A Split-and-Merge Bayesian Variable Selection Approach for Ultra-high dimensional Regression

Faming Liang

Texas A&M University

July 8, 2014
Abstract

This talk presents a Bayesian variable selection approach for ultra-high dimensional linear regression based on the strategy of split-and-merge. The proposed approach consists of two stages: (i) split the ultra-high dimensional dataset into a number of lower dimensional subsets and select relevant variables from each of the subsets, and (ii) aggregate the variables selected from each subset and then select relevant variables from the aggregated dataset. Since the proposed approach has an embarrassingly parallel structure, it can be easily implemented in a parallel architecture and applied to the big data problems with millions or more of explanatory variables. Under mild conditions, we show that the proposed approach is consistent; that is, the true explanatory variables can be correctly identified by the proposed approach as the sample size becomes large. Extensive comparisons of the proposed approach have been made with penalized likelihood approaches, such as Lasso, elastic net, SIS and ISIS. The numerical results show that the proposed approach generally outperforms the penalized likelihood approaches: The models selected by the proposed approach tend to be more sparse and closer to the true model.
The problem

Consider the problem of variable selection for the linear regression model:

$$y = X \beta^* + \varepsilon,$$  \hspace{1cm} (1)

where $y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is the response vector, $n$ is the sample size, $X = (x_1, \ldots, x_{P_n}) \in \mathbb{R}^{n \times P_n}$ is the design matrix, $P_n$ is the number of predictors, $\beta^*$ is the vector of unknown regression coefficients, and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T \sim N_n(0, \sigma^* 2 I_n)$ is the Gaussian random error.

- $P_n$ is a non-decreasing function of $n$.
- $P_n \gg n$. 
Penalized Likelihood Approach

This approach is to find a model $\xi$ that minimize the penalized likelihood function

$$-\log f(y; \mathbf{X}_\xi, \beta_\xi) + p_\lambda(\beta_\xi),$$

where $p_\lambda(\cdot)$ is the penalty function and the tuning parameter $\lambda$ can be determined through cross-validation.

- Lasso (Tibshirani, 1996)
- Adaptive Lasso (Zou, 2006)
- Smoothly clipped absolute deviation (SCAD, Fan and Li, 2001)
- Minimax concave penalty (MCP, Zhang, 2010)
- Extended BIC (Chen and Chen, 2008)
- Sure independence screening (SIS, Fan and Lv, 2008)
Bayesian approaches advance either in a sophisticated model search algorithm or in a prior specification that is particularly suitable for high dimensional problems:

- evolutionary stochastic search (Bottolo and Richardson, 2010)
- Lasso prior (Park and Casella, 2008)
- horseshoe prior (Carvalho et al., 2010)
- Ising prior (Li and Zhang, 2010)
- nonlocal prior (Johnson and Rossel, 2012; Johnson, 2012)
Bayesian Approach

Two issues need to be addressed for Bayesian approaches:

- Variable selection Consistency (Jiang, 2007; Bondell and Reich, 2012; Liang et al., 2013)
- Computational demand

A major concern with the Bayesian approach is its high computational demand. Since the volume of the model space increases geometrically with the dimension $P_n$, the CPU time for a Bayesian approach should increase accordingly or even faster.
Split-and-Merge (SaM) Approach

(i) Split the high dimensional data into a number of lower dimensional subsets and perform Bayesian variable selection for each subset.

(ii) Aggregate the variables selected from each subset and perform Bayesian variable selection for the aggregated dataset.

We will show

- Consistency: True model will be identified in probability 1 as the sample size becomes large.
- It has a favorable time complexity compared to the conventional Bayesian approaches and even penalized likelihood approaches.
Advantages of SaM

- Since SaM has an embarrassingly parallel structure, it can be easily implemented in a parallel architecture and applied to the big data problems with millions or more of predictors. This has been beyond the ability of conventional Bayesian approaches, as they directly work on the full dataset.
- SaM shares the same asymptotics, such as sure screening and model selection consistency, as SIS (Fan and Lv, 2008).
SaM versus SIS

• They are similar in spirit: Their first stage serves the purpose of dimension reduction, screening out uncorrelated predictors; and the second stage refines the selection of predictors.

• Compared to SIS and ISIS, SaM can often lead to more accurate selection of true predictors. SaM screens uncorrelated predictors based on the marginal inclusion probability, which has incorporated the joint information of all predictors contained in a subset. While SIS makes use of only the marginal information of each predictor.

• SIS needs to sort the marginal utilities of all predictors, while SaM never needs to process all predictors altogether.
Sparsity Assumption

- $P_n \succ n^\theta$ for some $\theta > 0$, where $b_n \succ a_n$ means $\lim_{n \to \infty} a_n/b_n = 0$.

- Weak sparsity: most components of $\beta^*$ are very small in magnitude such that

$$\lim_{n \to \infty} \sum_{j=1}^{P_n} |\beta_j^*| < \infty,$$

where $\beta_j^*$ denotes the $j$th entry of $\beta^*$.

- Strict sparsity: Most components of $\beta^*$ are zero such that

$$\lim_{n \to \infty} \sum_{j=1}^{P_n} 1\{\beta_j^* \neq 0\} < \infty.$$
Prior Specification

- We assume that each predictor has a prior probability $\lambda_n = r_n/P_n$, independent of other predictors, to be included in the model $\xi$. Further, we impose a constraint on the model size such that $|\xi| \leq \bar{r}_n$, where the upper bound $\bar{r}_n$ is pre-specified. Then

$$
\pi(\xi) \propto \lambda_n^{|\xi|}(1 - \lambda_n)^{P_n-|\xi|}I[|\xi| \leq \bar{r}_n].
$$

- We let the variance $\sigma^2$ be subject to an Inverse-Gamma prior distribution with the hyper-parameters $a_0$ and $b_0$; i.e., $\sigma^2 \sim IG(a_0, b_0)$.

- Conditioned on the model $\xi$ and $\sigma^2$, we let $\beta_\xi$ be subject to a Gaussian prior,

$$
\beta_\xi | \xi, \sigma^2 \sim N_{|\xi|}(0, \sigma^2 V_\xi),
$$

where $V_\xi$ is a positive definite matrix depending on $\xi$. 
The Posterior

Integrating out $\beta_\xi$ and $\sigma^2$, we get the posterior of $\xi$:

$$\pi(\xi|D_n) \propto (r_n/P_n)^{|\xi|} (1 - r_n/P_n)^{P_n - |\xi|} \frac{\sqrt{\det(V_\xi^{-1})}}{\sqrt{\det(X_\xi^T X_\xi + V_\xi^{-1})}}$$

$$\times \left\{ 2b_0 + y^T (I - X_\xi (X_\xi^T X_\xi + V_\xi^{-1})^{-1} X_\xi^T ) y \right\}^{-n/2 - a_0} I[|\xi| \leq \bar{r}_n].$$

(5)

If we set $V_\xi^{-1} = X_\xi^T X_\xi / n$, $a_0 = b_0 \approx 0$, and $r_n = P_n^{1-\gamma}$ for a constant $\gamma \in (0, 1)$, then $\log \pi(\xi|D_n) \approx -EBIC(\xi) + C$ for some constant $C$ independent of $\xi$. 
Multiplicity Control

- Fully Bayesian variable selection approach: $\lambda_n$ is usually subject to a Beta prior, which will induce an automatic multiplicity adjustment for variable selection: The penalty for adding an extra variable increases as $P_n$ increases.
- We assume $\lambda_n \to 0$ as $P_n \to \infty$, which provides an automatic mechanism of multiplicity control for SaM.
Posterior Consistency for Correctly specified Models

**Theorem 1** [Posterior Consistency] Assume that the dataset $D_n$ is drawn from model (1) and all predictors $|x_i| \leq 1$. If there exists a sequence $\{\epsilon_n\} \in (0, 1)$ such that $n\epsilon_n^2 > 1$, and the following conditions hold:

\[
\Delta(r_n) \prec \epsilon_n, \\
\bar{r}_n \ln(1/\epsilon_n^2) \prec n\epsilon_n^2, \\
\bar{r}_n \ln(P_n) \prec n\epsilon_n^2, \\
\bar{r}_n \ln(n\epsilon_n^2 \tilde{B}_n) \prec n\epsilon_n^2, \\
1 \leq r_n \leq \bar{r}_n < P_n, \\
r_n \prec P_n, \\
B(r_n) \prec n\epsilon_n^2, \tag{12}
\]

then the posterior consistency holds, i.e., there exists a constant $c_1 > 0$ such that

\[
P^*\{\pi[d(f, f^*) > \epsilon_n | D_n] > e^{-c_1 n\epsilon_n^2}\} < e^{-c_1 n\epsilon_n^2}, \tag{13}\]

for sufficiently large $n$, where $d(\cdot, \cdot)$ denotes the Hellinger distance of $f$ and $f^*$, $f^*$ denotes the true density of data generation, and $f$ is the density for a model simulated from the posterior,
Posterior Consistency for Correctly specified Models

Theorem 1 assumes that all the predictors are uniformly bounded. This condition can be relaxed as follows:

\( (A_1) \) There exist some constants \( M > 0 \) and \( \delta > 0 \) such that for any subvector of \( x, (x_{j_1}, \ldots, x_{j_r})^T \) with \( 1 \leq j_1 < \cdots < j_r \leq P_n, \)
\( r \leq \bar{r}_n, \) and \( |a_i| \leq \delta \) for \( i = 1, \ldots, r, \) we have

\[
E \exp \left\{ \left( \sum_{i=1}^{r} \frac{a_i x_{j_i}}{r} \right)^2 \right\} \leq \exp(M).
\]

It controls the tail distribution of \( x \) such that the marginal distribution of \( y \) does not change dramatically with respect to the change of \( \beta. \)
If \( x \) follows a Gaussian process with zero mean and variance=1, then

\[
E \exp \left\{ \left( \sum_{i=1}^{r} a_i x_{j_i} \right)^2 / r^2 \right\} \leq (1 - 2\delta^2)^{-1/2}.
\]
Theorem 2  Assume that the data set $D_n$ is generated from model (1) and all the predictors satisfy condition $(A_1)$. Given a sequence $\{\epsilon_n\}$, $\epsilon_n \rightarrow 0$, and $n\epsilon_n^2 \rightarrow \infty$. If the conditions (7)-(12) of Theorem 1 hold, the model is strictly sparse, and $r_n \succ 1$, then the posterior consistency still holds, i.e., there exists a constant $c_1 > 0$ such that for sufficiently large $n$,

$$P^*\left\{\pi[d(f, f^*) > \epsilon_n | D_n] > e^{-c_1 n\epsilon_n^2}\right\} < e^{-c_1 n\epsilon_n^2}.$$
Posterior Consistency for Correctly specified Models

If the distribution of a predictor is heavy-tailed, then the condition \((A_1)\) fails. To address this issue, a much weaker condition is proposed below:

\((A_2)\) There exist some constants \(M > 0\) and \(\delta > 0\) such that for any subvector of \(x\), \((x_{j_1}, \ldots, x_{j_r})^T\) with \(1 \leq j_1 < \cdots < j_r \leq P_n\), \(r \leq \bar{r}_n\), and \(|a_i| \leq \delta\) for \(i = 1, \ldots, r\), we have

\[
E \left( \sum_{i=1}^{r} a_i x_{j_i} / r \right)^2 \leq M.
\]

The condition \((A_2)\) only imposes conditions on the second moments of predictors. Then a slightly weaker result of Theorem 2 can be obtained as follows:

\[
\lim_{n \to \infty} P^* \{ \pi[d(f, f^*) > \epsilon_n | D_n] > e^{-c_1 n \epsilon_n^2} \} = 0. \quad (14)
\]
Identifiability Condition

It controls the severeness of multicollinearity through the sequence \( \{ \epsilon_n \} \):

\((B_1)\) A predictor \( x_k \) is said to be identifiable among all other predictors, if, for any \( 1 \leq j_1, \ldots, j_{\bar{r}_n} \leq P_n \) \( (j_i \neq k \text{ for all } i) \) and \( b_i \in \mathbb{R} \),

\[
E \exp \left\{ - (x_k + \sum_{i=1}^{\bar{r}_n} b_i x_{j_i})^2 \right\} \leq 1 - \delta_n, \text{ and } \delta_n \succ \epsilon_n^2,
\]

where \( x_l \) denotes a generic observation of the predictor \( x_l \). This condition states that if a predictor \( x_k \) is identifiable, then there does not exist a linear combination of other predictors in \( X \) which can mimic it.
Theorem 3 [Sure Screening] Assume that all the conditions of Theorem 2 hold. If a true predictor $x_t \ (t \in t)$ is identifiable, then

$$P^* \{ \pi [t \in \xi | D_n] < 1 - e^{-c_1 n} \epsilon_n^2 \} < e^{-c_1 n} \epsilon_n^2,$$

where $\xi$ denotes a model sampled from the posterior distribution. Furthermore, if condition $(A_2)$ holds, then the weaker convergence holds:

$$\lim_n P^* \{ \pi [t \in \xi | D_n] < 1 - e^{-c_1 n} \epsilon_n^2 \} = 0.$$
Variable screening for misspecified models

Let \( D_n^s = \{y, X_s\} \) denote a fixed subset of observations, where \( s \subset \{1, 2, \ldots, P_n\} \) and \( X_s \) contains only \( s = |s| < P_n \) predictors with the indices belonging to \( s \). Assume that \( X_s \) does not include all of the true predictors of model (1). Therefore, the model

\[
y = X_s \beta_s + \varepsilon
\]

is misspecified.

Let \( f_0 \) denote the minimization point of the Kullback-Leibler divergence in \( \mathcal{P}_s \), i.e.

\[
f_0 = \arg \min_{f \in \mathcal{P}_s} \int \ln(f^*/f)f^*.
\]

The idea is to show that the density \( f_0 \) can be consistently estimated by the models sampled from the posterior \( \pi(\xi | D_n^s) \) as \( n \to \infty \).
Variable screening for misspecified models

Parallel to \( (A_1) \), we introduce the following condition regarding the range of \( x_i \)'s.

\[ (A'_1) \] There exists a constant \( \delta > 0 \) such that for any \( a = (a_1, \ldots, a_s)^T \in \mathbb{R}^s \), with \( |a_i| \leq \delta \) for \( i = 1, \ldots, s \), we have

\[
E[(a^T x_s)^2] < \infty,
\]

where \( x_s \) denote a generic observation of the predictors included in \( X_s \).

**Theorem 4** (Posterior consistency for misspecified models)
Assume the condition \( (A'_1) \) holds for a given subset \( s \). Under the prior setting as described above, for any \( \epsilon > 0 \),

\[
\pi(\{f \in \mathcal{P}_s : d(f_0, f) > \epsilon\}|D_n^s) \to 0, \quad \text{a.s.} \quad (16)
\]
as \( n \to \infty \), where \( f_0 \) is as defined in (15), and \( f \) is a parameterized density proposed from posterior.
Variable screening for misspecified models

Let $\tilde{s}$ denote the subset of $s$ corresponding to nonzero entries of $\beta_0$.

**Theorem 5** Assume the conditions of Theorem 4 hold. If $x_s$ does not have exact multicollinearity between variables, i.e., there does not exist a nonzero vector $a \in \mathbb{R}^s$ such that $P(a^T x_s = 0) = 1$, then for the posterior probability of model $\xi$, conditioned on the subset $D_n^s$, we have

$$\pi(\tilde{s} \subset \xi | D_n^s) \xrightarrow{p} 1.$$
**SaM Approach**

1. **Split** $P_n$ predictors into $K_n$ groups, $s_1, \ldots, s_{K_n}$, with $\max_{i=1,\ldots,K_n} |s_i| \leq s$ for a pre-specified value of $s$.

2. **Stage I**: Select predictors from each subset $D_n^{s_i} = \{y, X_{s_i}\}$, $i = 1, \ldots, K_n$, at a FDR level of $\alpha_1$, subjecting to the prior setting: $\pi(\sigma) \sim IG(a_0, b_0)$,
   \[
   \pi(\xi) = \lambda_{\bar{s}} |\xi| (1 - \lambda_s)^{|\xi|-|\xi|} I(|\xi| \leq \bar{s}),
   \]
   and $\pi(\beta_\xi | \xi, \sigma) \sim N(0, \sigma^2(X_\xi^T X_\xi + \tau I)/n)$, where $\bar{s}$ denotes the upper bound of the size of the models considered for each subset data. Let $\tilde{s}_1, \ldots, \tilde{s}_{K_n}$ denote the sets of selected predictors from the $K_n$ subsets, respectively.

3. **Stage II**: Merge the sets $\tilde{s}_1, \ldots, \tilde{s}_{K_n}$ into a single set $\tilde{S} = \bigcup_{i=1}^{K_n} \tilde{s}_i$, and define $p_n = |\tilde{S}|$. Perform Bayesian variable selection on the aggregated dataset $D_n^{\tilde{S}} = \{y, X_{\tilde{S}}\}$ at a FDR level of $\alpha_2$, subjecting to the prior: $\pi(\sigma) \sim IG(a_0, b_0)$,
   \[
   \pi(\xi) = (r_n/p_n)^{|\xi|} (1 - r_n/p_n)^{p_n-|\xi|} I(|\xi| \leq \bar{r}_n),
   \]
   and $\pi(\beta_\xi | \xi, \sigma) \sim N(0, \sigma^2(X_\xi^T X_\xi + \tau I)/n)$.
SaM Simulation

- The stochastic approximation Monte Carlo (SAMC) algorithm (Liang et al., 2007) was used to simulate from the posterior in SaM.
- SAMC belongs to the class of adaptive MCMC algorithms; it adapts the invariant distribution at each iteration. As explained in Liang, the self-adjusting mechanism of the invariant distribution enables SAMC to be immune to local trap problems.
- As shown in Liang (2009), SAMC is essentially a dynamic importance sampling algorithm, and quantities of interest can be estimated by weighted averaging its importance samples.
Hyperparameter Setting

• Our choice $\lambda_n = P_n^{-\gamma}$, where $\gamma$ is the hyperparameter to set.
• We suggest

$$
\gamma = \inf \left\{ \tilde{\gamma} : \arg \max_{|\xi|} \pi(|\xi||D_n, \tilde{\gamma}) = |\xi_{MAP, \tilde{\gamma}}| \right\}, \tag{17}
$$

that is, choosing the minimum value of $\gamma$ such that the mode of $\pi(|\xi||D_n)$ coincides with the size of the MAP model $\xi_{MAP}$.
• In practice, one may try a sequence of $\gamma$ values, and then choose the smallest one for which the mode of posterior of model size coincides with the size of the MAP model.
Example 1

This example consists of multiple datasets with different values of $n$ ranging from 20 to 120. For each value of $n$, we set $P_n = n^{1.5}$ and simulated 100 datasets independently. For each dataset, the design matrix $X$ was generated from a multivariate normal distribution. The variance of each column of $X$ was set to be 1, and the correlation coefficient between different columns of $X$ was set to be 0.25, which represents a strong correlation for real gene expression data. For each dataset, we set $\sigma^* = 1$ and chose the first 5 columns of $X$ as the true predictors with the regression coefficients being 0.7, 0.9, 1.1, 1.3, 1.5, respectively.
Example 1

Figure: Marginal inclusion probabilities of six predictors for the case of Example 1 that the mutual correlation coefficient between different predictors is 0.25.
Example II

This example illustrates the SaM approach. It consists of 100 simulated datasets, each consisting of \( n = 150 \) observations and \( P_n = 1000 \) predictors. For each dataset, the design matrix \( \mathbf{X} \) was generated from a multivariate normal distribution. The first 100 columns of \( \mathbf{X} \) are mutually correlated with an equal correlation coefficient of 0.25, and independent of the rest 900 columns. The rest 900 columns are mutually independent. The first three columns were chose as the true predictors with the regression coefficients being 1.5, 3.0 and 4.5, respectively. We randomly split each dataset into 50 subsets with \( s = 20 \).


Example II

Table: Results of SaM for Example II: \(^a\) average number of predictors selected in stage I; \(^b\) average number of predictors that are correlated with the true predictors and selected in stage I; \(^c\) average number of predictors selected in stage II; \(^d\) average number of true predictors selected in stage II. The values reported in parentheses are standard deviations of the corresponding estimate.

<table>
<thead>
<tr>
<th></th>
<th>Stage I</th>
<th>Stage II</th>
</tr>
</thead>
<tbody>
<tr>
<td>size(^a)</td>
<td>correlated predictors(^b)</td>
<td>size(^c)</td>
</tr>
<tr>
<td>166.6(.87)</td>
<td>98.2(.22)</td>
<td>3.2(.05)</td>
</tr>
</tbody>
</table>
Massive Data Example

We simulated 100 datasets with $\sigma^2 = 0.25$. Each dataset was generated from a multivariate normal distribution with $n = 100$ observations and $P_n = 500,000$ predictors, among which 2,000 predictors are equally correlated with a correlation coefficient of $\rho = 0.25$ and the rest predictors are uncorrelated. Among those 2,000 correlated predictors, 5 of them were chosen as the true predictors with the regression coefficients (0.2, 0.3, 0.4, 0.5, 0.6). Each dataset was randomly split into 1,000 subsets with $s = 500$. 
Massive Data Example

Table: Comparison of SaM with Lasso, SIS and ISIS for the massive data example: \(^a\) average number of predictors selected by different methods; \(^b\) average number of true predictors selected by different methods; \(^c\) mean squared error of \(\hat{\beta}_\xi\), i.e., \(\|\hat{\beta}_\xi - \beta^*\|^2\), produced by different methods; \(^d\) the posterior mean of \(\beta\) by model average is used as \(\hat{\beta}\) for SaM method; The numbers in the parentheses are the corresponding standard deviations.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SaM</th>
<th>Lasso</th>
<th>SIS</th>
<th>ISIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(|\xi|)^d</td>
<td>5.3(0.22)</td>
<td>44.62(0.98)</td>
<td>11.47(0.29)</td>
<td>13.57(0.59)</td>
</tr>
<tr>
<td>(|\xi \cap t|)^b</td>
<td>3.53(0.073)</td>
<td>4.24(0.070)</td>
<td>3.31(0.083)</td>
<td>3.37(0.086)</td>
</tr>
<tr>
<td>fsr(%)</td>
<td>25.5(2.30)</td>
<td>89.7(0.38)</td>
<td>67.7(1.58)</td>
<td>65.8(2.50)</td>
</tr>
<tr>
<td>nsr(%)</td>
<td>29.4(1.46)</td>
<td>15.2(1.40)</td>
<td>33.8(1.65)</td>
<td>32.6(1.72)</td>
</tr>
<tr>
<td>MSE(^c)</td>
<td>0.146(.013)(^d)</td>
<td>0.263(.011)</td>
<td>0.246(.016)</td>
<td>0.342(.022)</td>
</tr>
</tbody>
</table>
CPU Cost

- SaM cost for each dataset about 16 hours in a serial implementation on a single core of Intel® Xeon® CPU E5-2690(2.90Ghz). This converts to about half an hour if 100 such cores are run in parallel, after accounting for the CPU times spent on each subset and the aggregated subset.

- Bayesian SSVS algorithm with a horse shoe prior for the full dataset: It took up to one day to run only 20+ iterations on the same computer.
Lan et al. (2006) conducted an experiment which examines the genetics of two inbred mouse population (C57BL/6J and BTBR). A total of 60 F2 samples, with 31 female and 29 male mice, were used to monitor the expression levels of 22,575 genes.

- SaM: The data was divided into 45 subsets with $s = 502$. In stage I, SaM selected 1113 genes from 22575 genes. In stage II, SaM selected 6 genes under the setting $r_n = P_n^{0.3}$. The first five selected genes also compose the MAP model. The leave-one-out cross validation mean square error of the model of the six genes is 0.084.

- SIS-SCAD selected 17 genes with the leave-one-out mean square error 0.204.

- ISIS-SCAD selected 9 genes with the leave-one-out mean square error 0.112.
PCR Example

Figure: Comparison of SaM with Lasso, SIS-Lasso, and ISIS-Lasso for the PCR data: leave-one-out cross validation median square error of the MAP$_1$ – MAP$_{35}$ models produced by SaM (black dot with dashed line), Lasso (triangle with dashed line), SIS-Lasso ("+" with solid line), and ISIS-Lasso (hollow dot with solid line).
Time Complexity Analysis

Let $K_n$ denote the number of subsets, and let $\bar{s}_n$ denote the upper bound of the size of the models considered for each subset. Then, for each subset, the computational complexity of the first stage screening is bounded by

$$O(nP_n^2/K_n^2) + \{O(\bar{s}_n^3) + O(P_n/K_n)\} T,$$

where the first term is for the overhead computation of the matrix/vector products $X_s^T X_s$, $X_s^T y$ and $y^T y$ for simulations of the posterior; $T$ is the number of iterations; $O(\bar{s}_n^3)$ is for evaluation of the posterior, which involves inversion of the matrix $X_\xi^T X_\xi + V_\xi^{-1}$; and $O(P_n/K_n)$ is for evaluation of the proposal distribution.
Time Complexity Analysis

If we

(i) let the number of iterations increase with the subset size in a low power, say $T = O((P_n/K_n)^\alpha)$ for some $\alpha \geq 1$,

(ii) choose $K_n$ to minimize the time complexity of SaM, i.e., minimizing

$$O((P_n/K_n)^{1+\alpha}) + O((K_n\bar{r})^{1+\alpha}) + O(n(P_n/K_n)^2) + O(n(K_n\bar{r})^2),$$

then the time complexity can be minimized at

$$K_n = O(\sqrt{P_n})$$

for any $\alpha > 0$.

If $K_n = O(\sqrt{P_n})$, then the resulting subset size is $O(\sqrt{P_n})$, and the resulting time complexity of SaM is $O(P_n^{(1+\alpha)/2}) + O(nP_n)$. That is, SaM can work in a linear or lower order polynomial time of $P_n$. 
Time Complexity Analysis

- SaM has a time complexity of $O(P_n^{(1+\alpha)/2}) + O(nP_n)$. In practice, we need to restrict $K_n$ to a finite number according to our computer resources, say, set $K_n = \min(K, \sqrt{P_n})$.

- Lasso has a computational complexity of $O(nP_n^2)$ when $P_n > n$. In a parallel implementation with $\sqrt{P_n}$ processors, its time complexity should be lower bounded by $O(nP_n^{3/2})$.

- The Bayesian SSVS algorithm with a local shrinkage prior has a time complexity of at least $O(P_n^4)$, given that its computational complexity at each iteration is $O(P_n^3)$. Even in a parallel implementation with $\sqrt{P_n}$ processors, its time complexity should be lower bounded by $O(P_n^{7/2})$. 
Time Complexity Analysis

- Even if implementing on a serial computer, SaM can still have a favorable time complexity compared to the direct simulation of the full data posterior (i.e., setting $K_n = 1$).
- The time complexity of SaM in a serial implementation is $\sqrt{P_n} O(P_n^{1+\alpha}/2 + nP_n)$ if $\sqrt{P_n}$ subsets are used, while the time complexity of the direct simulation of the full data posterior is $O(P_n^{1+\alpha}) + O(nP_n^2)$.
- The split-and-merge strategy leads to a CPU time saving of order at least $O(\sqrt{P_n})$ in serial implementations!
Discussion

- Consistency: Theoretical guarantee!
- Embarrassingly parallel structure: Easy implementation!
- Favorable time complexity: A linear or low order polynomial time of \( P_n \).
- Better screening criterion: SaM makes use of the joint information of multiple predictors in predictor screening, while SIS makes use of only the marginal utility of each predictor.
- Bayesian can potentially do better than frequentists for high dimensional data!
Acknowledgments

- NSF grants
- KAUST grant
- Student: Qifan Song (Assistant Professor at Purdue University)