# Predictor Selection Algorithm for Bayesian Lasso

Quan Zhang*

May 16, 2014

## 1 Introduction

The Lasso [1] is a method in regression model for coefficients shrinkage and model selection. It is often used in the linear regression model

$$\boldsymbol{y} = \mu \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{y}$ is the response vector with the length of $n$, $\mu$ is the overall mean, $\boldsymbol{X}$ is the $n \times p$ standardized design matrix, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)'$, and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of iid normal errors with zero mean and unknown variance $\sigma^2$. The Lasso estimates, $\hat{\boldsymbol{\beta}}$, minimizes the sum of the squared residuals subject to $|\boldsymbol{\beta}| \leq t$. It can be described as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta})'(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} |\beta_i|$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \bar{y}\mathbf{1}_n$, and $\lambda \geq 0$ is the tuning parameter relating to the bound $t$. It is indicated that $\hat{\boldsymbol{\beta}}$ can be interpreted as a Bayesian mode estimate when the parameters $\beta_i$'s have iid double exponential priors, i.e.,

$$\pi(\boldsymbol{\beta}) = \prod_{i=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_i|}. \tag{1}$$

The double exponential distribution can be writen as a scale mixture of normals:

$$\frac{\alpha}{2} e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}} e^{-z^2/(2s)} \frac{a^2}{2} e^{-a^2 s/2} ds,\ a > 0. \tag{2}$$

The detailed hierarchical model formulation and the Gibbs sampler implementation can been seen in Park and Casella's Bayesian Lasso paper [2]. Note that $\lambda$ can be not only pre-specified, but also estimated if a hyperprior is placed. A slightly more complicated Bayesian hierarchical model and its Gibbs sampler implementation will be given in Section 4.

However, as Park and Cassella said [2], this method cannot automatically perform predictor selection, in other words, it cannot force some of the regression coefficients to zero. We propose an algorithm to automatically select predictors in the process of the Gibbs sampler. We applied this method to ordinary linear regression problem with variable selection. Another important application is the penalized spline using the $l_1$ penalty. Using the Bayesian Lasso with predictor selection, we can not only fit the spline model, but also automatically select knots.

*Division of Biostatistics, School of Public Health, University of Minnesota.

## 2 Algorithm for predictor selection

As described above, we have $p$ predictors, $X_1, \cdots, X_p$ and denote the corresponding regression parameters of the predictors as $\beta_1, \cdots, \beta_p$. The main idea of the algorithm is to throw out one predictor whose coefficient is the most insignificant after some iterations of the Gibbs sampler till all the remaining predictors' coefficients are significant based on a specific Bayesian credible interval (BCI). The algorithm can be fulfilled through the following steps:

1. Run $N_1$ iterations of the Gibbs sampler, where $N_1$ is a large number, say $10,000$. Note that the $N_1$ iterations are just burn-in and we found it neccessary especially when $p$ is large.

2. Run another $M$ iterations where $M$ is relatively small, like $1,000$. For a specific $p^*$, for example, $0.05$, if the $(1-p^*)$ Bayesian credible interval based on the last $M$ iterations of all the $p$ $\beta$'s exclude zero, then continue running the Gibbs sampler till convergence. Otherwise, go to Step 3.

3. Define

$$k_{drop} = \underset{k}{\operatorname{argmax}} \left\{ min( \sum_{g=N_1+1}^{N_1+M} I(\beta_k^{(g)} > 0), \sum_{g=N_1+1}^{N_1+M} I(\beta_k^{(g)} < 0)) \right\}$$

   where $\beta_k^{(g)}$ is the sample value of $\beta_k$ in the $g$th iteration of the Gibbs sampler for $k = 1, \cdots, p$. Throw out $X_{k_{drop}}$. That is to say, we throw out the $\beta$ that is most centered to $0$ based on the last $M$ iterations of the Gibbs sampler.

4. Replace $N_1$ by $N_1 + M$ and $p$ by $p - 1$. Let $k = 1, \cdots, k_{drop} - 1, k_{drop} + 1, \cdots, p$ and go to step 2.

This algorithm will select very few predictors if $X_i$'s are highly correlated, even though there are a large number of predictors. We can define the number of predictors to be selected in the algorithm rather than defining $p^*$. This may result in some of the selected predictors' coefficient of insignificance.

## 3 Linear regression

We applied our algorithm to a linear regression problem on the diabetes data [3], which has $n = 442$ and $p = 10$. The predictors are $10$ baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of $442$ diabetes patients, and the response of interest is a quantitative measure of disease progression one year after baseline.

We performed a Bayesian Lasso with the predictor selection algorithm described in Section 2, setting $p^* = 0.05$ and placing a hyperprior on $\lambda$ to estimate it simultaneously. Then we perform the Bayesian Lasso without selecting predictors and the ordinary Lasso both of which have the tuning parameter matching the value of $\lambda$ estimated by the algorithm. The results are summarized in Table 1 along with the ordinary Lasso whose $\lambda$ is chosen by $n$-fold cross-validation. Comparing the Bayesian Lasso using the algorithm with the one not using the algorithm, the algorithm threw out all predictors that have an insignificant coefficient based on the $95\%$ credible interval, except one predictor, (7) hdl. hdl has a marginally significant coefficient in the Bayesian Lasso without the algorithm, but has a highly significant coefficient if using the algorithm. If we compare the Bayesian Lasso with the algorithm and the ordinary Lasso with $\lambda$ matched, we find the algorithm tends to select fewer predictors: (5) tc and (10) glu are non-zero in the ordinary Lasso but are thrown out by the algorithm. If we look at the last two columns, we find the Lasso with $\lambda$ matched shinks $\boldsymbol{\beta}$ more than that by the cross-validation.

Furthermore, we performed the Bayesian Lasso without the predictor selection algorithm and estimate $\lambda$ simultaneously by placing a hyperprior. We also fit the model by the Bayesian Lasso using the algorithm and the ordinary Lasso

| predictor | Bayesian Lasso (predictor selection) | $95\%$ BCI | Bayesian Lasso | $95\%$ BCI | Lasso ($\lambda$ matched) | Lasso ($n$-fold c.v.) |
|---|---|---|---|---|---|---|
| (1) age | 0 | NA | -3.41 | (-107.68, 104.08) | 0 | 0 |
| (2) sex | -206.18 | (-325.21, -89.66) | -204.32 | (-321.71, -91.80) | -178.92 | -195.13 |
| (3) bmi | 520.06 | (396.68, 653.10) | 519.28 | (391.97, 649.30) | 520.02 | 521.95 |
| (4) map | 313.71 | (181.85, 437.34) | 303.78 | (176.65, 428.40) | 287.35 | 295.79 |
| (5) tc | 0 | NA | -146.16 | (-552.47, 135.28) | -81.13 | -100.76 |
| (6) ldl | 0 | NA | -11.44 | (-269.57, 312.16) | 0 | 0 |
| (7) hdl | -267.55 | (-392.72, -134.28) | -160.04 | (-378.52, 64.75) | -217.80 | -223.07 |
| (8) tch | 0 | NA | 81.12 | (-120.75, 359.53) | 0 | 0 |
| (9) ltg | 472.74 | (335.30, 598.74) | 512.75 | (325.79, 720.73) | 501.06 | 512.84 |
| (10) glu | 0 | NA | 59.88 | (-52.77, 195.31) | 45.40 | 53.46 |

Table 1: Comparison of linear regression parameters for the diabetes data using the $\lambda$ estimated by the Bayesian lasso with the predictor selection algorithm.

both of which match the tuning parameter. The results are given in Table 2. We found the Bayesian Lasso using the algorithm selects 5 predictors while the ordinary Lasso selects 7. From the 5th and 10th rows in both Table 1 and 2 we can see, given a tuning paramter, for a predictor that the Bayesian Lasso using the algorithm throws out whereas the ordinary Lasso selects, the estimate of its coefficent by the ordinary Lasso always has a larger absolute value than by the Bayesian Lasso without the algorithm.

| predictor | Bayesian Lasso (predictor selection) | $95\%$ BCI | Bayesian Lasso | $95\%$ BCI | Lasso ($\lambda$ matched) |
|---|---|---|---|---|---|
| (1) age | 0 | NA | -3.30 | (-110.02, 109.24) | 0 |
| (2) sex | -216.05 | (-333.69, -98.77) | -218.93 | (-340.17, -87.23) | -195.39 |
| (3) bmi | 521.11 | (396.80, 651.19) | 523.03 | (395.94, 652.81) | 521.99 |
| (4) map | 316.00 | (186.78, 447.18) | 309.39 | (183.42, 439.15) | 295.92 |
| (5) tc | 0 | NA | -199.04 | (-666.78, 137.80) | -101.08 |
| (6) ldl | 0 | NA | 10.51 | (-290.87, 418.43) | 0 |
| (7) hdl | -274.46 | (-400.46, -151.25) | -144.30 | (-387.20, 92.73) | -223.16 |
| (8) tch | 0 | NA | 92.90 | (-135.19, 340.37) | 0 |
| (9) ltg | 472.56 | (349.14, 593.77) | 534.82 | (335.42, 749.72) | 513.03 |
| (10) glu | 0 | NA | 63.30 | (-49.00, 196.43) | 53.59 |

Table 2: Comparison of linear regression parameters for the diabetes data using the $\lambda$ estimated by the Bayesian Lasso without predictor selection.

## 4 Penalized Spline using $l_1$ penalty

The Bayesian Lasso with the predictor selection algorithm can be applied to penalized spline models using $l_1$ penalty so that we can fit the model and select knots from a set of candidates simultaneously. The model is constructed as

$$\boldsymbol{y} = \mu \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\varepsilon} \tag{3}$$

where $\boldsymbol{Z}$ is the $n \times K$ standardized design matrix, $\boldsymbol{u} = (u_1, \cdots, u_K)'$ and other notations are defined as in Section 1. Then $(\boldsymbol{\beta}, \boldsymbol{u})$ are estimated as

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{u}}) = \underset{(\boldsymbol{\beta}, \boldsymbol{u})}{\operatorname{argmin}} (\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u})'(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u}) + \lambda \sum_{i=1}^{K} |u_i|$$

where $\tilde{\boldsymbol{y}}$ and $\lambda$ has the same meaning as in Section 1.

### 4.1 Bayesian hierarchical model and Gibbs sampler implementation

Analougous to Bayesian Lasso for linear regression model [2], the hierarchical representation of the full model is

$$\begin{aligned}
\tilde{\boldsymbol{y}} | \boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{u}, \sigma^2 &\sim N_n(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \sigma^2 \boldsymbol{I}_n) \\
\boldsymbol{\beta} &\sim \pi(\boldsymbol{\beta}) d\boldsymbol{\beta}, \quad \pi(\beta) \propto 1 \\
\boldsymbol{u} | \tau_1^2, \cdots, \tau_K^2, \lambda, \sigma^2 &\sim N_K(\boldsymbol{0}_K, \sigma^2 D_\tau), \quad D_{\boldsymbol{\tau}} = \mathsf{diag}(\tau_1^2, \cdots, \tau_K^2) \\
\tau_1^2, \cdots, \tau_K^2 | \lambda &\sim Gamma(1, \frac{\lambda^2}{2}) \\
\lambda^2 &\sim Gamma(r, \delta) \\
\sigma^2 &\sim \pi(\sigma^2) d\sigma^2, \quad \pi(\sigma^2) \propto \frac{1}{\sigma^2}
\end{aligned}$$

with $\boldsymbol{\tau}$ and $\sigma^2$ independent. According to (2), integrating out $\boldsymbol{\tau}$, the conditional prior on $\boldsymbol{u}$ will be

$$\pi(\boldsymbol{u} | \lambda, \sigma^2) = \prod_{i=1}^{K} \frac{\lambda}{2\sigma} e^{-\lambda |u_i| / \sigma}. \tag{4}$$

Park and Casella [2] explained in detail why the prior in (4) is preferred to the one in (1). As for the Gamma prior on $\lambda^2$, we need the prior variance of $\lambda$ large enough to make the prior relatively flat. So set $r = 1$ and $\delta = 0.1$ so that the prior mean of $\lambda^2$ is much larger than $\hat{\lambda}$ ($\hat{\lambda}$'s are less than 1 throughout all models in this paper). The full conditional distributions of the parameters are

$$\begin{aligned}
\boldsymbol{\beta} | \cdots &\sim N_p \left( (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}'(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{u}), \sigma^2 (\boldsymbol{X}'\boldsymbol{X})^{-1} \right) \\
\boldsymbol{u} | \cdots &\sim N_K \left( (\boldsymbol{Z}'\boldsymbol{Z} + D_\tau^{-1})^{-1} \boldsymbol{Z}'(\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta}), \sigma^2 (\boldsymbol{Z}'\boldsymbol{Z} + D_\tau^{-1})^{-1} \right) \\
\sigma^2 | \cdots &\sim InvGamma \left( \frac{p + n - 1}{2}, \frac{||\tilde{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z}\boldsymbol{u}||^2 + \boldsymbol{u}' D_\tau^{-1} \boldsymbol{u}}{2} \right) \\
\frac{1}{\tau_j} | \cdots &\sim \text{Inverse Gaussian with mean parameter as } \sqrt{\frac{\lambda^2 \sigma^2}{u_j^2}} \text{ and scale parameter as } \lambda^2, \ j = 1, \cdots, K \\
\lambda^2 | \cdots &\sim Gamma \left( K + r, \frac{\sum_{j=1}^{K} \tau_j^2}{2} + \delta \right)
\end{aligned}$$

The Gibbs sampler samples parameters cyclically from their full conditionals above. We construct the hierarchical model using $\tilde{\boldsymbol{y}}$ instead of $\boldsymbol{y}$ since $\mu$ is often of secondary interest. But we make inference of $\mu$, we can still draw samples from its full conditional distribution, $N(\frac{1}{n}\sum_{i=1}^{n}, \frac{\sigma^2}{n})$, if each column of $\boldsymbol{X}$ and $\boldsymbol{Z}$ is centered.

## 4.2 Penalized spline model for the global temperature data using B-spline Basis

The Global temperature data [4] recorded $125$ years ($\boldsymbol{x}$) of global temperature ($\boldsymbol{Y}$) from year 1881 to 2005. The design matrix $\boldsymbol{X}$ is obtained by standardizing $(x, x^2, x^3)$ so that each column of $\boldsymbol{X}$ has mean zero and $l_2$ norm one. If $\boldsymbol{Z}$ is a polynomial basis matrix, its columns will be highly correlated. It is known that Lasso is not good at dealing with problems when the predictors are highly correlated [5], so we use a B-spline basis matrix generated by the `R` function `splineDesign`: set the option $order = 4$ and $knots$ as $23$ equally spaced points on the interval $[min(\boldsymbol{x})+2, max(\boldsymbol{x})-2]$ such that the function returns a design matrix with $21$ columns. We obtain $\boldsymbol{Z}$ by standardizing this matrix.

### 4.2.1 B-spline and truncated cubic spline using $l_2$ penalty

This section is to show how B-spline fits (3) using the $l_2$ penalty and compare it to the truncated cubic spline. Define

$$\boldsymbol{W} = \left((\boldsymbol{x} - \kappa_1)_+^3, \cdots, (\boldsymbol{x} - \kappa_{21})_+^3\right)$$

where the knots $\kappa_1, \cdots, \kappa_{21}$ are equally spaced points on the interval $[min(\boldsymbol{x}) + 2, max(\boldsymbol{x}) - 2]$. Then we obatin $\boldsymbol{Z}^*$ by standardizing $\boldsymbol{W}$. We use $\boldsymbol{Z}$ to fit (3) using $l_2$ penalty on $\boldsymbol{u}$ by the `R` function `lme`, and compare the fitted curve with the one using $\boldsymbol{Z}^*$. The curves are plotted in Figure 1 and we can see the curve fitted using B-spline are more bumpy.

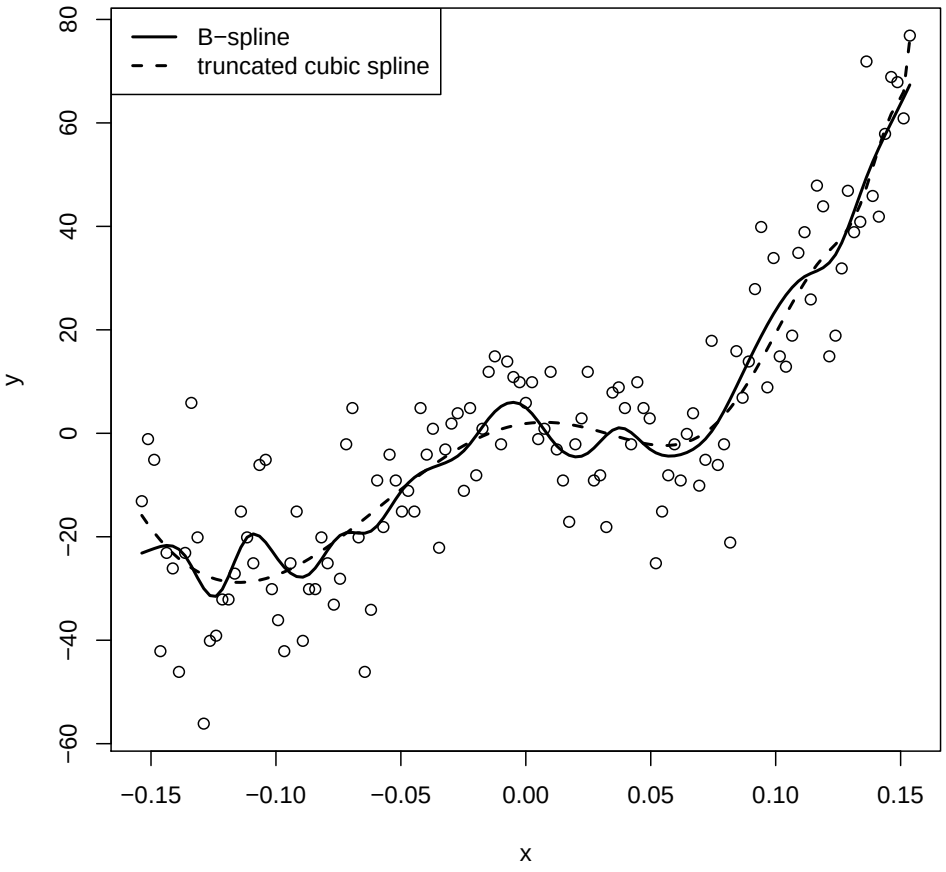


Figure 1: B-spline and truncated cubic spline using $l_2$ penalty.

### 4.2.2 B-spline knots selection

We can apply the predictor selection algorithm to the penalized spline model to select knots automatically. We use the design matrix $\boldsymbol{X}$ and the B-spline basis matrix $\boldsymbol{Z}$ using $l_1$ penalty to fit (3). Running the Gibbs sampler with the algorithm, $3$, $5$, $6$ and $6$ knots are selected corresponding to $p^*$ equal to $0.01$, $0.05$, $0.1$ and $0.2$, respectively. The fitted curves are plotted in Figure 2. We can see the smoothed curves do not fit very well; the curve tends to fit better as $p*$ increases. We can also specify the number of knots to select in the algorithm, rather then setting $p^*$. Figure 3 plotted

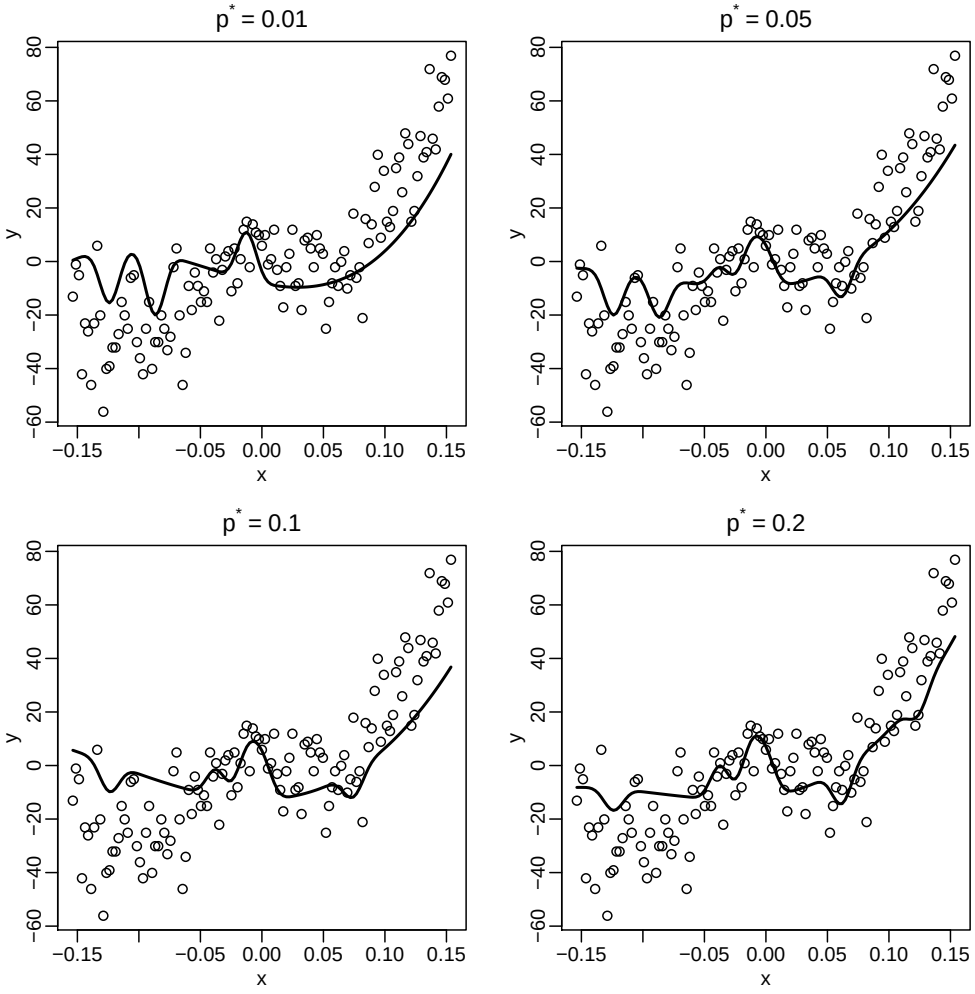


Figure 2: B-spline using $l_1$ penalty and knots selection for $p^* = 0.01, 0.05, 0.1, 0.2$.

the fitted curve with $10$ knots. It fits much better than all in Figure 2.

## 5 Discussion

Like ordinary Lasso, the Bayesian Lasso with the predictor selection algorithm does not select predictors stably; even if we run the Gibbs sampler several times with all the same settings, the results of predictor selection and parameter estimation may differ a lot especially when the predictors are highly correlated. Another problem of the algorithm is the computing speed. First, we need a burn-in of $N_1$ iterations and $N_1$ may be a very big number, say $100000$, when we initially have a lot of predictors. Furthermore, $M$ should also be large, say $10000$, because the coefficient of a predictor may change a lot after throwing out a correlated predictor.

When we stop throwing out predictors, we need to run the Gibbs sampler till convergence. In this paper, we just run a big number of iterations and use the trace plot to check convergence. The method of Monte Carlo standard errors

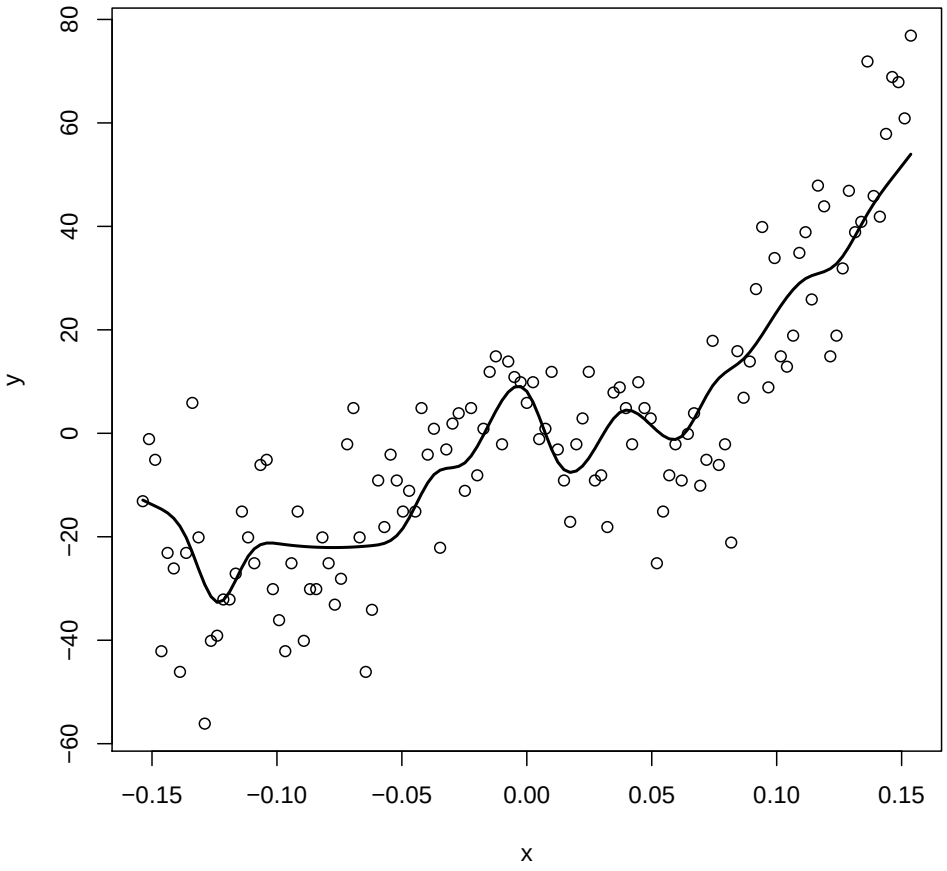


Figure 3: B-spline using $l_1$ penalty with $10$ knots.

can be used to determine when to stop the sampler [6].

## References


[1] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[2] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[3] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[4] James S Hodges. *Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects*. CRC Press, 2013.

[5] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[6] James M Flegal, Murali Haran, Galin L Jones, et al. Markov chain monte carlo: Can we trust the third significant figure? *Statistical Science*, 23(2):250–260, 2008.