Lookahead and Piloting Strategies for Variable Selection

Citation

Zhang, Junni L., Ming T. Lin, Jun S. Liu, and Rong Chen. 2007. "Lookahead and Piloting Strategies for Variable Selection." Statistica Sinica 17: 985-1003.

Permanent link

http://nrs.harvard.edu/urn-3:HUL.InstRepos:14121859

Terms of Use

Share Your Story

LOOKAHEAD AND PILOTING STRATEGIES FOR VARIABLE SELECTION

Junni L. Zhang[1], Ming T. Lin[2], Jun S. Liu[3] and Rong Chen[1,2]

[1]*Peking University*, [2]*University of Illinois at Chicago, and* [3]*Harvard University*

Abstract: The traditional variable selection problem has attracted renewed attention from statistical researchers due to the recent advances in data collection, especially in fields such as bioinformatics and marketing. In this paper, we formulate regression variable selection as an optimization problem, propose and study several deterministic and stochastic sequential optimization methods with lookahead. Using several synthetic examples, we show that the stochastic sequential method with lookahead robustly and significantly outperforms a few close competitors, including the popular stepwise methods. When applied to analyze a yeast amino acid starvation microarray experiment, this method can find many transcription factors that are known to be important for yeast to cope with stress and starvation.

Key words and phrases: AIC, Akaike information criterion, BIC, Bayesian information criterion, gene regulation, Gibbs sampler, microarray data, sequential Monte Carlo, TFBM, transcription factor binding-site motif.

1. Introduction

Variable selection in regression modelling is a long-standing problem in statistics. Recently there has been a significant surge of interest in analytically accurate, numerically robust, and algorithmically efficient variable selection methods, largely due to the tremendous advance in data collection techniques such as those in biology and marketing and those associated with the internet.

This article focuses on the variable selection problem in linear regression. Suppose Y, an $n \times 1$ vector, is the response variable, and $\{X_1, \dots, X_p\}$, of which each is an $n \times 1$ vector, is a set of potential predictors. Let $\boldsymbol{r}$ index a subset of $\{1, \dots, p\}$ with $|\boldsymbol{r}|$ denoting the size of subset $\boldsymbol{r}$, and let $\boldsymbol{X_r}$ (an $n \times |\boldsymbol{r}|$ matrix) be the potential predictors in subset $\boldsymbol{r}$. We are interested in finding the "best" subset $\boldsymbol{r}$ to fit the linear model

$$Y = \boldsymbol{X_r}\boldsymbol{\beta_r} + \epsilon, \tag{1}$$

so that a criterion function $H(\boldsymbol{r})$ is optimized. Here $\boldsymbol{\beta_r}$ is a $|\boldsymbol{r}| \times 1$ vector of regression coefficients, ϵ is a $n \times 1$ vector of errors, and $\epsilon \sim N(0, \sigma^2 I)$.

There have been many discussions in the statistical literature concerning the choice of the selection criterion $H(\boldsymbol{r})$. Information-based criteria have been very popular, among which the Akaike information criterion (AIC) (Akaike (1981)) and the Bayesian information criterion (BIC) (Schwarz (1978)) are best known. In linear regression settings, the AIC value for a model with subset $\boldsymbol{r}$ is $n\log(SS_{\boldsymbol{r}})+ 2|\boldsymbol{r}| + \text{constant}$, and its BIC value is $n\log(SS_{\boldsymbol{r}}) + 2|\boldsymbol{r}|\log(n) + \text{constant}$, where $SS_{\boldsymbol{r}}$ is the residual sum of squares (RSS) from model (1). The smaller the AIC or BIC values of a model are, the "better" the model is. Other selection criteria that penalize the number of unknown coefficients in the model (equivalent to the L^0-norm of $\boldsymbol{\beta_r}$) include the adj-R^2, Mallows' C_p (Mallows (1973)), the Risk inflation criterion (RIC) (Foster and George (1994)), etc. Ridge regression (Hoerl and Kennard (1970)) stems from penalizing the L^2-norm of $\boldsymbol{\beta_r}$ which, however, does not necessarily lead to a reduced number of predictors. Recent years have seen the rising popularity of criteria based on an L^1 penalty such as "LASSO", see Tibshirani (1996). These L^1-penalty based methods can do bona fide model selection much as information-based criteria, and are also computationally tractable (Efron, Hastie, Johnstone and Tibshirani (2004)).

The development of Bayesian approaches to address variable selection has opened up a different path. In these approaches, the posterior probability is used as the model selection criterion, and MCMC algorithms are used to sample models with high posterior probabilities. The difficulties lie in prior specifications and in the slow-mixing property of standard MCMC schemes. Fully specified and carefully chosen proper priors have been used in George and McCulloch (1993), Carlin and Chib (1995), and Fernández, Ley and Steel (2001); methods for reducing the influence of the priors on the final selected model have been proposed by Lindley (1968), O'Hagan (1995), Laud and Ibrahim (1995), and Berger and Pericchi (1996). To improve MCMC mixing, Liang and Wong (2000) suggested an evolutionary Monte Carlo method that samples directly from the Boltzmann-like distribution

$$\pi(\boldsymbol{r}) \propto exp\{-H(\boldsymbol{r})/\tau\}, \tag{2}$$

where $H(\boldsymbol{r})$ was taken as Mallows' $C_p(\boldsymbol{r})$. It was also shown in Liang and Wong (2000) that, with an appropriate prior setting, sampling from the posterior distribution is approximately equivalent to sampling from (2) with $\tau = 2$ and $H(\boldsymbol{r}) = C_p(\boldsymbol{r})$.

We note here that the sampling framework based on the Boltzmann distribution (2) can easily be tuned to conduct model optimization and to accommodate any chosen model selection criterion, thus unifying the criterion-based and the Bayesian model-based approaches. On one hand, by letting $H(\boldsymbol{r})$ be the desired variable selection criterion and τ converge to zero, sampling from (2) is

equivalent to minimizing $H(\boldsymbol{r})$; on the other hand, by equating $-H(\boldsymbol{r})/\tau$ to the log-posterior distribution of $\boldsymbol{r}$, which can be derived analytically by integrating out the regression coefficients and the variance parameter when appropriate conjugate priors are used, sampling from (2) is essentially sampling from the posterior model distribution. We focus here on the optimization aspect of the variable selection problem. If model variations should be accounted for, one can conduct further MCMC sampling from (2) using the optimal model we obtained as the starting value.

We content ourselves with addressing the following question: suppose a selection criterion such as *AIC* or *BIC* is given, how should one find the optimal model under the criterion in the space of 2^p possible models. The proposed approach for finding the optimal model is sequential in nature, adopting the lookahead and piloting ideas in sequential Monte Carlo (e.g., see Zhang and Liu (2002)), and some ideas from the Gibbs sampler. Although it is not our intention to study the virtues of different selection criteria in this paper, we show in an example the differences between the models selected by using the BIC criterion and those selected by LASSO.

The rest of the paper is organized as follows. In Section 2, we reformulate the variable selection problem. We then construct several sequential algorithms with lookahead and piloting strategies, including greedy search algorithms in Section 3 and stochastic search algorithms in Section 4. Some of the algorithms are quite standard and serve as preliminary methods and benchmarks. Those using the pilot lookahead strategy are the main proposed algorithms. In Section 5, we present some simulated examples to compare the performance of various methods, and in Section 6 we apply the best of our methods to analyze a yeast microarray data in Gasch, Spellman, Kao, Carmel-Harel, Eisen, Storz, Botstein and Brown (2000). We conclude the article with a brief discussion.

2. Problem Formulation

Let $\boldsymbol{r} = (r_1, \ldots, r_p)$ denote a vector of indicators for the predictors: $r_j = 1$ if X_j is included in the regression model, and $r_j = 0$ otherwise. Let $\mathcal{R}_p$ be the set of all possible such vectors. The variable selection problem based on criterion $H(\boldsymbol{r})$ (e.g., *AIC* or *BIC*) is then to solve

$$\boldsymbol{r}^* = \arg\min_{\boldsymbol{r} \in \mathcal{R}_p} H(\boldsymbol{r}) \tag{3}$$

or, equivalently,

$$\boldsymbol{r}^* = \arg\max_{\boldsymbol{r} \in \mathcal{R}_p} \pi(\boldsymbol{r}), \tag{4}$$

where $\pi(\boldsymbol{r})$ is defined as in (2), in which $\tau > 0$ controls the landscape of the distribution.

In modern applications, the number of potential predictors p is often in the range of hundreds to tens of thousands, making an exhaustive search impossible. The leap and bound algorithm (Furnival and Wilson (1974)) uses a tree structure to enumerate models, and can find the globally optimal one. But, with the current desktop computing power, the leap and bound algorithm can only be applied to problems with $p \leq 60$ and the number of variables in the true model not exceeding 15. Stepwise methods and their variants provide fast but suboptimal solutions. Among them, the forward selection method adds the predictors progressively, with each variable being chosen to provide the largest reduction in RSS (or other criterion, such as $H(\boldsymbol{r})$ itself). Backward elimination starts with the full model and removes one predictor a time according to the insignificance of that variable. The forward-backward method (Efroymson (1960)) is a variation of the forward selection: after each predictor is added to the set of selected predictors, a test is conducted to see whether any of the previously selected predictors can be deleted without appreciably increasing the RSS. Among the series of models obtained using stepwise methods, the optimal model with respect to some adopted criterion is then chosen.

Following the stepwise concept and the growth principle (Rosenbluth and Rosenbluth (1955) and Liu and Chen (1998)), we decompose the problem of determining $\boldsymbol{r}$ into a sequence of simpler problems: determining the indicator for one predictor at a time. Because our procedure is sequential in nature, we found that putting the variables in an appropriate order can significantly increase the efficiency of the procedure. We tested using three methods to pre-arrange the predictors: (a) forward stepwise selection, by which we arrange the predictors according to the order of selection into the model; (b) the backward stepwise elimination, by which we arrange the predictors according to the reverse order of elimination from the model; (c) random ordering. We assume hereafter that the predictors have always been pre-arranged and just use $X_1, X_2, \ldots, X_p$ to denote them.

For a positive integer j, we let $[j] = (j \bmod p)$, i.e., the remainder of j divided by p, and let the index vector $[j_1 : j_2] = ([j_1], [j_1 + 1], \ldots, [j_2])$ for $j_1 \leq j_2$. Hence, for example, $[(p-1) : (p+1)] = (p-1, p, 1)$, achieving a wrap around operation. Furthermore, we let I denote a nonempty subset of $\{1, \ldots, p\}$, let $\boldsymbol{r}_I$ and $\boldsymbol{r}_{-I}$ denote the subsets of $\boldsymbol{r}$ whose indices are in I and I^c, respectively, and let $\boldsymbol{X}_I$ and $\boldsymbol{X}_{-I}$ denote the corresponding subsets of $(X_1, \ldots, X_p)$.

3. Greedy Search with Lookahead

Iterative conditional minimization (ICM) is probably one of the simplest ways of finding a suboptimal solution to (3). It iteratively determines whether to include variable X_j in the model so as to yield a smaller $H(\boldsymbol{r})$, while fixing the

inclusion/exclusion status of other predictors. Formally, this approach iteratively finds the conditional minimum

$$r_j^* = \arg \min_{r_j' \in \{1,0\}} \{H(r_j', \boldsymbol{r}_{-j})\}$$

and uses it to replace the current value of r_j for $j = 1, \ldots, p$. In most problems, ICM tends to be too greedy to result in a good global result.

The lookahead concept from the sequential Monte Carlo literature (Meirovitch (1982), Meirovitch (1985), Chen, Wang and Liu (2000) and Zhang and Liu (2002) has been shown to be useful to reduce greediness and promote adaption. A combination of lookahead and ICM yields the following algorithm, termed the ICM(δ), where δ is the lookahead step. Specifically, at each step, we explore all $2^{\delta+1}$ possible ways of including/excluding predictors $\boldsymbol{X}_{[j:j+\delta]} = \{X_j, X_{[j+1]}, \ldots, X_{[j+\delta]}\}$ while fixing the inclusion/exclusion status of the remaining predictors. The decision of inclusion or exclusion of X_j is based on the best model among the $2^{\delta+1}$ models explored.

Algorithm ICM(δ).

Let the initial variable indicator vector be $\boldsymbol{r}^{(0)}$. For iterations $t = 1, 2, \ldots$ and for $j = 1, \ldots, p$, compute

$$r_j^* = \arg \min_{r_j' \in \{1,0\}} \left[\min_{\boldsymbol{r}_{[j+1:j+\delta]}' \in \{1,0\}^\delta} H\left(r_j', \boldsymbol{r}_{[j+1:j+\delta]}', \boldsymbol{r}_{-[j:j+\delta]}\right) \right], \quad (5)$$

and use it to replace the current value of r_j.

The computational cost of ICM(δ) increases exponentially with δ, which effectively limits the number of steps one can afford to use. On the other hand, a small δ is still too greedy and the algorithm can be easily trapped in a local mode. Note that the idea of lookahead is to explore a large space and make decision based on more information. In order to gather information from more predictors while maintaining low computational cost, we propose to employ a "cheaper" exploration procedure, called the "pilot search," which is similar to the one proposed in the SMC literature (Wang, Chen and Guo (2002) and Zhang and Liu (2002)). In each step of the algorithm, we still consider the $2^{\delta+1}$ possible combinations of the predictors $\boldsymbol{X}_{[j:j+\delta]}$ for δ step lookahead. However, for each combination, the inclusion/exclusion status of the remaining predictors are reevaluated through a fast but greedy process, using ICM(δ^*) with a very small δ^*. Hence one complete interaction of ICM(δ^*) is performed to search through these remaining predictors so as to reach a more (but locally) optimized $H(\boldsymbol{r})$ value, and the ICM(δ^*) steps form the pilot search. The decision of whether to include X_j in the model is made based on the best model explored as follows.

Algorithm ICMP(δ, δ*).

Let $\boldsymbol{r}^{(0)}$ be an initial model. For iterations $t = 1, 2, \ldots$, and for $j = 1, \ldots, p$, compute

$$r_j^* = \arg \min_{r_j' \in \{1,0\}} \left[\min_{\boldsymbol{r}_{[j+1:j+\delta]}' \in \{1,0\}^\delta} H\left(r_j', \boldsymbol{r}_{[j+1:j+\delta]}', \widehat{\boldsymbol{r}}_{-[j:j+\delta]} \right) \right]$$

and use it to replace the current value of r_j, where $\widehat{\boldsymbol{r}}_{-[j:j+\delta]}$ is obtained by **ICM(δ*)** with the initial indicator vector $(r_j', \boldsymbol{r}_{[j+1:j+\delta]}', \boldsymbol{r}_{-[j:j+\delta]})$, and $p - \delta - 1$ iteration steps from $[j + \delta + 1]$ to $[j + p - 1]$ (i.e., going over $\boldsymbol{r}_{-[j:j+\delta]}$).

Both ICM and ICMP are deterministic algorithms. We execute the algorithms until the selection criterion H values does not change in two iterations.

4. Stochastic Optimization with Lookahead

4.1. The stochastic optimization algorithms

In this section we present stochastic counterparts of ICM(δ) and ICMP(δ,δ*), using the Boltzmann-like distribution $\pi(\cdot)$ in (2) as a guide for sampling distribution selection. Similar to ICM(δ), the algorithm of iterative conditional sampling with δ-step lookahead (ICS(δ)) explores all $2^{\delta+1}$ possible models with $\boldsymbol{X}_{[j:j+\delta]}$, while fixing the inclusion/exclusion status of the remaining predictors. These models are then grouped according to whether X_j is included or not. The inclusion/exclusion status of X_j is then sampled according to the total sum of Boltzmann probabilities (2) in each of the two groups.

Algorithm ICS(δ).

Let $\boldsymbol{r}^{(0)}$ be an initial model. For iterations $t = 1, 2, \ldots$, and for $j = 1, \ldots, p$, compute

$$q_0 = \sum_{\boldsymbol{r}_{[j+1:j+\delta]}' \in \{1,0\}^\delta} \exp\left\{ -\frac{1}{\tau} H\left(r_j' = 0, \boldsymbol{r}_{[j+1:j+\delta]}', \boldsymbol{r}_{-[j:j+\delta]} \right) \right\},$$

$$q_1 = \sum_{\boldsymbol{r}_{[j+1:j+\delta]}' \in \{1,0\}^\delta} \exp\left\{ -\frac{1}{\tau} H\left(r_j' = 1, \boldsymbol{r}_{[j+1:j+\delta]}', \boldsymbol{r}_{-[j:j+\delta]} \right) \right\};$$

and draw $r_j^{(t)}$ from Bernoulli($q_1/(q_0 + q_1)$).

Note that when $\delta = 0$, each r_j is updated by drawing from

$$\pi(r_j | \boldsymbol{r}_{-j}) = \frac{\pi(r_j, \boldsymbol{r}_{-j})}{\sum_{r_j' \in \{1,0\}} \pi(r_j', \boldsymbol{r}_{-j})},$$

hence the ICS(0) algorithm is just the systematic-scan Gibbs sampler (Liu (2001)). The ICS(δ) algorithm is a generalized Gibbs sampling procedure that iteratively samples from the conditional distribution

$$\pi(r_j \mid \boldsymbol{r}_{-[j:j+\delta]}), \tag{6}$$

which also forms an irreducible and reversible Markov chain. We show below that this generalized Gibbs sampling method also has π as its invariant distribution.

Theorem 1. *Let $\pi(\boldsymbol{z})$ be the target distribution of interest, with $\boldsymbol{z} = (z_1, \ldots, z_p)$. Suppose we iteratively update each component by*

$$z_j \sim \pi(z_j \mid \boldsymbol{z}_{-[j:j+\delta]}), \tag{7}$$

for $j = 1, \ldots, p$. Then π is the invariant distribution of this procedure, with the caveat that at the last iteration one should update $\boldsymbol{z}_{[j:j+\delta]} \sim \pi(\boldsymbol{z}_{[j:j+\delta]} \mid \boldsymbol{z}_{-[j:j+\delta]})$.

Proof. Consider an "overlapping" systematic-scan Gibbs sampler that updates as follows:

$$\boldsymbol{z}_{[j:j+\delta]} \sim \pi(\boldsymbol{z}_{[j:j+\delta]} \mid \boldsymbol{z}_{-[j:j+\delta]}), \quad j = 1, \ldots, p. \tag{8}$$

It is easy to see that π is the invariant distribution of this algorithm. Sampling from (8) can be done by first generating z_j from (7), and then generating $\boldsymbol{z}_{[j+1:j+\delta]}$ from $\pi(\boldsymbol{z}_{[j+1:j+\delta]} \mid z_j, \boldsymbol{z}_{-[j:,j+\delta]})$. However, the generated value of $\boldsymbol{z}_{[j+1:j+\delta]}$ is not involved in the next step (and all future steps) of the updating scheme (8). Hence, the algorithm based on (8) is identical to that based on (7) except for the last step. Thus, as long as we update at the last step according to (8), $\pi(\cdot)$ is the invariant distribution of the Markov chain based on (7).

Again, due to computational constraints, δ cannot be too large. The piloting idea can be used also in the stochastic search, which gives rise to the following ICSP(δ, δ^*) algorithm.

Algorithm ICSP(δ, δ^*).

Let $\boldsymbol{r}^{(0)}$ be an initial model. For iterations $t = 1, 2, \ldots$, and for $j = 1, \ldots, p$, compute

$$q_0 = \sum_{\boldsymbol{r}'_{[j+1:j+\delta]} \in \{1,0\}^\delta} \exp\left\{ -\frac{1}{\tau} H\left(r'_j = 0, \boldsymbol{r}'_{[j+1:j+\delta]}, \widehat{\boldsymbol{r}}_{-[j:j+\delta]} \right) \right\},$$

$$q_1 = \sum_{\boldsymbol{r}'_{[j+1:j+\delta]} \in \{1,0\}^\delta} exp\left\{ -\frac{1}{\tau} H\left(r'_j = 1, \boldsymbol{r}'_{[j+1:j+\delta]}, \widehat{\boldsymbol{r}}_{-[j:j+\delta]} \right) \right\},$$

*where $\widehat{\boldsymbol{r}}_{-[j:j+\delta]}$ is obtained by **ICM(δ^*)** with the starting vector $(r'_j, \boldsymbol{r}'_{[j+1:j+\delta]}, \boldsymbol{r}_{-[j:j+\delta]})$ and $p - \delta - 1$ iterative steps from $[j + \delta + 1]$ to $[j + p - 1]$. Then sample a new r_j from Bernoulli($q_1/(q_0 + q_1)$).*

In general, ICSP(δ,δ^*) does not form an irreducible and reversible Markov chain. Nevertheless it turns out to be a better optimization algorithm than ICS(δ), as shown by our empirical studies in Section 5. The intuition is that it uses information from more predictors than ICS(δ) at each step, making it less susceptible to local modes.

The choice of the temperature parameter τ in the Boltzmann distribution is important for stochastic optimization methods. When τ is too large, the Boltzmann distribution becomes too flat, resulting in a decreased probability in the region near the global mode. If τ is too small, the stochastic methods can be easily trapped in a local minimum. For both ICS and ICSP algorithms, we use multiple chains with difference temperatures and choose the best solution among them. Alternatively, one could follow the simulated annealing approach (SA; Kirkpatrick, Gerlatt and Vecchi (1983)). We have found in our simulation studies that algorithms with SA are not as effective as the multiple temperature scheme.

Both ICS and ICSP are stochastic search algorithms. We terminate the algorithms when the H value does not decrease in M_{SR} iterations, where M_{SR} is a pre-determined stopping criterion. Based on our experiences we found it sufficient to choose M_{SR} between 10 to 50.

5. Simulation Studies and Performance Comparisons

Throughout the examples, we use the *BIC* as the model selection criterion, and use the notations and the settings listed in Table 1 unless stated otherwise. One hundred data sets were simulated for each example. For *ICS*, we used 20 temperatures given by a geometric series $\tau_v = 10 \log(n) \times 1,000^{-(v-1)/19}$, $v = 1, \ldots, 20$. For *ICSP*, we used the last ten temperatures $\{\tau_{11}, \ldots, \tau_{20}\}$. All algorithms were initiated with $\boldsymbol{r}^{(0)} = (0, \ldots, 0)$, and we studied the effect of pre-ordering of the predictors based on forward selection, backward elimination, and random ordering.

Table 1. Notation for various variable selection methods.

LB	exhaustive search with the leap and bound algorithm
F	forward selection method
B	backward elimination method
FB	forward-backward method
ICM	**ICM**(3)
ICMP	**ICMP**(2, 1)
ICS	**ICS**(3), 20 temperatures, 5 chains for each temperature, $M_{SR} = 10$ iterations
ICSP	**ICSP**(2, 1), 10 temperatures, one chain for each temperature, $M_{SR} = 3$ iterations

5.1. Example 1

We generated $p = 60$ predictors according to

$$X_{10(i-1)+j} = X^*_{10(i-1)+j} + e_0 + e_i, \quad i = 1, \ldots, 6, \ j = 1, \ldots, 10.$$

where the X^*_j ($n \times 1$ vectors), $j = 1, \ldots, 60$, were drawn independently from $N(0, I)$, and $e_0 \sim N(0, I)$ and $e_i \sim N(0, 2I)$ are independent noises. The predictors can be grouped into six clusters. The theoretical correlation coefficient between two predictors in the same cluster is 0.75, and that in different clusters is 0.25. The dependent variable was then generated from the model

$$Y = X_1 + X_2 + X_3 + X_{11} + X_{12} + X_{21} + X_{22} + \epsilon,$$

where $\epsilon \sim N(0, 4^2 I)$. We used $n = 150$ observations.

Table 2 reports the number of times each method reached the true global minimum BIC value found by the leap and bound (LB) algorithm, and the average computation time of each algorithm. Although the standard stepwise procedures (F, B, FB) and ICM needed ignorable amounts of computing time, their performances were clearly inferior to other methods even for this simple example. LB requires significantly more computing time, and this grows exponentially with the size of the true model and the total number of predictors. It also appears that pre-ordering based on both forward selection and backward elimination are better than random ordering, especially for the ICM method.

Table 2. For Example 1, each entry is the number of times each method reached the global minimum BIC value (found by LB). Rows 2–4: three ways of pre-ordering the predictors. Row 5: the average CPU time used by each method.

	LB	F	B	FB	ICM	$ICMP$	ICS	$ICSP$
forward ordering	100	58	53	61	83	99	100	100
backward ordering	100	58	53	61	72	99	100	100
random ordering	100	58	53	61	55	97	97	99
time(sec.)	88.24	-	-	-	-	0.43	10.16	5.57

5.2. Example 2

This example is taken from George and McCulloch (1993). A total of 60 predictors were generated according to $X_j = X^*_j + e$, $j = 1, \ldots, 60$, where the X^*_j's and e were i.i.d. $N(0, I)$, giving rise to a theoretical correlation of 0.5 between all pairs of predictors. Three hundreds observations ($n = 300$) were generated from

$$Y = 1_n + \sum_{s=16}^{30} X_s + 2 \sum_{s=31}^{45} X_s + 3 \sum_{s=46}^{60} X_s + \epsilon,$$

where 1_n is a $n \times 1$ vector with all elements 1, and $\epsilon \sim N(0, 20^2 I)$. Table 3 summarizes the performance of all the methods. Since it is infeasible to obtain the global minimum using the LB algorithm in this example, we report the number of times each method reached the minimum BIC value found by all the tested methods collectively. The results are similar to those of Example 1.

Table 3. For Example 2, each entry is the number of times each method reached the minimum BIC value found by all the methods collectively. Rows 2–4: three ways of pre-ordering the predictors; Row 5: the average CPU time used by each method.

	F	B	FB	ICM	$ICMP$	ICS	$ICSP$
forward ordering	7	58	50	56	98	99	100
backward ordering	7	58	50	80	96	100	100
random ordering	7	58	50	53	96	100	100
time(sec.)	-	-	-	-	1.42	16.61	15.11

5.3. Example 3

In this example, we generated $p = 100$ predictors as follows: we first generated X_j^*'s, V_j's and e independently from $N(0, I)$, $N(0, 2I)$ and $N(0, I)$, respectively, and then took

$$
\begin{aligned}
X_j &= X_j^* + e, & j &= 1, \ldots, 60, \\
X_{60+j} &= X_j + V_j + X_{60+j}^*, & j &= 1, \ldots, 20, \\
X_{80+j} &= X_j - V_j + 0.5 X_{80+j}^*, & j &= 1, 2, \ldots, 20.
\end{aligned}
$$

The dependent variable was generated by

$$
Y = \sum_{s=11}^{20} X_s + \sum_{s=61}^{70} X_s + \sum_{s=81}^{90} X_s + \epsilon,
$$

where $\epsilon \sim N(0, 20^2 I)$. We used $n = 1,000$ observations.

In general, we expect better results with more iterations for each algorithm. For forward ordering, Table 4 reports the number of times each method reaches the minimum BIC value, defined as the one found collectively by all the methods under all different settings, at Mth iteration for various M. The algorithms appear to converge rather quickly, and the boldfaced numbers in the table correspond to the values of M at which the algorithms ceased to make any further improvement. It is instructive to see, for example, that in 98 out of 100 cases the ICSP algorithm reached the minimum BIC value in 3 iterations.

Furthermore, we compared the effect of using different look-ahead sizes δ in these methods. Setting the pilot look-ahead size to $\delta^* = 1$ for $ICMP$ and $ICSP$,

and using forward ordering, we report in Table 5 the number of times each method reached the minimum BIC obtained by all δ and all methods combined, and the average CPU time. We can see that a larger δ does improve the performance, albeit at the expense of higher computational cost. It is also seen that the improvement of $\delta = 3$ over $\delta = 2$ is limited. Hence, we recommend taking δ no greater than 3 in practice.

Table 4. For Example 3 with forward ordering, each entry is the number of times the corresponding method with the corresponding value of M reaches the minimum BIC value. The boldfaced numbers correspond to the values of M needed for convergence.

M	2	3	6	10	15	25	40	60	100
ICM	9	**16**	16	-	-	-	-	-	-
$ICMP$	73	**82**	82	-	-	-	-	-	-
ICS	27	35	46	56	74	82	92	**93**	93
$ICSP$	93	98	99	**100**	100	100	-	-	-

Table 5. For Example 3 with forward ordering, each entry is the number of times each method, with different δ and with the automatic stopping rule, reaches the minimum BIC value obtained by all methods and all δ combined, and the average CPU time used by each method.

		ICM	$ICMP$	ICS	$ICSP$
$\delta = 0$	performance	11	81	75	99
	time(sec.)	-	2.53	32.08	58.89
$\delta = 1$	performance	16	82	85	99
	time(sec.)	-	5.31	61.68	121.32
$\delta = 2$	performance	16	82	91	100
	time(sec.)	-	10.82	135.54	249.33
$\delta = 3$	performance	16	82	93	100
	time(sec.)	-	21.93	304.66	512.84

Using our search algorithms as tools, we can perform comparison studies of different selection criteria. In this example we compared BIC with LASSO (Tibshirani (1996)). LASSO finds coefficients β_j's that minimize

$$\left(Y - \sum_{j=1}^{p} X_j \beta_j\right)^T \left(Y - \sum_{j=1}^{p} X_j \beta_j\right)$$

subject to

$$\sum_{j=1}^{p} |\beta_j| \leq \rho \sum_{j=1}^{p} |\beta_j^{OLS}|,$$

where β_j^{OLS} is OLS coefficients of the full model, and where the parameter $\rho \in [0,1]$ controls the size of model selected by LASSO.

We first used $\rho = 0.20$ so that the models selected by LASSO have similar sizes as those selected by BIC (using the $ICSP$ method). On average, a model selected by BIC has 18.7 variables with 11.9 selected variables in the true model, and a model selected by LASSO has 21.2 variables with 8.9 selected variables in the true model.

We generated a testing set $(\widetilde{X}, \widetilde{Y})$ with $n_1 = 10{,}000$ observations from the same model, and then compared the prediction error rate

$$PE = \frac{(\widetilde{Y} - \sum_{j=1}^p \widetilde{X}_j \beta_j)^T (\widetilde{Y} - \sum_{j=1}^p \widetilde{X}_j \beta_j)}{\widetilde{Y}^T \widetilde{Y}}$$

of models selected by BIC with those selected by LASSO. Figure 1 shows the histogram of PE_{BIC}/PE_{LASSO} over the 100 training sets. We can see that models selected by BIC outperform those selected by LASSO ($PE_{BIC}/PE_{LASSO} < 1$).
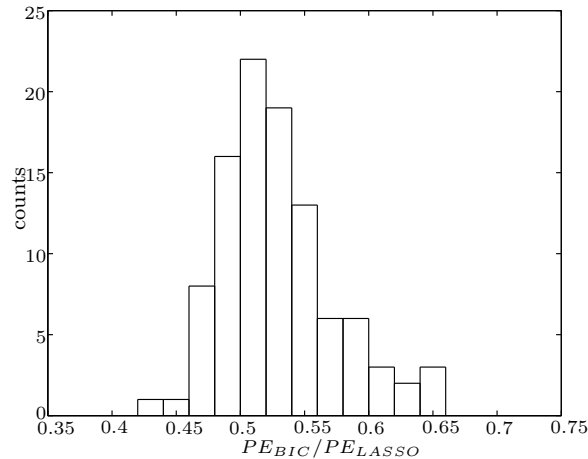
Figure 1. Histogram of PE_{BIC}/PE_{LASSO} for Example 3 ($\rho = 0.20$).

We then used a five-fold cross-validation procedure to choose the best parameter ρ in LASSO for each training set. Now, on average, a model selected by LASSO has 48.6 variables, among which 21.2 variables are in the true model. The histogram of PE_{BIC}/PE_{LASSO} is shown in Figure 2, where it can be seen that models selected by LASSO perform slightly better than those selected by BIC ($PE_{BIC}/PE_{LASSO} > 1$). Note that this comparison is not fair for BIC, because the tuning parameter in the penalty term of BIC is not optimized, while it was in LASSO. Further comparison is beyond the scope of this paper, but we want to emphasize that it is our search algorithms that allow such comparisons to be made.
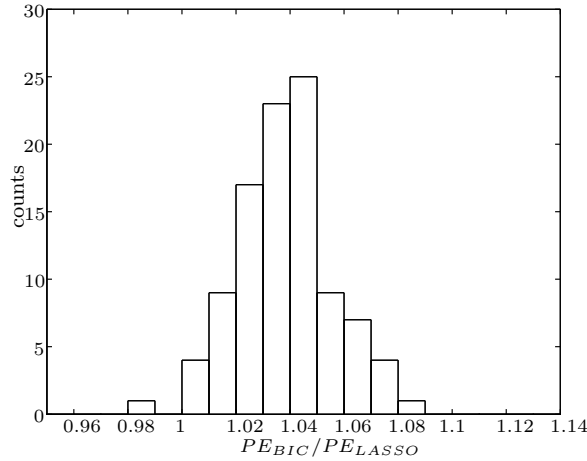
Figure 2. Histogram of PE_{BIC}/PE_{LASSO} for Example 3 (ρ chosen by five-fold cross-validation).

5.4. Example 4

Now we push our methods further to test the case with 1,000 predictors, using an extended model from Fernández, Ley and Steel (2001): we first drew X_j^*, $j = 1, \ldots, 1,000$, and e_0 independently from $N(0, I)$, and then took

$$X_j = X_j^* + e_0, \qquad\qquad\qquad\qquad\qquad j = 1, \ldots, 600,$$
$$X_{600+j} = 0.3X_j + 0.5X_{j+1} + 0.7X_{j+2} + 0.9X_{j+3} + 1.1X_{j+4} + X_{600+j}^*,$$
$$j = 1, \ldots, 100,$$
$$X_{700+j} = 0.3X_j - 0.5X_{j+1} + 0.7X_{j+2} - 0.9X_{j+3} + 1.1X_{j+4} + X_{700+j}^*,$$
$$j = 1, \ldots, 100,$$
$$X_{800+j} = 0.3X_{j+100} + 0.5X_{j+101} + 0.7X_{j+102} + 0.9X_{j+103} + 1.1X_{j+104} + X_{800+j}^*,$$
$$j = 1, \ldots, 100,$$
$$X_{900+j} = 0.3X_{j+100} + 0.5X_{j+101} - 0.7X_{j+102} + 0.9X_{j+103} - 1.1X_{j+104} + X_{900+j}^*,$$
$$j = 1, \ldots, 100.$$

The dependent variable was generated by

$$Y = 10 \times 1_n + \sum_{s=1}^{10} X_{s+600} + \sum_{s=1}^{10} X_{s+700} + \sum_{s=1}^{10} X_{s+800} + \sum_{s=1}^{10} X_{s+900} + \epsilon,$$

where $\epsilon \sim N(0, 30^2 I)$. The sample size is $n = 5,000$. We repeated the experiment 100 times. Performances of different methods with the automatic stopping rule (with $M_{SR} = 20$) are shown in Table 6.

Table 6. The number of times each method with automatic stopping rule reaches the minimum BIC value obtained by all methods combined for Example 4. Rows 2 and 4: different ways of pre-ordering the variables; Rows 3 and 5: the average CPU time used by each method.

	F	B	FB	ICM	$ICMP$	ICS	$ICSP$
forward ordering	0	7	0	13	74	53	95
time(sec.)	-	31.42	-	4.38	998	4097	11389
backward ordering	0	7	0	12	74	58	95
time(sec.)	-	31.42	-	2.92	1187	4186	10966

We also compared the BIC values between the true and the BIC-selected models (using the $ICSP$ method) for all 100 simulated cases, and observed that the selected model was always "better" than the true model. Figure 3 shows the histogram of $BIC_{select} - BIC_{true}$. We also found that the selected model tended to have fewer variables than the true model, indicating that BIC may have penalized the number of parameters too much in this large-p-moderate-n case too much. A systematic study of performances of different model selection criteria in this setting is under way. To our knowledge, the method we developed here is the first one that enables us to conduct such large-scale studies.
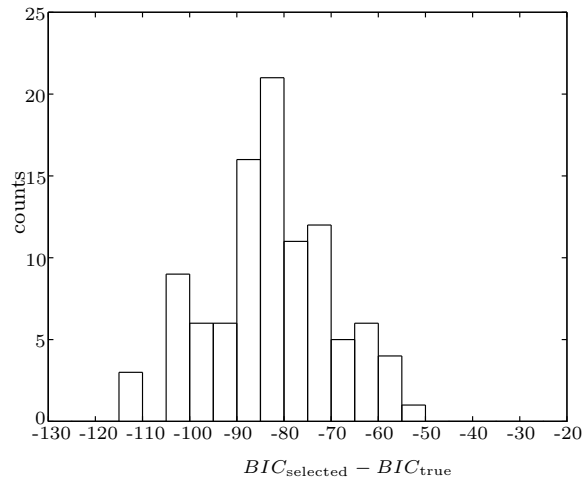

Figure 3. Histogram of the difference between the BIC value of the model selected by $ICSP$ and that of the true model for Example 4.

In all of our simulation examples, ICSP outperformed all other methods, especially when one considers both the performance in finding global modes and the CPU time spent.

6. An Application and Discussion

Conlon, Liu, Lieb and Liu (2003) analyzed the yeast amino acid starvation data reported by Gasch et al. (2000) and intended to discover genomic "features" in the form of short DNA sequence motifs in the promoter region of each gene (about 500-800 base pairs upstream of a gene) that can help explain the change in mRNA expression level of different genes before and after amino acid starvation. The basic biology theory postulates that protein molecules called "transcription factors (TFs)" bind to certain short DNA segments (i.e., motifs) in the promoter region of a gene to help turn the gene's transcription on or off (Jensen, Liu, Zhou and Liu (2004), Biao and van der Laan (2004)). Thus, it is reasonable to suspect that those differentially expressed genes may share certain common motif patterns in their upstream sequences.

Conlon et al. (2003) developed a strategy called "Motif Regressor" for analyzing the data. They first applied a fast motif discovery algorithm MD-scan (Liu, Brutlag and Liu (2002)) to the sequences upstream of yeast genes whose expression values were significantly changed after 30 min of amino acid starvation (Gasch et al. (2000)). MDscan found 414 motifs, with lengths ranging from 5 to 15, from these sequences. Each gene g's upstream region (up to 800 base pairs before the translation start site) is scanned by each motif m and its likelihood score for containing this motif, $S_{g,m}$, is computed. Here g ranges from 1 to 5,970 and m ranges from 1 to 414. Treating gene expression levels as the dependent variable and $S_{g,m}$ as the predictors, they then used stepwise regression to select 25 motifs from the 414 candidates, resulting in a linear model with an R-square of 19.8%. After grouping similar motif patterns together, these 25 motifs were clustered into 15 distinctive patterns, among which 8 are experimentally verified TF binding motifs with biological functions consistent with the cell's regulation of amino acid starvation.

Built on Conlon et al. (2003)'s analysis, we here apply our model selection method to the same data set, with a minor modification. Based on the BIC value, our method, $ICSP$ with automatic stopping rule and forward or backward ordering, selects the final model with 29 motifs, spending 33.3 minutes of CPU time. These motifs can be clustered into 24 distinctive patterns, showing that most of the redundant patterns found by the stepwise regression strategy in Conlon et al. (2003) have been avoided. Among the 24 distinctive motifs, 14 correspond to experimentally known TF binding motifs. The new model has an R-square of 21%, and a BIC value of 51,384.4 (the BIC value of the model selected by the forward-backward method is 51,7387.9). Of interesting, when ordered by the significance of their coefficients, 15 of the 18 most significant motifs correspond to experimentally verified ones, which is also a strong indication of the biological validity of our result. Compared with stepwise regression, our

method is more successful in reducing highly correlated covariates, which then gives room for other biologically important factors to be selected. The motifs we selected are displayed in Figure 4 as sequence logos.

Figure 4. Motifs selected by $ICSP$ (with automatic stopping rule). In the first column, the first number is the rank of the motif, ordered by the magnitude of t-statistics, and the second number is the index of its group. The second column is the sequence logo of the motif. The third column lists the names of experimentally known TF binding motifs that match with the selected ones. A negative motif coefficient suggests that the corresponding TF plays a repression role.

Motif #,Group	Motif Sequence Logo	Known Motif Names	Motif Coefficients	t Statistics	p-value
1,1		M3B	-0.03288	-12.37	<0.0001
2,2		RAP1	-0.03065	-10.13	<0.0001
3,3		M3A	-0.02311	-7.18	<0.0001
4,3		M3A	-0.02231	-5.77	<0.0001
5,4		MET4	0.03099	5.74	<0.0001
6,5		STRE	0.03684	5.57	<0.0001
18,5		STRE	0.01967	3.78	0.0002
14,5		STRE	0.03322	4.30	<0.0001
11,6		PHO4	0.03027	4.45	<0.0001
7,6		PHO4	0.03333	5.07	<0.0001
15,6		PHO4	-0.03174	-4.19	<0.0001
8,7			-0.02600	-4.93	<0.0001
9,8		GCN4	0.02647	4.79	<0.0001
10,9		URS1	0.03401	4.63	<0.0001
12,10			-0.03424	-4.41	<0.0001
13,11		REB1	-0.01709	-4.40	<0.0001
16,12		NRG1	0.03148	4.17	<0.0001
17,13			-0.02670	-4.11	<0.0001
19,14			-0.02566	-3.74	0.0002
20,15		HSF1	-0.02743	-3.72	0.0002

Motif #,Group	Motif Sequence Logo	Known Motif Names	Motif Coefficients	t Statistics	p-value
21,16		GLN3	0.02816	3.69	0.0002
22,17		MET31	0.01849	3.61	0.0003
23,18			-0.02221	-3.46	0.0005
24,19			0.02529	3.42	0.0006
25,20			0.02120	3.33	0.0009
26,21			0.02243	3.32	0.0009
27,22			-0.02022	-3.26	0.0011
28,23			-0.02053	-3.24	0.0012
29,24		RFX1	-0.02139	-3.10	0.0019

The new known motif patterns we discovered correspond to the binding sites of transcription factors REB1, GLN3, NRG1, HSF1, Met31, and Rfx1. It is known that HSF1p (Heat shock transcription factor 1 protein) plays a major role in stress protection. It represses genes involved in growth and differentiation, and has an elevation in expression during stress conditions (so that its targets are repressed). Our model infers a negative coefficient for the HSF1p targets, which fits perfectly with HSF1's biological function. REB1p regulates mostly genes involved in cell growth, and the negative coefficient for its targets suggests that a cell slows down its growth during starvation. GLN3p plays a role in nitrogen catabolite repression. Mutant GLN3p yeast is viable, but does not grow well on poor nitrogen sources. NRG1 is a glucose-dependent repressor, and is important to stress responses. Met31p (Blaiseau, Isnard, Surdin-Kerjan, and Thomas (2002)) regulates expression of the methionine biosynthetic genes, and Rfx1p is a repressor of DNA damage-inducible genes. Our model suggests that during amino acid starvation, the cell becomes more active in producing methionine, one of the amino acids, and represses the DNA-repair mechanism.

7. Summary

In this paper, we proposed several lookahead and piloting strategies to tackle the variable selection problem. Using several synthetic examples and an application, we demonstrated the superiority of these algorithms over deterministic stepwise methods. The optimization framework given in Section 2-4 can work for any variable selection criterion, and therefore is not comparable with algorithms that are designed for a specific selection criterion, such as LASSO (Tibshirani (1996)) or LARS (Efron et al. (2004)). The methodology we developed here can

also be extended, for example, to dealing with the variable selection problem for generalized linear models such as logistic regression. Because *BIC* was not designed for the case when the sample size n is smaller than the number of predictors p, our search algorithms, when combined with *BIC*, cannot cope with the $n < p$ case, but they can potentially be applied in combination with variable selection criteria that have good properties in the $n < p$ case. More research is needed to verify this speculation.

Acknowledgement

References

Akaike, H. (1981). Likelihood of a model and information criteria. *J. Econ.* **16**, 3-14.

Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *J. Amer. Statist. Assoc.* **91**, 109-122.

Biao, X. and van der Laan, M. J. (2004). A statistical method for constructing transcriptional regulatory networks using gene expression and sequence data. *University of California, Berkeley.*

Blaiseau, P. L., Isnard, A. D., Surdin-Kerjan, Y. and Thomas, D. (2002). Met31p and Met32p, two related zinc finger proteins, are involved in transcriptional regulation of yeast sulfur amino acid metabolism. *Mol. Cell. Biol.* **17**, 3640-3648.

Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov Chain Monte Carlo methods. *J. Roy. Statist. Soc. Ser. B* **57**, 473-484.

Chen, R., Wang, X. and Liu, J. S. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Trans. Information Theory* **46**, 2079-2094.

Conlon, E. M., Liu, X. S., Lieb, J. D. and Liu, J. S. (2003). Integrating regulatory motif discovery and genome-wide expression analysis. *Proc. Nat'l Acad. Sci.* **100**, 3339-3344.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.* **32**, 407-499.

Efroymson, M. A. (1960). Multiple regression analysis. In *Mathematical Methods for Digital Computer* (Edied by A. Ralston and H. S. Wilf). Wiley, New York.

Fernández, C., Ley, E. and Steel, M. F. J. (2001). Benchmark priors for Bayesian model averaging. *J. Econ.* **100**, 381-427.

Foster, D. P. and George, E. I. (1994). The risk inflation criterion for multiple regression. *Ann. Statist.* **22**, 1947-1975.

Furnival, G. M. and Wilson, R. W. J. (1974). Regressions by leaps and bounds. *Technometrics* **16**, 499-511.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D. and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell* **11**, 4241-4257.

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* **88**, 881-889.

Hoerl, A. and Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12**, 55-67.

Jensen, S. T., Liu, X. S., Zhou, Q. and Liu, J. S. (2004). Computational discovery of gene regulatory binding motifs: A Bayesian perspective. *Statist. Sci.* **19**, 188-204.

Kirkpatrick, S., Gerlatt, C. D. and Vecchi, M. P. J. S. (1983). Optimization by simulated annealing. *Science* **220**, 671-680.

Laud, P. W. and Ibrahim, J. G. (1995). Predictive model selection. *J. Roy. Statist. Soc. B* **57**, 247-262.

Liang, F. and Wong, W. H. (2000). Evolutionary Monte Carlo: applications to C_p model sampling and change point problem. *Statist. Sinica* **10**, 317-342.

Lindley, D. V. (1968). The choice of variables in multiple regression (with discussion). *J. Roy. Statist. Soc. Ser. B* **30**, 31-66.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing.* Springer, New York.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.* **93**, 1032-1044.

Liu, S. X., Brutlag, D. L. and Liu, J. S. (2002). An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray expriments. *Nat. Biotechnol.* **20**, 835-839.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics* **15**, 661-676.

Meirovitch, H. (1982). A new method for simulation of real chains: scanning future steps. *J. Phys. A* **15**, 735-742.

Meirovitch, H. (1985). Scanning method as an unbiased simulation technique and its application to the study of self-attracting random walks. *Physic. Rev. A* **32**, 3699-3708.

O'Hagan, A. (1995). Fractional Bayes factors for model selection (with discussion). *J. Roy. Statist. Soc. Ser. B* **57**, 99-138.

Rosenbluth, M. N. and Rosenbluth, A. W. (1955). Monte Carlo calculation of the average extension of molecular chains. *J. Chem. Phys.* **23**, 356-359.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6**, 461-464.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58**, 267-288.

Wang, X., Chen, R. and Guo, D. (2002). Delayed pilot sampling for mixture Kalman filter with application in Fading Channels. *IEEE Trans. Sig. Proc.* **50**, 241-264.

Zhang, J. L. and Liu, J. S. (2002). A new sequential importance sampling method with its application to the 2D hydrophobic-hydrophilic model. *J. Chem. Phys.* **117**, 3492-3498.

Department of Business Statistics and Econometrics, Guanghua School of Management, Peking University, Beijing 100871, P. R. China.

E-mail: zjn@gsm.pku.edu.cn

Department of Information and Decision Sciences (M/C 294), College of Business Administration, The University of Illinois at Chicago, 601 Morgan Street, Chicago, IL 60607, U.S.A.

E-mail: linming@uic.edu

Department of Statistics, Harvard University, 1 Oxford St., Cambridge, MA 02138, U.S.A.

E-mail: jliu@stat.harvard.edu

Department of Information and Decision Sciences (M/C 294), College of Business Administration, The University of Illinois at Chicago, 601 Morgan Street, Chicago, IL 60607, U.S.A.

E-mail: rongchen@uic.edu