A Fast New Bayesian Approach to High-Dimensional Nonparametric Regression Without MCMC

Ray Bai

Department of Biostatistics, Epidemiology, and Informatics, University of Pennsylvania

Joint work with Gemma E. Moran (co-first author), Joseph Antonelli (co-first author), Yong Chen, and Mary R. Boland

April 2, 2019

Overview

Classical Linear Regression

Consider the classical linear regression model,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_n)$$

where:

- $\boldsymbol{y}$ is an n-dimensional response vector,
- $\boldsymbol{X}_{n\times(p+1)} = [\mathbf{1}, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_p]$ is a design matrix with n samples and p covariates (and a column $\mathbf{1} = (1, \ldots, 1)^T$ for the intercept).
- We are mainly interested in estimating $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ and performing model selection from the p covariates.

Classical Linear Regression

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_n)$$

PROS:

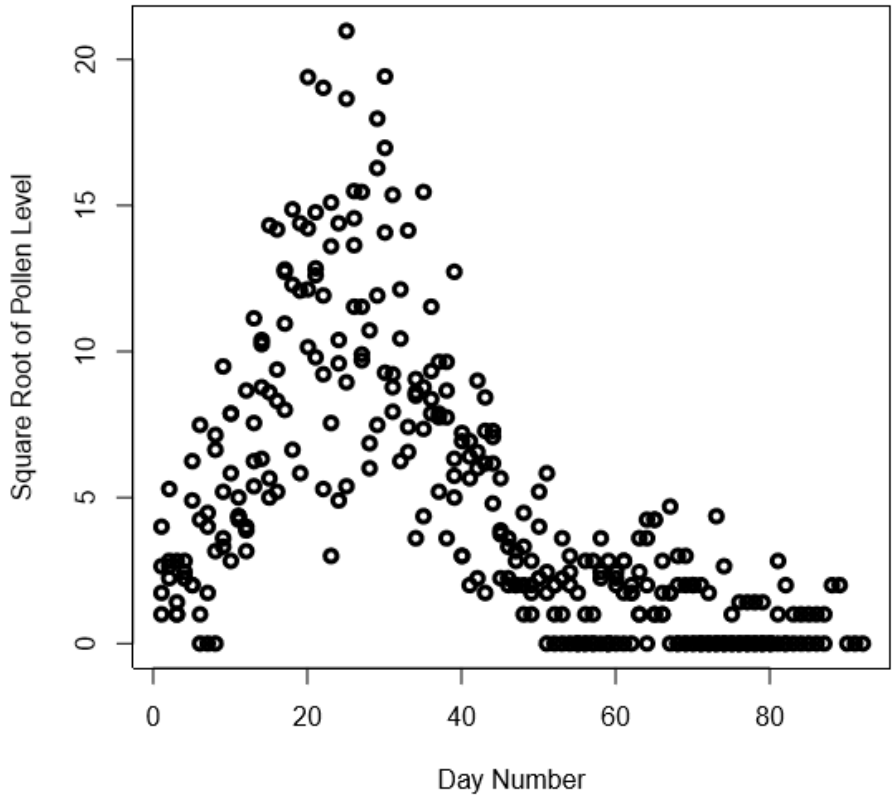
- Widely used.
- Relatively easy to interpret.

CONS:

- The assumption that the covariates have a linear relationship with the response is very restrictive.
- We typically need to check model diagnostics like residual plots to ensure that the linear model is a good fit. What if it's *not*?

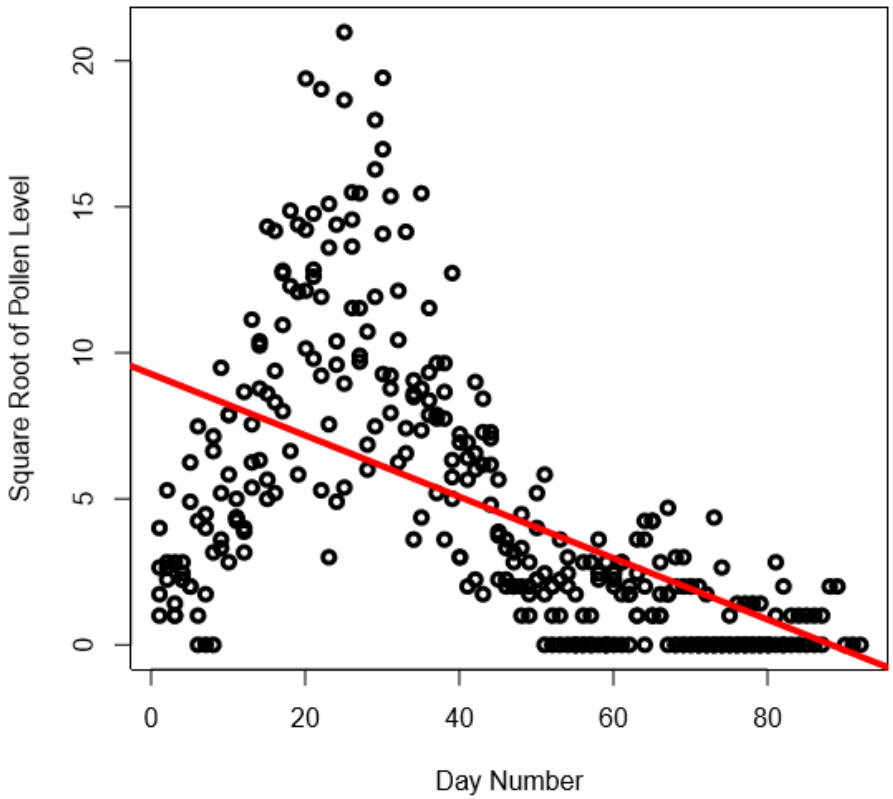
Example Where the Linear Model Fails

The below plot shows ragweed pollen levels plotted against the day in the current ragweed season. There seems to be a relationship between pollen levels and day in the ragweed season, with a peak around day 25 and then a plateau.



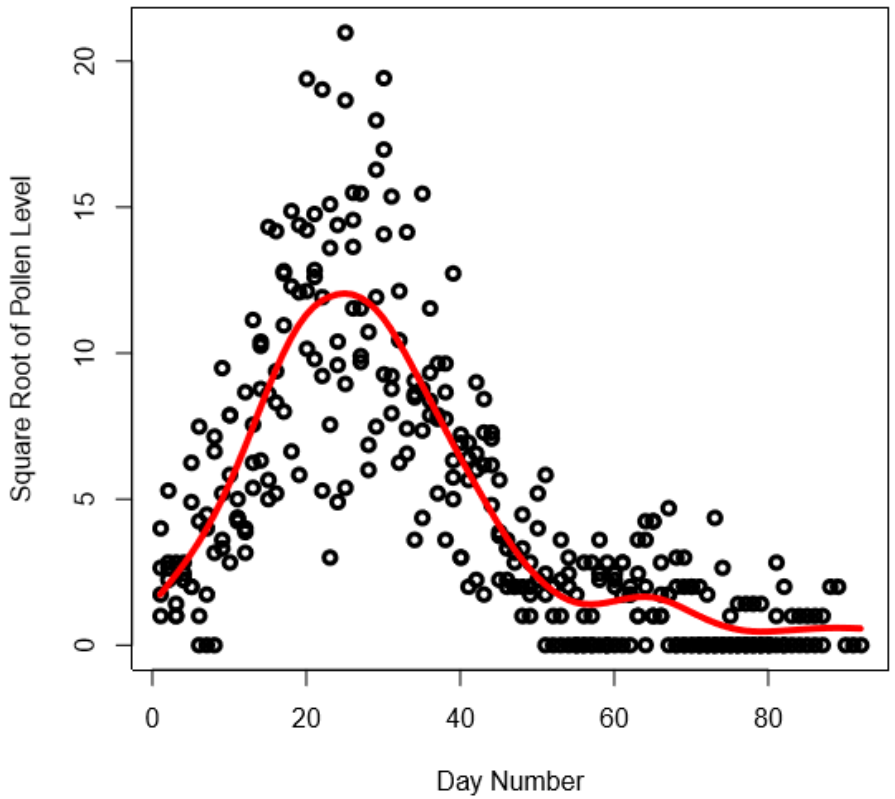
Example Where the Linear Model Fails

Below, we plot the ordinary least squares (OLS) linear model for this data set: $\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$. As you can see, the linear model *fails* to capture the true relationship between day and pollen level.



Nonparametric Regression

Below, we fit a smoothing spline between day and pollen level instead, i.e. $\widehat{y} = \widehat{\beta}_0 + \widehat{f}(x)$, where $\widehat{f}$ is a **nonlinear** function of x. As we can see, the nonparametric method provides a *much* better fit.



Nonparametric Regression

Henceforth, we assume that $\boldsymbol{y} = (y_1, \ldots, y_n)'$ has been centered, i.e. $\sum_{i=1}^n y_i = 0$, so there is no intercept in our model.

We can model the response as a (possibly nonlinear) **function** of the covariates. Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})'$ denote the vector of the p observed predictor values for observation i. We have the following model:

$$y_i = f(\boldsymbol{x}_i) = f(x_{i1}, \ldots, x_{ip}) + \varepsilon_i, \quad \mathbb{E}(\varepsilon_i) = 0, \mathbb{E}(\varepsilon_i^2) < \infty.$$

For nonparametric regression (as opposed to linear regression), we make minimal or no assumptions about the specific functional form of f.

Ways to Perform Nonparametric Regression

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i, \quad \mathbb{E}(\varepsilon_i) = 0, \mathbb{E}(\varepsilon_i^2) < \infty.$$

We can model the function $f(\cdot)$ using a number of methods:

- random forests
- gradient boosting
- neural networks
- kernel smoothing
- **generalized additive models** (the topic of this talk)

We can *also* perform model selection from the p covariates using these methods, so there is no real loss in interpretability. *And* we have added flexibility.

These methods are gaining popularity in machine learning because they often outperform linear regression for both estimation and prediction.

Parametric vs. Nonparametric Statistics

Table: Parametric vs. nonparametric statistics.

Parametric	**Nonparametric**
Parameter space is **finite**-dimensional ($p+2$ unknowns) $p+1$ coefficients in $\beta \in \mathbb{R}^{p+1}$ and noise variance σ^2	Parameter space is **infinite**-dimensional e.g. the set of all continuous functions f on [0,1] (say)
Strong assumptions about relationship (e.g. linearity) and/or distributional family (e.g. normality)	**Minimal or no assumptions** about relationship (can be any shape) or the distributional family

Generalized Additive Models (GAMs)

Suppose we have p covariates. Let $(x_{i1}, \ldots, x_{ip})'$ denote the p observed predictor values for the ith observation. Using *generalized additive models*, we model the response y_i as follows:

$$y_i = f_1(x_{i1}) + f_2(x_{i2}) + \ldots + f_p(x_{ip}) + \varepsilon_i = \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i,$$

where the f_j's can be smooth, nonlinear functions and we assume $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Question: How can we estimate the f_j's, $j = 1, \ldots, p$?

Generalized Additive Models

Assume that each f_j can be approximated by a linear combination of basis functions $\mathcal{B}_j = \{g_{j1}, \ldots, g_{jd}\}$, i.e.,

$$f_j(X_{ij}) \approx \sum_{k=1}^{d} g_{jk}(X_{ij})\beta_{jk}$$

We have a system of equations for each f_j. For the jth covariate, we are approximating for the n observations:

$$\begin{aligned}
f_j(X_{1j}) &\approx g_{j1}(X_{1j})\beta_{j1} + g_{j2}(X_{1j})\beta_{j2} + \ldots + g_{jd}(X_{1j})\beta_{jd}, \\
f_j(X_{2j}) &\approx g_{j1}(X_{2j})\beta_{j1} + g_{j2}(X_{2j})\beta_{j2} + \ldots + g_{jd}(X_{2j})\beta_{jd}, \\
&\vdots \\
f_j(X_{nj}) &\approx g_{j1}(X_{nj})\beta_{j1} + g_{j2}(X_{2j})\beta_{j2} + \ldots + g_{jd}(X_{nj})\beta_{jd}.
\end{aligned}$$

We denote the unknown weight vectors $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jd})^T$.

Matrix Representation of GAMs

Let $\widetilde{\boldsymbol{X}}_j$ denote the $n \times d$ matrix with the (i,k)th entry $\widetilde{\boldsymbol{X}}_j(i,k) = g_{jk}(X_{ij})$. Let $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jd})^T$ be the unknown weight vectors. Then we may represent the GAM in matrix form as

$$\boldsymbol{y} - \delta = \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n),$$

where δ is an $n \times 1$ vector of the lower-order bias (or approximation error). The approximation error is typically $O(n^{-\alpha})$ for some $\alpha > 0$, so the bias goes to zero as sample size increases.

We have p design matrices $\widetilde{\boldsymbol{X}}_j, j = 1, \ldots, p$, each of dimension $p \times d$. These are matrices of *basis functions* of the covariates.

Choosing a Basis Expansion

How do we choose the set of basis functions $\mathcal{B}_j = \{g_{j1}, \ldots, g_{jd}\}$ to approximate f_j? There are a lot of possibilities:

- Hermite polynomials
- Laguerre polynomials
- Fourier series
- splines

Of the above, splines are the most commonly used in practice since they are the most flexible (although Fourier series are useful for wavelet analysis and modeling data that is *known* to be periodic).

Splines

Splines are piecewise polynomial functions over an interval $[a, b]$, separated into sub-intervals. The endpoints of the sub-intervals are called *knots*. **Cubic splines** impose the conditions that the piecewise polynomials are cubic and that they continuous over $[a, b]$, C^1-continuous, and C^2-continuous (that is, the first and second derivatives are also continuous at the inner knots).

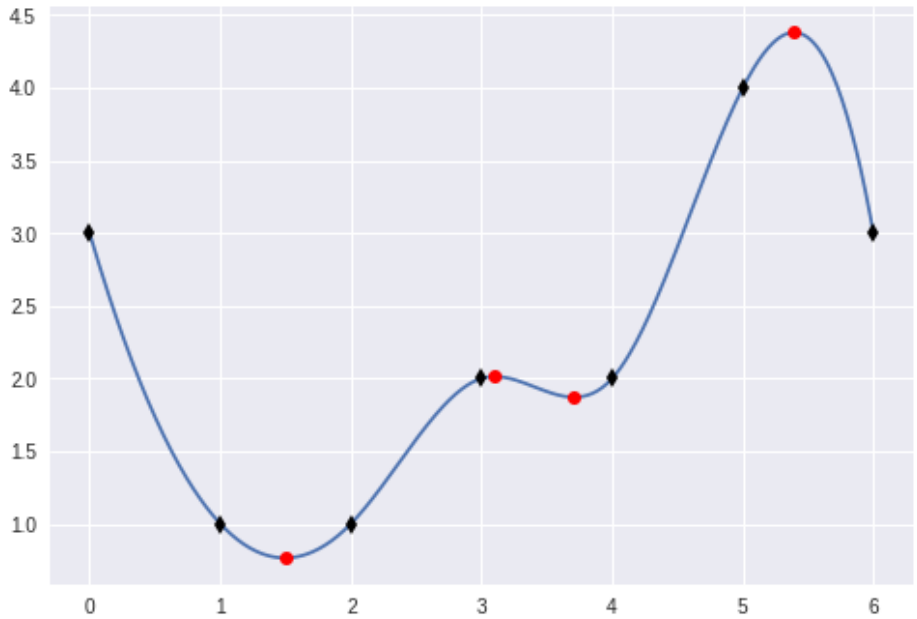


Figure: Image retrieved from `https://calculus7.org/tag/spline/`. Accessed 12 Mar. 2019.

Natural Cubic Splines

- Suppose that the interval $[a, b]$ is partitioned into $n+1$ knots: $a := t_0 < t_1 < t_2 < \ldots < t_{n-1} < t_n := b$. Most software uses equidistant points for the knots and chooses a "default" number of knots but allows the practitioner to override these defaults.
- Define the piecewise polynomials as

$$\begin{cases} S_0(x), & t_0 \leq x \leq t_1, \\ S_1(x), & t_1 \leq x \leq t_2, \\ \vdots & \\ S_{n-1}(x), & t_{n-1} \leq x \leq t_n. \end{cases}$$

The **natural cubic spline** imposes the condition that $S_0''(t_0) = S_{n-1}''(t_n) = 0$.

Variable Selection with GAMs

Letting $f_j(\boldsymbol{X}_j)$ denote an $n \times 1$ vector with ith component $f_j(x_{ij})$ and letting $\widetilde{\boldsymbol{X}}_j$ denote the jth design matrix of basis functions corresponding to the jth predictor, i.e. $\widetilde{\boldsymbol{X}}_j(i,k) = g_{jk}(X_{ij})$, recall that we have chosen to model our data as:

$$\boldsymbol{y} = \sum_{j=1}^{p} f_j(\boldsymbol{X}_j) + \boldsymbol{\varepsilon} = \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \boldsymbol{I}_n),$$

When the dimensionality of the covariates p is high (including $p \gg n$), we often want to impose a low-dimensional structure such as **sparsity**.

That is, we assume that the response y depends on only a few of the p covariates. Thus, *most* of the f_j's are $f_j = 0$ and thus do not contribute to predicting the response. This is equivalent to assuming that most of the weight vectors $\boldsymbol{\beta}_j = \mathbf{0}_d$.

Bayesian Model for GAMs

To conduct Bayesian analysis of the model,

$$\boldsymbol{y} = \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n),$$

we need to place priors on the weight vectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_p$, and a prior on the noise variance σ^2. For the noise variance, we put the noninformative Jeffrey's prior,

$$\pi(\sigma^2) \propto \sigma^{-2}.$$

What about the prior for the $\boldsymbol{\beta}_j$'s, $j = 1, \ldots, p$?

Group Lasso Density

Let $\boldsymbol{\beta}_j \in \mathbb{R}^d$ denote a real-valued vector of length d. We define the *group lasso density* as

$$\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda) = \frac{\lambda^d e^{-\lambda\|\boldsymbol{\beta}_j\|_2}}{2^d \pi^{(d-1)/2}\Gamma\left((d+1)/2\right)},$$

where $\|\boldsymbol{\beta}_j\|_2 = \sqrt{\beta_{j1}^2 + \ldots + \beta_{jd}^2}$.

This expression looks complicated but it can be derived as the marginal density for $\boldsymbol{\beta}_j$ of the hierarchical mixture

$$\boldsymbol{\beta}_j|\tau \sim \mathcal{N}_d(\mathbf{0}, \tau \boldsymbol{I}_d), \tau \sim \mathcal{G}\left((d+1)/2, \lambda^2/2\right),$$

and has been used by other authors before (e.g. Kyung et al. (2010) and Xu and Ghosh (2015)).

The Spike-and-Slab Group Lasso

Let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T)^T \in \mathbb{R}^{dp}$. We define the *spike-and-slab group lasso* (SSGL) as:

$$\begin{aligned} \pi(\boldsymbol{\beta}|\theta) = & \prod_{j=1}^{p} \left[(1-\theta)\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0) + \theta\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1) \right], \\ \theta \sim & \ \mathcal{B}(a,b), \end{aligned}$$

where $\boldsymbol{\Psi}(\cdot|\lambda)$ denotes the group lasso density indexed by hyperparameter λ, and $\theta \in (0,1)$ is a mixing proportion endowed with the prior $\mathcal{B}(a,b)$.

The Spike-and-Slab Group Lasso

$$\begin{aligned} \pi(\boldsymbol{\beta}|\theta) = & \prod_{j=1}^{p} \left[(1-\theta)\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0) + \theta\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1)\right], \\ \theta \sim & \ \mathcal{B}(a,b). \end{aligned}$$

$\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0)$ corresponds to the "spike" which shrinks all the entries in the vector $\boldsymbol{\beta}_j$ towards 0, while $\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1)$ corresponds to the "slab," which stabilizes vectors with large coefficients so they are not downward biased or shrunk as heavily.

The mixing proportion θ is the probability that the vector $\boldsymbol{\beta}_j \neq \mathbf{0}_d$ (or the probability that $\boldsymbol{\beta}_j, 1 \leq j \leq p$, belongs to the slab rather than the spike).

Hyperparameters in the Spike-and-Slab Group Lasso

$$\begin{aligned} \pi(\boldsymbol{\beta}|\theta) = & \prod_{j=1}^{p} \left[(1-\theta)\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0) + \theta\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1)\right], \\ \theta \sim & \ \mathcal{B}(a, b). \end{aligned}$$

We fix λ_1 to be small, so that the slab has large variance and prevents overshrinkage of large coefficients.

We choose λ_0 to be large, so that the spike has small variance and is heavily concentrated around $\mathbf{0}_d$.

Under sparsity, we want θ to be small with high probability, so that most of the weight vectors $\boldsymbol{\beta}_j$ belong to the spike. We can set $a = 1, b = p$ so that θ is concentrated near zero and most of the weight vectors will belong to the spike.

Advantages of the Spike-and-Slab Group Lasso

$$\begin{aligned} \pi(\boldsymbol{\beta}|\theta) = & \prod_{j=1}^{p} \left[(1-\theta)\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0) + \theta\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1)\right], \\ \theta \sim & \mathcal{B}(a,b). \end{aligned}$$

- The two-component spike-and-slab model ensures that **the amount of shrinkage applied to each weight vector $\boldsymbol{\beta}_j, j = 1, \ldots, p$, is adaptive**. Rather than applying the same amount of shrinkage to each $\boldsymbol{\beta}_j$, the SSGL will apply *less* shrinkage if the coefficients are larger.
- The prior on θ allows our model to **self-adapt to the true sparsity pattern** of the data.
- The prior on θ ensures that our model favors parsimonious models when p is very large. We thus **avoid the curse of dimensionality**.

Bayesian Computation

Another advantage of the SSGL is that the posterior mode for the weight vector $\boldsymbol{\beta}_j$ under the SSGL can be exactly $\mathbf{0}_d$. We can thus **use the posterior mode to automatically threshold** out insignificant functional components f_j to be zero.

Our model performs simultaneous variable selection *and* estimation. Let $\widehat{f}_j(\boldsymbol{X}_j)$ denote the $n \times 1$ vector of the function $\widehat{f}_j$ evaluated at each observation $X_{ij}, i = 1, \ldots, n$.

- If $\widehat{\boldsymbol{\beta}}_j = \mathbf{0}_d$, then our function estimate is $\widehat{f}_j(\boldsymbol{X}_j) = 0$.
- If we estimate $\widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}_d$, then the ith component of $\widehat{f}_j(\boldsymbol{X}_j)$ is $f_j(X_{ij}) = \sum_{k=1}^{d} g_{jk}(X_{ij})\widehat{\beta}_{jk} \neq 0$.

In order to implement our Bayesian model, we *only* need to find the posterior mode. We do *not* need to use MCMC.

Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) has been a staple in Bayesian analysis for decades.

PROS OF MCMC:

- It is both theoretically sound and "exact." If you run the algorithm for enough iterations, it will eventually converge to the target distribution.
- We can perform automatic inference (estimation *and* uncertainty quantification) from the estimated posterior distribution.

CONS OF MCMC:

- It often involves expensive matrix inversions.
- In high dimensions, it can be be very slow (takes hours or even days to run). This may not be acceptable for practitioners.
- Even in moderate dimensions, it can be slow if the posterior is multimodal (chain can become trapped at a local mode).

Faster Alternatives to MCMC

There has been some work done to make MCMC more scalable. An alternative is to bypass MCMC completely. It all boils down to this: **Turn Bayesian computation into an optimization problem.**

- Bayesian coordinate ascent/EM algorithms to perform maximum *a posteriori* (MAP) estimation: Find the most probabilistically likely value of the unknown parameters.
- Variational inference. Approximate the posterior with a *variational density* that belongs to an exponential family. Optimize the hyperparameters in the variational density.
- Expectation-propagation. Another way to approximate the posterior with an approximate density by iteratively minimizing the Kullback-Leibler divergence.

Posterior MAP Estimation

In Bayesian analysis, we use Bayes' theorem to obtain the posterior:

$$\pi(\boldsymbol{\beta}, \sigma^2 | \boldsymbol{y}) \propto f(\boldsymbol{y} | \boldsymbol{\beta}, \sigma^2) \pi(\boldsymbol{\beta}) \pi(\sigma^2).$$

The log of the posterior under our model is

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2} \|\boldsymbol{y} - \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j\|_2^2 - \frac{n}{2} \log \sigma^2 + \log \pi(\boldsymbol{\beta}) + \log \pi(\sigma^2).$$

The posterior mode for $\boldsymbol{\beta}$ is the $\widehat{\boldsymbol{\beta}}$ which maximizes $L(\boldsymbol{\beta}, \sigma^2)$ with respect to $\boldsymbol{\beta}$ and is the "most likely" parameter values under our prior.

This has connections with penalized likelihood, where $\log \pi(\boldsymbol{\beta})$ is a sparsity-inducing prior and thus can be thought of as *penalty function* on the regression coefficients $\boldsymbol{\beta}$.

Posterior MAP Estimation

We treat the log-posterior $L(\boldsymbol{\beta}, \sigma^2)$ as our objective function to maximize.

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j\|_2^2 - \frac{n}{2}\log(\sigma^2) + \log \pi(\boldsymbol{\beta}) + \log \pi(\sigma^2).$$

A potential problem is that if the prior $\pi(\boldsymbol{\beta})$ is not log-concave, then $L(\boldsymbol{\beta}, \sigma^2)$ will *not* be concave and the posterior will be multimodal. Thus, merely numerically solving

$$\frac{\partial L}{\partial \beta_j} = 0, j = 1, \ldots, p, \qquad \frac{\partial L}{\partial \sigma^2} = 0,$$

may give you only a local posterior mode rather than the *global* mode.

The SSGL prior is *not* log-concave and hence it results in a multimodal posterior. Fortunately, we can use theory developed by Zhang and Zhang (2012) for non-concave penalties to find the **global** posterior mode rather than merely a local mode.

The SSGL Penalty

Recall that $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T)^T \in \mathbb{R}^{dp}$ and that $\pi(\sigma^2) \propto \sigma^{-2}$. Instead of using $\log \pi(\boldsymbol{\beta})$, we will use a slightly modified penalty, i.e. the objective function we will optimize is

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j\|_2^2 - (n+2)\log\sigma + pen(\boldsymbol{\beta}),$$

where we ensure that $pen(\mathbf{0}_{dp}) = 0$.

Doing this only adds an additive constant to $L(\boldsymbol{\beta}, \sigma^2)$ and therefore does not affect the final solution. However, it turns out that centering the penalty allows us to obtain a *much* more refined characterization of the posterior mode.

The SSGL Penalty

With the prior on θ, $\theta \sim \mathcal{B}(a, b)$, the marginal prior for the regression coefficients is:

$$\pi(\boldsymbol{\beta}) = \int_0^1 \prod_{j=1}^{p} [(1-\theta)\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_0) + \theta\boldsymbol{\Psi}(\boldsymbol{\beta}_j|\lambda_1)] d\pi(\theta).$$

We then define $pen(\boldsymbol{\beta})$ as

$$pen(\boldsymbol{\beta}) = \log\left[\frac{\pi(\boldsymbol{\beta})}{\pi(\mathbf{0}_{dp})}\right]$$

This ensures that $pen(\mathbf{0}_{dp}) = 0$ and only adds an additive constant of $-\log \pi(\mathbf{0}_{dp})$ to the objective $L(\boldsymbol{\beta}, \sigma^2)$.

The Global Posterior Mode Under the SSGL

Assume that the design matrices have been orthonormalized, i.e. $\widetilde{\boldsymbol{X}}_j^T \widetilde{\boldsymbol{X}}_j = \boldsymbol{I}_d, j = 1, \ldots, p$. This is not an issue – it just means that our basis functions are taken to be orthonormal basis functions. Define the quantities,

$$p^\star_{\theta_j}(\boldsymbol{\beta}_j) = \frac{\theta_j \boldsymbol{\Psi}(\boldsymbol{\beta}_j | \lambda_1)}{\theta_j \boldsymbol{\Psi}(\boldsymbol{\beta}_j | \lambda_1) + (1 - \theta_j)\boldsymbol{\Psi}(\boldsymbol{\beta}_j | \lambda_0)},$$
$$\lambda^\star_{\theta_j}(\boldsymbol{\beta}_j) = p^\star_{\theta_j}(\boldsymbol{\beta}_j)\lambda_1 + [1 - p^\star_{\theta_j}(\boldsymbol{\beta}_j)]\lambda_0,$$

where $\theta_j = \mathbb{E}[\theta | \boldsymbol{\beta}_{\setminus j}]$ is the conditional mean for θ given the subvector of $\boldsymbol{\beta}$ that excludes the d elements corresponding to the jth covariate. Finally, we define the function $h : \mathbb{R}^d \to \mathbb{R}$ as

$$h(\boldsymbol{\beta}_j) = \left[\lambda^\star_{\theta_j}(\boldsymbol{\beta}_j) - \lambda_1\right]^2 + \frac{2}{\sigma^2} \log p^\star_{\theta_j}(\boldsymbol{\beta}_j).$$

We can uniquely characterize the global mode in the next theorem.

The Global Posterior Mode Under the SSGL

Theorem

Let $\boldsymbol{z}_j = \widetilde{\boldsymbol{X}}_j'(\boldsymbol{y} - \sum_{l \neq j} \widetilde{\boldsymbol{X}}_l \widehat{\boldsymbol{\beta}}_l)$, *and let* $\theta_j = \mathbb{E}[\theta | \widehat{\boldsymbol{\beta}}_{\setminus j}]$. *Further, suppose that* $(\lambda_0 - \lambda_1) > 2/\sigma$ *and* $h(\mathbf{0}_d) > 0$. *Then the global posterior mode under* $pen(\boldsymbol{\beta})$ *satisfies*

$$\widehat{\boldsymbol{\beta}}_j = \begin{cases} \mathbf{0}_d & \text{when } \|\boldsymbol{z}_j\|_2 \leq \Delta_j, \\ \left(1 - \frac{\sigma^2 \lambda^\star_{\theta_j}(\widehat{\boldsymbol{\beta}}_j)}{\|\boldsymbol{z}_j\|_2}\right)_+ \boldsymbol{z}_j & \text{when } \|\boldsymbol{z}_j\|_2 > \Delta_j, \end{cases}$$

where the threshold Δ_j *satisfies*

$$\Delta_j = \inf_{\boldsymbol{\beta}_j} \left\{ \frac{\|\boldsymbol{\beta}_j\|_2}{2} - \frac{\sigma^2 pen(\boldsymbol{\beta})}{\|\boldsymbol{\beta}_j\|_2} \right\}$$

The Global Posterior Mode Under the SSGL

The threshold Δ_j is difficult to compute, but we have the following lower and upper bounds for Δ_j.

Theorem

The threshold Δ_j can be bounded as

$$\Delta_j^L < \Delta_j < \Delta_j^U,$$

with

$$\Delta_j^L = \sqrt{2\sigma^2 \log[1/p^\star_{\theta_j}(\mathbf{0}_d)] - \sigma^4\nu} + \sigma^2\lambda_1,$$
$$\Delta_j^U = \sqrt{2\sigma^2 \log[1/p^\star_{\theta_j}(\mathbf{0}_d)]} + \sigma^2\lambda_1,$$

and

$$0 < \nu < \frac{2}{\sigma^2} - \left(\frac{1}{\sigma^2(\lambda_0 - \lambda_1)} - \frac{\sqrt{2}}{\sigma}\right)^2.$$

The Global Posterior Mode Under the SSGL

- For large λ_0, $\nu \to 0$, and so, $\Delta_j^L \approx \Delta_j^U$.
- When the number of covariates p is large, $\mathbb{E}[\theta|\widehat{\boldsymbol{\beta}}_{\setminus j}]$ is approximately the same as $\widehat{\theta} = \mathbb{E}[\theta|\widehat{\boldsymbol{\beta}}]$. For practical purposes, we may thus replace every instance of θ_j with $\widehat{\theta}$. Consequently, we can replace all the individual thresholds $\Delta_j, j = 1, \ldots, p$, with a single threshold,

$$\Delta^U = \sqrt{2\sigma^2 \log[1/p_{\widehat{\theta}}(\mathbf{0}_d)]} + \sigma^2\lambda_1.$$

- Let $\widehat{q}$ be the number of weight vectors where $\widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}_d$. In Bai et al. (2019), it is shown that the conditional mean $\widehat{\theta}$ satisfies

$$\mathbb{E}[\theta|\widehat{\boldsymbol{\beta}}] = \frac{a + \widehat{q}}{a + b + p}$$

when $\lambda_0 \to \infty$.

Coordinate Ascent Algorithm

With all these ingredients, we can define our coordinate ascent algorithm to find the posterior mode estimator $\widehat{\boldsymbol{\beta}}$. There are some additional technical details, but the general gist is the following:

1. Initialize $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^*, \theta^{(0)} = \theta^*, \sigma^{2(0)} = \sigma^{2*}, \Delta^U = \Delta^*$.
2. For each iteration k, repeat the following updates for $(\boldsymbol{\beta}^{(k)}, \theta^{2(k)}, \sigma^{2(k)}, \Delta^U)$ until convergence:

$$\boldsymbol{\beta}_j^{(k)} \leftarrow \left(1 - \frac{\sigma^{2(k)}\lambda^*(\boldsymbol{\beta}_j^{(k-1)}; \theta^{(k)})}{\|\boldsymbol{z}_j\|_2}\right)_+ \boldsymbol{z}_j \mathbb{I}(\|\boldsymbol{z}_j\|_2 > \Delta^U),$$

$$\theta^{(k)} \leftarrow \frac{a + \widehat{q}^{(k)}}{a + b + p},$$

$$\sigma^{2(k)} \leftarrow \frac{\|\mathbf{y} - \widetilde{\boldsymbol{X}}\boldsymbol{\beta}^{(k)}\|_2^2}{n+2},$$

$$\Delta^U \leftarrow \begin{cases} \sqrt{2\sigma^{2(k)} \log[1/p^*(\mathbf{0}_d; \theta^{(k)})]} + \sigma^{2(k)}\lambda_1 & \text{if } h(\mathbf{0}_d; \theta^{(k)}) > 0 \\ \sigma^{2(k)}\lambda^*(\mathbf{0}_d; \theta^{(k)}) & \text{otherwise} \end{cases}$$

We fix λ_1 to be a small value and tune λ_0 from $\lambda_0 \in \{1, 2, \ldots, 25\}$ and the degrees of freedom d from a reasonable range of values.

Uncertainty Quantification

The coordinate ascent algorithm will find the global posterior mode $\widehat{\boldsymbol{\beta}}_j$, $j = 1, \ldots, p$, and most of these will be estimated as $\widehat{\boldsymbol{\beta}}_j = \mathbf{0}_d$, meaning that most of the function estimates, $\widehat{f}_j(\boldsymbol{X}_j) = \sum_{k=1}^{d} g_{jk}(\boldsymbol{X}_j)\widehat{\beta}_{jk} = 0$. However, this only gives a point estimate. How can we do inference?

Define $\widehat{\boldsymbol{\Sigma}} = \widetilde{\boldsymbol{X}}^T \widetilde{\boldsymbol{X}}/n$ and $\widehat{\boldsymbol{\Theta}}$ be an approximate inverse of $\widehat{\boldsymbol{\Sigma}}$. With $\widehat{\boldsymbol{\beta}}$ the global posterior mode of $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T)^T$, define the vector,

$$\widehat{\boldsymbol{\beta}}^{\mathrm{DB}} = \widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\Theta}}\widetilde{\boldsymbol{X}}^T(\boldsymbol{Y} - \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}})/n.$$

In van de Greer et al. (2014), it was shown that this "de-biased" vector has the asymptotic distribution,

$$\sqrt{n}(\widehat{\boldsymbol{\beta}}^{\mathrm{DB}} - \boldsymbol{\beta}) \sim \mathcal{N}(\mathbf{0}, \sigma^2 \widehat{\boldsymbol{\Theta}}\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Theta}}^T).$$

De-biasing for Uncertainty Quantification

Let $\beta_{jk}, k = 1, \ldots, d$, be the kth component of vector $\boldsymbol{\beta}_j, j = 1, \ldots, p$. Taking the modal estimate of $\widehat{\sigma}^2$ and the neighborhood selection covariance estimator $\widehat{\boldsymbol{\Theta}}$ given in Meinshausen and Bühlmann (2006), we have as the asymptotic $100(1-\alpha)\%$ pointwise confidence intervals for β_{jk}:

$$[\widehat{\beta}^L_{jk}, \widehat{\beta}^U_{jk}] := [\widehat{\beta}^{\mathrm{DB}}_{jk} - c(\alpha, n, \widehat{\sigma}^2), \widehat{\beta}^{\mathrm{DB}}_{jk} + c(\alpha, n, \widehat{\sigma}^2)],$$

where $c(\alpha, n, \widehat{\sigma}^2) = \Phi^{-1}(1-\alpha/2)\sqrt{\widehat{\sigma}^2(\widehat{\boldsymbol{\Theta}}\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Theta}}^T)_{jk,jk}/n}$.

To construct $100(1-\alpha)\%$ confidence bands for $f_j(\boldsymbol{X}_j), j = 1, \ldots, p$, we take $[\widehat{f}^L_j(\boldsymbol{X}_j), \widehat{f}^U_j(\boldsymbol{X}_j)]$, where

$$\widehat{f}^L_j(\boldsymbol{X}_j) = \sum_{k=1}^{d} g_{jk}(\boldsymbol{X}_j)\widehat{\beta}^L_{jk},$$

$$\widehat{f}^U_j(\boldsymbol{X}_j) = \sum_{k=1}^{d} g_{jk}(\boldsymbol{X}_j)\widehat{\beta}^U_{jk}.$$

Modeling Nonlinear Interactions with GAMs

In recent years, there has been a great deal of interest in identifying higher-order nonlinear interaction terms between covariates. Below, we consider all two-way interactions between the covariates, as well as main effects of the individual covariates.

We assume that the interaction effects may be decomposed into the sum of bivariate functions of each pair of covariates, yielding the model:

$$y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \sum_{k=1}^{p-1} \sum_{l=k+1}^{p} f_{kl}(X_{ik}, X_{il}) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Modeling Nonlinear Interactions with GAMs

For the bivariate functions, we approximate f_{kl} using the outer product of the basis functions of the interacting covariates:

$$f_{kl}(X_{ik}, X_{il}) \approx \sum_{s=1}^{d^*} \sum_{r=1}^{d^*} g_{ks}(X_{ik}) g_{lr}(X_{il}) \beta_{klsr},$$

where $\boldsymbol{\beta}_{kl} = (\beta_{kl11}, \ldots, \beta_{kl1d^*}, \beta_{kl21}, \ldots, \beta_{kld^*d^*})^T \in \mathbb{R}^{d^{*2}}$ is the vector of unknown weights. We let $\widetilde{\boldsymbol{X}}_{kl}$ denote the $n \times d^{*2}$ matrix with rows

$$\widetilde{\boldsymbol{X}}_{kl}(i, \cdot) = \text{vec}(\boldsymbol{g}_k(X_{ik}) \boldsymbol{g}_l(X_{il})^T),$$

where $\boldsymbol{g}_k(X_{ik}) = (g_{k1}(X_{ik}), \ldots, g_{kd^*}(X_{ik}))^T$. Our model can be represented in matrix form as

$$\boldsymbol{y} - \delta = \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j + \sum_{k=1}^{p-1} \sum_{l=k+1}^{p} \widetilde{\boldsymbol{X}}_{kl} \boldsymbol{\beta}_{kl} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n),$$

where δ is a vector of the lower-order bias.

Interaction Detection with the SSGL

In this interaction model, we either include $\boldsymbol{\beta}_{kl}$ in the model (i.e. estimate $\widehat{\boldsymbol{\beta}}_{kl} \neq \mathbf{0}_{d^{*2}}$) if there is a non-negligible interaction between the kth and lth covariates, or we estimate $\widehat{\boldsymbol{\beta}}_{kl} = \mathbf{0}_{d^{*2}}$ if such an interaction is negligible.

To implement the SSGL model, we maximize the following objective function with respect to $(\boldsymbol{\beta}, \sigma^2)$:

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2} \left\| \boldsymbol{y} - \sum_{j=1}^{p} \widetilde{\boldsymbol{X}}_j \boldsymbol{\beta}_j - \sum_{k=1}^{p-1} \sum_{l=k+1}^{p} \widetilde{\boldsymbol{X}}_{kl} \boldsymbol{\beta}_{kl} \right\|_2^2 + pen(\boldsymbol{\beta}) - (n+2) \log \sigma,$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T, \boldsymbol{\beta}_{12}^T, \ldots, \boldsymbol{\beta}_{(p-1)p}^T)^T \in \mathbb{R}^{pd + p(p-1)d^{*2}/2}$ and $pen(\boldsymbol{\beta})$ is the SSGL penalty.

Simulation Studies

We will compare our SSGL approach with the following methods:

1. GroupLasso: the group lasso of Yuan and Lin (2010)
2. BSGS: Bayesian sparse group selection of Chen et al. (2016)
3. SoftBart: the soft BART approach of Linero and Yang (2018)
4. RandomForest: the random forest approach of Breiman (2001)
5. SuperLearner: the super learner of van der Laan et al. (2007)

We will look at:

- **predictive accuracy**: the mean squared error (MSE) for estimating $f(\boldsymbol{X}_{\text{new}})$ averaged over a new sample of data $\boldsymbol{X}_{\text{new}}$.
- **variable selection accuracy**: the F1 score, which is a measure that balances precision and recall. A higher F1 score means the model is doing a better job selecting the relevant covariates, while thresholding out irrelevant covariates.

Sparse GAM Simulation

We first generate independent covariates from a standard uniform distribution and we let the true regression surface take the following form:

$$\mathbb{E}(Y|\boldsymbol{X}) = 5\sin(\pi X_1) + 2.5(X_3^2 - 0.5) + e^{X_4} + 3X_5,$$

with variance $\sigma^2 = 1$. The rest of the functions are set to $f(X_j) = 0, j = 6, \ldots, p$.

We vary $n \in \{100, 300\}$ and the number of covariates $p = 300$. We use **natural cubic splines** as the basis functions and tune the degree of freedom $d \in \{2, 3, 4\}$ using cross-validation. Thus, we are estimating a total of between 600 and 1200 unknown basis coefficients.

We repeat each simulation 1000 times and average the MSE and the F1 score across the 1000 replications.

Sparse GAM Simulation with $n = 100$

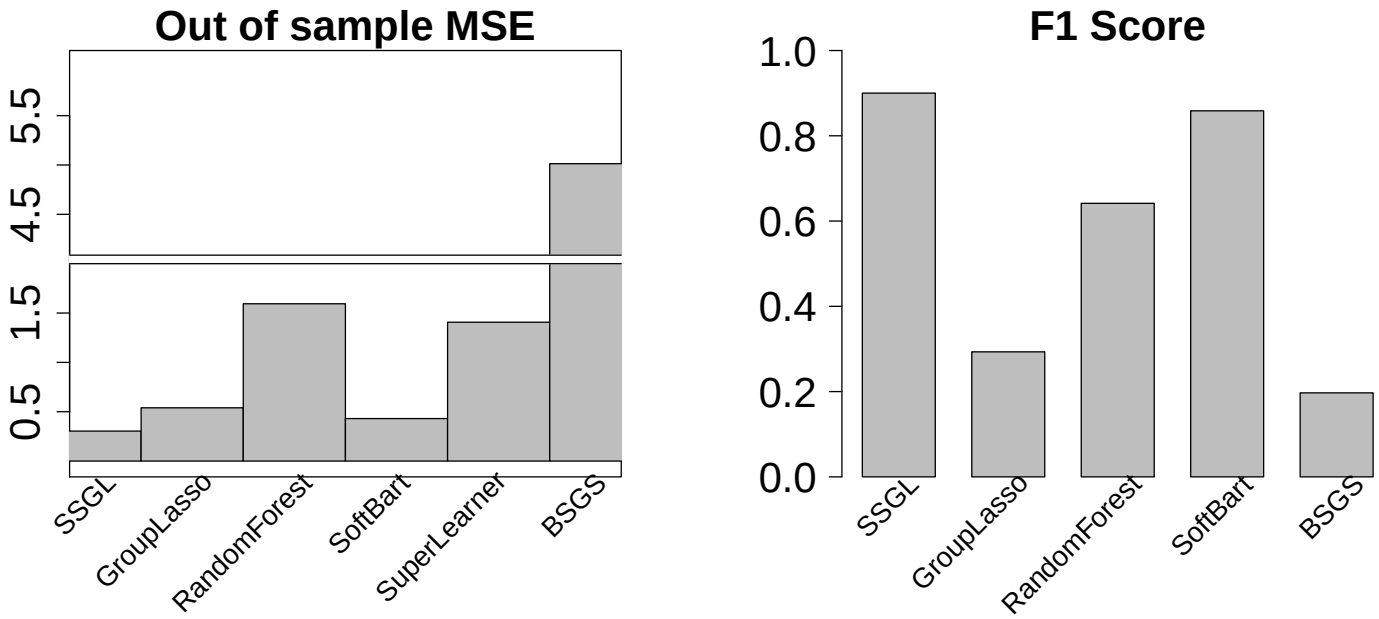


Figure: Simulation results with $n = 100$. The left panel presents the out-of-sample mean squared error and the right panel shows the F1 score to evaluate variable selection. The SSGL has the lowest MSE and the higher F1 score, meaning it has the best performance.

Sparse GAM Simulation with $n = 100$

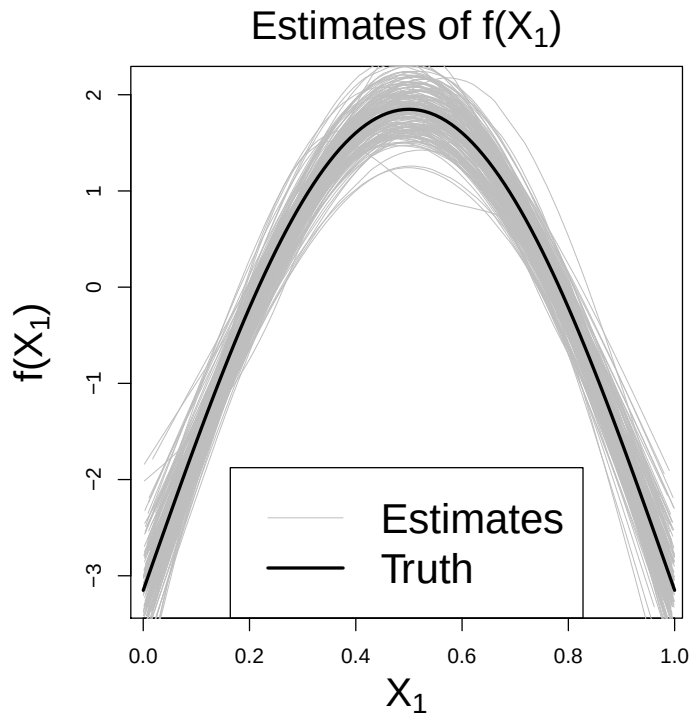


Figure: Plot of the estimates from each simulation of $f_1(X_1)$ against the truth $f_1(X_1) = 5\sin(\pi X_1)$.

Sparse GAM Simulation with $n = 300$

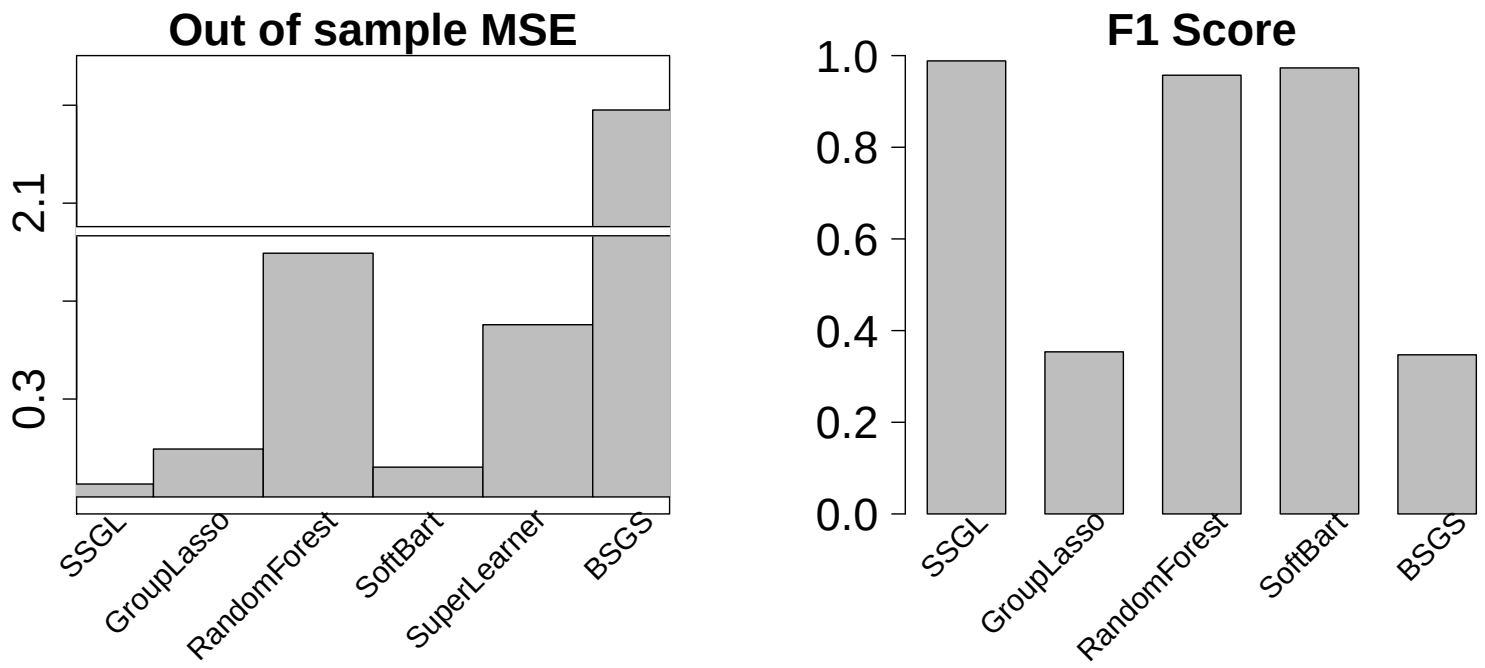


Figure: Simulation results from the sparse setting with $n = 300$. The left panel presents the out-of-sample mean squared error, and the right panel shows the F1 score to evaluate variable selection.

Interaction Detection Simulation

We now explore the ability of the SSGL approach to identify important interaction terms in a nonparametric regression model. We generate 25 independent covariates from a standard uniform distribution with a sample size of $n = 300$. Data is generated from the following model:

$$\mathbb{E}(Y|\boldsymbol{X}) = 2.5\sin(\pi X_1 X_2) + 2\cos(\pi(X_3 + X_5)) + 2(X_6 - 0.5) + 2.5X_7,$$

with $\sigma^2 = 1$.

This may not seem like a high-dimensional scenario, but it actually *is*, because we will consider all two-way interactions. There are 300 such interactions. The important two-way interactions are between X_1 and X_2 and between X_3 and X_5. We again use **natural cubic splines** as the basis functions.

Interaction Detection Simulation

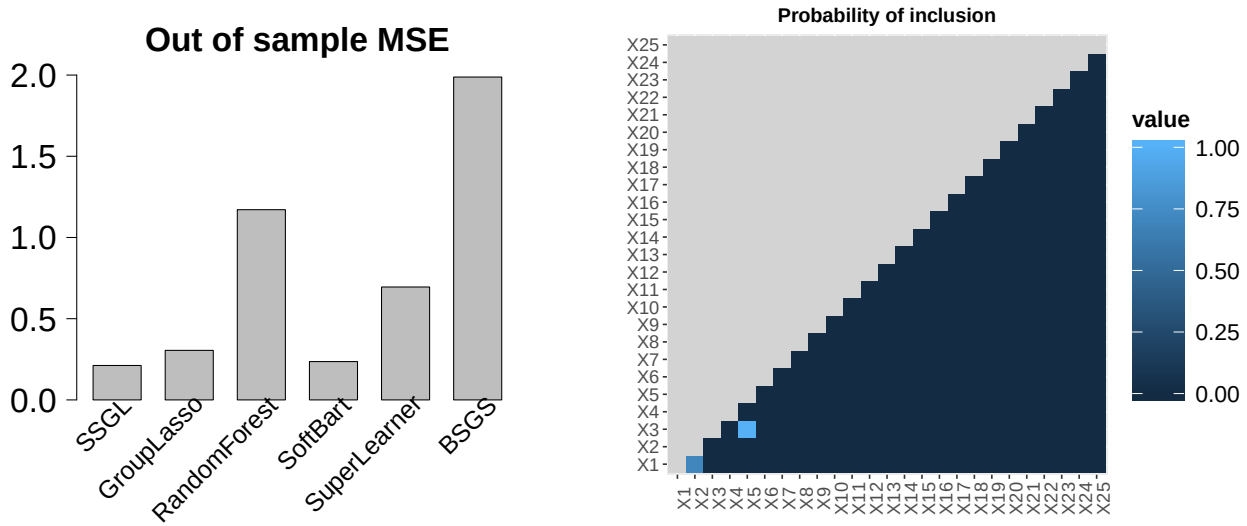


Figure: The left panel shows out-of-sample MSE , while the right panel shows the probability of a two-way interaction being included into the SSGL model for all pairs of covariates.

SSGL does a very good job at identifying the two important interactions. The (X_1, X_2) interaction is included in 70% of simulations, while the (X_3, X_5) interaction is included 100% of the time. All other interactions are included less than 1% of the time.

Estimating Health Effects of Environmental Exposures

We analyze data from the 2001-2002 cycle of the National Health and Nutrition Examination Survey. We aim to identify which organic pollutants are associated with changes in leukocyte telomere length (LTL) levels. Telomeres are segments of DNA that help to protect chromosomes, and LTL levels are commonly used as a proxy for overall telomere length.

We may expect that high levels of two toxins may have an even more adverse effect on a person's health than having high levels of either of the two toxins, so we include interactions in our model.

In addition to the 18 exposures, there are 18 additional demographic variables (possible confounding variables) which we adjust for in our model.

Estimating Health Effects of Environmental Exposures

We model the effects of the 18 exposures on LTL length using the SSGL prior, with natural cubic splines with 2 degrees of freedom as the basis functions. For the interaction terms, this leads to 4 terms for each pair of interactions, and we orthogonalize these terms with respect to the main effects so that the interaction terms will not be included simply due to there being main effects of the pollutants.

- Our model identifies one important main effect, which is the main effect of the first furan. We identify a positive association between Furan1 and LTL levels.
- Our model also identifies two interactions as having nonzero regression coefficients in the model. These are between Dioxin1 and PCB126, as well as between Dioxin2 and PCB99.

Estimating Health Effects of Environmental Exposures

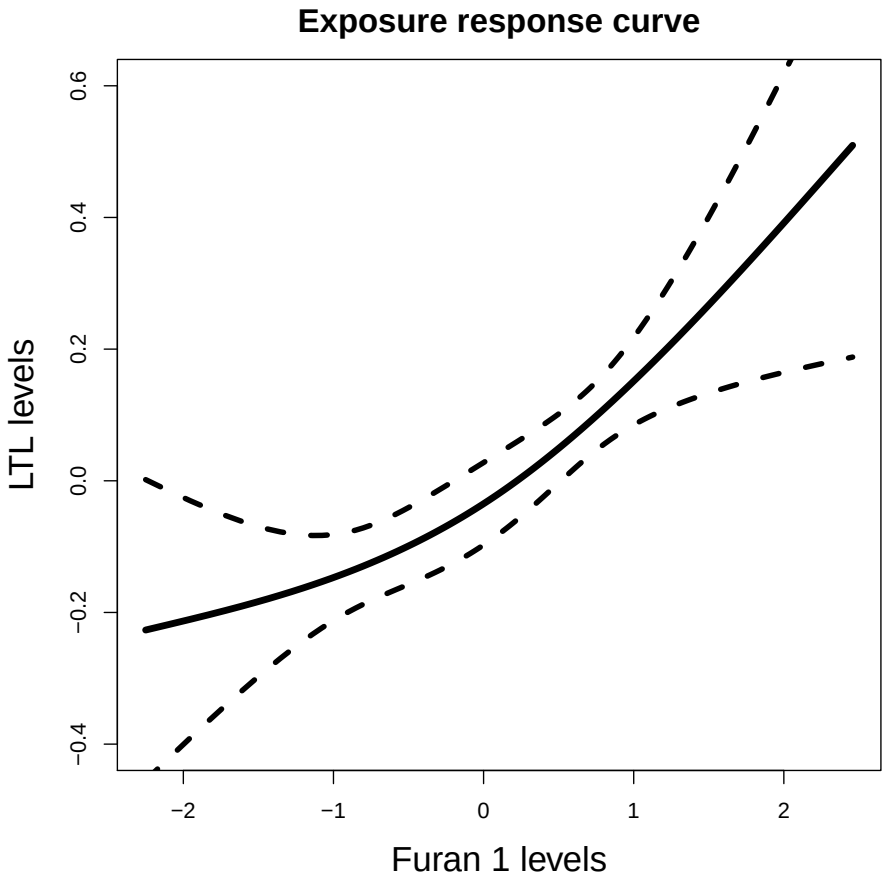


Figure: Estimated exposure-response curve between Furan1 and LTL levels in the NHANES data, along with the 95 % confidence bands.

Summary

We have introduced the spike-and-slab group lasso (SSGL) for Bayesian estimation and variable selection in sparse generalized additive models (GAMs).

- The posterior mode of the SSGL shrinks vectors of basis coefficients to $\mathbf{0}_d$, while stabilizing estimates of nonzero vectors. So we can perform estimation and variable selection *simultaneously*.
- The prior on the mixing proportion $\theta \sim \mathcal{B}(1, p)$ helps our model to avoid the curse of dimensionality and makes the SSGL penalty fully self-adaptive to the true sparsity pattern of the data.
- We can implement the SSGL model and provide uncertainty quantification without using MCMC. We use a coordinate ascent algorithm to find the MAP estimator and use de-biasing methods for uncertainty quantification instead.
- The SSGL can rapidly identify important nonlinear main effects as well as nonlinear interaction effects.

Pre-print of Paper

The paper associated with this presentation is available at:
`https://arxiv.org/abs/1903.01979`

Our paper contains more simulation studies and data analyses. We also prove a handful of theorems about the SSGL model (posterior contraction rates, etc.). If you are interested in the theoretical aspects of our model, please refer to the above pre-print.

This presentation will be available for download on my website:
`http://www.raybai.net`

References


Antonelli, J., Mazumdar, M., Bellinger, D., Christiani, D. C., Wright, R., and Coull, B. A. (2017). Estimating the health effects of environmental mixtures using bayesian semiparametric regression and sparsity inducing priors. *arXiv preprint arXiv:1711.11239*.


Bai, R., Moran, G. E., Antonelli, J., Chen, Y., and Boland M. R. (2019). Spike-and-slab group lassos for grouped regression and sparse generalized additive models. *arXiv preprint arXiv:1903.01979*.


Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.


Chen, R.-B., Chu, C.-H., Yuan, S., and Wu, Y. N. (2016). Bayesian sparse group selection. *Journal of Computational and Graphical Statistics*, **25**:665-683.


Javanmard, A. and Montanari, A. (2018). Debiasing the lasso: Pptimal sample size for gaussian designs. *The Annals of Statistics*, **46**:2593-2622.


Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2010). Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, **5**:369-411.


Linero, A. R. and Yang, Y. (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **80**:1087-1110.

References


Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, **34**:1436–1462.


Moran, G. E., Ročková, V., and George, E. I. (2019). Variance prior forms for high-dimensional Bayesian variable selection. *Bayesian Analysis (to appear)*.


Ročková, V. and George, E. I. (2018). The spike-and-slab lasso. *Journal of the American Statistical Association*, **113**:431–444.


van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, **42**:1166–1202.


van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, **6**:1544-6115.


Xu, X. and Ghosh, M. (2015). Bayesian variable selection and estimation for group lasso. *Bayesian Analysis*, **10**:909–936.


Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: series B (Statistical Methodology)*, **68**:49–67.


Zhang, C.-H. and Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, **27**: 576-593.

Questions?