group 0 is not critical.

In general, variable selection is more difficult when the candidate set is large, when collinearity among the predictors is strong, and when the sample size is small. In such cases, there often exist a number of competing models with similar BIC values and many local modes. The simulation results show that the proposed algorithm works well, even in difficult cases. Its performance can be further improved by using larger k and q values, at the expense of long computation time.

Table 7. Performance of the proposed GBK algorithm with various v, that is, the number of consecutive iterations without BIC improvement needed before termination. The results are based on 100 datasets from Scenario C with sample size $N = 1{,}000$ and 90 predictors.

v	4		6		8		10	
Initial Group 0	Random	Empty	Random	Empty	Random	Empty	Random	Empty
Number of successes	94	97	96	95	98	94	97	96
Average time (seconds)	7,326	7,790	10,412	8,178	11,451	11,648	12,652	15,818

4. EMPIRICAL PROPERTIES OF BIC WHEN THE NUMBER OF PREDICTORS IS LARGE

A by-product of the proposed algorithm is that it enables us to study the properties of BIC (or other information criterion) when the number of predictors is large. It is well known that BIC is consistent (Haughton 1988). That is, under the assumption of a true model, the probability that BIC selects the "true" model converges to 1 as the sample size goes to infinity. Hence, given a set of predictors **X** and a sufficiently large sample size N, the optimal model selected by BIC should converge to the true model. However, there has not been any study concerning the effect of a large number of "junk" variables in **X** on the performance of the BIC criterion. The GBK algorithm with its relatively good performance and efficiency enables us to conduct such a study. Specifically, we intend to gain a better understanding of the properties of BIC under difficult model selection conditions.

Again we consider Scenario C. Using $q = 30$ and $k = 7$, we start with the basic set of predictors $\mathbf{X} = G_{60} = \{X_1, \ldots, X_{60}\}$, and gradually add extra groups of "junk" variables to **X**, each group containing 30 variables. The extra predictors are added by one or two groups at a time. More specifically, the predictors **X** are in $\{G_{60}, G_{90}, G_{120}, G_{180}, G_{210}, G_{270}\}$ so that it evolves to include more "junk" variables, where G_h are defined sequentially as

$$G_{90} = G_{60} \cup \{X_1^{(1)}, \ldots, X_{30}^{(1)}\}, \qquad G_{120} = G_{90} \cup \{X_1^{(2)}, \ldots, X_{30}^{(2)}\}, \qquad \ldots.$$

We then monitor the sample size needed for BIC to select the true variable set $\{X_{31}, \ldots, X_{60}\}$.

Table 8 shows the selection results as the set of predictors **X** expands from G_{60} to G_{270}, under various sample sizes N ranging from 20,000 to 391,875. In the table, "Diff" denotes $1000 \times (BIC(0) - BIC(M))$ with $BIC(0)$ and $BIC(M)$ being the BIC value of the true and selected model, respectively. Thus, a positive "Diff" value indicates that the selected model has a BIC value smaller than that of the true model. When the true model is identified, "Diff" is 0. A negative "Diff" would indicate that the GBK algorithm fails to find the optimal model in terms of the BIC criterion. It is assuring that there are no negative "Diff" values in the table.

In the table, "Size" denotes the number of predictors in the selected model and the number in parentheses shows the number of true predictors in the selected model. For example, "30 (30)" means the selected model is the true model and "17 (10)" means that

Table 8. Variables selected by the GBK algorithm using the BIC criterion for different sample sizes N and different sets of predictors $\mathbf{X}$ from Scenario C, where "Diff" denotes the 1,000 × difference between the BIC values of the true model and the selected model, and "Size" p (h) shows the selected model has p predictors among which h variables are in the true model.

	G_{60}		G_{90}		G_{120}		G_{180}		G_{210}		G_{270}	
N	Diff	Size	Diff	Size	Diff	Size	Diff	Size	Diff	Size	Diff	Size
20,000	0.17	27 (27)	4.89	17 (10)	5.24	16 (11)	5.90	19 (7)	7.06	15 (9)	7.06	15 (9)
30,000	0.00	30 (30)	2.81	20 (14)	3.15	17 (12)	3.76	17 (8)	4.71	16 (8)	4.71	16 (8)
45,000	0.00	30 (30)	1.33	21 (16)	1.92	17 (12)	2.26	16 (11)	2.44	16 (8)	2.74	19 (5)
67,500	0.00	30 (30)	0.60	21 (14)	0.82	21 (13)	1.11	18 (11)	1.32	19 (12)	1.58	20 (9)
101,250	0.00	30 (30)	0.10	26 (22)	0.29	27 (20)	0.39	25 (15)	0.50	24 (15)	0.67	23 (16)
151,875	0.00	30 (30)	0.00	30 (30)	0.08	29 (24)	0.15	28 (22)	0.21	27 (19)	0.28	26 (17)
211,875	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.07	29 (22)	0.07	29 (22)	0.08	28 (20)
271,875	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.01	29 (21)
331,875	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)
391,875	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)	0.00	30 (30)

the selected model has 17 predictors; among them 10 are true predictors used to generate the response variable. A clear pattern emerging from the table is that the larger the number of candidate predictors is, the more observations are needed for BIC to identify the true model. For G_{60}, it needs about $N = 30{,}000$ to identify the correct model, whereas for G_{270} it needs more than 330,000 observations. Furthermore, when the sample size is smaller than needed, BIC tends to select a model that has a smaller number of variables than the true model and often contains "junk" variables.

The table also shows that the GBK algorithm works consistently as seen from the fact that, for a fixed sample size N, the "Diff" value increases as the number of candidate predictors increases. Note that if a procedure often converges to a local minimum, such a decreasing pattern is not guaranteed.

5. COMPARISON OF BIC AND LASSO

Although a careful and thorough comparison of variable selection criteria is beyond the scope of this article, we use Scenario C of the previous section and the proposed GBK algorithm to make a quick comparison between BIC and LASSO (Tibshirani 1996) similar to that in Zhang et al. (2007). The goal is to demonstrate that the proposed algorithm makes it possible to carry out such a comparison. Note that LASSO and other regularization procedures are based on the "sparseness" condition which is not required by BIC. In the following simulation we use 120 highly correlated candidate predictors. Among them 30 are true predictors (in generating the response variable). This is a mild sparse situation. Specifically, 100 training datasets and one testing dataset are randomly generated from Scenario C with sample size 1,000 and 5,000, respectively, with the 120 predictors given by

$$\{X_1, \ldots, X_{60}, X_1^{(1)}, \ldots, X_{30}^{(1)}, X_1^{(2)}, \ldots, X_{30}^{(2)}\}.$$

For each training set, models are selected by employing BIC and LASSO and then applied to the testing set to compute out-of-sample prediction error rate PE as

$$PE_c = \frac{(\widetilde{Y} - \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}_c)'(\widetilde{Y} - \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}_c)}{\widetilde{Y}'\widetilde{Y}},$$

where $(\widetilde{Y}, \widetilde{\mathbf{X}})$ denotes the testing dataset, and $\widehat{\boldsymbol{\beta}}_c$ is the vector of coefficient estimates based on the selection criterion c, which is either BIC or LASSO. In particular, $\widehat{\boldsymbol{\beta}}_{LASSO}$ is obtained by minimizing

$$(Y - \mathbf{X}\boldsymbol{\beta})'(Y - \mathbf{X}\boldsymbol{\beta})$$

subject to

$$|\boldsymbol{\beta}|'\mathbf{1} \leq \rho|\widehat{\boldsymbol{\beta}}_{full}|'\mathbf{1},$$

where $\mathbf{1} = (1, \ldots, 1)'$ is an $m \times 1$ vector, $\widehat{\boldsymbol{\beta}}_{full}$ is the least squares estimate from the full model, and $\rho \in [0, 1]$ is a tuning constant that controls the size of the LASSO model (Tibshirani 1996; Zhang et al. 2007).

Following Zhang et al. (2007), two models are selected by LASSO for each of the 100 datasets with $\rho = 0.2$ and a ρ chosen from a fivefold cross-validation process, respectively. The ratio of PE, namely PE_{BIC}/PE_{LASSO}, is then computed to measure the performance of BIC and LASSO. Figure 1 presents the histogram of PE_{BIC}/PE_{LASSO} for $\rho = 0.2$. From the figure, a majority of the PE ratios are less than 1 and only about 13% of PE are larger than 1. The result indicates that BIC outperforms LASSO when ρ is fixed at 0.2.

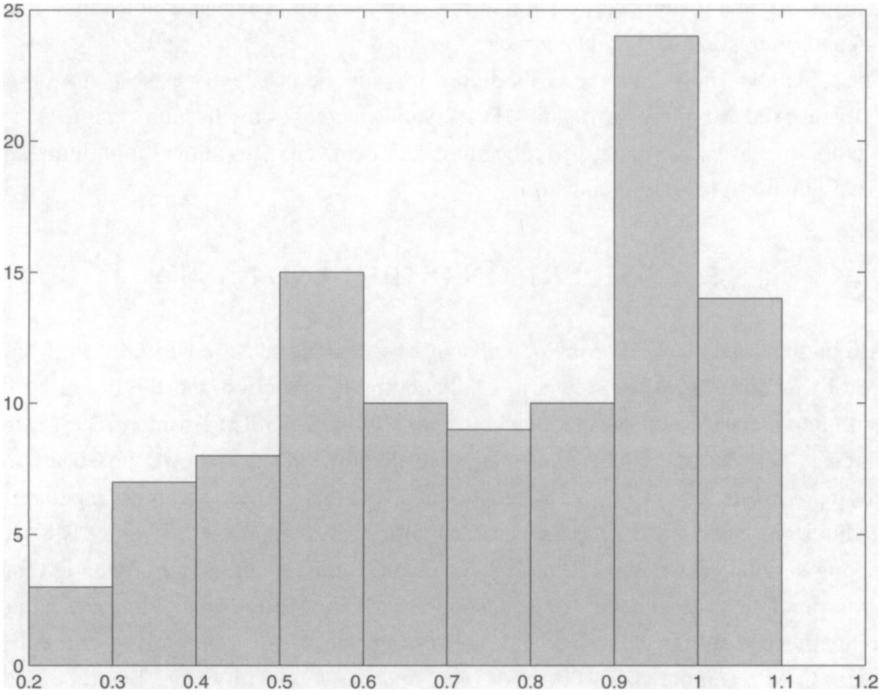

Figure 1. Histogram: Ratio of prediction errors PE_{BIC}/PE_{LASSO}. For LASSO, $\rho = 0.2$ is used. The result is based on 100 training datasets and one forecasting dataset generated from Scenario C.

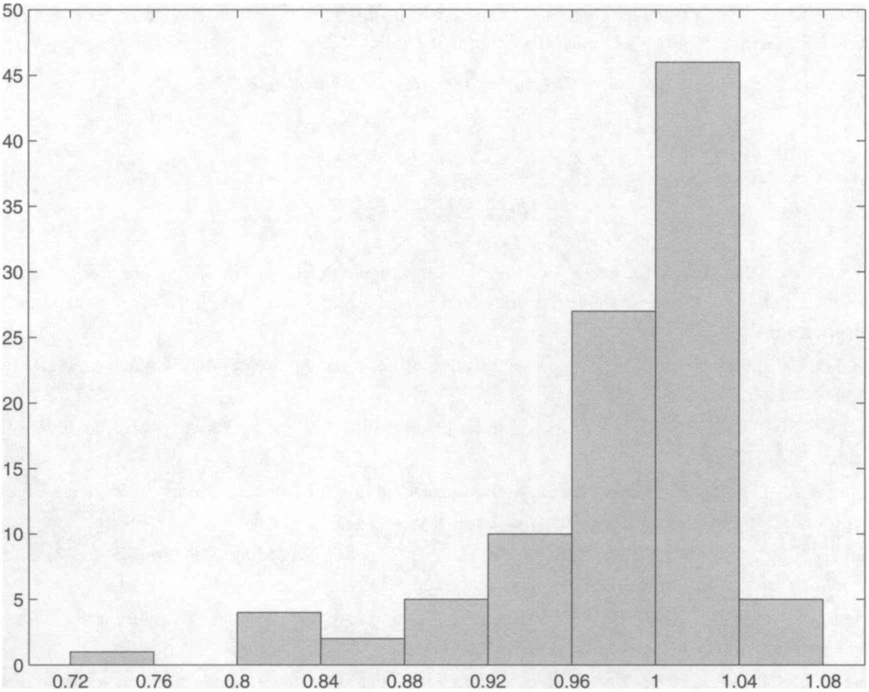

Figure 2. Histogram: Ratio of prediction errors between BIC and LASSO, where ρ of LASSO is chosen by a fivefold cross-validation. The result is based on 100 training datasets and one forecasting dataset generated from Scenario C.

When ρ is chosen by cross-validation, the BIC continues to perform better, even though the performance of LASSO is much improved; see Figure 2. Compared with Figure 1, the minimum *PE* ratio between BIC and LASSO increases from 0.23 to 0.74. In addition, the distribution of *PE* ratio is more concentrated around 1. The average model sizes selected by BIC is 14.9. On average, 7.02 of the selected variables are in the set of true predictors $\{X_{31}, \dots, X_{60}\}$. On the other hand, LASSO selects 15.24 variables on average with 6.4 in the true set when $\rho = 0.2$ and selects 28.29 variables with 13.97 in the true set when ρ is chosen by cross-validation.

SUPPLEMENTAL MATERIALS

Main C++ code: (main.cpp, C++ file)
Sample Parameter Setting File: (sample_par.txt, ascii file)
Test data: (sample_data.txt, ascii file)
Sample Output File: (sample_output.txt, ascii file)

ACKNOWLEDGMENTS

The authors gratefully acknowledge the insightful comments and suggestions from the editor, an associate editor, and two referees, which resulted in significant improvement of the article. The C++ code of the procedure is available at the *Journal of Computational and Graphical Statistics* website. Rong Chen's research was sponsored in part by NSF Grant DMS 0244541 and NIH Grant R01 Gm068958.

[Received December 2006. Revised September 2008.]

REFERENCES

Aeberhard, S., de Vel, O., and Coomans, D. (1993), "Fast Variable Selection," in *Computing Science and Statistics: Proceedings of the 25th Symposium on the Interface*, Fairfax Station, VA: Interface Foundation of North America, pp. 210–212.

Akaike, H. (1973), "Maximum Likelihood Identification of Gaussian Autoregressive Moving Average Models," *Biometrika*, 60, 255–265.

——— (1974), "A New Look at Statistical Model Identification," *IEEE Transactions on Automatic Control*, AC-19, 716–723.

Allen, D. M. (1971), "The Prediction Sum of Squares as a Criterion for Selecting Predictor Variables," Technical Report 23, Dept. of Statistics, University of Kentucky.

Antoch, J. (1986), "Algorithmic Development in Variable Selection Procedures," in *Proceedings in Computational Statistics, 1986*, Heidelberg: Physica-Verlag, pp. 83–90.

Antoniadis, A., and Fan, J. (2001), "Regularization of Wavelests Approximations" (with discussion), *Journal of the American Statistical Association*, 96, 939–967.

Candes, E., and Tao, T. (2007), "The Dantzig Selector: Statistical Estimation When p Is Much Larger Than n," *The Annals of Statistics*, 35, 2313–2351.

Chatterjee, S., Laudato, M., and Lynch, L. A. (1996), "Genetic Algorithms and Their Statistical Applications: An Introduction," *Computational Statistics and Data Analysis*, 22, 633–651.

Donoho, D. (2000), "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality," Aide-Memoire of a lecture at AMS conference *Math Challeges of the 21st Century*.

Draper, H., and Smith, H. (1998), *Applied Regression Analysis* (3rd ed.), New York: Wiley.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 402–451.

Fan, J., and Li, R. (2006), "Statistical Challenges With High Dimensionality: Feature Selection in Knowledge Discovery," in *Proceedings of the International Congress of Mathematicians*, Vol. III, eds. M. Sanz-Sole, J. Soria, J. Varona, and J. Verdera, Zürich: European Mathematical Society, pp. 595–622.

Fernandez, C., Ley, E., and Steel, M. (2001), "Benchmark Priors for Bayesian Model Averaging," *Journal of Econometrics*, 100, 381–427.

Ferri, M., and Piccioni, M. (1992), "Optimal Selection of Statistical Units. An Approach via Simulated Annealing," *Computational Statistics and Data Analysis*, 13, 47–61.

Frank, I. E., and Friedman, J. H. (1993), "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109–148.

Furnival, G. M., and Wilson, R. W. J. (1974), "Regressions by Leaps and Bounds," *Technometrics*, 4, 499–511.

Gatu, C., and Kontoghiorghes, E. J. (2006), "Branch-and-Bound Algorithms for Computing the Best-Subset Regression Models," *Journal of Computational and Graphical Statistics*, 15, 139–156.

George, E. I., and McCulloch, R. E. (1993), "Variable Selection via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881–889.

Haughton, D. M. A. (1988), "On the Choice of a Model to Fit Data From an Exponential Family," *The Annals of Statistics*, 16, 342–355.

Johnsson, T. (1992), "A Procedure for Stepwise Regression Analysis," *Statistical Papers*, 33, 21–29.

Jornsten, R. (2007), "Simultaneous Model Selection via Rate-Distortion Theory, With Applications to Clustering and Significance Analysis," technical report, Dept. of Statistics, Rutgers University.

Mallows, C. L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–675.

Mendieta, G. R., Boneh, S., and Walsh, R. (1994), "A Simulation Study to Evaluate the Performance of a New Variable Selection Method in Regression," in *Computing Science and Statistics. Computationally Intensive Statistical Methods. Proceedings of the 26th Symposium on the Interface*, Fairfax Station, VA: Interface Foundation of North America, pp. 515–518.

Miller, A. (1990), *Subset Selection in Regression*, London: Chapman & Hall.

Miller, T. W., and Ribic, C. A. (1995), "Tree-Structured Variable Selection Methods," in *ASA Proceedings of the Statistical Computing Section*, Alexandria, VA: American Statistical Association, pp. 142–147.

Mitchell, T. J., and Beauchamp, J. J. (1988), "Bayesian Variable Selection in Linear Regression," *Journal of the American Statistical Association*, 83, 1023–1032.

Pauler, D. K. (1998), "The Schwarz Criterion and Related Methods for Normal Linear Models," *Biometrika*, 85, 13–27.

Pearson, K. (1896), "Mathematical Contributions to the Theory of Evolution: III. Regression, Heredity and Panmixia," *Philosophical Transactions of the Royal Society of London*, 187, 253–318.

Rao, C. R., and Wu, Y. (1989), "A Strongly Consistent Procedure for Model Selection in a Regression Problem," *Biometrika*, 76, 369–374.

Ronchetti, E., and Staudte, R. G. (1994), "A Robust Version of Mallows' c_p," *Journal of the American Statistical Association*, 89, 550–559.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Shen, X., and Ye, J. (2002), "Adaptive Model Selection," *Journal of the American Statistical Association*, 97, 210–221.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Ser. B, 58 (1), 267–288.

Tibshirani, R., and Knight, K. (1999), "The Covariance Inflation Criterion for Adaptive Model Selection," *Journal of the Royal Statistical Society*, Ser. B, 61, 529–546.

Zhang, L. J., Lin, T. M., Liu, S. J., and Chen, R. (2007), "Lookahead and Piloting Strategies for Variable Selection," *Statistica Sinica*, 17 (3), 985–1003.

Zheng, X., and Loh, W.-Y. (1997), "A Consistent Variable Selection Criterion for Linear Models With High-Dimensional Covariates," *Statistica Sinica*, 7, 311–325.

Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429.