2014

Frequent approximation of the Bayesian Posterior Inclusion Probability by stochastic subsampling

Ernest Fokoue

VI Ly

Frequentist Approximation of the Bayesian Posterior Inclusion Probability by Stochastic Subsampling

Vi Ly
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive, Rochester, NY 14623, USA

Ernest Fokoué
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive, Rochester, NY 14623, USA

Abstract

This paper uses stochastic subsampling of the dataset to provide a frequentist approximation of what is known in the Bsyesian framework as the posterior inclusion probability. The unique merit of this contribution lies in the fact that it makes it easier for typically frequentist-minded practitioners, of which there are very many, to relate to the way the Bayesian paradigm allows the computation of the nicely interpretable variable importance. Despite its computationally intensive nature due to the need to fit a very large number of models, the proposed approach is readily applicable to both classification and regression tasks and can be done in comparatively comparative computational times thanks to the availability of parallel computing facilities through cloud and cluster computing. Finally, the scheme proposed is very general and can therefore be easily adapted to all kinds of statistical prediction tasks. Application of the proposed method to some very famous benchmark datasets shows that it mimics the Bayesian counterpart quite well in the important context of variable selection.

I. Introduction

Modern statistical learning and data mining are filled with thousands of studies where the main statistical task revolves around estimation and prediction based on the traditional multiple linear regression (MLR) model given by

$$M_f: \quad \boldsymbol{Y} = \alpha \mathbf{1}_n + \boldsymbol{X}_f \boldsymbol{\beta}_f + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\beta}_f = (\beta_1, \beta_2, \cdots, \beta_p)^\top$, $\boldsymbol{Y} = (\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_n)^\top$, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n)^\top \sim \texttt{MVN}(0, \sigma^2 \boldsymbol{I}_n)$, the design matrix $\boldsymbol{X}_f$ is an $n \times p$ matrix, and $\mathbf{1}_n = (1, 1, \cdots, 1)^\top$ is a $n \times 1$ dimensional vector of 1's. We shall refer to (1) as the *full model*. We assume that many of the β_j's are essentially zero, so that the intrinsic rank of the design matrix $\boldsymbol{X}_f$ is a number $q \in \mathbb{N}$ with $q \ll p$. Many data mining problems do exhibit such a characteristic of rank deficiency, main because variables are typically pick up as they are available, and therefore will turn out to be either noise variable (no relationship with the response) or redundant variables. A basic result in the linear model analysis

shows that when $\boldsymbol{X}_f$ is rank deficient, the ordinary least squares estimator

$$\hat{\boldsymbol{\beta}}_f^{(\texttt{OLS})} = (\boldsymbol{X}_f^\top \boldsymbol{X}_f)^{-1} \boldsymbol{X}_f^\top \boldsymbol{Y}$$

of $\boldsymbol{\beta}_f$ will tend to exhibit a high (inflated) variance, thereby corrupting all predictions and inferences with the computed model. It is therefore crucial to determine (if possible) the intrinsic model that generated the data, i.e. the model made up of only the most significant and non redundant variables. For many decades, both frequentist and Bayesian statisticians have contributed substantially to this topic of *variable selection*. In elementary statistical regression analysis courses, the method of choice for variable selection has been overwhelmingly frequentist with *stepwise regression heuristic* occupying a prominent place, and *best subsets selection* occasionally used whenever possible. While a heuristic like stepwise regression does provide a workable approach to variable selection, it is not a principled method, and does have the extra limitation of not providing any measure of variable importance. Besides, when the number of variables p is larger than the sample size n (a setting now knows as large p small n or short fat data), the stepwise regression heuristic cannot be used because the submodels cannot even be built, let alone compared. In recent years, both frequentists and Bayesians have developed new methods for handling some of the most formidable variable selection tasks, many of which arose from the statistical learning and data mining community. Various Bayesian statisticians have contributed a wealth of scholarly research work covering both the traditional setting of variable selection where n is much larger than p and the now popular and more tricky short fat data context where p is much larger than n.

II. Bayesian Approach to Variable Selection

The vast majority of Bayesian contributions to variable selection of late have concentrated on the use of conjugate prior, with the typical choice of prior on $\boldsymbol{\beta}$ being a Gaussian prior of the form

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{W} \sim \texttt{MVN}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{W}^{-1}\right),$$

where $\boldsymbol{W}$ is the prior precision matrix. Of course, the use of a zero mean prior expresses the assumption of many insignificant coefficients. However, even more important is the use of a vector of indicator variables that ultimately provides a mechanism (device) for performing variable selection. One of the key building blocks of the Bayesian variable selection machinery is the use of a vector of indicator variables. With the p original predictor variables, there are $2^p - 1$ non empty models corresponding each to a subset of the provided variables. We shall use a vector $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_p)^\top$ to denote the index of a given model, with each γ_j being an indicator of the variable's presence in the model under consideration, namely

$$\gamma_j = \begin{cases} 1 & \texttt{If variable } \ X_j \ \texttt{ appears in the model} \\ 0 & \texttt{Otherwise} \end{cases}$$

Clearly, $\boldsymbol{\gamma} = (1, 1, \cdots, 1)^\top$ corresponds to the *full model* M_f, while $\boldsymbol{\gamma} = (0, 0, \cdots, 0)^\top$ corresponds to the empty model also referred to as the *null model*, and given by

$$M_n: \quad \boldsymbol{Y} = \alpha \boldsymbol{1}_n + \boldsymbol{\epsilon}. \tag{2}$$

Equipped with this index, $p_\gamma = \sum_{j=1}^{p} \gamma_j$ is the number of predictor variables in model M_γ, and $\boldsymbol{\beta}_\gamma$ is the subset of $\boldsymbol{\beta}$ made up of only the β_j's picked up by $\boldsymbol{\gamma}$. Finally, $\boldsymbol{X}_\gamma$ is the submatrix of $\boldsymbol{X}$ whose columns are only those p_γ columns of $\boldsymbol{X}$ picked up by $\boldsymbol{\gamma}$, so that $\boldsymbol{X}_\gamma$ is really an $n \times p_\gamma$ matrix, and the corresponding model M_γ is given by

$$M_\gamma : \quad \boldsymbol{Y} = \alpha \mathbf{1}_n + \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma + \boldsymbol{\epsilon}. \tag{3}$$

For the normal linear model, we have $[\boldsymbol{y}|\alpha, \boldsymbol{\beta}_\gamma, \sigma^2, M_\gamma] \sim \texttt{MVN}(\alpha \mathbf{1}_n + \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma, \sigma^2 \boldsymbol{I}_n)$, which means that

$$p(\boldsymbol{y}|\boldsymbol{\theta}_\gamma, M_\gamma) = \frac{1}{\sqrt{((2\pi)\sigma^2)^n}} \exp\left\{ -\frac{1}{2} (\boldsymbol{y} - \alpha \mathbf{1}_n - \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma)^\top (\boldsymbol{y} - \alpha \mathbf{1}_n - \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma) \right\}$$

where $\boldsymbol{\theta}_\gamma = \{\alpha, \boldsymbol{\beta}_\gamma, \sigma^2\}$. When it comes to Bayesian variable selection, arguably the most crucial ingredient is the posterior density of a given model, given by

$$p(M_\gamma|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|M_\gamma) p(M_\gamma)}{\sum_{\gamma \in \boldsymbol{\Gamma}} p(\boldsymbol{y}|M_\gamma) p(M_\gamma)}$$

where $\boldsymbol{\Gamma} = \{0,1\}^p$ and $p(\boldsymbol{y}|M_\gamma)$ is the marginal density of the data, also referred to as the marginal likelihood of model M_γ, and defined by

$$p(\boldsymbol{y}|M_\gamma) = \int_\Theta p(\boldsymbol{y}|\boldsymbol{\theta}_\gamma, M_\gamma) p(\boldsymbol{\theta}_\gamma|M_\gamma) d\boldsymbol{\theta}_\gamma.$$

In some special cases, it is possible to derive closed-form (analytical) expressions for $p(\boldsymbol{y}|M_\gamma)$, but in general, it must be approximated using a variety of schemes. The posterior probability $p(M_\gamma|\boldsymbol{y})$ of model M_γ, plays a central role in Bayesian learning.

$$p(\boldsymbol{z}|\boldsymbol{y}) = \sum_{\gamma \in \boldsymbol{\Gamma}} p(\boldsymbol{z}|M_\gamma, \boldsymbol{y}) p(M_\gamma|\boldsymbol{y})$$

and also

$$\mathbb{E}(\boldsymbol{z}|\boldsymbol{y}) = \sum_{\gamma \in \boldsymbol{\Gamma}} \mathbb{E}(\boldsymbol{z}|M_\gamma, \boldsymbol{y}) p(M_\gamma|\boldsymbol{y})$$

Among Bayesian statisticians, there are those who suggest that when it comes to model selection, one must choose the model with the highest posterior density model, i.e.,

$$\gamma_{\texttt{HPM}} = \underset{\gamma \in \boldsymbol{\Gamma}}{\texttt{argmax}} \{p(M_\gamma|\boldsymbol{y})\}$$

Barbieri and Berger (2004) have suggested selecting instead the so-called median probability model (MPM) given $\gamma_{\texttt{MPM}}$, such that

$$[\gamma_j]_{\texttt{MPM}} = \begin{cases} 1 & \texttt{if} \, \pi_j = \Pr[\gamma_j = 1|\boldsymbol{y}] \geq \frac{1}{2} \\ 0 & \texttt{otherwise} \end{cases} \tag{4}$$

In practice, the estimated posterior inclusion probability is given by

$$\widehat{\texttt{PIP}}_j = \hat{\pi}_j = \Pr[\widehat{\gamma_j = 1}|\boldsymbol{y}] = \sum_{\gamma : \gamma_j = 1} p(M_\gamma|\boldsymbol{y}), \tag{5}$$

which means that the importance of a variable is measured in terms of its relative frequency of appearance in models. In Equation (5), it is crucial to be able to compute the posterior density of a given model. Empirically, this means being able to generate a least a representative subset (sample) of all the models, and then computing estimates of the posterior density from them.

III. Main result

The frequentist's alternative to Bayesian PIP takes on the following form:

1. Split the dataset into training / test sets by random sampling (a good rule of thumb is 70% of data for training set and the remaining 30% for test set). Repeat this process numerous times to generate multiple training / test sets.

2. Define a class of models to build from training sets (Simple Linear Regression (SLR), Multiple Linear Regression (MLR), Logistic Regression etc.)

3. Apply variable selection techniques to training sets (stepwise regression via BIC, stepwise regression via AIC, forward selection via BIC, etc.)

4. Scan through the newly built models and calculate the percentage in which each explanatory variable was deemed significant (p-value $\leq \alpha$ with standard α of 0.05). Rank variables by their percentages to determine their importance to a model. Retain variables which are deemed significant in $\geq 50\%$ of the replicates (This is comparable to the median probability model used in a Bayesian PIP framework). Discard variables which are deemed significant in $< 50\%$ of the replicates.

5. Build new models using only the variables retained from step 4. Apply these newly built models on their corresponding test sets to ascertain out of sample prediction error. Select the model which satisfies the user's end goal (lowest prediction error, highest accuracy, most parsimonious model, best convergence rate).

IV. Computational demonstrations

IV.1 Computational demonstration on pattern recognition

According to Microsoft, spam is a term used to classify unwanted email. Spam may contain viruses or other malicious programs that can harm a computer. Furthermore, spam may be used as scams to acquire vital personal information such as credit card accounts, bank accounts, social security numbers, etc. Spam filters have been developed as preventative measures to protect the end user from ever opening these emails. Text categorization is one of the several techniques used to create spam filters. A number of terms are identified as indicators of spam/non-spam from a training set of emails. In the simplest spam filters, the frequencies of these terms in an email are determined and used to flag emails as spam/non-spam Fumera et al. (2006). The purpose of this paper is threefold:

1. Use logistic regression to generate a spam / non-spam classification algorithm and determine its effectiveness

2. Present a frequentist alternative to Bayesian Posterior Inclusion Probability (PIP) for variable selection

3. Compare logistic regression classification accuracy with accuracies of newer machine learning algorithms

Data used for this regression analysis came from the ubiquitous Spambase dataset at the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets/Spambase`). This dataset was donated by Forman et al. (1999) from Hewlett-Packard Labs. The collection of emails was provided from Forman et al. (1999)'s email account at Hewlett-Packard. The dataset consisted of $4,601$ observations with 57 explanatory variables and 1 response variable. 54 of the 57 explanatory variables measure the percentage of an email in which specific words or characters appear. The remaining 3 explanatory variables measure the average length of uninterrupted sequences of capital letters, length of longest uninterrupted sequence of capital letters, and the total number of capital letters in an email. The response variable was binary coded 1 for spam and 0 for non-spam. Out of the $4,601$ observations, $2,788$ emails were non-spam (60.60%) and $1,813$ emails were categorized as spam (39.40%). The Spambase dataset webpage on the UCI Machine Learning Repository cited an average $\sim 7\%$ misclassification error. The goal of this project is to generate a classification model using regression techniques while matching or beating the 7% misclassification error.
The figure below shows the linear pairwise correlations between all of the explanatory variables. Based on the correlation plot, there is potential for multicollinearity to affect regression results. Due to the multicollinearity, variable selection will be required during the model building phase. Even with variable selection, there are still some questions that remain unanswered. What is the optimal model size to achieve model complexity-testing accuracy tradeoff? How much confidence should be placed on the variables identified by variable selection as significant variables? Furthermore, how does one characterize the importance of a variable's contribution to the model? We will provide some insight into these questions through a frequentist approach to variable selection.

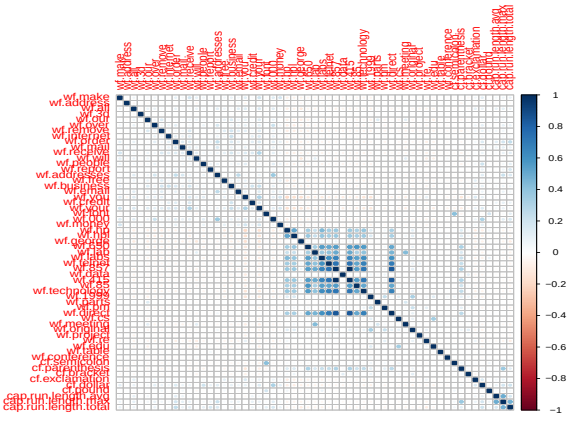

Figure 1: *Figure: Correlation Plot*

The frequentist's alternative to Bayesian PIP takes on the following form:

1. Split the dataset into training / test sets by random sampling (a good rule of thumb is 70% of data for training set and the remaining 30% for test set). Repeat this process numerous times to generate multiple training / test sets.

2. Define a class of models to build from training sets (Simple Linear Regression (SLR), Multiple Linear Regression (MLR), Logistic Regression etc.)

3. Apply variable selection techniques to training sets (stepwise regression via BIC, stepwise regression via AIC, forward selection via BIC, etc.)

4. Scan through the newly built models and calculate the percentage in which each explanatory variable was deemed significant (p-value $\leq \alpha$ with standard α of 0.05). Rank variables by their percentages to determine their importance to a model. Retain variables which are deemed significant in $\geq 50\%$ of the replicates (This is comparable to the median probability model used in a Bayesian PIP framework). Discard variables which are deemed significant in $< 50\%$ of the replicates.

5. Build new models using only the variables retained from step 4. Apply these newly built models on their corresponding test sets to ascertain out of sample prediction error. Select the model which satisfies the user's end goal (lowest prediction error, highest accuracy, most parsimonious model, best convergence rate).

The frequentist's alternative approach to PIP will now be applied to the Spambase dataset. The entire dataset consisted of $4,601$ observations. For step 1, 70% of the observations ($3,221$ observations) were randomly sampled from the entire dataset to form the training set. The remaining 30% of the observations ($1,380$ observations) formed the test set. This process was repeated 100 times to form 100 replicates of 70/30 training/test split. Since the main goal will be binary classification of spam / non-spam, the logistic regression model was selected for step 2. A model was built for each of the 100 training replicates using logistic regression with Logit link function on all 57 explanatory variables. The 100 models were then applied on their corresponding test sets to calculate the out-of-sample accuracies. The left-most figure below is a comparative boxplot between the training and test set accuracies of the Logit link function on all 57 explanatory variables. The same process was repeated using the Probit (center plot below) and Cauchit (right plot below) link functions. Based on the plots and number summaries below, the Cauchit link function provided the best family of models with respect to training and test set accuracies.

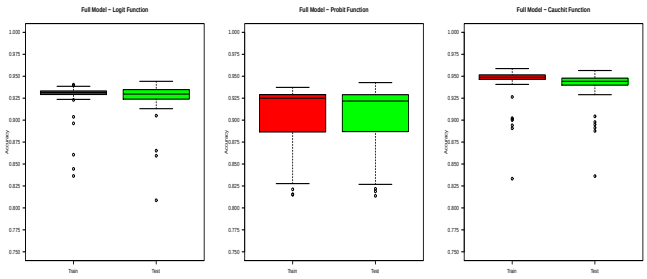

Figure 2: *Figure: Comparative Boxplots of Training/Testing Accuracies - Logit (Left); Probit (Center); Cauchit (Right)*

With our link function in hand, we will now approach the daunting task of variable selection in step 3. Stepwise regression via Akaike Information Criterion (AIC) was applied to each of the 100 replicates. This process was repeated using stepwise regression via Bayesian Information

Model	Min	1st Qu	Median	Mean	3rd Qu	Max
Logit	83.64%	92.92%	93.14%	92.83%	92.83%	93.32%
Probit	81.50%	88.66%	92.50%	90.50%	92.90%	93.73%
Cauchit	69.95%	94.65%	94.88%	94.28%	95.16%	95.87%

`Link function`	Min	1st Qu	Median	Mean	3rd Qu	Max
`Logit`	80.87%	92.39%	92.97%	92.69%	93.48%	94.42%
`Probit`	81.38%	88.73%	92.17%	90.43%	92.90%	94.28%
`Cauchit`	69.86%	93.99%	94.42%	93.84%	94.78%	95.65%

Criterion (BIC), forward selection via BIC, and the full model. The following figure provides boxplots for the number of variables deemed significant in a replicate. For example, in AIC stepwise regression method, 1 of the 100 replicates deemed 30 of the 57 variables significant while another replicate deemed 18 of the 57 variables significant. The figure provides insight that the optimal model size should be in the low 20′*s*. However, there is a now a dilemma. The plot also shows the variability in identifying significant variables due to the variability of training set data. By choosing the results from a single replicate, it is possible that a noise variable was deemed significant or a significant variable was missed due simply to the sample training data. This potential error is further demonstrated by the next two examples.

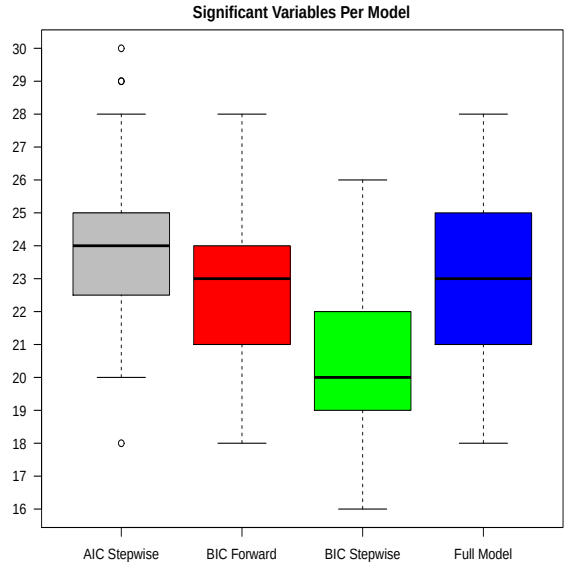

Figure 3: *Figure: Number of Variables Deemed Significant in Replications*

The first example below shows 2 of the 100 replicates after performing AIC stepwise regression. In the replicate on the left, 18 variables were deemed significant, while 30 variables were deemed significant in the replicate on the right.
The second example below shows 2 of the 100 replicates after performing BIC stepwise regression. In the replicate on the left, 17 variables were deemed significant, while 26 variables were deemed significant in the replicate on the right.

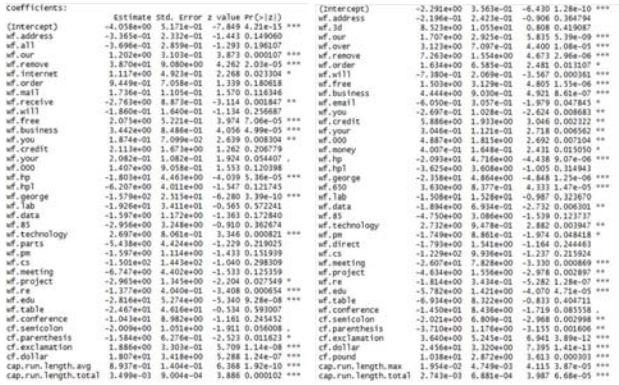

Figure 4: *Figure:* 2 *Replicates After AIC Stepwise Regression*

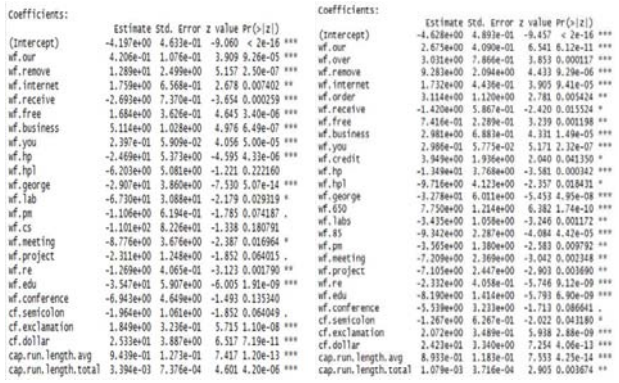

Figure 5: *Figure:* 2 *Replicates After BIC Stepwise Regression*

In the next section, we will examine how does one come with a more robust method to truly identify significant variables?

The percentage in which each explanatory variable was deemed significant (p-value ≤ 0.05) out of the 100 training replicates was calculated and plotted in the figure below (step 4). There are several advantages to this approach. Firstly, we have identified, with high confidence, the 18 significant variables to comprise our core model. These variables are wf.remove, wf.hp, wf.re, wf.our, wf.free, wf.edu, cf.exclamation, cf.dollar, wf.business, wf.george, wf.project, cap.run.length.total, wf.your, wf.000, wf.internet, wf.receive, wf.over, and wf.money (note: wf is acronym for word frequency, cf is acronym for character frequency, and cap.run.length.total is the total number of capital letters in an email); for the remainder of this analysis, they will be referred to as the 18 core variables. There is high confidence that these 18 variables are significant variables due to their robustness. They were consistently identified as significant in $\geq 50\%$ of the replicates in all 4 of the variable selection methods despite the variability in the training sets' observations.

Secondly, this approach also affords flexibility to the end user in modeling. There were 7 variables (marked by red dotted vertical lines) which were deemed significant in $\geq 50\%$ of the replicates by at least 1 of the 4 variable selection methods but not by all 4. These variables are wf.technology, cap.run.length.avg, wf.meeting, wf.order, wf.you, wf.credit, and cap.run.length.max. Depending on the end user's threshold for model complexity-accuracy tradeoff, the user can experiment building models with any combination of these 7 variables in addition to the 18 core variables. Furthermore, each variable's importance can now be characterized by the percent of replicates in which

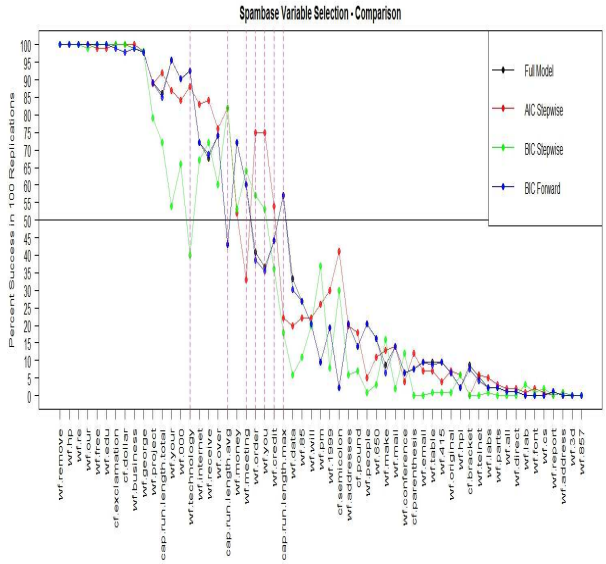

Figure 6: *Figure: Percentage in Which Each Explanatory Variable Deemed Significant*

they are deemed significant. For example, it may not be cost-effective for a business to measure all 18 core variables or may require too much computing power; as a result, the user may be constrained to using only 10 variables. How would the end user decide which 10 variables to use? Based on the plot, the user should select the first 10 variables (wf.remove, wf.hp, wf.re, wf.our, wf.free, wf.edu, cf.exclamation, cf.dollar, wf.business, and wf.george) because they were statistically significant in $>$ 90% of the replicates in all 4 variable selection techniques. One can view this approach as an alternative to Mallow's Cp.

The remaining 32 variables (the variables to the right of the rightmost red dotted line) can be discarded because they never exceeded 50% in the replicate sets for any of the 4 selection methods. These 32 variables are more susceptible to the randomness of the observations in the training replicates. When one compares multiple replicates, these variables will not be consistently deemed significant. The potential error in using only one variable selection method becomes evident. Suppose a modeler selected the model with the highest test set accuracy after performing only BIC stepwise regression (represented by green line in the plot above). There is $\sim$ 20% chance the chosen model would not include wf.project (the percentage of an email in which the word "project" appears). However, when looking at variable selection from an ensemble point of view, there is high confidence that wf.project is a significant variable despite not being deemed significant in a replicate. Alternatively, there is a 30% chance the chosen model would include cf.semicolon (the percentage of an email in which the character ";" appears). However, when looking at variable selection from an ensemble point of view, there is low confidence that cf.semicolon is a significant variable because it never reached $\geq$ 50% in any of the variable selection methods.

Five families of models were built in increasing complexity (step 5). For the first family, a model was built for each training replicate using the 18 core variables. The accuracies for calculated for both training and test sets. This process was repeated 4 more times, in which different variables were added onto the 18 core variables (the variables used are listed below).

Let M_c denote the core model containing the 18 variables that always appear in every replication.

Model Complexity	Variables Used
18 Variables	M_c
19 Variables	$M_{19} = M_c \cup \{\texttt{wf.tecnology}\}$
20 Variables	$M_{20} = M_{19} \cup \{\texttt{cap.run.length.avg}\}$
21 Variables	$M_{21} = M_{20} \cup \{\texttt{wf.meeting}\}$
23 Variables	$M_{23} = M_{21} \cup \{\texttt{wf.you}, \texttt{wf.credit}\}$

Model Complexity	Min	1st Qu	Median	Mean	3rd Qu	Max
18	91.09%	93.04%	93.51%	93.50%	93.93%	94.93%
19	91.09%	93.04%	93.48%	93.46%	93.91%	94.86%
20	91.67%	93.32%	93.94%	93.78%	94.22%	95.29%
21	91.09%	93.62%	94.13%	94.05%	94.49%	95.65%
23	90.94%	93.84%	94.13%	94.12%	94.42%	93.51%

For nonlinear models like the ones that arise in the generalized linear model framework, it is often the case that the Fisher scoring algorithm used to estimate the parameters of the model does not converge. When that happens, the solution delivered is at best suboptimal, and may lead to misleading conclusions. For instance, the gist of the method proposed in this paper lies is scanning the variables and choosing the model made up of variables whose p-values are less than 0.05. With a suboptimal solution, it is unwise and misleading to consider the p-values are meaningful. For that reason, we systematically tract all the estimations throughout the totality of our random replications, and we provide an estimate of the percentage of times the estimates of the model are meaningful. It makes sense to us that only the cases where convergence is achieved should be used for inference, because in a sense that measures an aspect of the quality of the model space search. Below is a partial table of the percentages:

Model Complexity	% Converge	% Non-Converge
18	73	27
19	80	20
20	68	32
21	37	63
23	28	72

The table above shows the test set accuracies for all 5 models built. The last two columns in the table above list the percentage of 100 models which converged and the percentage of 100 models which did not converge. All 5 of the models have very respectable accuracies; furthermore, there is relatively little variation in test set accuracies throughout the 100 replicates. The end user now has several models to choose from. If the end user sought the most parsimonious model and is willing to accept a slight loss in accuracy, he/she can select the 18 variable model. If, on the other hand, the end user sought the highest prediction rate, he/she would select the 23 variable model. If the modeler sought the most computationally stable model, he/she would select the 19 variable model; this model had 80% of its replicates converge. For comparison with current machine learning methods, we selected the 20 variable model. The 20 variable model achieved the best tradeoff between model accuracy and convergence rate; this model complexity is in agreement with the predicted optimal model size. The following two figures plot the ROC curves for all 5 models. In the first figure, all 5 ROC curves achieve "right angle" shape. The

second figure provides a close-up of the upper-left corner of the first figure. The 20 variable model (green line) tracks well with the 23 variable model (purple dotted line); both of these have more area under the curve compared to the remaining 3 models.

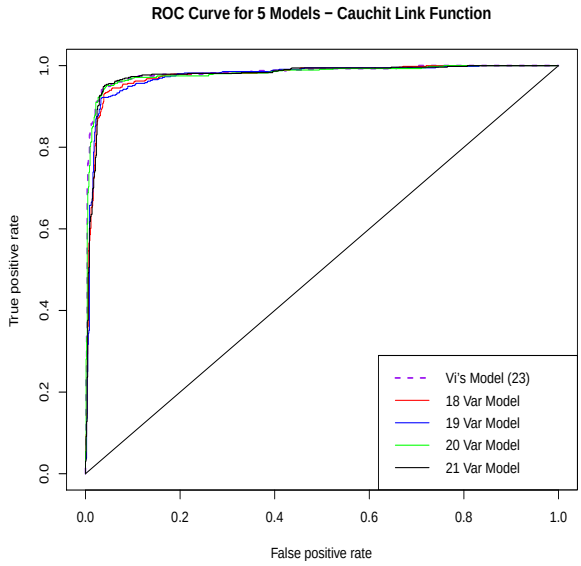

Figure 7: *Figure: ROC Curves for 5 Models*

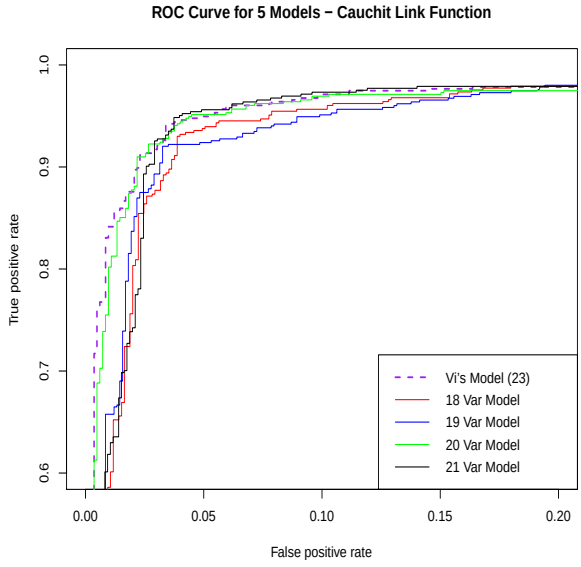

Figure 8: *Figure: Close Up of ROC Curves for 5 Models*

The 20 explanatory variables used to generate the model were wf.remove, wf.hp, wf.re, wf.our, wf.free, wf.edu, cf.exclamation, cf.dollar, wf.business, wf.george, wf.project, cap.run.length.total, wf.your,

wf.000, wf.internet, wf.receive, wf.over, wf.money, wf.technology, and cap.run.length.avg. 6 out of the 20 explanatory variables are indicators of non-spam (as their frequency in an email increases, the more likely an email is not spam); these variables are wf.hp, wf.re, wf.george, wf.edu, wf.project, and wf.technology. 14 out of the 20 explanatory variables are indicators of spam (as their frequency in an email increases, the email is more likely to be spam); these variables are wf.000, wf.money, cf.dollar, wf.free, wf.remove, wf.business, wf.your, cf.exclamation, cap.run.length.avg, cap.run.length.total, wf.internet, wf.over, wf.our, and wf.receive.

When one looks at both classes of indicators, certain patterns arise which can provide further explanation. The variables wf.hp, wf.re, wf.george, wf.project, and wf.technology in the non-spam indicators, suggest a personal or professional relationship with the recipient. Since the dataset was donated by Forman et al. (1999) at Hewlett Packard labs, it makes sense that emails containing "hp" and "George" indicate that the sender either knew the recipient and/or the email was work-related. The characters "re" are often used in emails as replies. Therefore, email replies are flagged as non-spam because the recipient is receiving a reply to an earlier email sent out by the recipient. Additionally, the words meeting and project are usually work-related terms and hence why they are also indicators of non-spam. The variables wf.000, wf.money, cf.dollar, wf.free, and wf.business are associated with money. This makes sense since most spam emails are attempts to get money from the recipient.

In Kiran and Atmosukarto (2005), different machine learning techniques were applied to the Spambase dataset with the goal of optimizing correct classification rate. The following table lists the testing set accuracy for the different machine learning techniques used in the paper. Our 20 variable model, with a median test accuracy of 93.94%, was only beat by Ensemble Decision Tree and Adaboost methods.

Classifier	Accuracy
Ensemble Decision Tree (Nb of trees = 25)	96.40%
Adaboost	95.00%
Stacking	93.80%
SVM	93.40%
Bagging	92.80%
Decision Tree	92.58%
Neural Network	90.80%
Naive Bayes	89.57%
Nearest Neighbor (k=5)	89.40%

Independently, Goosen and Du Toit (2009) applied Generalized Additive Neural Networks (GANNs) to the Spambase dataset to classify email as spam/non-spam. The spambase dataset was broken in 67% training and 33% test. The AutoGANN method used by Goosen and Du Toit (2009) attained a 95.8% accuracy. While this methodâĂŹs accuracy beat the 20 variable model (95.8% $>$ 93.94%), it required higher model complexity with 41 explanatory variables used.

Sivanadyan (2003) used a neural network method called Multi-Layer Perceptron (MLP). In this paper, the Spambase dataset was broken down into training set with $4,025$ observations ($\sim 87.48\%$) and test set with 576 observations ($\sim 12.59\%$). The author applied this technique on 3 different scenarios. In the first scenario, Sivanadyan (2003) used MLP on the training set using all 57 exploratory variables. Due to the high dimensionality, the author noted that the MLP method had trouble converging and consequently, generated poor classification rates. A similar effect was observed in our approach. For our 5 models, the percentage of models that converged drastically decreased when model complexity increased above 20 variables. The following table lists the

accuracies for different models built by the author using all 57 models.

MLP Architecture	Learning Rate	Momentum	Average Classification Rate
$20-10-10-10-5$	0.1	0.80	62.40%
$20-10-10-10-5$	0.1	0.95	63.20%
$20-10-10-5$	0.1	0.85	60.61%
$15-15-15-5$	0.1	0.85	60.59%

The author used the same MLP method after reducing dimensionality to 21 variables. The test accuracies listed below are very comparable to the test set accuracies for our 20 variable model.

MLP Architecture	Learning Rate	Momentum	Average Classification Rate
$20-10-10-10-5$	0.1	0.80	93.50%
$20-10-10-10-5$	0.1	0.95	90.30%
$20-10-10-10-7$	0.1	0.80	93.80%

Lastly, in the third attempt, MLP was applied after reducing dimensionality to 9 variables. The test accuracies are listed below.

MLP Architecture	Learning Rate	Momentum	Average Classification Rate
$20-10-10-8$	0.1	0.80	92.40%
$20-10-10-5$	0.1	0.80	91.70%
$20-10-10-5$	0.1	0.95	91.80%

In a fourth comparison, Sharma and Arora (2013) utilized 9 different machine learning algorithms for their paper *Adaptive Approach for Spam Detection*. In their approach, the data was transformed into $1's$ and $0's$. If a certain word appeared, that exploratory variable was a 1; of the certain word did not appear in an email, the exploratory variable was a 0. This was done for 55 of the 57 exploratory variables. The following table lists the performance of different algorithms after ten-fold cross validation. Our 20 variable model, with a median test accuracy of 93.94%, was only beat by Random Committee and essentially tied with Random Forest technique.

`Algorithm`	`Accuracy`
Bayes Network	88.56%
Logic Boost	89.70%
Random Tree	91.54%
JRip	92.32%
J48	92.34%
Multilayer Perceptron	93.28%
Kstar	93.56%
Random Forest	93.89%
Random Committee	94.28%

There are several advantages to performing frequentist approach. First, this method provides a more robust variable selection by examining how often a variable is deemed significant by multiple traditional variable selection methods given random samples of observation data. Second, it also provides an approximation to the optimal model size. Third, it allows the modeler to

characterize a variableâĂŹs importance to the model through the frequency in which a variable is deemed significant. Lastly, it affords the modeler flexibility in choosing certain variables to retain or discard depending on the modeler's threshold for model-complexity accuracy tradeoff. The main downside to this method is computational intensity. On a Windows 7 64-Bit Laptop with Intel i7 processor and 16GB RAM, this process required $\sim$ 12 hours to just apply BIC stepwise regression to 100 replicates and another 12 hours to apply AIC stepwise regression to the 100 replicates. Running the full model and forward selection via BIC on the 100 replicates was markedly faster and completed within minutes. By incorporating more variable selection methods to the ensemble, the modeler will have a serious tradeoff in computing time. However, this dilemma may be alleviated through the use of parallel processing in which multiple tasks are dispersed over multiple workstations rather than running the tasks sequentially on one computer. With further advances in parallel processing and increases in computing power, the ensemble variable selection method's advantages will significantly dominate over its main weakness.

IV.2 Computational demonstration on regression analysis

In the Spambase dataset, the frequentist approach was applied to classification. In the next example, the frequentist approach was applied for multiple linear regression (MLR) on the Bodyfat dataset. The dataset, which was originally donated by Penrose et al. (1985), attempts to estimate body fat percentage by underwater weighing and various body circumference measurements for 252 men; this dataset may be found in the R package mfp[1]. The dataset contained 2 response variables: brozek and siri. The brozek response variable calculated body fat percentage through the equation:

$$\texttt{brozek} = \frac{457}{\texttt{density}} - 414.2$$

The siri response variable calculated body fat percentage through the equation:

$$\texttt{siri} = \frac{495}{\texttt{density}} - 450$$

There were 14 explanatory variables. The first 3 variables are density (density determined from underwater weighing), age, and weight. The remaining 11 explanatory variables are body circumference measurements for neck, chest, abdomen, hip, thigh, knee, ankle, biceps, forearm, and wrist. The linear pairwise correlation plot below indicates a significant amount of multicollinearity and redundant variables.

The purpose of this section will be to compare the results from the frequentist approach for variable selection against results from Bayesian PIP. There were 5 cases identified in the dataset as erroneous observations/outliers, and as a result, were excluded during analyses. Cases 48, 76, and 96 were identified as having errors in body fat values. In case 42, the man weighed 200 lbs with a height of 3 ft. For case 182, the body fat percentage was rounded to 0 after having a negative body fat percentage. Additionally, the density variable was excluded from MLR because formulas to calculate brozek and siri response variables require density variable; consequently, the inclusion of density variable in MLR would dominate and bias the selection of other significant variables. The dataset was broken into 500 replicates of 70% training / 30% test sets (step 1 of frequentist approach). The linear model (MLR) was selected for the class of models (step 2). AIC stepwise regression, BIC stepwise regression BIC forward selection, and full model were applied to the 500 training sets (step 3). This section will concentrate solely on the brozek response

[1]This bodyfat data set can also be found at `http://lib.stat.cmu.edu/datasets/bodyfat`

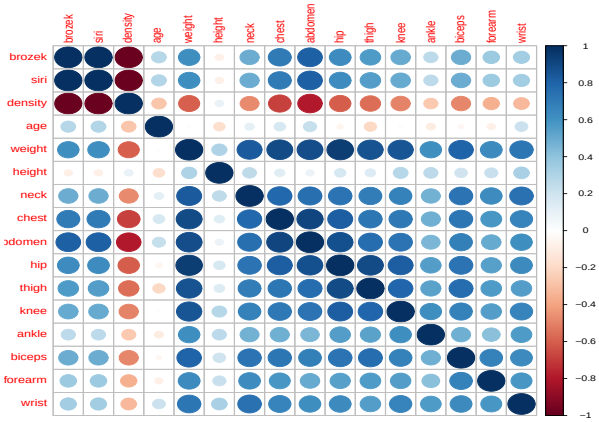

Figure 9: *Figure: Linear Correlation Plot for Bodyfat Dataset*

variable. After scanning through the 500 replicates, the percentage in which each variable was deemed significant is listed below for each of the variable selection methods (step 4).

Table 1: *Estimated Percentage of Inclusion of each variable*

Variable	Full Model	Stepwise BIC	Forward BIC	Stepwise AIC
abdomen	1	1	1	1
wrist	0.8	0.698	0.8	0.842
weight	0.13	0.576	0.13	0.53
forearm	0.386	0.36	0.386	0.516
neck	0.21	0.222	0.21	0.304
hip	0.082	0.22	0.082	0.226
height	0.01	0.166	0.01	0.156
age	0.124	0.08	0.0124	0.2
biceps	0.036	0.08	0.036	0.1
thigh	0.07	0.038	0.07	0.208
chest	0.012	0.028	0.012	0.052
ankle	0.014	0.016	0.014	0.034
knee	0	0.002	0	0.006

The following section will compare the results from our frequentist approach against the results from Bayesian PIP. In order for a more objective comparison, only the stepwise BIC portion will be compared against the Bayesian PIP. The Bayesian PIP results were acquired using the BMS package in R and shown below.

The frequentist approach (looking at stepwise BIC only) and the Bayesian PIP both identified abdomen as the most significant explanatory variable; in both methods, the abdomen variable was deemed significant in 100% of the 500 training sets. Additionally, in both methods, weight

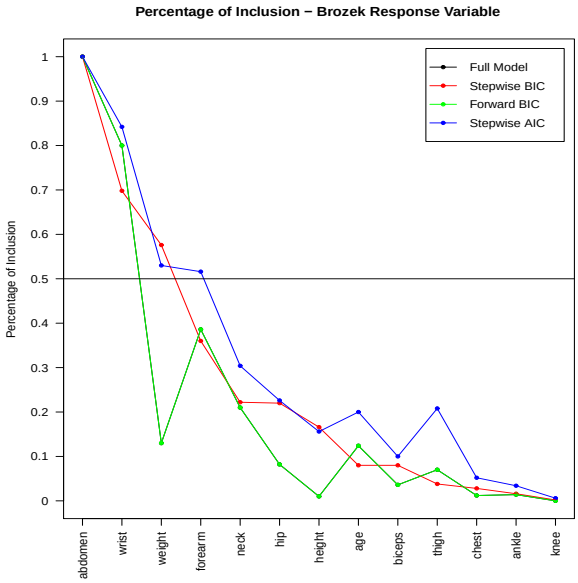

Figure 10: *Figure: Percentage of Inclusion of Explanatory Variables for* brozek *Response Variable - Frequentist Approach*

	PIP	Post Mean	Post SD	Cond.Pos.Sign	Idx
abdomen	1.00000000	0.8994546675	0.05909395	1.00000000	6
weight	0.94254054	-0.1171392909	0.04085303	0.00000000	2
wrist	0.59224949	-0.7619333211	0.72426259	0.00000000	13
forearm	0.28606611	0.1181661995	0.20943063	0.99999662	12
neck	0.14135764	-0.0570056583	0.16396585	0.00000000	4
biceps	0.13742721	0.0382317610	0.11145263	0.99984420	11
hip	0.09782832	-0.0222768348	0.08206601	0.00000000	7
thigh	0.07307283	0.0110396174	0.05035270	0.99731043	8
age	0.05365942	0.0015489301	0.01020422	0.86215779	1
height	0.04282675	-0.0046580449	0.04223018	0.08660439	3
ankle	0.03962332	0.0056336034	0.05104796	0.98806057	10
knee	0.03664630	0.0043967238	0.04736250	0.96027152	9
chest	0.03268106	-0.0005889387	0.01716922	0.28242364	5

Mean no. regressors	Draws	Burnins	Time
"3.4760"	"8192"	"0"	"1.356216 secs"
No. models visited	Modelspace 2^K	% visited	% Topmodels
"8192"	"8192"	"100"	"6.1"
Corr PMP	No. Obs.	Model Prior	g-Prior
"NA"	"247"	"random / 6.5"	"UIP"
Shrinkage-Stats			
"Av=0.996"			

Time difference of 1.356216 secs

Figure 11: *Figure: R Output - Bayesian PIP of Explanatory Variables for* brozek *Response Variable*

and wrist variables exceeded the median probability model (deemed significant in $\geq$ 50% of 500 replicates) and will be retained as significant variables. The main difference between the frequentist approach and PIP is evident in the weight variable. The weight variable was deemed significant in $\sim$ 94% of the 500 models by Bayesian PIP but only $\sim$ 58% by the frequentist method. The following plot shows the distribution of model size across the 500 replicates for the

frequentist approach. The optimal model size should include 3 to 4 variables. The average model size across 500 replicates was 3.49.

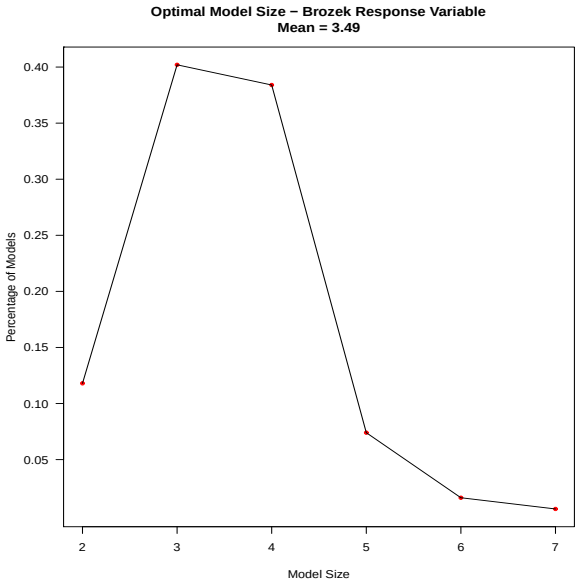

Figure 12: *Figure: Optimal Model Size - Frequentist Approach*

The following plot shows the distribution of model size across 500 replicates for the Bayesian PIP approach. The PIP approach also indicated an optimal model size between 3 to 4 variables with an average model size of 3.48 across 500 replicates. The results between the two methods are very similar.

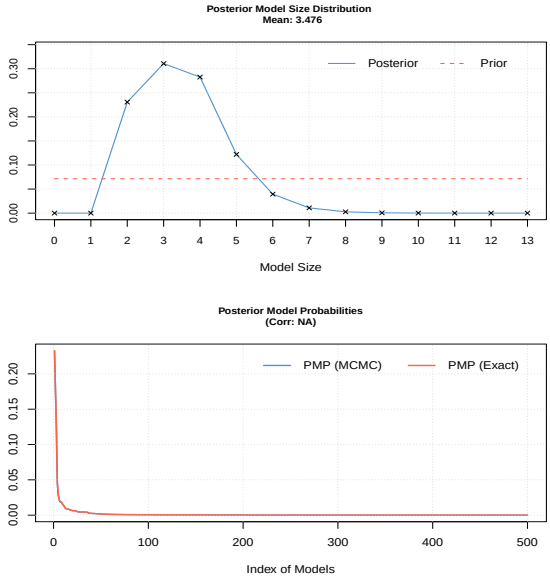

Figure 13: *Figure: Optimal Model Size - Bayesian PIP Approach*

The frequentist approach (looking at stepwise BIC only) and the Bayesian PIP both identified abdomen, weight and wrist as significant variables. Building 500 linear models using only the 3 variables, we obtain the following results shown below. As expected, the out of sample prediction results for the frequentist and Bayesian PIP because we are applying the same variables to the training sets.

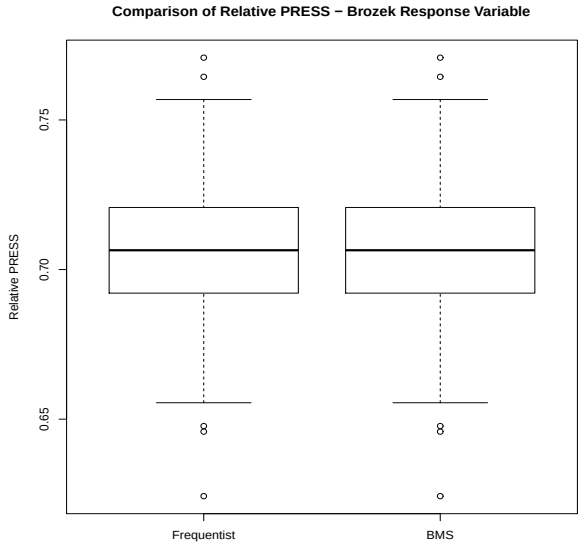

Figure 14: *Figure: Comparison of Relative PRESS - Both Approaches Use* abdomen, weight *and* wrist *Variables*

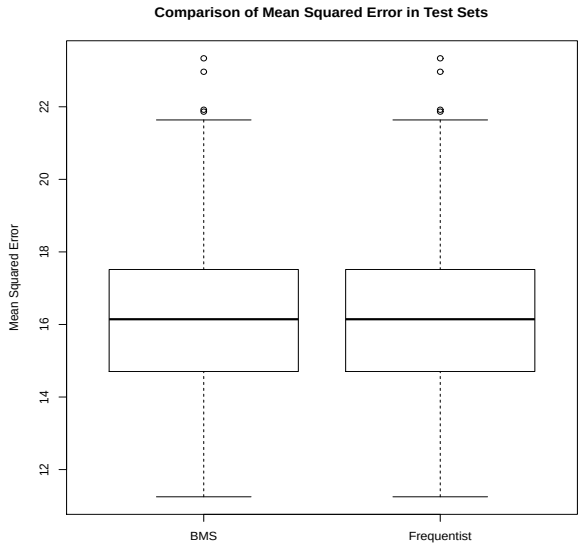

Figure 15: *Figure: Comparison of MSE - Both Approaches Use* abdomen, weight *and* wrist *Variables*

We will now build new sets of models to compare the frequentist and Bayesian PIP approaches. From a meta-analysis standpoint, abdomen and wrist variables were deemed significant in $\geq 50\%$ of the replicates for all 4 variable selection techniques in the frequentist approach. As a result, 500 linear models were built using only the abdomen and wrist explanatory variables. From the PIP standpoint, abdomen, weight, and wrist variables exceeded the median probability model. As a result, 500 linear models were built using abdomen, weight, and wrist explanatory variables. We could have justifiably included weight as an additional third variable in our frequentist approach because it was deemed significant by at least one variable selection technique; however, in doing so, we would get the same end results as the PIP linear models since both methods would now use the same variables. By comparing a two variable model (abdomen and wrist) against a three variable model (abdomen, weight, and wrist), we hope to achieve a more distinction between the two model types. In the next section, we will compare the out of sample predictive performance of the frequentist and the PIP approaches. The following table and plot will compare the relative PRESS values between the two methods.

Model	Min	1st Qu	Median	Mean	3rd Qu	Max
Frequentist	0.6025	0.6638	0.6801	0.6812	0.6983	0.7585
BMS PIP	0.6243	0.6921	0.7065	0.7061	0.7208	0.7708

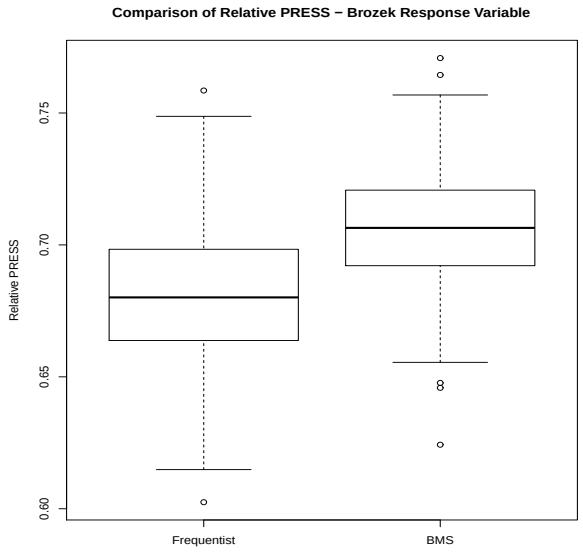

Figure 16: *Figure: Comparison of Relative PRESS*

The 500 linear models built for each of the approaches were applied on their corresponding test sets (step 5). The following table and plot will compare the Mean Squared Error (MSE) across the 500 test sets for both methods.

While weight achieved a PIP of 0.94, its addition into the three variable model did not provide a practical improvement. The results between the PIP and frequentist approaches are very comparable and provides validity of using the frequentist approach as an alternative to the Bayesian PIP.

Model	Min	1st Qu	Median	Mean	3rd Qu	Max
Frequentist	11.31	15.84	17.52	17.68	19.27	25.31
BMS PIP	11.25	14.71	16.14	16.18	17.51	23.34

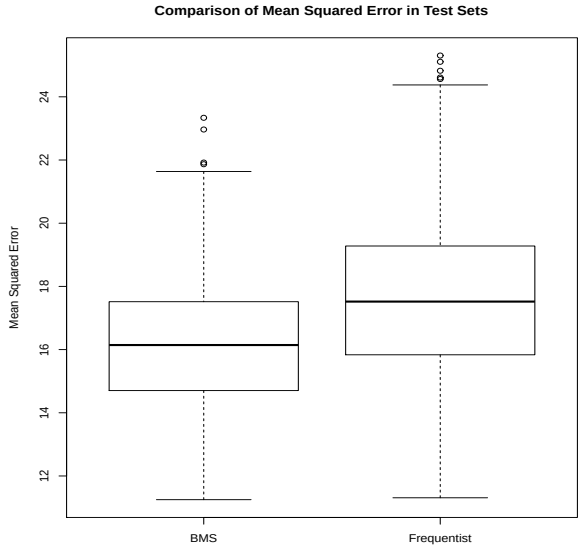

Figure 17: *Figure: Comparison of MSE*

The results from using siri as the response variable are almost identical to the results attained above using brozek as the response variable. As a result, the comparison between the frequentist and PIP approaches for siri response variable will not be provided.
Applying the frequentist approach on linear models was significantly faster than on generalized linear models. Running stepwise regression via BIC and AIC, forward selection via BIC, and the full model on the 500 replicates for both brozek and siri response variables only took minutes.

V. Conclusion and discussion

We have used a straightforward, quite general and easily interpretable subsampling scheme to provide a frequentist approximation of the celebrated Bayesian posterior inclusion probability. Despite the relatively higher computational burden arising in the use of the proposed method on high dimensional classification tasks, it is fair to say that the present method mimics the Bayesian framework quite well. All the scores, judging from the numerical values and the corresponding plots appear identical or at least very similar in shape and form. One would be particularly be excited to use this approach because it is easier to understand since it builds up on the widely used framework of variable selection by the stepwise regression heuristic. Even more importantly, it does not run into the some of the challenges of the Bayesian framework like the difficulty in computing the marginal density of the data. As we said earlier, the great challenge for this method is the heavy computational burden. However, with the availability of distributed and high performance parallel computing resources, this method becomes even more attractive for high dimensional data mining problems since once can perform the independent random split

on different CPUs. Indeed, our future work will focus on substantially reducing the computing time by a careful use of the parallel computing resources.

References

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

Forman, G., M. Hopkins, and E. Reeber (1999). UCI machine learning repository.

Fumera, G., I. Pillai, and F. Roli (2006). Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research 7*, 2699–2720.

Goosen, J. C. and J. V. Du Toit (2009, August). Spam detection with generalised additive neural networks. In *Southern Africa Telecommunication Networks and Applications Conference*.

Kiran, R. and I. Atmosukarto (2005). Spam or not spam âĂŞ that is the question. Technical report, University of Washington.

Penrose, K., A. Nelson, and A. Fisher (1985). Generalized body composition prediction equation for men using simple measurement techniques. *Medicine and Science in Sports and Exercise 17*(2), 189.

Sharma, S. and A. Arora (2013, July). Adaptive approach for spam detection. *International Journal of Computer Science Issues 10*(1), 23–26.

Sivanadyan, T. (2003). Spam? not any more! detecting spam emails using neural networks. Technical report, University of Wisconsin.