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?

$$\begin{aligned} n &>> p \\ n &> p \\ p &> n \\ p &>> n \end{aligned}$$

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

Bayesian Model Selection

Bayesian Model Averaging

Prior Methods

Spike-and-Slab Priors

Stochastic Search Variable Selection

Normal Mixture of Inverse Gamma

Shrinkage Priors

LASSO

Normal-Gamma

Horseshoe

Expectation-Maximization Variable Selection

Overview

Algorithm

Deterministic Annealing

Experiment

Methodology

Results

Discussion

References

Appendix

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

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