Journal of Data Science **11**(2013), 343-370

modelSampler: An R Tool for Variable Selection and Model Exploration in Linear Regression

Tanujit Dey
College of William & Mary

Abstract:

We have developed a tool for model space exploration and variable selection in linear regression models based on a simple spike and slab model (Dey, 2012). The model chosen is the best model with minimum final prediction error (FPE) values among all other models. This is implemented via the R package **modelSampler**.

However, model selection based on FPE criteria is dubious and questionable as FPE criteria can be sensitive to perturbations in the data. This R package can be used for empirical assessment of the stability of FPE criteria. A stable model selection is accomplished by using a bootstrap wrapper that calls the primary function of the package several times on the bootstrapped data. The heart of the method is the notion of model averaging for stable variable selection and to study the behavior of variables over the entire model space, a concept invaluable in high dimensional situations.

Key words: FPE analysis, model exploration, model uncertainty, rescaled spike and slab model, variable selection.

1. Introduction

Variable selection in linear regression models is an important aspect of many scientific analyses. A classic approach to the problem is to use what is often referred to as final prediction error (FPE) approach (Akaike, 1969). The FPE criteria is one which takes the residual sum of squares and tacks on a penalty related to the number of variables. Classical examples are AIC (Akaike, 1973) and BIC (Schwarz, 1978). The best subset of variables or the "optimal model" is found by searching all models and finding that model with the smallest FPE criteria. Unfortunately all subset searches are not feasible in high dimensions as this severely constrains FPE methods. For example, one of the most popular all subset procedure implemented in R is the **leaps** (Lumley, 2010) package. However, **leaps** restricts the user to approximately 30 predictors. Restricted all subset

searches can be used to get around this problem (for instance, using the `step` function in R). Since models are fit in a forward stagewise fashion, the concern is that models tend to be avid of a few high signal variables entering early on, thus masking the ability to identify other influential predictors. Methods such as boosting (Schapire, 1990) addresses this issue of greed and acquisitiveness, but since their primary goal is to minimize prediction error, interpretability suffers and variable selection is not properly addressed.

In order to address these issues we have developed a Bayesian method based on rescaled spike and slab (RSS) models (Dey, 2012). For more details of spike and slab model, we refer readers to Ishwaran and Rao (2003; 2005a,b). This method is implemented via the R package **modelSampler** (The R package is available at http://tdey.people.wm.edu/modelSampler.html). The function `modelSampler` implements a Gibbs sampling procedure for drawing values from the Bayesian posterior. The Gibbs sampler is highly efficient and is able to effectively search over the relevant model space, enabling a "smart type" of restricted all subsets search. This way restricted AIC and BIC models can be found.

A pertinent question could be why is Bayesian hierarchical model being used to implement FPE method. To implement FPE based methods, we need to build all possible models based on the data set. This is impossible when the data dimension is moderate or large. With the advancement of technology, we are now dealing with more complex data with very high to ultra-high dimensional data sets. So proper implementation of the FPE methods are literally impossible. In this context, the use of RSS model is quite helpful to implement restricted search and is able to find a small subset of the entire model space. Once we have this subset, we can easily implement FPE methods for variable selection.

One can also argue using more advanced and popular methods like lasso (Tibshirani, 1996) and its several modified versions, elastic net (Zou and Hastie, 2005), SCAD (Fan and Li, 2001), adaptive elastic net (Zou and Zhang, 2009), SIS (Fan and Lv, 2008), to name a few. These methods are referred to as penalized likelihood techniques. The AIC and BIC are also based on l_0-penalization. In penalized likelihood based approaches, tuning parameter/s associated with the penalty term/s play an important role. For example, the tuning parameter for AIC is 2, whereas for BIC it is $\log(n)$, where n is the sample size. In case of other penalized procedures, the choice of tuning parameter/s is data driven and most of the time it is selected using methods like AIC, BIC or cross validation type techniques. It is interesting to note that even if we are leaning towards using these newly introduced techniques, FPE based methods are still in use and play a significant role in modern day practices.

One should be cautious of selecting variables solely on the basis of an FPE analysis; as the FPE criteria can be sensitive to perturbations in the data. In

fact, the **modelSampler** package can be used to assess instability of the FPE criteria, as well as to provide a more stable solution. The function `modelSampler` makes use of model averaging for stable variable selection. Selecting a more stable set of variables is accomplished using the function `boot.modelSampler`, which is essentially a bootstrap wrapper that calls the primary function `modelSampler`. The user specifies B, the number of bootstrap draws to use and the wrapper then makes B calls to the primary function. Each call uses a bootstrap draw of the original data.

A hard shrunk posterior mean is computed for each bootstrap draw for each model size visited by the Gibbs sampler. The hard shrunk estimators are then combined over the B draws to form a hard shrunk ensemble for each given model size. Using out-of-bagging we estimate the prediction error for each hard shrunk predictor, and select the predictor with the smallest prediction error. The dimension of the selected predictor is the optimal model size $\hat{k}$. The optimal model is then chosen by selecting the first ordered $\hat{k}$ variables. Ordering is based on an ensemble Bayesian model averaged (BMA) predictor, formed by averaging the BMA estimator over the B bootstrap draws. Ordering is based on the BMA ensemble because of the inherent stability that is gained by model averaging. Note that unlike traditional BMA where the goal is prediction (Hoeting *et al.*, 1999), our ensemble is derived solely for purposes of variable selection. This type of analysis is very different from the linear regression model implementation via `bicreg` function of the R package **BMA** (Rafttery *et al.*, 2010) for Bayesian model averaging.

The **modelSampler** package has four distinct features:

1. The core function `modelSampler` implements a Gibbs sampler to draw posterior values. The Gibbs sampler keeps track of different models as they are being sampled. The unique feature of the bimodal prior in the RSS model (details discussed later) is that it creates a unique mapping between posterior sample and a visited model (for details see the Gibbs sampler in the Appendix). This helps to perform FPE based variable selection. The output from the core function also provides posterior probabilities for each variable in the data set. This can be used to perform Bayesian variable selection, for example highest posterior variable selection, variable selection based on median model. Therefore if one is interested in fast and effective variable selection procedure, the `modelSampler` function is equipped to do so. It is to be noted that variable selection based on `modelSampler` is as competing as the popular penalized likelihood based procedures like lasso, see Dey (2012). An advantage of using this method is that there is no need to rely on proper choice of tuning parameter, as is the case in penalized likelihood based procedures.

2. If one is interested in only FPE based variable selection, there is some uncertainty involved in this selection procedure. `modelSampler` package can also be used to foresee the instability of the FPE based criteria. To overcome the uncertainty around FPE based variable selection procedure, we use a prediction error based variable selection method by using another function `boot.modelSampler`. The salient feature of our method implemented via this function is that it not only perform variable selection, but also the predictive performance is quite competing to the well known predictive methods like Random Forest, Boosting, and BMA.
3. Another important concern related to variable selection procedure is model uncertainty, particularly in higher dimensional setting. We are able to show (empirically) that even if we get an "optimal model" using some method, there are other models that have similar predictive performance (based on prediction error). Hence the question is which model should we choose as the best model! Using `boot.modelSampler`, we are able to show that there are many competing models with very similar prediction error. So we propose using both numeric and graphical outputs to choose a set of competing models and use them solely for prediction purposes.
4. This R package produces high dimensional graphics to visualize several salient features related to variable selection procedure, such as importance of variables with respect to total number of variables in the data set, visualizing the entire model space, the instability of FPE criteria, prediction error plot, etc.

1.1 Organization of the Article

The article is organized as follows. Section 2 presents an overview of the data generation model and formally defines rescaled spike and slab models. Section 3 has mathematical definitions for BMA estimators, ensemble BMA estimators, "hard shrunk" predictors and out-of-bag estimation of prediction error. Section 4 shows data analysis using the R package. Section 5 is the empirical study to compare predictive performance of the stable variable selection technique based on RSS model with other existing popular methods. Section 6 introduces the issue of model uncertainty as relates to "optimal" model selection, with several graphical examples based on **modelSampler** package along with data analysis. Section 7 concludes the article with a discussion.

2. A Bimodal Spike and Slab Model

Our discussion focuses on linear regression models. We assume that the responses $Y_1, \cdots, Y_n$ are independent with corresponding K-dimensional predictors

$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ such that

$$Y_i = \beta_1 x_{i,1} + \cdots + \beta_K x_{i,K} + \varepsilon_i = \boldsymbol{x}_i^t \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \cdots, n. \tag{1}$$

Here $\{\varepsilon_i\}$ are independent variables such that $\mathbb{E}(\varepsilon_i) = 0$ and $\mathbb{E}(\varepsilon_i^2) = \sigma^2 > 0$. It is also assumed that $\boldsymbol{x}_i$'s are standardized, so that $\sum_{i=1} x_{i,k} = 0$ and $\sum_{i=1} x_{i,k}^2 = n$ for each k and $\sum_{i=1}^n y_i = 0$. Notice that the last constraint is satisfied by centering $\{y_i\}_{1 \le i \le n}$ by the mean. Since y is assumed as centered, no intercept term is included in (1). From a model selection point of view, the focus is to identify predictors that are non-zero in (1). RSS models were introduced in Ishwaran and Rao (2003; 2005a, b) as a method for selecting variables from (1). A rescaled spike and slab model refers to a Bayesian model specified by the following prior hierarchy

$$\begin{aligned} (\tilde{\boldsymbol{Y}}_i | \boldsymbol{x}_i, \boldsymbol{\beta}) &\overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{x}_i^t \boldsymbol{\beta}, n), \quad i = 1, \cdots, n, \\ (\boldsymbol{\beta} | \boldsymbol{\gamma}) &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \\ \boldsymbol{\gamma} &\sim \pi(d\boldsymbol{\gamma}), \end{aligned} \tag{2}$$

where $\tilde{\boldsymbol{Y}}_i = \hat{\sigma}^{-1} n^{1/2} Y_i$ and $\hat{\sigma}^2$ is an estimator for σ^2, $\mathbf{0}$ is the K-dimensional zero vector, $\boldsymbol{\Gamma}$ is the $K \times K$ diagonal matrix $\text{diag}(\gamma_1, \cdots, \gamma_K)$ and π is the prior measure for $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_K)^t$.

We consider a simple prior for γ_k, different than that used by Ishwaran and Rao (2003; 2005a, b). We are using a two-component mixture model with one component taking a very small value $v_0 > 0$, while the other component takes a very large value $V > 0$. Such a choice induces a prior for $\boldsymbol{\beta}$, which is a mixture of near-degenerate multi-normal distributions. Each near-degenerate distribution acts like a flat prior over a subspace (which can be thought of as model), and the posterior mean conditioned in this subspace closely approximates the constrained ordinary least squares (OLS) estimator (for details see Dey, 2012). Under this prior, the Gibbs sampler produces different models and each model gives an estimate for the constrained OLS estimator. By doing so we have an effective way to implement a Monte Carlo all subset search which allows us to compute classical FPE estimates for these models. This helps us to study the empirical performance of classical methods, such as AIC and BIC without being severely restricted to the number of predictors that can be entertained.

The prior π for $\boldsymbol{\gamma}$ is specified as

$$\begin{aligned} (\gamma_k | v_0, V, w) &\overset{\text{iid}}{\sim} (1 - w)\, \delta_{v_0}(\cdot) + w\, \delta_V(\cdot), \quad k = 1, \cdots, K, \\ w &\sim \nu(dw). \end{aligned} \tag{3}$$

Here ν is a prior measure for $w \in [0, 1]$. One can think of w as a complexity parameter that controls the overall size of the model.

The rescaling of the response variable induces a non-vanishing penalization effect, and ensures a selective shrinkage property in orthogonal models when used in tandem with a continuous bimodal prior (Ishwaran and Rao, 2005a; 2011). This property allows the posterior mean for the coefficients to shrink towards zero for truly zero coefficients, while for non-zero coefficients the posterior estimates are similar to the OLS estimator.

Remark 1. One way to estimate $\hat{\sigma}^2$ is by using the unbiased estimator of σ^2 from the least squares technique. Let $\hat{y}_{\text{OLS}}$ be the predictor for y based on OLS estimator. Then $\hat{\sigma}^2 = ||y - \hat{y}_{\text{OLS}}||^2/(n-p)$. If OLS does not exist, a more general approach could be used to get $\hat{\sigma}^2$ estimator by using both parametric or nonparametric methods. We recommend using the method Random Forest to get the estimator from test-set mean squared error. For details see Ishwaran and Rao (2011). Both the proposed estimation procedures are implemented in the R package.

3. Variable Selection Based on modelSampler

Here we discuss the stable variable selection technique based on the bootstrapped wrapper `boot.modelSampler`. In `boot.modelSampler`, we call `modelSampler` B times on bootstrapped data. For each bootstrap iteration we calculate the BMA estimators. Based on B iterations we calculate ensemble BMA estimators. The ensemble BMA could be used to generate a stable ordered list of variables such that for any given k, we can then determine the best model of size $\hat{k}$. As model selection ultimately forces us to choose a single model from our class of models, it generally rules out the BMA estimators that do not correspond to any one specific model. Therefore, it does not help us to find our optimal $\hat{k}$ because due to averaging the actual size $\hat{k}$ gets ruled out.

Here we formally define BMA estimator under RSS model (2). The BMA estimator for $\boldsymbol{\beta}$ is $\widehat{\boldsymbol{\beta}} = \mathbb{E}(\boldsymbol{\beta}|\tilde{\boldsymbol{Y}})$:

$$\begin{aligned}
\widehat{\boldsymbol{\beta}} &= \sum_{\alpha} \mathbb{E}(\boldsymbol{\beta}|\alpha, \tilde{\boldsymbol{Y}})\pi(\alpha|\tilde{\boldsymbol{Y}}) \\
&= \sum_{k=1}^{K} \sum_{\alpha \in \Delta_k} \mathbb{E}(\boldsymbol{\beta}|\alpha, \tilde{\boldsymbol{Y}})\pi(\alpha|\tilde{\boldsymbol{Y}}), \quad \Delta_k = \{\alpha : K_\alpha = k\} \\
&= \sum_{k=1}^{K} \mathbb{E}(\boldsymbol{\beta}|\alpha \in \Delta_k, \tilde{\boldsymbol{Y}})\pi(\Delta_k|\tilde{\boldsymbol{Y}}).
\end{aligned}$$

The last line of the above expression follows from

$$\begin{aligned}\mathbb{E}(\boldsymbol{\beta}|\alpha \in \Delta_k, \tilde{\boldsymbol{Y}}) &= \int \boldsymbol{\beta} \frac{f(\boldsymbol{\beta}, \Delta_k|\tilde{\boldsymbol{Y}})}{f(\Delta_k|\tilde{\boldsymbol{Y}})} d\boldsymbol{\beta} \\ &= \frac{1}{\pi(\Delta_k|\tilde{\boldsymbol{Y}})} \sum_{\alpha \in \Delta_k} \int \boldsymbol{\beta} f(\boldsymbol{\beta}, \alpha|\tilde{\boldsymbol{Y}}) d\boldsymbol{\beta} \\ &= \frac{1}{\pi(\Delta_k|\tilde{\boldsymbol{Y}})} \sum_{\alpha \in \Delta_k} \left[\int \boldsymbol{\beta} f(\boldsymbol{\beta}, \alpha, \tilde{\boldsymbol{Y}}) d\boldsymbol{\beta} \right] \pi(\alpha|\tilde{\boldsymbol{Y}}).\end{aligned}$$

Consequently we have

$$\widehat{\boldsymbol{\beta}} = \sum_{k=1}^{K} \widehat{\boldsymbol{\beta}}_k \pi(\Delta_k|\tilde{\boldsymbol{Y}}),$$

where $\widehat{\boldsymbol{\beta}}_k = \mathbb{E}(\boldsymbol{\beta}|\alpha \in \Delta_k, \tilde{\boldsymbol{Y}})$ is the "conditional posterior mean".

For each bootstrap b, where $b = (1, \cdots, B)$, there is a BMA estimator which is denoted as $\hat{\beta}_b^*$. The ensemble BMA estimates for these bootstrapped estimator is: $\widehat{\boldsymbol{\beta}}^e = B^{-1} \sum_{b=1}^{B} \widehat{\boldsymbol{\beta}}_b^*$. In the case of the ensemble BMA estimator $\widehat{\boldsymbol{\beta}}^e = (\widehat{\beta}_1^e, \cdots, \widehat{\beta}_K^e)^t$, we rank variables by their absolute coefficients estimates. So if

$$|\widehat{\beta}_{j_1}^e| \geq |\widehat{\beta}_{j_2}^e| \geq \cdots \geq |\widehat{\beta}_{j_k}^e|, \tag{4}$$

then variable j_1 is the top variable, j_2 is second best variable, and so on. In particular, the best model of size k is $\alpha_k = \{j_1, j_2, \cdots, j_k\}$. We don't know the actual k because we are using an ensemble technique and model size specific information is lost due to averaging.

As the above procedure is not useful, we use the hard shrinkage method to find optimal $\hat{k}$. We find $\hat{k}$ by using a class of "hard shrunk predictors". From each bootstrap iteration, we estimate "hard shrunk estimators". After B iterations we calculate ensemble "hard shrunk predictors". Then we use out-of-bagging to estimate the prediction error for each hard shrunk ensemble. The dimension of the predictor is $\hat{k}$ (the optimal model size) that has the smallest prediction error.

3.1 Optimal Model Size Determination via Hard Shrinkage and Model Averaging

To determine $\hat{k}$ we are not going to use a traditional Bayesian approach, we prefer to take on a frequentist approach. Because of averaging across all the models, BMA estimators lose model size specific information. It can be shown that the posterior mean approximates constrained OLS estimates which is why we use the conditional posterior mean of (4), see Dey (2012) for details. We indirectly use $\widehat{\boldsymbol{\beta}}_k$ by defining a hard shrinkage estimator:

$$\widehat{\boldsymbol{\beta}}_k^H = (\boldsymbol{X}_{\alpha_k}^t \boldsymbol{X}_{\alpha_k} + n\boldsymbol{I}_{\alpha_k})^{-1} \boldsymbol{X}_{\alpha_k}^t \boldsymbol{Y},$$

where $\boldsymbol{I}_{\alpha_k}$ is an identity matrix of size α_k and $\alpha_k = \{j_1(k), j_2(k), \cdots, j_k(k)\}$ and $j_l(k)$, for $l = 1, \cdots, K$, is determined from the ranking of the coefficients of $\widehat{\boldsymbol{\beta}}_k = (\hat{\beta}_{1,k}, \cdots, \hat{\beta}_{K,k})^t$:

$$|\hat{\beta}_{j_1(k),k}| \geq |\hat{\beta}_{j_2(k),k}| \geq \cdots \geq |\hat{\beta}_{j_k(k),k}|. \tag{5}$$

Based on this estimator, the hard shrunk predictor is

$$\widehat{\boldsymbol{\mu}}_k^H(\boldsymbol{x}) = \boldsymbol{X}^t \widehat{\boldsymbol{\beta}}_k^H.$$

Considering the linear regression model (1). Let $(Y_{1,b}^*, \boldsymbol{x}_{1,b}^*), \cdots, (Y_{n,b}^*, \boldsymbol{x}_{n,b}^*)$ be the b-th bootstrap sample of size n, and $(Y_{1,b}^{**}, \boldsymbol{x}_{1,b}^{**}), \cdots, (Y_{n_b,b}^{**}, \boldsymbol{x}_{n_b,b}^{**})$ be the corresponding out-of-bag (OOB) sample of size n_b. While bootstrapping the hard shrunk estimator is estimated for each bootstrap data as

$$\widehat{\boldsymbol{\beta}}_{k,b}^{H*} = \left(\boldsymbol{X}_{\alpha_k,b}^{*t} \boldsymbol{X}_{\alpha_k,b}^* + n\boldsymbol{I}_{\alpha_k}\right)^{-1} \boldsymbol{X}_{\alpha_k,b}^{*t} \boldsymbol{Y}^*.$$

Based on this estimator we define hard shrunk ensemble predictor

$$\widehat{\boldsymbol{\mu}}_k^{H,(i)}(\boldsymbol{x}) = \left(\sum_{b=1}^{B} \mathbb{I}_{i,b} \hat{\boldsymbol{\mu}}_{k,b}^H(\boldsymbol{x}^*)\right) \Big/ n_{i,b},$$

where

$$\mathbb{I}_{i,b} = \begin{cases} 1, & \text{if } n_{i,b} = 0 \text{ and } k\text{-th model is visited,} \\ 0, & \text{if } n_{i,b} \geq 1 \text{ and } k\text{-th model is not visited,} \end{cases}$$

$n_{i,b} = \sum_{b=1}^{B} \mathbb{I}_{i,b}$ and $\hat{\boldsymbol{\mu}}_{k,b}^H(\boldsymbol{x}^*)$ bootstrapped hard shrunk predictor. Note that $\widehat{\boldsymbol{\mu}}_k^{H,(i)}(\boldsymbol{x})$ is the ensemble hard shrunk predictor with "i" removed. This is called out-of-bag (OOB) ensemble predictor.

Once we get this class of hard shrunk predictors we are able to estimate the prediction error (PE). Formally we define this as

$$\hat{\mathscr{P}}_k{}^H = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{\boldsymbol{\mu}}_k^{H,(i)}(\boldsymbol{x}_i)\right)^2.$$

In fact $\hat{\mathscr{P}}_k{}^H$ is nothing more than the leave-one-out bootstrap estimate of prediction error for $\widehat{\boldsymbol{\beta}}_k^H$. Based on this estimated PE we find the optimal model of size $\hat{k}$, where $\hat{k}$ is determined as

$$\hat{k} = \arg\min_k \{\hat{\mathscr{P}}_k{}^H\}. \tag{6}$$

Finally the "best" subset of variables are $\alpha_{\hat{k}} = \{j_1, j_2, \cdots, j_{\hat{k}}\}$ using the selection procedure as described in (4).

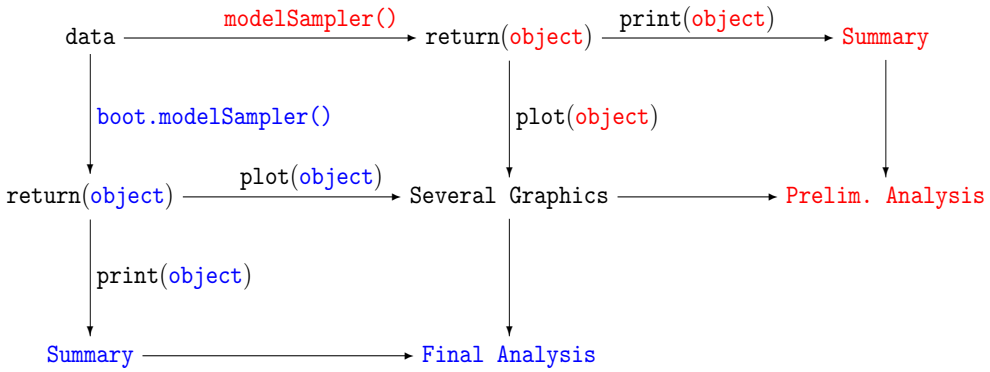
Remark 2. The same OOB technique is applied to get an OOB estimate of PE for the ensemble BMA estimator (even though we don't use it). When we bootstrap the data, for each iteration we calculate BMA estimator $\hat{\boldsymbol{\beta}}_b^* = (\hat{\boldsymbol{\beta}}_{b,1}^*, \cdots, \hat{\boldsymbol{\beta}}_{b,K}^*)^t$. Based on this estimator we define an OOB predictor $\hat{\boldsymbol{\mu}}_b(\boldsymbol{x}_b^{**}) = \boldsymbol{x}_b^{**t}\hat{\boldsymbol{\beta}}_b^*$. The corresponding estimate for PE of OOB predictor is $\hat{\mathscr{P}}_b^{**} = n^{-1}\sum_{i=1}^n (Y_i - \hat{\boldsymbol{\mu}}_{i,b}(\boldsymbol{x}_{i,b}^{**}))^2$.

As bootstrap iteration continues, our BMA estimators approach to ensemble BMA estimators. We follow similar step to define predictors, updating $\hat{\boldsymbol{\beta}}_b^*$ and to subsequently estimate PE's. At the last stage of iterations, our estimated PE for B-th iteration is $\hat{\mathscr{P}}_B^{**} = n^{-1}\sum_{i=1}^n (Y_i - \hat{\boldsymbol{\mu}}_{i,B}(\mathbf{x}_{i,B}^{**}))^2$, where $\hat{\boldsymbol{\mu}}_{i,B}(\mathbf{x}_{i,B}^{**}) = \mathbf{x}_B^{**t}\hat{\boldsymbol{\beta}}^e$. Note that $\hat{\boldsymbol{\beta}}^e$ is the ensemble BMA estimator. The final estimated PE for ensemble BMA estimator is $\hat{\mathscr{P}}^e = B^{-1}\sum_{b=1}^B \hat{\mathscr{P}}_b^{**}$.

4. Description of modelSampler and Usage in R

4.1 Roadmap of modelSampler

The work flow of the package can be visualized in two distinct phases. In the initial phase, the user calls `modelSampler` for a restricted FPE search. In phase two, `boot.modelSampler` is used to generate a final stable set of variables.



In R package **modelSampler**, the core function `modelSampler` implements a Gibbs sampling procedure as described in Section 3. This function outputs an object that comprises of several results, namely models that are selected based on the FPE selection criteria, frequencies of each model visited by Gibbs sampler, the "best" subset of variables which corresponds to a specific model size and the posterior inclusion probability of each variable in the data set. This object is visualized via several interesting graphics to support the analysis. For further analysis `boot.modelSampler` function is implemented in the package. It calls the

core function B times, and finally outputs the "best" subset of variables. This output is then used via graphics to study the model space of the data, instability of the FPE selection criteria, optimal size of the model and model uncertainty.

4.2 Example: Diabetes Data

The example uses the Diabetes data (Efron *et al.*, 2004). The data set comprises of 442 diabetes patients with 10 baseline variables. `Y` is the outcome variable. For details of the data:

```
R> install.packages("modelSampler")
R> library("modelSampler")
R> data("Diabetes")
R> ?Diabetes
```

We first use the core function modelSampler for preliminary analysis of the data which gives the following output:

```
R> library("modelSampler")
R> data("Diabetes")
R> ms.out <- modelSampler(Y~., Diabetes, n.iter1=2500,
+ n.iter2=10000,verbose=FALSE)
R> print(ms.out)
-----------------------------------------------------------------------
No. predictors               : 10
No. sampled values           : 10000
Estimated complexity         : 0.6 +/- 0.168
Prob. visiting new model     : 0.016

Model selection results:
         hpm     aic     bic
s5    0.9701    TRUE    TRUE
bmi   0.9697    TRUE    TRUE
bp    0.9991    TRUE    TRUE
s1      0.66    TRUE   FALSE
sex   0.9323    TRUE    TRUE
s2    0.4999    TRUE   FALSE
s3    0.5491   FALSE    TRUE
s4    0.3534   FALSE   FALSE
s6    0.1732   FALSE   FALSE
age   0.1017   FALSE   FALSE

Top models stratified by size:
          1       2       3       4       5       6       7       8       9    10
s5    FALSE   FALSE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE  TRUE
bmi   FALSE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE  TRUE
bp     TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE  TRUE
s1    FALSE   FALSE   FALSE    TRUE   FALSE    TRUE    TRUE    TRUE    TRUE  TRUE
```

```
sex   FALSE   FALSE   FALSE   FALSE    TRUE    TRUE    TRUE    TRUE    TRUE  TRUE
s2    FALSE   FALSE   FALSE   FALSE   FALSE    TRUE    TRUE    TRUE    TRUE  TRUE
s3    FALSE   FALSE   FALSE   FALSE    TRUE   FALSE   FALSE   FALSE    TRUE  TRUE
s4    FALSE   FALSE   FALSE   FALSE   FALSE   FALSE    TRUE    TRUE    TRUE  TRUE
s6    FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE    TRUE    TRUE  TRUE
age   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE  TRUE
             1         2         3         4         5         6         7
freq     4.000    22.000   177.000   357.000  1933.000  3764.000  2484.000
mss   4774.120  3581.692  3083.057  3012.299  2913.764  2876.889  2868.567
aic   4788.958  3611.369  3127.572  3071.653  2987.956  2965.920  2972.437
bic   4819.313  3672.078  3218.635  3193.071  3139.728  3148.047  3184.918
             8         9        10
freq   943.000   263.000    53.000
mss   2861.571  2860.862  2860.682
aic   2980.279  2994.408  3009.067
bic   3223.114  3267.598  3312.611
-----------------------------------------------------------------
```

After completion of 10,000 sampling, the probability of visiting a new model by Gibbs sampler is 0.016, which supports the convergence of the Gibbs sampler. Even though it suffices the convergence of the Gibbs sampler, an in-build function is available in the package to graphically diagnose the issue of convergence (we will discuss this later). Six variables are selected by AIC, whereas only five variables are selected by BIC. We can also measure the posterior inclusion probabilities of each variable. By observing the subset of variables corresponding to each model, one can estimate the significance of variables in the data. For instance, the variable `bp` is the most significant variable as it appears in the model of size one and is present till the model acquires a full size. Also its posterior inclusion probability is 0.999, which shows that it is the most significant variable among all variables in the data set. The next significant variable is `s5` and so on. Looking at the total frequencies of each model, we impart that model size 5 to 7 is probably the right dimension of the data.

```
R> plot.modelSampler(ms.out)
```

Based on the output generated by `modelSampler` we get Figure 1, generated from the above command. Five plots are produced going from top to bottom and left to right:

1. The function `modelSampler` estimates a complexity parameter. A complexity plot is provided to view the range of estimated complexity parameter that allows the user to interpret the dimensionality of model space. A high value is characteristic of a larger model. Estimated complexity for Diabetes data is 0.6 ± 0.16.

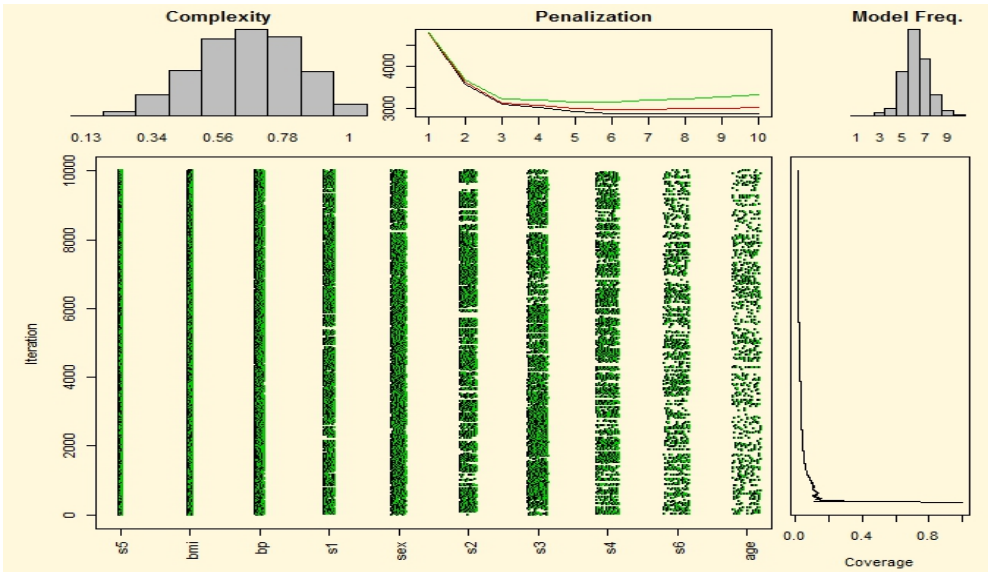


Figure 1: Ensemble graphical analysis of Diabetes data

2. A penalization plot represents a restricted dimension-specific FPE values for instance, minimum residual sum of squares, AIC and BIC. The black line corresponds to minimum residual sum of squares, the red line is for AIC values and the green line is for BIC values.

3. A dimensionality plot is used to depict the size of models visited by `modelSampler`. From the plot it seems that for the Ozone data, the actual size of the model will be approximately between 5 to 7.

4. An image plot is provided to visualize the different models called on, as a function of the number of Monte Carlo iterations. During sampling `bp` was present almost all the time, whereas `age` hardly appeared. Based on this plot we infer that `bp` is the most significant variable while `age` is the least significant variable.

5. A coverage plot is provided for the user to assess convergence of Gibbs sampler. This plot portrays the probability of visiting a new model. As iteration continues the line levels to almost a horizontal line. This shows that `modelSampler` converges very fast.

4.3 Convergence of modelSampler

In the earlier section we were tracking the convergence of the Gibbs sampler via coverage plot. Now we are going to use Gelman-Rubin multiple sequence

diagnostics for our Gibbs sampler. The `plot.gelman` function of the R package is used to implement graphical diagnostics for the convergence of the Gibbs Sampler. For this particular function the package depends on the R package **coda**. In general, the Gibbs sampler of the **modelSampler** reports final samples after some burn-in samples. To use Gelman-Rubin diagnostics, we run Gibbs sampler for three different chains with three different starting values, increasing the sample size from 200 to 1500. For each sample size corresponding to the three chains, we get p (total number of variables in the data set) potential scale reduction factor (psrf) values. If psrf values are close to 1, we believe that samples converge to the stationary distribution. The following command generates the Gelman-Rubin diagnostics plot of the convergence of the Gibbs sampler:

```
R> library("modelSampler")
R> data(Diabetes)
R> plot.gelman(Y~., data=Diabetes)
```

Figure 2 is the graphical representation of the convergence of the Gibbs sampler. In the figure corresponding to each sample size, we draw a circle with radius one, and psrf values are plotted from the center of the circle. As the sample size increases, the psrf values get close to one and they rest on the circumference of the circle. It is quite distinct that when sample size equals 1500 all psrf values are close to one; which suffices that Gibbs sampler converges really fast for moderate number of sampled values from the Gibbs sampler. The `Diabetes` data has moderate number of variables. For higher dimensional data, we notice that sample size of 7500 is reasonable for the convergence of the Gibbs sampler. In fact this option is also available in the `plot.gelman` function; we refer readers to use this function for the Ozone interaction data (`OzoneI`) available in the package.

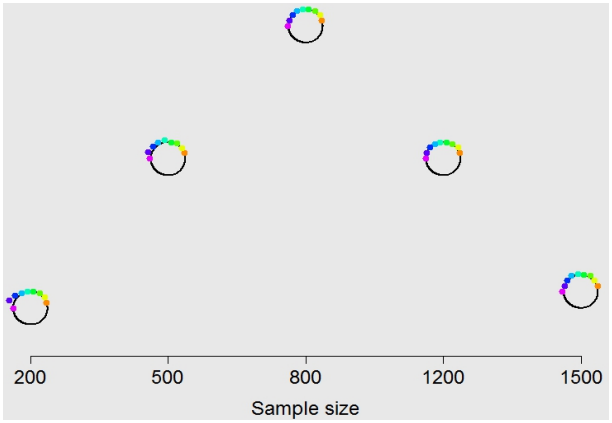


Figure 2: Gelman-Rubin diagnostics plot for `modelSampler` based on Diabetes data

4.4 Diabetes Data: Final Phase of Analysis

Here we are going to use the `boot.modelSampler` for stable variable selection technique based on `modelSampler`.

```
R> library("modelSampler")
R> data("Diabetes")
R> ms.boot <- boot.modelSampler(Y~., Diabetes, n.iter1=2500,
+ n.iter2=2500, B=100, verbose=FALSE)
```

This returns an `object` to do further graphical analysis, and at the same time returns the "best" subset of variables based on our ensemble selection technique. We use this `object` to study the stability of the FPE model selection criteria. FPE models are highly sensitive to the perturbation of data as demonstrated in Figure 3. The R-program of **modelSampler** package keeps track of each FPE model visited for each bootstrap draw. The goal is to measure the stability of the FPE criteria for selecting models when data are being perturbed through bootstrapping. We study this using an AIC-BIC instability plot. The following command generates such a plot indicating significant instability:

```
R> plot.FPE(ms.boot)
```

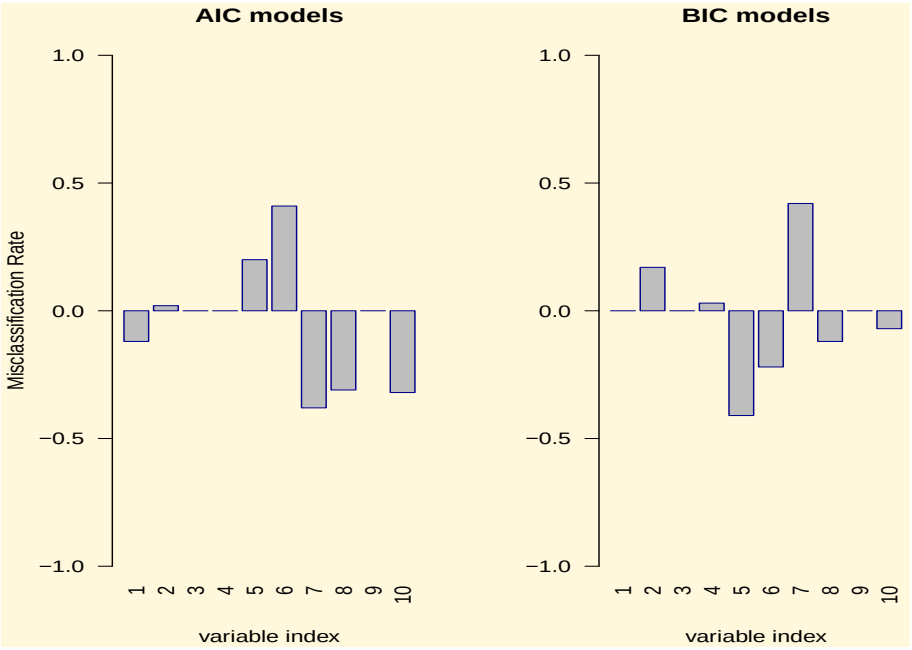


Figure 3: The instability feature of the FPE model selection criteria for Diabetes data. Results are based on 100 bootstrap iterations

In Figure 3, positive values less than 1 indicate that a given variable was selected by the FPE criteria in both the full data set and the bootstrapped data set, but was not selected at all times during bootstrap iterations. Negative values greater than -1 imply that a particular variable is selected several times by the FPE criteria for bootstrapped data but was not selected in the original data set.

4.5 Icicle Plot

Figure 4 is a graphical depiction of the Diabetes data model space called an icicle plot. The following command from the **modelSampler** package generates Figure 4.

```
R> plot.icicle(ms.boot,"Diabetes data")
```

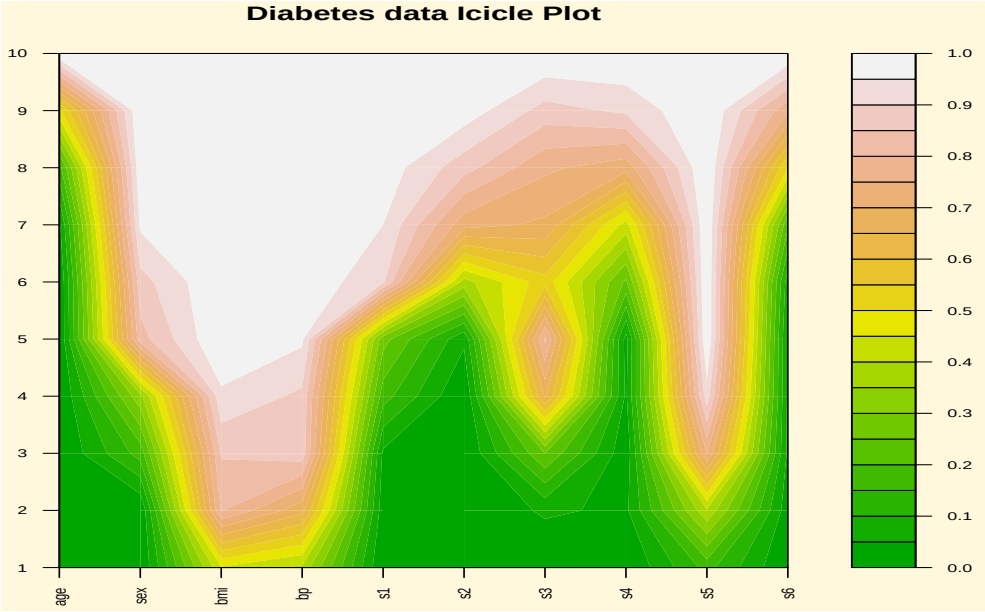


Figure 4: Graphical depiction of the Diabetes data model space. Results are based on 100 bootstrap iterations

This plot is a three dimensional (x, y and z) depiction of the Diabetes data model space. In the left frame of the plot, x displays all variables of the data set, y displays different dimensions of the model space and z (the color) depicts the relative frequency of a selected variable in a given hard shrinkage model. The right frame of the plot represents a color palette of the relative frequencies of the z values. For example, the variables `bp`, `s5` and `bmi` appear in all hard shrinkage models of size 4 or higher in all 100 bootstrap iterations, thereby indicating that they are highly stable and informative.

The mathematical explanation of the plot is as follows: Let $\mathbb{I}_{k,b} = 1$ if the Gibbs sampler visits a model of size k for bootstrap draw b, otherwise $\mathbb{I}_{k,b} = 0$. Not all model sizes are visited, so $\mathbb{I}_{k,b}$ can often be zero. Now we define a binary variable for each variable j of the model size k, to indicate whether that variable was in the conditional posterior mean $\hat{\beta}_k$ in the model α_k. In particular, let

$$a_{k,j,b} = \begin{cases} 1, & \text{if } \mathbb{I}_{k,b} = 1 \text{ and } j \in \alpha_k, \\ 0, & \text{otherwise.} \end{cases}$$

Define $A = (a_{k,j})$ be a $K \times K$ matrix where

$$a_{k,j} = k \sum_{b=1}^{B} a_{k,j,b} \Big/ \sum_{j=1}^{K} \sum_{b=1}^{B} a_{k,j,b}.$$

The icicle plot is a plot of A^c.

4.6 Out-of-bagging and the Best Subset of Variables

An out-of-bag (OOB) technique is used to calculate prediction error (PE). The following PE plot helps to visualize the optimal model size based on the smallest PE. The following command generates the OOB PE plot as shown in Figure 5 (the plot shows the optimal size of the model being 7):

```
R> plot.ooberror(ms.boot,"Diabetes data")
```

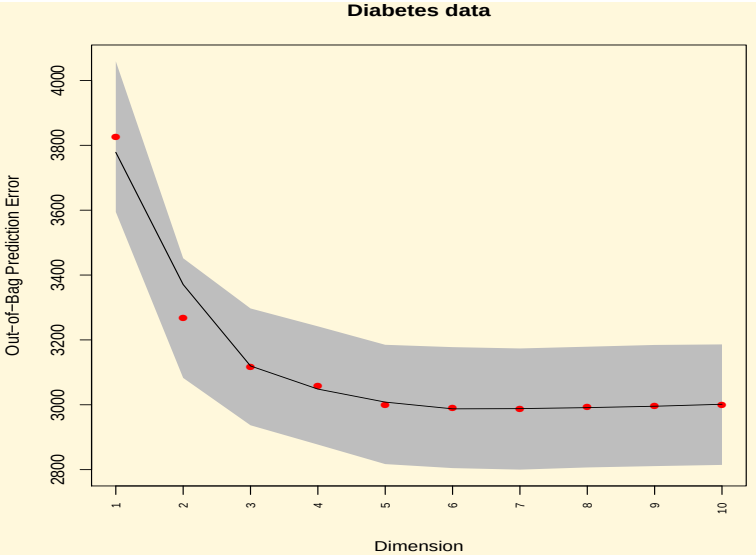


Figure 5: Out-of-bag prediction error plot for Diabetes data. Results are based on 100 bootstrap iterations. Red points are model size specific estimated PE. Gray band surrounding red points are PE $\pm$ standarderror (PE). Smooth black line represents loess estimated line of PE

Once the optimal model size is determined via (6), we can determine the "best" subset of variables $\alpha_{\hat{k}}$ using (5) as $\alpha_{\hat{k}} = \{j_1, j_2, \cdots, j_{\hat{k}}\}$. Using our R package the "best" subset of variables for Diabetes data are:

```
R> print(ms.boot)
-----------------------------------------------------------
Optimal model obtained via ensemble out-of-bagging:
[1] "s5"  "bmi"  "bp"  "s1"  "sex" "s3"  "s2"
-----------------------------------------------------------
```

The seven above selected variables are also ordered based on their importance in the selection. For example, `s5` is the most important variable whereas `s2` is the least significant variable in the selected subset of variables in the `Diabetes` data set.

5. Empirical Study

This section is devoted towards numerical experiments to evaluate our proposed method in different circumstances. The experimentations involve a simulation study and a real data analysis. The simulation study evaluates the variable selection performance of our method compared to other methods. The real data analysis is implemented to illustrate the predictive performance of our method. For comparison purposes three different methods, which are well known for their predictive performances have been incorporated.

5.1 Simulation Study

A simulation study is conducted to illustrate the performance of our proposed method in correlated situation. For comparison, we included the following methods in the simulation study: lasso and elastic net (enet). Lasso solutions were calculated using the LARS algorithm (Efron *et al.*, 2004) by invoking the lasso implementation with a 10-fold validation to estimate the shrinkage parameter. Computations are implemented using the `lars` R package. The elastic net method is implemented using `enet` R package. For enet tuning parameters are selected using BIC stopping rule.

We follow the simulation design as mentioned in Example 1 of Zou and Zhang (2009). The data are generated from the following linear model: $Y = X^T\beta + \nu$, where β is a p-dimensional vector and $\nu \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 6$ and X generated from a p-dimensional multivariate normal distribution with mean zero and covariance matrix Σ; (j,k)-th entry of Σ is $\rho^{|j-k|}$, $\quad 1 \leq j, k \leq p$. We are considering two values of $\rho = 0.5, 0.75$. The dimension of models are taken as $p = [4n^{1/2}] - 5$ for $n = 100, 200$. If $\mathbf{1}_m/\mathbf{0}_m$ denote a m-vector of 1's/0's, the true coefficients are $\beta = (3.\mathbf{1}_q, 3.\mathbf{1}_q, 3.\mathbf{1}_q, \mathbf{0}_{p-3q})^T$, $q = [p/9]$, and d is the numbers of true non-zero

coefficients of the models. We generate 100 data sets based on this simulation design.

The simulation results are promising. Table 1 summarizes the following findings: in moderately correlated settings, our method selects a slightly larger model than the true model; whereas in highly correlated situations, our method opts for smaller models than the true model. Lasso selects bigger model than all other methods; enet performs better than lasso. From a variable selection perspective, the metric ZAZ is an important criterion. In the model the lower value of ZAZ, as compared to the total number of true zero variables initiates larger false positive rate. Our method consistently produces ZAZ values closer to the true value than the other methods.

Table 1: Simulation study results. All results are reported based on averaging over 100 data sets. (a) For $n = 100$, $p = 35$, $d = 9$, true $ZAZ = 26$. (b) For $n = 200$, $p = 51$, $d = 15$, true $ZAZ = 36$

(a)

Model	ρ	$\hat{p}$	ZAZ
lasso	0.50	14.86	20.14
	0.75	13.11	21.72
enet	0.50	10.34	24.66
	0.75	9.61	25.19
modelSampler	0.50	9.34	25.13
	0.75	8.44	25.78

(b)

Model	ρ	$\hat{p}$	ZAZ
lasso	0.50	23.16	27.84
	0.75	21.08	29.91
enet	0.50	16.64	34.36
	0.75	16.03	34.95
modelSampler	0.50	16.06	34.91
	0.75	15.65	35.02

5.2 Real Data Application

Here we demonstrate the predictive performance of the stable variable selection technique based on the RSS model. For comparison purposes we have considered three different methods: Random Forest, Boosting and Bayesian Model Averaging (BMA) methods; the first two methods are frequentist methods while BMA is based on Bayesian methodology. The prediction error is calculated using OOB technique. Random Forest (RF) computations are implemented using the R package **randomForest** (Liaw and Wiener, 2002). In all cases, 1000 trees were grown under default settings. For Boosting we have used R package **gbm** (Ridgeway, 1999). We use a shrinkage (learning) parameter of 0.01, a tree depth of five (the base learner), and 10-fold validation to determine the optimal number of boosting iterations. For BMA we have used the `bicreg` function of the R package **BMA** (Rafttery *et al.*, 2010). We have used 1500 iterations to compute OOB prediction error.

Figure 6 is the beanplot representation of the four different methods considered in this empirical study. The green horizontal lines corresponding to each bean represents 1500 data points of OOB PEs. Our method outperforms all the

three methods considered with respect to the prediction error. We know that RF and Boosting are popular for their predictive performance; so is the BMA from a Bayesian perspective. Note that out of these four methods only RSS and BMA does variable selection, so OOB PE computations are always based on a subset of variables, whereas the RF and Boosting methods use all variables for PE computation. Figure 6 shows that the RSS method exhibits stable performance with respect to other methods. In all three methods the variation of the OOB PEs are very high, whereas for RSS model the data points are exceptionally close to each other.od. The green horizontal lines corresponding to each bean represents 1500 data points of OOB PEs. As we know that RF is popular for its predictive performance, here too for the `Diabetes` data set, it has the lowest PE. RSS model outperforms the Boosting and This empirically suffices that the variable selection method based on RSS model is a consistent and coherent meth the BMA methods with respect to predictive performances. Note that out of these four methods only RSS and BMA does variable selection, so OOB PE computations are always based on a subset of variables whereas the RF and Boosting methods use all variables for PE computation, in spite of this RSS PE performance is quite competitive to the RF method. Figure 6 shows that RSS method exhibits stable performance with respect to other methods. In all three methods the variation of the OOB PEs are very high, whereas for RSS model the data points are very close to each other. This empirically suffices that variable selection method based on RSS model is a consistent method.

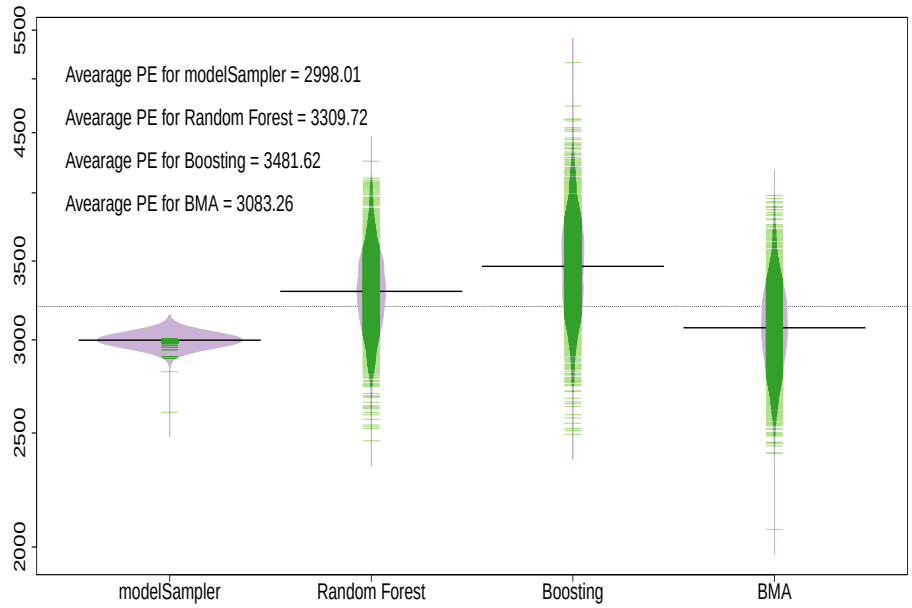


Figure 6: Beanplot for PE performances of four different methods using Diabetes data

6. Variable Stability and Model Space Revisited

Our method offers an optimal variable selection technique, but sometimes the "optimal solution" is not the best solution. The traditional approach is to wring every bit of information from the data, but an attempt to store and evaluate enormous sets of data into a computer database bogs resources and at this point the "optimal" model seems meaningless. Thus by not choosing a single "best" model, we can also study the best set of competing models. A subset of models are selected by the model that has the best predictive performance and where variables begin to stabilize across the entire model space. We illustrate our concept by using two data sets: one with moderate dimension and the other with higher dimension.

Figure 5 shows that the prediction error line for the `Diabetes` data is quite flat and the standard error is large for models of size 5 and onwards. Therefore there is not much benefit in choosing a model of size 5 over a model of size 7 or 8 from a predictive point of view. In a given situation any model in that sub-model space can be chosen as the "best" model.

This motivates us to examine the behavior of estimators across the model space and to outline a pattern of variable stabilization across the model space. The following command from the **modelSampler** package generates such a plot called the variable stability plot.

```
R> plot.var.stability(ms.boot)
```

Figure 7 is a variable stability plot of the Diabetes data set. In the left frame of the plot, the left horizontal axis represents the dimension of the model space, the left vertical axis represents the model size specific ensemble hard shrunk estimator, the right vertical axis displays the indices of variables which are selected in the "best" model from predictive point of view. The right frame of the plot represents RGB-color palette for model size specific prediction error. Mathematical explanation of model size specific ensemble hard shrunk estimators, as plotted in Figure 7 is as follows: For each bootstrap draw b, let $(\boldsymbol{X}^*_{\alpha_k,b}, \boldsymbol{Y}^*_b)$ be the pair of bootstrapped samples drawn from the original sample. Here $\boldsymbol{X}^*_{\alpha_k,b}$ is the $n \times \alpha_k$ design matrix formed from the first k columns of the re-ordered $\boldsymbol{X}$ where ordering is based on (4). The corresponding bootstrapped response from the original $\boldsymbol{Y}$ is $\boldsymbol{Y}^*_b$ and $\boldsymbol{I}_{\alpha_k}$ is a $k \times k$ identity matrix. The hard shrunk estimator of model size k for the b-th bootstrap is

$$\widehat{\boldsymbol{\beta}}^*_{k,b} = (\boldsymbol{X}^{*t}_{\alpha_k,b}\boldsymbol{X}^*_{\alpha_k,b} + \boldsymbol{I}_{\alpha_k})^{-1}\boldsymbol{X}^{*t}_{\alpha_k,b}\boldsymbol{Y}^*_b. \tag{7}$$

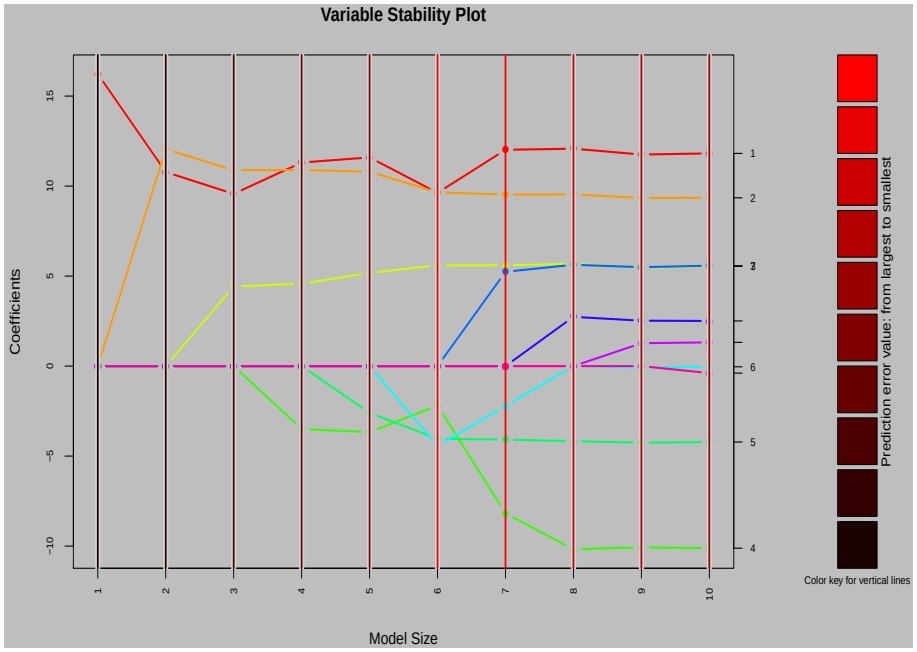


Figure 7: Variable stability plot for the Diabetes data. Results are based on 100 bootstrap iterations. Horizontal lines with points are model size specific ensemble hard shrunk estimators. Shaded vertical lines are model size specific PE. The vertical line with no shade represents the model with smallest PE

The ensemble hard shrunk estimator of model size k is

$$\widehat{\boldsymbol{\beta}}_k^{\text{stable}} = \frac{1}{B}\sum_{b=1}^{B}\widehat{\boldsymbol{\beta}}_{k,b}^{*}. \tag{8}$$

The estimators of (8) are plotted in Figure 7.

The estimators in the plot are standardized so they are fairly comparable to each other. Variables actually begin to stabilize from models of size eight. Other variables are showing stability beyond models of size eight but these estimators are very close to zero. Based on Figures 5 and 7, models of size five to eight can be elected as the "best" set of competing models.

Figure 7 also brings forth another important issue related to the model selection problem. The entire model space for a given data set can be represented in two categorizes (Shao, 1993): Category I model space which consists of underfit and incorrect models and Category II model space that contains overfit models that also have the "true" model. In Figure 7, variables begin to stabilize once in the Category II model space. For example, from model size 5 and onwards the variables stabilize as one moves to the Category I model space (which is model size 1 to model size 6).

6.1 Another Example of Variable Stability Plot

We continue this discussion with a second data set which is of higher dimension, ozone interaction data which is available in the package as `OzoneI`. The original `Ozone` (Breiman and Friedman, 1985) data set is modified to encompass all pairwise interactions of main effects, and B-spline basis functions of up to 6 degrees of freedom for all original predictors. The original data set consists of 203 observations on 12 variables with `ozone` being the outcome variable. With this modification the `OzoneI` data has 203 observations and 89 variables.

```
R> library("modelSampler")
R> data("OzoneI")
R> ms.boot.I <- boot.modelSampler(ozone~., OzoneI, n.iter1=2500,
+ n.iter2=2500, B=100, verbose=FALSE)
R> plot.ooberror(ms.boot.I,"Ozone interaction data")
```

Figure 8 is the out-of-bag prediction error plot for the ozone interaction data set. The "best" model for this data set is found to be the model of size 13. For model size 9 and 13, the PE is almost same and in between region almost flat. Again for model size 13 and onwards the PE is quite flat till model size 36. Therefore selection based on the PE may lead to choosing any model within this region. The data set comprises of interactions between independent variables, and it is evident that there is significant amount of association between the variables. Therefore for better prediction in the final model,we embrace for interaction by negotiating the interplay between variable selection and prediction. For this reason the variable stability plot plays a decisive role in selecting the model based on the PE plot. Figure 9 represents the variable stability plot for the ozone interaction data set. It looks like variables begin to stabilize from models of size 13 (which is the best model according to the analysis), but the plot suggests that larger models win over selection based on their stability. Models of size 35 or 36 may be the desired model based on the stabilization criterion. The following command generates the variable stability plot for the ozone interaction data

```
R> plot.var.stability(ms.boot.I)
```

We have generated another graphics where only those variables are plotted that emerge as the most significant variables from our analysis using the option `filter.flag=TRUE` in the `plot.var.stability` function.

From Figure 10 it is now clear that the important variables begin to stabilize once crossing the model of size 13, but if we proceed further to models of size 35 or 36 stabilization occurs. In a nutshell one can use a combination of the variable stability plot and the PE error plot as a graphical tool for model selection, as well as to get an overview of the best set of competing models in any given situation.

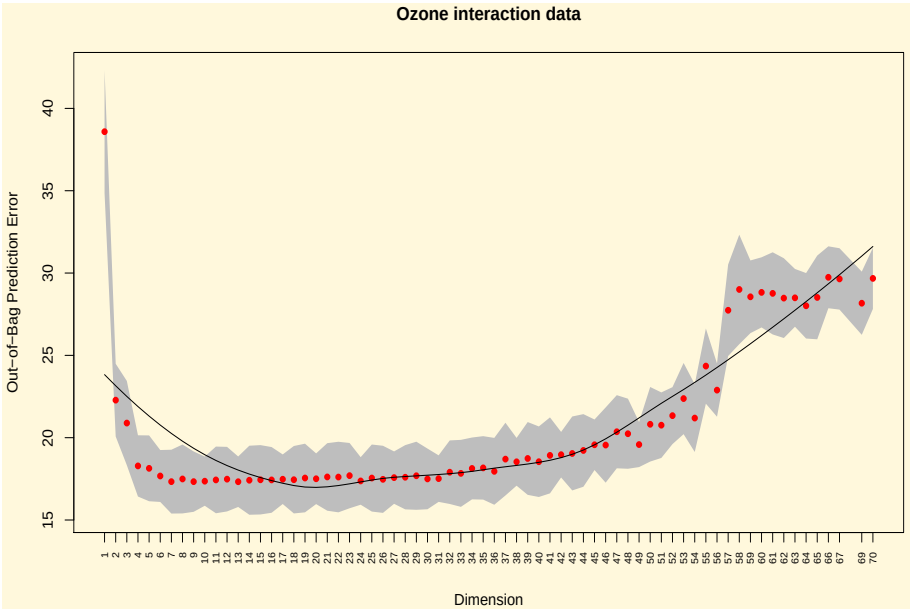


Figure 8: Out-of-bag prediction error plot for the Ozone interaction data. Results are based on 100 bootstrap iterations. Red points are model size specific estimated PE. Gray band surrounding the red points are PE $\pm$ standarderror (PE). Smooth black line represents loess estimated line of PE

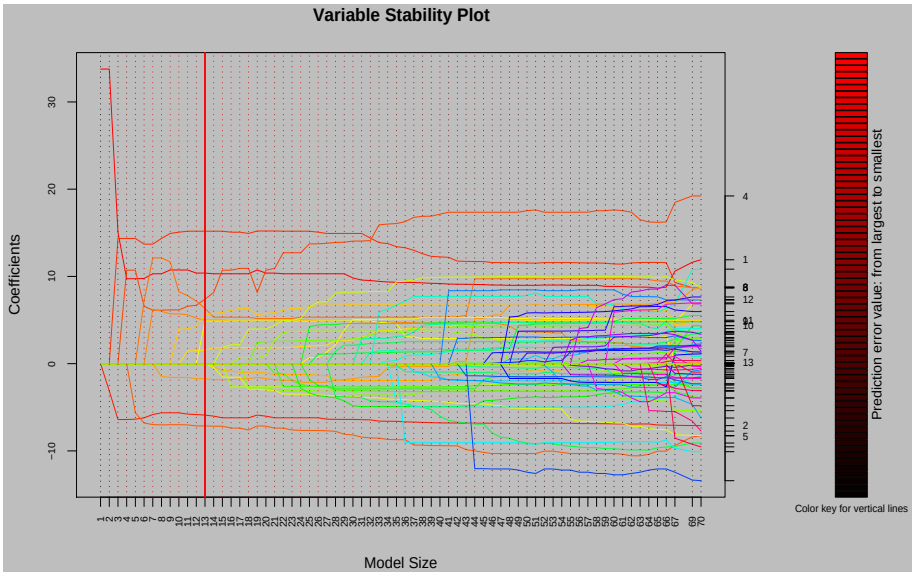


Figure 9: Variable stability plot for the Ozone interaction data. Results are based on 100 bootstrap iterations. Horizontal lines are model size specific ensemble hard shrunk estimators. Dashed vertical lines are model size specific PE. Vertical solid line represents the model with smallest PE

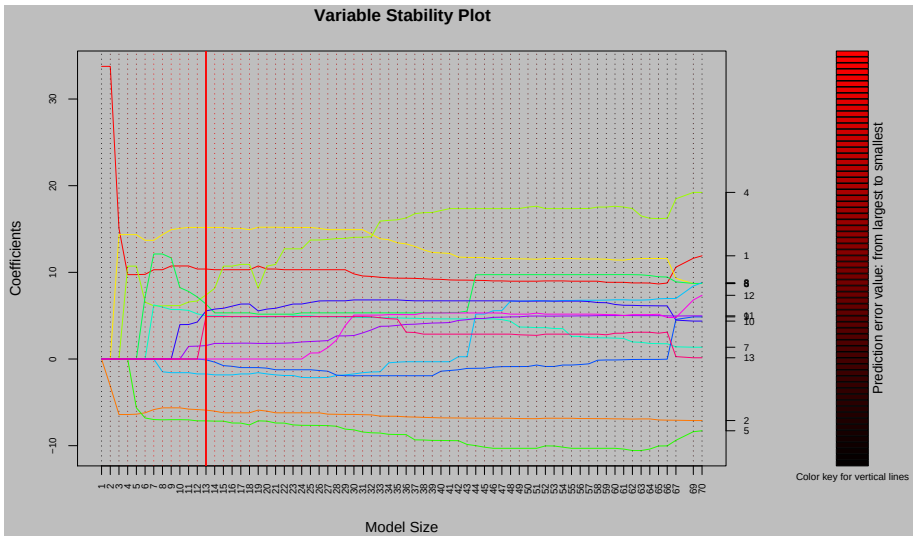


Figure 10: Variable stability plot for the Ozone interaction data. Results are based on 100 bootstrap iterations. Horizontal lines are model size specific ensemble hard shrunk estimators. Dash vertical lines are model size specific PE. The vertical solid line represents the model with smallest PE. Only the significant variables from the resulting analysis are displayed

```
R> plot.var.stability(ms.boot.I, filter.flag = TRUE)
```

The categorization of a model space is now quite distinct from Figure 10. In Figure 10, in the Category I space, variables are not stable, but proceeding towards Category II space, variables begin to stabilize. It is worthy to note that moving forward from our "best" model towards the full model, variables lose their stability because of these overfit models.

7. Discussion

This R package employs both frequentist and Bayesian approach to perform variable selection: variable selection can be performed via simple FPE analysis using the `modelSampler` function at the same time more stable variable selection can be done using ensemble technique using the `boot.modelSampler` function. Besides these frequentist techniques, highest posterior model selection and also variable selection by median model (Barbieri, 2004) (variables with posterior inclusion probability greater than or equal to 0.5) are available for Bayesian analysis (see the output from the `Diabetes` data analysis). Using Gelman-Rubin diagnostics we show that the `modeSampler` converges quickly. Besides stable variable selection based on empirical study, the **modelSampler** package displays

competitive prediction error performance, and it outperforms popular predictive methods like Random Forest, Boosting and BMA.

The proposed stable estimators discuss model uncertainty and helps to illustrate the model space graphically. The `boot.modelSampler` function offers an optimal solution but sometimes the "optimal solution" is not the best solution. Hence while dealing with massive quantities of data with the goal to obtain an optimal solution, `boot.modelSampler` would be a prolong and delayed approach, but `modelSampler` offers faster solution and better performance in the area of "data mining".

Appendix: Gibbs Sampler

Here we outline a Gibbs sampler for drawing posterior values from (2) under the prior (3). The prior for ν are considered as: a uniform prior for w.

When running the Gibbs sampler we keep track of the different α models as they are sampled. For this purpose we introduce the following notation. Defining

$$I_k = \begin{cases} 1, & \text{if } \gamma_k = V = n, \\ 0, & \text{otherwise.} \end{cases}$$

Thus each draw for $\boldsymbol{\gamma}$ has an associated binary K-tuple $(I_1, \cdots, I_K)$ which is identified with the model $\alpha = \{k : I_k = 1\}$. Using this notation the Gibbs sampler is defined as follows:

1. Draw $(\boldsymbol{\beta}|\boldsymbol{\gamma}, \tilde{\boldsymbol{Y}}) \sim \mathcal{N}(\boldsymbol{\mu}, n\boldsymbol{\Sigma})$, where $\tilde{\boldsymbol{Y}} = \{\tilde{Y}_1, \cdots, \tilde{Y}_n\}$, $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{X}^t\tilde{\boldsymbol{Y}}$ and $\boldsymbol{\Sigma} = (\boldsymbol{X}^t\boldsymbol{X} + n\boldsymbol{\Gamma}^{-1})^{-1}$.
2. Draw γ_k from

$$(\gamma_k|\beta_k, w) \overset{\text{ind}}{\sim} \frac{w_{1,k}}{w_{1,k} + w_{2,k}}\delta_{v_0}(\cdot) + \frac{w_{2,k}}{w_{1,k} + w_{2,k}}\delta_V(\cdot), \quad k = 1, \cdots, K,$$

   where $V = n$ and

$$w_{1,k} = (1 - w)v_0^{-1/2} \exp\left(-\frac{\beta_k^2}{2v_0}\right) \quad \text{and} \quad w_{2,k} = wV^{-1/2} \exp\left(-\frac{\beta_k^2}{2V}\right).$$

3. Draw w from $(w|\boldsymbol{\gamma}) \sim \text{Beta}(1 + \#\{k : I_k = 1\}, 1 + \#\{k : I_k = 0\})$.
4. This completes one iteration. Update I_k using $\boldsymbol{\gamma}_k$ and define α. Compute the RSS for the current draw: $\text{RSS}(\alpha) = ||\boldsymbol{Y} - \hat{\boldsymbol{Y}}(\alpha)||^2$, where

$$\hat{\boldsymbol{Y}}(\alpha) = \boldsymbol{X}_{+\alpha}(\boldsymbol{X}_{+\alpha}^t\boldsymbol{X}_{+\alpha} + \boldsymbol{I}_{+\alpha})^{-1}\boldsymbol{X}_{+\alpha}^t\boldsymbol{Y}.$$

5. Computing penalized RSS values is straightforward once RSS has been determined. For example, the AIC information criterion is defined as

$$\mathrm{AIC}(\alpha) = n^{-1}\mathrm{RSS}(\alpha) + 2n^{-1}\hat{\sigma}^2 K_\alpha,$$

whereas BIC is defined as

$$\mathrm{BIC}(\alpha) = n^{-1}\mathrm{RSS}(\alpha) + n^{-1}\hat{\sigma}^2 K_\alpha \log(n).$$

Acknowledgements

The author would like to thank the referee, the associate editor and the editor who helped to substantially improve the paper.

References

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics* **21**, 243-247.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory* (Edited by B. N. Petrov and F. Czáki), 267-281. Akadémiai Kiadó, Budapest.

Barbieri, M. M. and Berger, J. O. (2004). Optimal predictive model selection. *Annals of Statistics* **32**, 870-897.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation (with discussion). *Journal of the American Statistical Association* **80**, 580-619.

Dey, T. (2012). A bimodal spike and slab model for variable selection and model space exploration. *Journal of Data Science* **10**, 363-383.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion). *Annals of Statistics* **32**, 407-499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348-1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of Royal Statistical Society, Series B* **70**, 849-911.

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association* **88**, 881-889.

Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial (with discussion). *Statistical Science* **14**, 382-417.

Ishwaran, H. and Rao, J. S. (2003). Detecting differentially expressed genes in microarrays using Bayesian model selection. *Journal of the American Statistical Association* **98**, 438-455.

Ishwaran, H and Rao, J. S. (2005). Spike and slab gene selection for multigroup microarray data. *Journal of the American Statistical Association* **100**, 764-780.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of Statistics* **33**, 730-773.

Ishwaran, H. and Rao, J. S. (2011). Consistency of spike and slab regression. *Statistics & Probability Letters* **81**, 1920-1928.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *Rnews* **2**, 18-22.

Lumley, T. (2010). **leaps**: regression subset selection. R package version 2.9.

Raftery, A., Hoeting, J., Volinsky, C., Painter, I. and Yeung, K. Y. (2010). **BMA**: Bayesian model averaging. R package version 3.12.

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics* **31**, 172-181.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning* **5**, 197-227.

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

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association* **88**, 486-494.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* **58**, 267-288.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* **67**, 301-320.

Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics* **37**, 1733-1751.

Received July 10, 2012; accepted December 13, 2012.

Tanujit Dey
Department of Mathematics
College of William & Mary
Williamsburg, VA 23185, USA
tdey@wm.edu