Bayesian Methods of Variable Selection

Dav King

Introduction

Note: all notes and code used in this project can be found at github.com/davmking/bayesian-variable-selection.

Consider the traditional regression setting:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{1}$$

In this setting, β is the $p \times 1$ vector of predictors, $\mathbf{X}$ is the $n \times p$ feature matrix, $\mathbf{Y}$ is the $n \times 1$ vector of responses, and ϵ is the $n \times 1$ vector of random, irreducible, zero-mean noise. $p(\mathbf{Y}|\beta)$ is known as the **data-generative model**, i.e., the mechanism by which we believe the response variable $\mathbf{Y}$ is actually generated. Often, we have a great number of potential predictors, but believe that only a small handful of them are truly a part of the data-generative model. This is a statistical concept known as **sparsity**, where the predictors in the true data-generative model are **signals** and the remaining predictors are **noise**. Our goal is to separate the signals from the noise in order to truly understand the underlying data generative model.

Our goal, therefore, is to control the number of predictors by performing **variable selection**. By identifying only the subset of signals, we can reduce computational complexity, minimize variance in the model, and improve interpretability. This is especially useful in situations where we care more about interpretation than prediction: An understanding of the actual data-generative model allows us to understand how response variables come about, which can have highly meaningful applications in real-world settings. As computational power and data availability have simultaneously increased, variable selection processes have developed major applications in fields such as genomics and the social sciences.

Separating signals from noise is not a small feat, however. Though it is tempting to try all possible combinations of variables while solving this problem (known as best subset selection), this method is computationally intractable except in situations where p is very small. For a model with p parameters, there are 2^p possible model formulations, which compose the model space $\mathcal{M}$. Searching over all of $\mathcal{M}$ quickly becomes impossible, at least with modern

computing technology. Another possible solution is greedy algorithms such as forward- and back-selection, but these processes can miss important effects that arise when multiple related predictors are included in the model together. Other methods that penalize predictors, such as LASSO regression, have become standard across industries, but they are often not aggressive enough to select a sufficiently small subset of variables or lead to bias in the model's coefficient predictions.

One very active domain of research around these problems draws from the Bayesian perspective on statistics. At the core of Bayesian statistics is Bayes' theorem, which describes how beliefs about a random variable should be updated after encountering new information:

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A|B)\mathbb{P}(B)}{\mathbb{P}(A)} \tag{2}$$

As will be discussed below, Bayesian statistics allows for the statistician to impose some structure on the model based on their prior beliefs about its format, which can help stabilize calculations and ensure an output of the desired form. It also can search over high-probability regions of the model space much more effectively than frequentist approaches. Many papers have compared different aspects of Bayesian model averaging to one another (e.g., (Lu & Lou, 2022), (Rockova, 2013)), but few if any have compared different domains of Bayesian variable selection to one another. Thus, this paper describes the foundation of Bayesian variable selection methods, and performs a brief experiment to compare approaches across domains.

Research Questions

The central research questions for this paper were as follows:

1. What methods of Bayesian variable selection exist, and how do they relate to one another?
2. Which methods perform best in each of the following settings?

$$n >> p$$
$$n > p$$
$$p > n$$
$$p >> n$$

3. What are the computation time/accuracy tradeoffs?

Bayesian Modeling

Introduction

$$\mathbb{P}(\beta|\mathbf{Y}) = \frac{\mathbb{P}(\mathbf{Y}|\beta)\mathbb{P}(\beta)}{\int_\beta \mathbb{P}(\mathbf{Y}|\beta)\mathbb{P}(\beta)d\beta} \tag{3}$$

In a Bayesian modeling setting, we use Bayes' theorem to generate estimates of the model parameters. In this equation, $\mathbb{P}(\mathbf{Y}|\beta)$ is the data generative model, or the process by which we believe the response variable is generated. This is generally represented with a likelihood function. $\mathbb{P}(\beta)$ is the prior distribution, representing our beliefs about the structure of β before viewing any of the data. This is a useful concept in Bayesian variable selection, as we will see later. $\int_\beta \mathbb{P}(\mathbf{Y}|\beta)\mathbb{P}(\beta)d\beta$ is the normalizing constant which ensures the posterior distribution integrates to 1, and it can usually be absorbed into proportionality and thus omitted from these calculations. Finally, $\mathbb{P}(\beta|\mathbf{Y})$ is the posterior distribution, reflecting our beliefs about β after seeing the data - in other words, reflecting an update in our beliefs about β from our prior beliefs, based on the data likelihood function.

Markov Chain Monte Carlo

Let β and $\mathbf{Y}$ be as defined above, and let θ represent all other parameters in our modeling setup. In this setting, we would like to calculate the joint posterior distribution $p(\beta, \mathbf{Y}, \theta)$. In many cases, posterior distributions are difficult or impossible to compute by hand. In this case, we estimate the joint posterior by sampling repeatedly from the full conditional distributions of all involved parameters (full conditional means the distribution of one parameter given everything else). This is known as a **Markov Chain Monte Carlo (MCMC)** algorithm. It is conducted as follows: Initialize your values $\beta^{(1)}$, $\mathbf{Y}^{(1)}$, and $\theta^{(1)}$. Then, for every step s,

1. Sample $\beta^{(s+1)}$ from the full conditional $p(\beta|\mathbf{Y}^{(s)}, \theta^{(s)})$.
2. Sample $\mathbf{Y}^{(s+1)}$ from the full conditional $p(\mathbf{Y}|\beta^{(s+1)}, \theta^{(s)})$.
3. Sample $\theta^{(s+1)}$ from the full conditional $p(\theta|\beta^{(s+1)}, \mathbf{Y}^{(s+1)})$.

After repeating this algorithm for long enough, it will eventually converge to the joint posterior (reasonable initialization is key, or it may not reach the true joint posterior before the heat death of the universe). Usually, iterations on the order of tens of thousands are sufficient for the sampler to converge and give a reasonable estimate of the joint posterior. There are a couple things of importance here. First, the order in which these variables are sampled does not matter, and they can even be re-shuffled randomly at different steps. Second, the Markov Chain is memoryless - each sample only depends on the step immediately before it.

There are many different ways that values can be sampled in this process, but two of the most famous algorithms are Gibbs and Metropolis-Hastings. In a Gibbs sampler, the full conditional

of each variable is known and easy to compute, and so new values are simply drawn from that full conditional and accepted with probability 1. In a Metropolis-Hastings algorithm, the full conditional of each variable is harder to draw from, so the following steps are used:

1. Propose a new value of the parameter of interest from a proposal distribution, usually centered at around the current parameter value.
2. Calculate the likelihood ratio $\alpha = \ell(\text{proposal})/\ell(\text{current})$.
3. Accept the likelihood ratio with probability $\min\{1, \alpha\}$.

With a Metropolis-Hastings sampler, if the proposed value is more likely than the current value, it is always accepted; otherwise, it is accepted with probability corresponding to the likelihood ratio (which ensures that the sampler continues to explore parameter space). Note that it is possible to use both Gibbs and Metropolis-Hastings sampling for different parameters within the same MCMC sampler.

Non-Prior Methods

The first domain of Bayesian variable selection is general methods revolving around the problem setup, without modifying the priors placed on the βs. These still take place within the framework of MCMC sampling, but introduce additional structure or variables in order to reach the desired outcomes.

Bayesian Model Selection

Define a variable $\gamma = \gamma_1, \ldots, \gamma_p$. This characterizes a sub-model $\mathcal{M}_\gamma$,

$$\mathbf{Y} = \beta_0 \mathbf{1} + \mathbf{X}_\gamma \beta_\gamma + \epsilon, \tag{4}$$

where $\gamma_j = 1$ indicates that β_j is included in $\mathcal{M}_\gamma$, and $\gamma_j = 0$ indicates that β_j is not (Lu & Lou, 2022). The different specifications of γ characterize the entire model space $\mathcal{M}$. We can place a variety of priors on γ, but Bernoulli priors are among the most common.

In our MCMC sampler, we sample γ along with β and $\mathbf{Y}$. Typically, we sample γ with a procedure like Metropolis-Hastings, so that we can accept or reject the new choice of γ based on the model likelihood ratios (however, even Gibbs sampling will converge). There are a number of different criteria that can be used for comparing models, such as the Bayesian information criterion (BIC) or Chib's marginal likelihood estimator (Lu & Lou, 2022).

Regardless of how new values of γ are sampled, the sampler is able to identify probable models intuitively: More probable models will appear in the posterior distribution of γ more frequently. In fact, this procedure often converges to one (or a few) very strong candidate models rapidly,

even when only exploring a small fraction of model space. This makes Bayesian model selection a very useful procedure, since it is able to dramatically cut down on the amount of computation required to identify highly probable models (with some risk of failing to reach the areas of true high posterior density, if the procedure is done poorly). This is also a very simple procedure, and it requires relatively little computation work, making it one of the fastest algorithms considered here. However, it does have some downsides: The selection of only one model specification may place too much emphasis on random noise in the data sample, while other highly probable models are dropped from consideration. Still, this procedure is very effective and widely used.

Bayesian Model Averaging

Bayesian model averaging follows the exact same procedure as BMS: Define γ as a latent binary indicator variable, sample this along with β and $\mathbf{Y}$, and find the highly probable models. However, there is one key difference. In BMS, we look to identify the model with the highest posterior probability, and determine that to be our selected model. In BMA, we simply average over the coefficients of every sampled model, letting $\beta_j = 0$ if $\gamma_j = 0$. This results, more-or-less, in posterior estimates for β that are weighted by the probability of each model specification γ. Though unintuitive, this procedure often results in highly accurate estimates of the true model specification γ (Hoeting et al., 1999).

BMA, like BMS, has relatively little computational demand, and is widely applicable in a variety of situations. BMA also has the additional benefit of weighting β coefficients by model probabilities, which allows the model specification to be influenced by a variety of highly probable model specifications instead of just one. However, there may be situations where the averaging does not produce the desired result, depending on the circumstances of the data. It may also have difficulty converging to the correct values if initialized poorly. This procedure is useful for its low computational demand and incorporation of information from the full (sampled) model space.

Bayesian Subset Selection

Bayesian subset selection is very similar to the frequentist problem of best subset selection, with a few key differences. First, instead of focusing on the singular best subset, BSS identifies a family of "acceptable" subsets, or those within some tolerable ϵ of the best subset. This allows for the identification of near-optimal subsets, which may represent the true data generative model but might not be recognized under best subset selection because of patterns in the random noise. Second, it draws response estimates from the posterior predictive distribution of a Bayesian model, rather than simply the estimates from an OLS regression model over the subset. This allows for a better quantification of uncertainty in the variable selection process, and establishes the optimal coefficients for any subset of predictors. Third, it uses a more efficient branch-and-bound algorithm to explore only the promising subsets, and only

considers subsets up to a maximum size. Fourth, it summarizes across the acceptable subsets to generate the "best" subset, including the "best" (by cross-validation) predictive subset, the smallest acceptable subset, and metrics on variable co-importance, all of which the frequentist best subset selection lacks (D. R. Kowal, 2022).

However, as promising as BSS is with its many strengths over frequentist methods and its ability to compare subsets of models in the true spirit of variable selection, it still has its drawbacks. First, even with the computational benefits of the branch-and-bound algorithm, it is still computationally intractible above a certain number of predictors, which restricts the algorithm to subsets of an arbitrary maximum size (and may eliminate the true best subsets from consideration). Since there is no efficient implementation of this in R, it is excluded from this analysis. Additionally, this means it is not much more computationally tractable than frequentist best subset selection, where the branch-and-bound algorithm can also be applied. This puts BSS far behind BMS and BMA, which do not need to search over the full model space to identify promising models. Still, the Bayesian BSS method is very promising (and still very new), and offers a lot of advantages. Research in the coming years will likely make this one of the clearly preferred methods of Bayesian variable selection, particularly as searching over candidate subsets becomes more efficient.

Prior Methods

Suppose we have some prior beliefs about the structure of our βs. Specifically, we believe that not all of them are included in the true data-generative model, and many of their coefficients must be zero. We can place a prior distribution on the βs accordingly, reflecting this belief. This produces a posterior distribution of our desired form, where many of the βs are set to (near) zero. We can also introduce additional structure with this method, such as placing priors on different groups of βs, but that goes beyond the scope of this paper.

There are two major classes of priors used for variable selection: spike-and-slab priors and shrinkage priors.

Spike-and-Slab Priors

A spike-and-slab prior is a mixture distribution: A combination of two distributions, where β_j is drawn from one or the other according to some Bernoulli probability. Thus, a latent indicator variable γ is still used, but instead of explicitly denoting whether a specific β_j is included in the model, it denotes whether the prior distribution for β_j is drawn from the spike or the slab distribution.

Spike-and-slab priors have the general form

$$\beta_j|\gamma_j \sim (1-\gamma_j)\phi_0(\beta_j) + \gamma_j\phi_1(\beta_j), \tag{5}$$

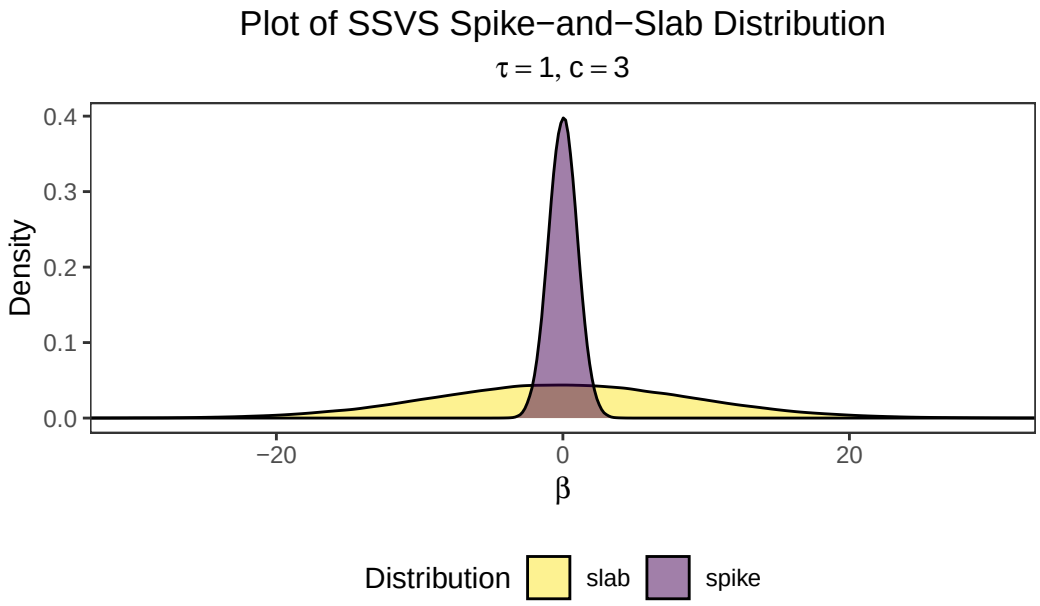
where $\phi_0(\beta_j)$ is a concentrated "spike" distribution that pulls some of the βs to (near) zero, while $\phi_1(\beta_j)$ is a diffuse "slab" distribution that allows the remaining βs to attain their true coefficients (Lu & Lou, 2022). By continuing to sample γ in the MCMC sampler, this specification returns a joint posterior where (in general) the signal coefficients are near their true values while the noise coefficients are near zero, with posterior inclusion probabilities (γ) that reflect the most probable model specifications. This allows for a little more uncertainty in the estimation of coefficients than BMS does, since near-zero coefficients do not have to be pulled to exactly zero. Overall, spike-and-slab approaches are well-studied, effective, and widely used. However, there is still the disadvantage of needing to include γ to sample over the model space $\mathcal{M}$, which can result in either extreme computational complexity or the possibility that the sampler never reaches the areas of true high posterior density.

Stochastic Search Variable Selection

The best-known implementation of the spike-and-slab prior is stochastic search variable selection (George & McCulloch, 1993). This follows the procedure outlined above, embedding the full model selection process in a hierarchical Bayesian MCMC scheme. In SSVS, the prior is represented as a mixture of normals:

$$\beta_j|\gamma_j \sim (1-\gamma_j)N(0,\tau_j^2) + \gamma_j N(0, c_j^2\tau_j^2) \tag{6}$$

Prior distributions can be placed on τ and c, or they can be held constant.



Note: spike is usually more concentrated and slab is usually more diffuse than depicted

Normal Mixture of Inverse Gamma

The normal mixture of inverse gamma expands on SSVS by moving the spike-and-slab formulation down one level in the hierarchy, to place it on the variance instead Rockova et al. (2012). The hierarchal specification of this model is

$$\beta_j|\eta_j \sim N(0, \eta_j) \tag{7}$$

$$\eta_j|\gamma_j \sim (1-\gamma_j)\text{IG}\left(a, \frac{\nu_0}{b}\right) + \gamma_j\text{IG}\left(a, \frac{\nu_1}{b}\right), \tag{8}$$

where IG is the inverse-gamma distribution, and $\nu_1 >> \nu_0$. A prior, typically Bernoulli, is placed on γ. By moving the sampling down one level in the hierarchy, we avoid having to put as much weight on the hyperparameters c and τ, often leading to better performance (Lu & Lou, 2022). However, due to a lack of an efficient implementation in R, we omit NMIG from the experiment in this paper.

Shrinkage Priors

Unlike spike-and-slab priors, which are mixture distributions, shrinkage priors are single-point, continuous distributions. Their goal is to pull some of the βs towards zero, while minimally penalizing the signal coefficients. Shrinkage priors have the benefit of being fully continuous, which means that they do not require a latent variable γ to search over $\mathcal{M}$, and this makes shrinkage priors less likely to leave an entire set of highly probable model specifications unexplored. However, this has its tradeoffs: approaches using shrinkage priors often have dramatically increased computation time, and research has suggested that they may not select variables aggressively enough or may overly penalize the signal coefficients in the process (i.e., they may be less effective at discerning between signal and noise predictors).

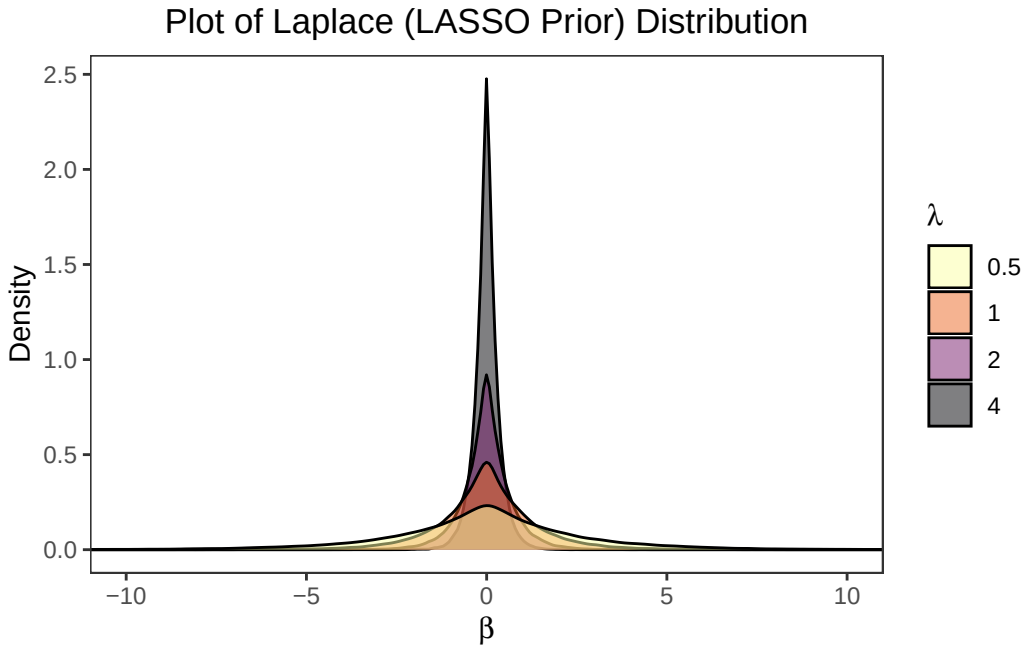
LASSO

In frequentist statistics, LASSO regression is a well-know regression setting that expands on OLS regression to penalize the ℓ_1 norm of the βs, using a hyperparameter λ to control the shrinkage:

$$\min_{\beta}(\mathbf{Y}-\mathbf{X}\beta)^T(\mathbf{Y}-\mathbf{X}\beta) + \lambda\sum_{j=1}^{p}|\beta_j| \tag{9}$$

In the Bayesian analogue to LASSO regression (Park & Casella, 2008), we place a conditional Laplace prior on β:

$$p(\beta|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} \exp\left\{\frac{-\lambda|\beta_j|}{\sqrt{\sigma^2}}\right\} \tag{10}$$



This results in the hierarchical specification

$$\beta_j|\tau_j \sim N(0, \sigma^2\tau_j^2) \tag{11}$$
$$\tau_j|\lambda \sim \exp(\lambda^2/2) \tag{12}$$

Under this distribution, there is a high probability of the βs falling at or near zero, which increases with λ as shown above. In this setting, we can also place a prior on λ and sample it along with β. The result is a posterior where many of the βs are at or near 0, and careful prior specification can ensure that the posterior distribution is unimodal (Park & Casella, 2008). This can also be generalized to other cases, such as bridge regression.

Normal-Gamma

While the Bayesian LASSO was one of the earliest methods of Bayesian variable selection and remains one of the best-used, it is not without its criticisms. The use of only a single hyperparameter reduces flexibility, meaning that either the LASSO will fail to perform adequate variable selection or the signal coefficients will be unnecessarily reduced. Thus, the

normal-gamma prior arises (Griffin & Brown, 2010). The specification for the normal-gamma is

$$\beta_j|\tau_j \sim N(0, \tau_j^2) \tag{13}$$
$$\tau_j^2|\lambda, \xi \sim \text{gamma}(\lambda, 1/(2\xi^2)) \tag{14}$$

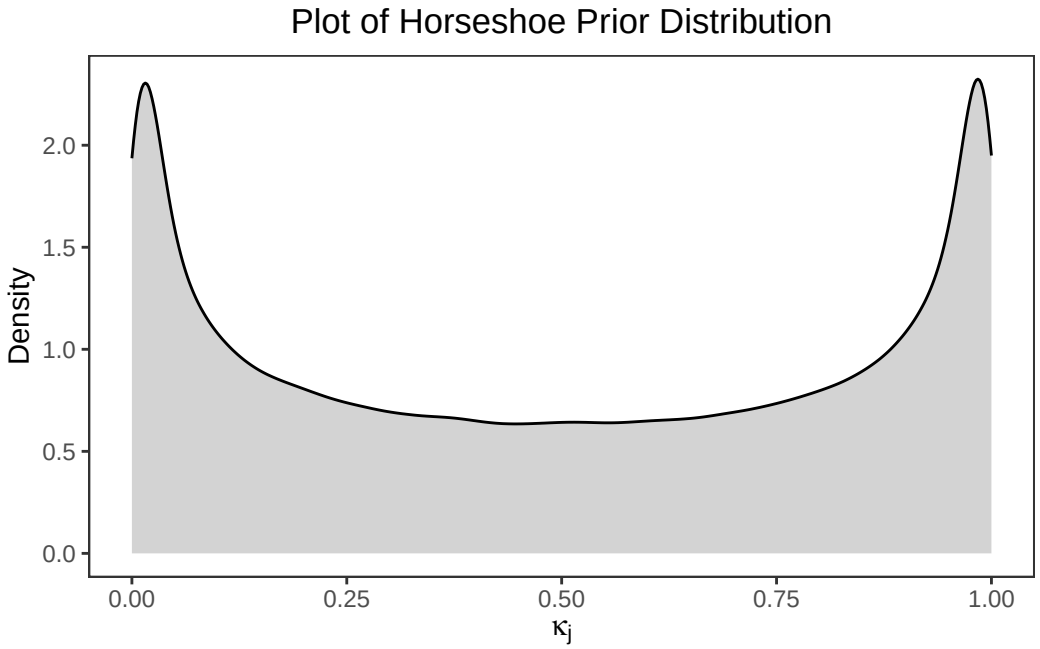
The introduction of the extra hyperparameter ξ and the gamma prior on the variance τ_j^2 allows this distribution to have a lot of mass close to zero, while maintaining heavy tails for the coefficients. This is generally an improvement on LASSO, though it does require the introduction of an extra parameter (potentially increasing computational complexity) and does not always outperform the LASSO in applied settings.

Horseshoe

Taking a slightly different approach, horseshoe priors fall into a class known as **global-local** shrinkage priors (Carvalho et al., 2010). These priors use a global hyperparameter to shrink all coefficients towards zero, while maintaining a local hyperparameter to adjust the scale of shrinkage for some coefficients at the local level. The hierarchical representation of the regression model under a horseshoe prior is

$$\beta_j|\eta_j \sim N(0, \eta_j^2) \tag{15}$$
$$\eta_j|\tau \sim C^+(0, \tau) \tag{16}$$
$$\tau|\sigma \sim C^+(0, \sigma), \tag{17}$$

where C^+ is a half-Cauchy distribution (restricted to positive values). In this specification, η_j is the local shrinkage parameter, while τ is the global shrinkage parameter. With some integration, this results in a shrinkage coefficient $\kappa_j = 1/(1+\eta_j^2)$. The half-Cauchy prior on η_j implies a Beta(1/2, 1/2) marginal distribution for κ_j, which gives the horseshoe its name.



The distribution for κ_j places a lot of mass near 0 (implying near-zero shrinkage, representing signals) and 1 (implying near-total shrinkage, representing noise). This specification allows for most of the coefficients to be pulled to zero, while allowing the remaining coefficients to reach their true values. It is a different specification than LASSO/NG, but one that is widely used and often performs better in practice.

Other Approaches

There are several other shrinkage priors that have been proposed. The horseshoe+ prior places another latent variable in the hierarchy for another level of local shrinkage (Bhadra et al., 2017). The Dirichlet-Laplace prior, another global-local shrinkage prior, draws local shrinkage coefficients from the joint Dirichlet distribution (Bhattacharya et al., 2014). The SSLASSO prior hybridizes spike-and-slab and shrinkage priors by drawing from a mixture of Laplace distributions (Rockova & George, 2018). While all of these approaches are promising and have their own strengths, they are excluded from further analysis here for the sake of time.

Expectation-Maximization Variable Selection

Overview

Though all other methods described here use the MCMC algorithm approach to Bayesian variable selection, EMVS is an algorithm that is formulated differently. EMVS is highly useful in high dimensional settings ($p >> n$), since it runs in a tiny fraction of the time that MCMC requires and can effectively identify the high-probability sparse models (Rockova, 2013). EMVS is anchored by the SSVS spike-and-slab approach (George & McCulloch, 1993), but it can be extended into other settings as well.

Algorithm

The EMVS algorithm is very straightforward. Begin by specifying the full Bayesian regression setting, typically using a spike-and-slab prior such as SSVS. Then, repeat two steps until convergence:

- **Expectation (E) Step:** Calculate the posterior inclusion probabilities (i.e., expectations) for each β_j given current parameter updates.
- **Maximization (M) Step:** Update the parameter estimates by maximizing the objective function of all model parameters, given the data.

More formal specifications of the equations required to execute this algorithm can be found in (Rockova, 2013). In practice, this algorithm generally converges very quickly - in the experiment below, all specifications of the algorithm converged within 15 iterations, which is stellar performance.

Deterministic Annealing

If the true posterior is multi-modal, it is very possible that poor initialization of the EMVS algorithm will lead to incorrect estimates of the posterior, as EMVS gets trapped in local modes. One potential solution to this is deterministic annealing. Instead of maximizing the objective function, we minimize a tempered version known as the negative free energy function. We can raise the "temperature", which smooths away the local modes of the negative free energy function, allowing only the true signals to shine through and giving us some robustness against poor initialization. As we progress through the algorithm, we lower the temperature progressively until its effects are completely removed, allowing us to view an equation that more truly resembles the posterior (Rockova, 2013). Some form of cross-validation is required to find the optimal values of initial temperature, but since the EMVS algorithm converges so quickly, this is still much faster than MCMC sampling in most cases.

Experiment

Methodology

In order to compare these methods against each other, I performed a brief experiment. This experiment was performed using synthetic data in order to be able to control and evaluate the experiment more effectively, though this does limit the generalizability of this experiment to real-world data which may not match this setting. Data were generated using the `simulate_lm()` function from the R package `BayesSubsets` (D. Kowal, 2024), with some modifications to have a little more diversity in coefficients. All data were generated with $n = 100$, for the sake of runtime when performing this experiment (greater amounts of data would likely change performance somewhat, but since the signal-to-noise ratio was fixed in these models, it likely would not have proven to be a huge change). Details of the model specifications, including the total number of predictors, the total number of significant predictors, and the true coefficients, can be found in the table below.

Model	n	p	p_sig	Coef
Model 1	100	10	3	3, 2, -2
Model 2	100	95	8	3, 2, -2, 1, -1, -1, -1, -1
Model 3	100	105	8	3, 2, -2, 1, -1, -1, -1, -1
Model 4	100	1000	8	3, 2, -2, 1, -1, -1, -1, -1

In this experiment, I focused only on the sparse signal setting. I did also consider the dense signal setting, but all of these algorithms performed poorly in that setting, so they are omitted from further exploration. Partial findings from the dense signal setting can be found in the appendix.

All analysis for this experiment was performed in R. I ran the modeling process for all four models with each of the variable selection methods. Runtimes are only reported in terms of orders of magnitude, as getting accurate timing on these tests would require numerous trials. Where applicable, I used ROC curves to determine the optimal thresholds (in posterior inclusion probability) for each model, based on true β values. While this is a luxury afforded to us in the synthetic setting that is not present in the real world, most of these models have forms of thresholding based on intersection points in the posterior distribution.

I evaluated models on four criteria: MSE in terms of estimated coefficients for the true (nonzero) coefficients, MSE in terms of estimated coefficients for all (including zero) coefficients, MSE in terms of the response variable on the training set, and MSE in terms of the response variable on a testing set (data generated with the same specification and a different random seed). I also explored true/false positive/negative rates, for researchers who are interested in those domains. All code to produce these results can be found in `experiment.qmd`, although some manipulation may be required to get the code to run sequentially.

Results

Discussion

Appendix A: Model Results

Model 1: $n >> p$

$n = 100$, $p = 10$, $p\_sig = 3$

Name	Size	TPR	FNR	FPR	TNR	Runtime
BMS	3	1	0	0	1	S
BMA	3	1	0	0	1	S
SSVS	3	1	0	0	1	S
LASSO	3	1	0	0	1	S
Normal-Gamma	3	1	0	0	1	S
Horseshoe	3	1	0	0	1	S
EMVS	3	1	0	0	1	Ms

Name	Size	MSE Coef Signal	MSE Coef All	MSE Train Y	MSE Test Y
BMS	3	NA	NA	NA	NA
BMA	3	0.00	0	0.92	0.80
SSVS	3	0.00	0	0.93	0.81
LASSO	3	0.00	0	0.92	0.82
Normal-Gamma	3	0.00	0	0.92	0.82
Horseshoe	3	0.00	0	0.92	0.81
EMVS	3	0.01	0	0.93	0.84

Model 2: $n > p$

$n = 100$, $p = 95$, $p\_sig = 8$

Name	Size	TPR	FNR	FPR	TNR	Runtime
BMS	4	0.38	0.62	0.01	0.99	S
BMA	73	0.88	0.12	0.76	0.24	S
SSVS, p = 0.021	5	0.50	0.50	0.01	0.99	S
LASSO, p = 0.767	4	0.50	0.50	0.00	1.00	M
Normal-Gamma, p = 0.768	4	0.50	0.50	0.00	1.00	M
Horseshoe, p = 0.623	8	0.75	0.25	0.02	0.98	M
EMVS, p = 0.994	4	0.50	0.50	0.00	1.00	Ms

Name	Size	MSE Coef Signal	MSE Coef All	MSE Train Y	MSE Test Y
BMS	4	NA	NA	NA	NA
BMA	73	1.33	1.50	0.18	130.24
SSVS, p = 0.021	5	0.76	0.07	7.21	10.03
LASSO, p = 0.767	4	0.63	0.07	4.12	9.68
Normal-Gamma, p = 0.768	4	0.62	0.06	4.19	9.53
Horseshoe, p = 0.623	8	0.52	0.05	4.90	8.46
EMVS, p = 0.994	4	0.34	0.21	2.23	16.44

Model 3: $p > n$

$n = 100$, $p = 105$, $p\_sig = 8$

Name	Size	TPR	FNR	FPR	TNR	Runtime
BMS	5	0.50	0.50	0.01	0.99	S
BMA	67	0.88	0.12	0.62	0.38	S
SSVS, p = 0.009	5	0.50	0.50	0.01	0.99	S
LASSO, p = 0.909	4	0.50	0.50	0.00	1.00	M
Normal-Gamma, p = 0.901	4	0.50	0.50	0.00	1.00	M
Horseshoe, p = 0.905	4	0.50	0.50	0.00	1.00	M
EMVS, p = 1	3	0.38	0.62	0.00	1.00	Ms

Name	Size	MSE Coef Signal	MSE Coef All	MSE Train Y	MSE Test Y
BMS	5	NA	NA	NA	NA
BMA	67	1.37	1.02	0.53	87.81
SSVS, p = 0.009	5	0.63	0.05	8.95	9.93
LASSO, p = 0.909	4	0.42	0.05	3.81	9.40
Normal-Gamma, p = 0.901	4	0.42	0.05	3.88	9.31
Horseshoe, p = 0.905	4	0.43	0.04	4.33	9.31
EMVS, p = 1	3	0.30	0.22	1.95	31.63

Model 4: $p >> n$

$n = 100$, $p = 1000$, $p\_sig = 8$

Name	Size	TPR	FNR	FPR	TNR	Runtime
BMS	5	0.38	0.62	0.00	1.00	M
BMA	26	0.25	0.75	0.02	0.98	S
SSVS, p = 0.214	3	0.38	0.62	0.00	1.00	M
LASSO, p = 0.418	5	0.50	0.50	0.00	1.00	H
Normal-Gamma, p = 0.421	5	0.50	0.50	0.00	1.00	H
Horseshoe, p = 0.426	5	0.50	0.50	0.00	1.00	H
EMVS, p = 0	3	0.38	0.62	0.00	1.00	Ms

Name	Size	MSE Coef Signal	MSE Coef All	MSE Train Y	MSE Test Y
BMS	5	NA	NA	NA	NA
BMA	26	1.33	0.05	0.16	64.29
SSVS, p = 0.214	3	1.29	0.01	9.98	17.38
LASSO, p = 0.418	5	0.58	0.01	0.95	13.16
Normal-Gamma, p = 0.421	5	0.58	0.01	1.12	13.10
Horseshoe, p = 0.426	5	0.55	0.01	2.31	13.71
EMVS, p = 0	3	0.53	0.01	4.56	14.29

References

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, *88*(422), 669–679. http://www.jstor.org/stable/2290350

Bhadra, A., Datta, J., Polson, N. G., & Willard, B. (2017). The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis*, *12*(4), 1105–1131. https://doi.org/10.1214/16-BA1028

Bhattacharya, A., Pati, D., Pillai, N. S., & Dunson, D. B. (2014). Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, *110*(512), 1479–1490. https://doi.org/10.1080/01621459.2014.960967

Carvalho, C. M., Polson, N. G., & Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, *97*(2), 465–480. https://doi.org/10.1093/biomet/asq017

George, E. I., & McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, *88*(423), 881–889. https://doi.org/10.1080/01621459.1993.10476353

Griffin, J. E., & Brown, P. J. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, *5*(1), 171–188. https://doi.org/10.1214/10-BA507

Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, *14*(4), 382–417. https://doi.org/10.1214/ss/1009212519

Kowal, D. (2024). *BayesSubsets: Bayesian subset selection and variable importance*. https://github.com/drkowal/BayesSubsets

Kowal, D. R. (2022). Bayesian subset selection and variable importance for interpretable prediction and classification. *Journal of Machine Learning Research*, *23*, 1–38. https://doi.org/10.48550/arXiv.2104.10150

Lu, Z., & Lou, W. (2022). Bayesian approaches to variable selection: A comparative study from practical perspectives. *The International Journal of Biostatistics*, *18*(1), 83–108. https://doi.org/10.1515/ijb-2020-0130

Park, T., & Casella, G. (2008). The Bayesian lasso. *Journal of the America Statistical Association*, *103*(482), 681–686. https://doi.org/10.1198/016214508000000337

Rockova, V. (2013). *Bayesian variable selection in high-dimensional applications*.

Rockova, V., & George, E. I. (2018). The spike-and-slab LASSO. *Journal of the American Statistical Association*, *113*(521), 431–444. https://doi.org/10.1080/01621459.2016.1260469

Rockova, V., Lesaffre, E., Luime, J., & Löwenberg, B. (2012). Hierarchical Bayesian formulations for selecting variables in regression models. *Statistics in Medicine*, *31*, 1221–1237. https://doi.org/10.1002/sim.4439