Advanced Bayesian Computation Weeks 3 and 4

Rajarshi Guhaniyogi
Winter 2018

February 2, 2018

$$y_i \sim N(\mu_i, 1/\phi), \;\; i = 1, ..., n.$$

- $\boldsymbol{x}_1, ..., \boldsymbol{x}_p$ correspond to p columns each of length n.
- Let $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p) \in \{0, 1\}^p$.
- $\boldsymbol{\mu} = (\mu_1, ..., \mu_n)'$ and $\boldsymbol{X}_{\boldsymbol{\gamma}}$ is an $n \times p_{\boldsymbol{\gamma}}$ dimensional matrix that includes columns corresponding to $\gamma_i = 1$.
- $\mathscr{M}_{\boldsymbol{\gamma}} : \boldsymbol{\mu} = \mathbf{1}_n \alpha + \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{\boldsymbol{\gamma}}$.
- $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is $p_{\boldsymbol{\gamma}}$-dimensional.
- $\boldsymbol{\Theta}_{\boldsymbol{\gamma}} = \{\boldsymbol{\beta}_{\boldsymbol{\gamma}}, \alpha, \phi\}$.

g-Prior Contd..

- g-prior was another class of approach that has surfaced long back due to its computational ease.
- Let ϕ be the precision parameter. The formulations of g-prior is

$$\boldsymbol{\beta}_{\gamma}|\phi \sim N(\mathbf{0}, \frac{g}{\phi}(\boldsymbol{X}'_{\gamma}\boldsymbol{X}_{\gamma})^{-1}), \ \pi(\phi) \propto \frac{1}{\phi}$$

- Let $\mathscr{M}_b$ be any base model. Then

$$BF[\mathscr{M}_{\gamma} : \mathscr{M}_{\zeta}] = \frac{BF[\mathscr{M}_{\gamma} : \mathscr{M}_b]}{BF[\mathscr{M}_{\zeta} : \mathscr{M}_b]}$$

- The marginal likelihood is given by

$$\pi(\boldsymbol{y}|\mathscr{M}_{\gamma}) = \frac{\Gamma((n-1)/2)}{\sqrt{\pi}^{n-1}\sqrt{n}}||\boldsymbol{y} - \bar{\boldsymbol{y}}||^{-(n-1)}\frac{(1+g)^{(n-1-p_{\gamma})/2}}{[1+g(1-R^2_{\gamma})]^{(n-1)/2}}.$$

g-Prior Contd..

- When $\mathscr{M}_b$ is the null model, denoted by $\mathscr{M}_N$

$$BF[\mathscr{M}_\gamma : \mathscr{M}_N] = (1+g)^{\frac{n-p_\gamma-1}{2}}[1+g(1-R^2_\gamma)]^{-(n-1)/2}.$$

- When $\mathscr{M}_b$ is the full model, denoted by $\mathscr{M}_F$

$$BF[\mathscr{M}_\gamma : \mathscr{M}_F] = (1+g)^{\frac{-n+p+1}{2}}[1+g\frac{(1-R^2_F)}{(1-R^2_\gamma)}]^{(n-p_\gamma-1)/2}.$$

- R^2_γ is the R^2 statistics for the model $\mathscr{M}_\gamma$.
- How to choose g? Can a fixed g be used?

Unit information prior

- Kass and Wasserman (1995) recommended choosing priors with the amount of information about the parameter equal to the amount of information contained in one observation.
- For regular parametric families, the amount of information is defined through Fisher information.
- In the normal regression case, the unit information prior corresponds to taking $g = n$, leading to Bayes factors that behave like BIC.

Risk inflation criterion

- Foster and George (1994) calibrated priors for model selection based on the Risk Inflation Criterion (RIC).
- They recommended the use of $g = p^2$ from a minimax perspective.

Choice of g

Benchmark prior

- Fernandez et al. (2001) did a thorough study on various choices of g with dependence on the sample size n or the model dimension p.
- They concluded with the recommendation to take $g = maxi(n, p^2)$.
- We refer to their benchmark prior specification as BRIC as it bridges BIC and RIC.

Local empirical Bayes

- The local EB approach can be viewed as estimating a separate g for each model.
- Using the marginal likelihood after integrating out all parameters, an EB estimate of g is the maximum (marginal) likelihood estimate constrained to be non-negative.

$$\hat{g}_{\gamma}^{EBL} = max\{F_{\gamma} - 1, 0\}, \;\; F_{\gamma} = \frac{R_{\gamma}^2/p_{\gamma}}{(1 - R_{\gamma}^2)/(n - 1 - p_{\gamma})}.$$

Global empirical Bayes

- The global EB procedure assumes one common g for all models, and borrows strength from all models by estimating g from the marginal likelihood of the data, averaged over all models.

$$\hat{g}^{EPB} = arg\max_{g>0} \sum_{\gamma} p(\mathscr{M}_\gamma) \frac{(1+g)^{\frac{n-p_\gamma-1}{2}}}{[1+g(1-R^2_\gamma)]^{(n-1)/2}}$$

Issues with choosing a fixed g

- Barlett paradox and information paradox.
- Som et al. (2016) proved that there is an additional paradox known as the conditional Lindley paradox.

Mixture of g-priors

- Letting $\pi(g)$ as the prior on g, mixtures of g prior yields

$$BF[\mathscr{M}_\gamma : \mathscr{M}_N] = \int_0^\infty (1+g)^{(n-p_\gamma-1)/2}[1+(1-R^2_\gamma)]^{-(n-1)/2}\pi(g)d$$

  under the null based approach.
- Expressions for the full model based approach can also be obtained easily.
- Under selection of a model $\mathscr{M}_\gamma \neq \mathscr{M}_N$, the posterior mean of $\boldsymbol{\mu}$,

$$E[\boldsymbol{\mu}|\mathscr{M}_\gamma, \boldsymbol{y}] = \mathbf{1}_n\hat{\alpha} + E[g/(1+g)|\mathscr{M}_\gamma, \boldsymbol{y}]\boldsymbol{x}_\gamma\hat{\boldsymbol{\beta}}_\gamma,$$

  $\hat{\alpha}$ and $\hat{\boldsymbol{\beta}}_\gamma$ are ordinary least square estimates under model $\mathscr{M}_\gamma$.
- The optimal Bayes estimate of $\boldsymbol{\mu}$ under squared error loss is

$$E[\boldsymbol{\mu}|\boldsymbol{y}] = \mathbf{1}_n\hat{\alpha} + \sum_{\gamma:\mathscr{M}_\gamma\neq\mathscr{M}_N} p(\mathscr{M}_\gamma|\boldsymbol{y})E[g/(1+g)|\mathscr{M}_\gamma, \boldsymbol{y}]\boldsymbol{x}_\gamma\hat{\boldsymbol{\beta}}_\gamma.$$

Mixtures of g-priors

- *Zellner-Siow prior*:
  $\pi(g) = \sqrt{(n/2)}\Gamma(1/2)g^{-3/2}e^{-n/(2g)}, g > 0$
- *Hyper-g prior*: $\pi(g) = \frac{a-2}{2}(1+g)^{-a/2}, g > 0.$
- Though the integral w.r.t g might not be in closed form, it is a one dimensional integral that assumes easy approximation through Laplace or quadrature techniques.

Model selection consistency

$$P(\mathscr{M}_\gamma|\boldsymbol{y}) \xrightarrow{P} 1, \text{ when } \mathscr{M}_\gamma \text{ is the true model.}$$

- By the relationship between posterior probabilities and Bayes factor

  $$BF[\mathscr{M}_{\gamma'} : \mathscr{M}_\gamma] \xrightarrow{P} 0,$$

  when $\mathscr{M}_\gamma$ is the true model and $\mathscr{M}_{\gamma'}$ be any other model that does not contain the true model.
- Zellner-Siow and hyper-g priors are consistent when $\mathscr{M}_\gamma$ is not the null model.
- When $\mathscr{M}_\gamma = \mathscr{M}_N$, hyper-g prior is not consistent.
- Lack of consistency of the hyper-g prior in the null model motivated a new prior hyper-g/n

  $$\pi(g) = \frac{(a-2)}{2n}(1+g/n)^{-a/2}.$$

Solving paradoxes

- Mixture of g-priors solve the information paradox.

Block hyper-g prior

- Model: $\boldsymbol{y} = \alpha \mathbf{1}_n + \boldsymbol{X}_1 \beta_1 + \cdots + \boldsymbol{X}_k \beta_k + \boldsymbol{\epsilon}$
- β_s is p_s dimensional.
- Let $\boldsymbol{g} = (g_1, ..., g_k)$.
- $\beta \,|\, \boldsymbol{g}, \tau^2 \sim N(\mathbf{0}, \boldsymbol{A}\tau^2)$,
  $\boldsymbol{A} = diag(g_1(\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1}, ..., g_k(\boldsymbol{X}_k'\boldsymbol{X}_k)^{-1})$.
- Assign block hyper-g prior on $\boldsymbol{g}$,
  $\pi(\boldsymbol{g}) = \prod_{i=1}^{k} \frac{(a-2)}{2}(1+g_i)^{-a/2}$.

Random Compression: Background

- When objects to deal with are of huge size, it is difficult to work with them or even merely store and transfer them.
- In those cases, it is natural to extract a few features of the data.
- For example, think of the summary statistics (or sufficient statistics idea from AMS 205B).
- In the case of matrices, it is a popular practice to represent a matrix with only eigenvectors corresponding to large eigenvalues, sometimes called the low-rank decomposition of a matrix.
- PCA can be computationally demanding. Sometimes the computational complexity of PCA with N vectors can be as high as $O(N^3)$.

Random Compression: Background

Johnson-Lindenstrauss Lemma

Given $0 < \epsilon < 1$, a set $\boldsymbol{X}$ of m points in $\mathbb{R}^N$, and a number $n > 8\ln(m)/\epsilon^2$, there is a linear map $f : \mathbb{R}^N \to \mathbb{R}^n$ such that

$$(1-\epsilon)\|\boldsymbol{u}-\boldsymbol{v}\|^2 \leq \|f(\boldsymbol{u})-f(\boldsymbol{v})\|^2 \leq (1+\epsilon)\|\boldsymbol{u}-\boldsymbol{v}\|^2,$$

for all $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{X}$.

- If our statistical method is based on distance between the data points, we may work with $f(\boldsymbol{u})$'s rather than $\boldsymbol{u}$'s.
- Easy to store, easy to manipulate and transfer.
- **Question:** How to find such an $f(\cdot)$?

Choice of $f(\cdot)$ through random compression

- It is difficult to find a deterministic function that satisfies the JL-lemma.
- Construct a matrix $\mathbf{\Phi} = ((\Phi_{ij}))$ of dimension $n \times N$ s.t $\Phi_{ij} \sim N(0, 1/n)$.
- A second option is to draw $\Phi_{ij} \sim N(0, 1)$ and then make n rows of $\mathbf{\Phi}$ orthogonal.
- Achlioptas suggested using

$$\Phi_{ij} = \begin{cases} -\sqrt{3} & w.p.\ 1/6 \\ 0 & w.p.\ 2/3 \\ \sqrt{3} & w.p.\ 1/6 \end{cases}$$

- With any such choice, define $f(\boldsymbol{u}) = \mathbf{\Phi}\boldsymbol{u}$.

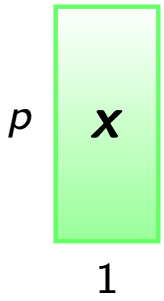
Usage of the Random Compression Matrix

- Suppose we have a $\boldsymbol{X}$ matrix of dimension $n \times p$, n is the sample size and p is the dimension of a sample.
- p is big, hence k-means clustering cannot be applied.
- Compress every row of $\boldsymbol{X}$ by a random compression matrix $\boldsymbol{\Phi}$.
- K-means clustering can the be safely applied on the compressed lower dimensional vectors.

Compressed Sensing

- Suppose $\boldsymbol{w} \in \mathscr{R}^N$ be compressed with $\boldsymbol{\Phi} \in \mathscr{R}^{n \times N}$ to $\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\epsilon}$.
- Can we recover $\boldsymbol{w}$ from $\boldsymbol{g}$?
- Yes, when $\boldsymbol{w}$ is sparse.
- Any vector $\boldsymbol{f}$ can be sparsified w.r.t. a basis function (e.g. wavelet).
- Mathematically, it is possible to find a matrix $\boldsymbol{B}$ s.t. $\boldsymbol{w} = \boldsymbol{B}'\boldsymbol{f}$ is sparse, though $\boldsymbol{f}$ is not.
- Possible to recover $\boldsymbol{w}$ through penalized optimization or Bayesian prior formulation.

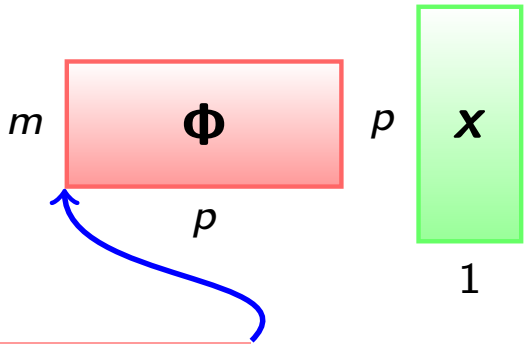
Can Compressing Predictors Help?



- $x_1, ..., x_n$ are of mammoth size.
- Storage is highly prohibitive, let alone computation.

Idea

Compressing predictors randomly in low dimension helps solve our problem.



- Random Projection Matrix

Random Projection Matrix: Lots of Zeroes Required in Φ

- Overwhelming literature on scaled Gaussian projection matrix ($\Phi_{ij} \sim N(0, 1/m)$)$\rightarrow$ **popular choice**.

Random Projection Matrix: Lots of Zeroes Required in Φ

- Overwhelming literature on scaled Gaussian projection matrix ($\Phi_{ij} \sim N(0, 1/m)$)$\rightarrow$ **popular choice**.
- We anticipate sparsity in the true predictor coefficients.

Random Projection Matrix: Lots of Zeroes Required in $\mathbf{\Phi}$

- Overwhelming literature on scaled Gaussian projection matrix ($\Phi_{ij} \sim N(0, 1/m)$)$\rightarrow$ **popular choice**.
- We anticipate sparsity in the true predictor coefficients.
- Might be important to have lots of zero entries in $\mathbf{\Phi}$.

Random Projection Matrix: Lots of Zeroes Required in $\mathbf{\Phi}$

- Overwhelming literature on scaled Gaussian projection matrix ($\Phi_{ij} \sim N(0, 1/m)$)$\rightarrow$ **popular choice**.
- We anticipate sparsity in the true predictor coefficients.
- Might be important to have lots of zero entries in $\mathbf{\Phi}$.

Random Projection (Dasgupta 2003, 2013)$\rightarrow$ Our Choice

$$\Phi_{ij} = \begin{cases} -1/\sqrt{\psi} & w.p.\ \psi^2 \\ 0 & w.p.\ 2\psi(1-\psi) \\ 1/\sqrt{\psi} & w.p.\ (1-\psi)^2 \end{cases}$$

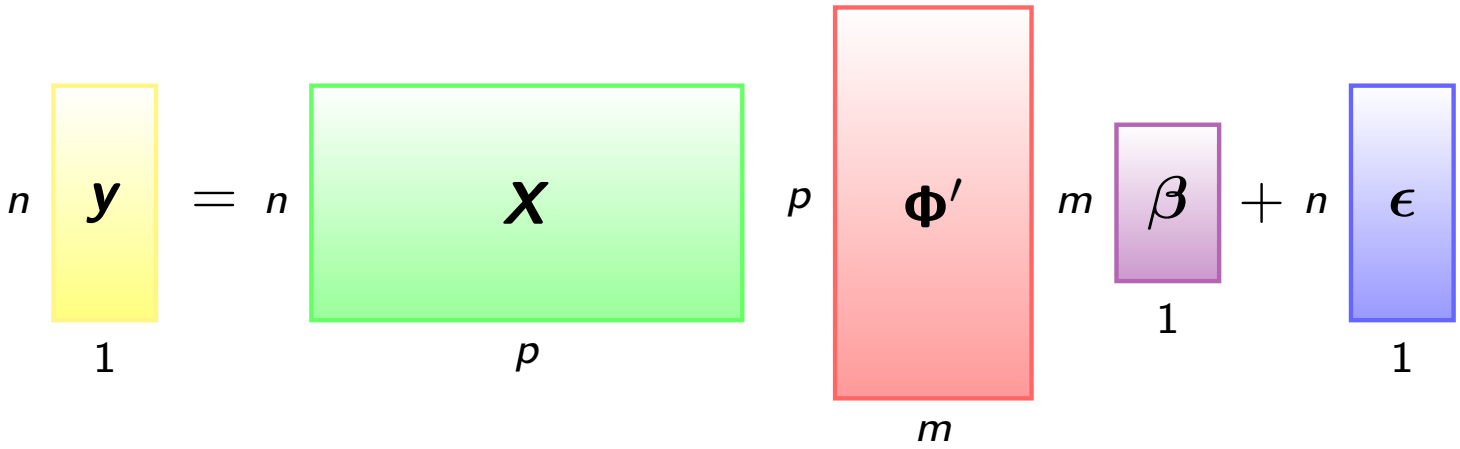
- Rows of $\mathbf{\Phi}$ are orthonormalized using Gram-Schmidt orthogonalization procedure.

Bayesian Compressed Regression

Compressed Regression

$$y = (\mathbf{\Phi} \boldsymbol{x})' \boldsymbol{\beta} + \epsilon, \ \epsilon \sim N(0, \sigma^2)$$

$\boldsymbol{\beta}$ is the low dimensional coefficients on compressed predictors.



- No longer in the high-dimensional setting, use conjugate prior.

Conjugate Prior

$$\boldsymbol{\beta} \,|\, \sigma^2 \sim N(\mathbf{0}, \sigma^2 \mathbf{\Sigma}_{\boldsymbol{\beta}}), \ \sigma^2 \sim IG(a, b).$$

Analytical Posteriors & Posterior Predictive Distributions

Posteriors

$$\beta \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi} \sim t_n\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \;\; \sigma^2 \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi} \sim IG(a_1, b_1)$$

Posterior Predictive Distribution

$$y_{n+1} \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi}, \boldsymbol{x}_{n+1} \sim t_n\left(\mu_{pred}, \sigma^2_{pred}\right)$$

- Only needs sufficient statistics $\boldsymbol{X'X}$, $\boldsymbol{X'y}$ and $\boldsymbol{y'y}$.
- Only matrix operation required $m \times m$ matrix inversion and $m \times p$, $p \times n$ matrix multiplication.

Analytical Posteriors & Posterior Predictive Distributions

Posteriors

$$\beta \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi} \sim t_n\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \;\; \sigma^2 \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi} \sim IG(a_1, b_1)$$

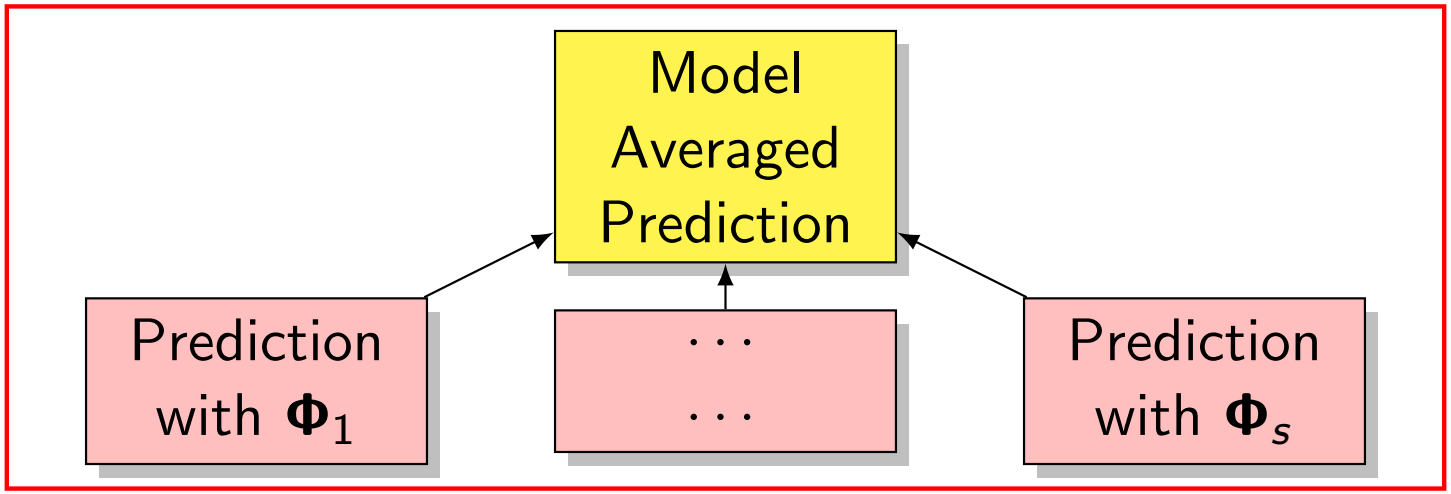
Posterior Predictive Distribution

$$y_{n+1} \,|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\Phi}, \boldsymbol{x}_{n+1} \sim t_n\left(\mu_{pred}, \sigma^2_{pred}\right)$$

- Only needs sufficient statistics $\boldsymbol{X'X}$, $\boldsymbol{X'y}$ and $\boldsymbol{y'y}$.
- Only matrix operation required $m \times m$ matrix inversion and $m \times p$, $p \times n$ matrix multiplication.

Bayesian Model Averaging Over Compression Matrix Dimension (m)

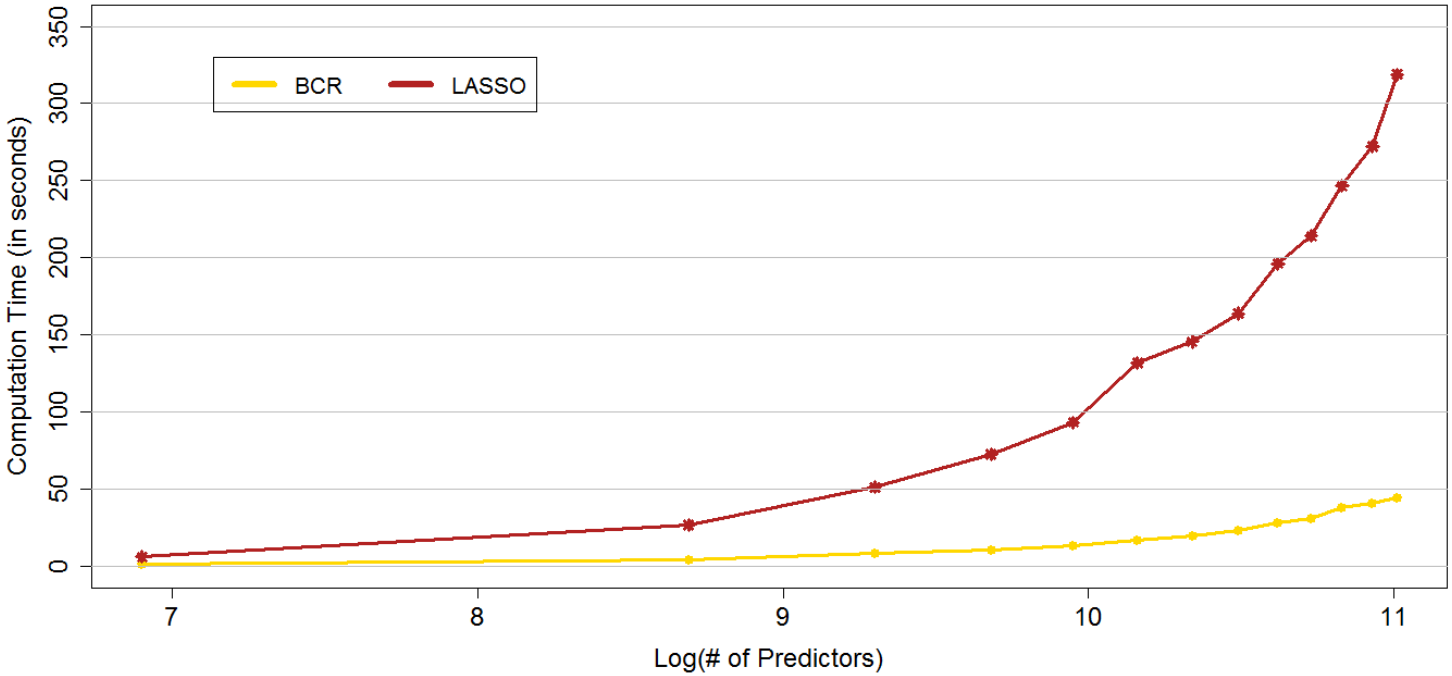
- To limit sensitivity to the randomly generated $\mathbf{\Phi}$ and choice of m, we use model averaging.



Model Weights are again functions of $\boldsymbol{X'X}$, $\boldsymbol{X'y}$ and $\boldsymbol{y'y}$ (sufficient statistics).

Computation Speeds: BCR (Guhaniyogi et al., 2016) Vs LASSO

Run time compared under similar predictive performance.



Random Compression for Frequentist Estimation with Big Data

- Computing least square estimator is time consuming with big data.
- However, one can compute the compressed least squares estimator.
- If $\boldsymbol{X}$ is $n \times p$ dimensional matrix and $\boldsymbol{y}$ is an n dimensional vector, $n >> p$.
- Rather than using the objective function $||\boldsymbol{y} - \boldsymbol{X}\beta||^2$, use $||\boldsymbol{\Phi}(\boldsymbol{y} - \boldsymbol{X}\beta)||^2$.
- The least squares solution is $(\boldsymbol{X}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{X})^{-1}(\boldsymbol{X}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{X})$.
- This was identified as not a very stable solution.
- Another estimator is $(\boldsymbol{X}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$.
- This comes from minimizing the objective function $\frac{1}{2}||\Phi\boldsymbol{X}\beta||^2 - \boldsymbol{y}'\boldsymbol{X}\beta$.
- This is called partial least squares solution and is much more stable.

High Dimensional Binary Regression

- $y_i \sim Ber(p_i)$, $i = 1, ..., n$.
- Use logistic link to model $p_i = \frac{\exp(\mu + \boldsymbol{x}_i'\boldsymbol{\beta})}{1+\exp(\mu+\boldsymbol{x}_i'\boldsymbol{\beta})}$.
- $\boldsymbol{x}_i$ is $p \times 1$ dimensional.
- $l_n(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{\exp(y_i\mu + y_i\boldsymbol{x}_i'\boldsymbol{\beta})}{1+\exp(\mu+\boldsymbol{x}_i'\boldsymbol{\beta})}$.
- Penalized optimization: $arg \min_{\boldsymbol{\beta}}[\log(l_n(\boldsymbol{\beta})) + \lambda \sum_{j=1}^{p} |\beta_j|]$.
- Bayesians would instead propose prior distribution on $\boldsymbol{\beta}$.

High dimensional Regression for Generalized Linear Model (Jiang, 2007)

- GLM with one natural parameter is given by the density

$$p(y|\boldsymbol{x}) = \exp[a(\boldsymbol{x}'\boldsymbol{\beta})y + b(\boldsymbol{x}'\boldsymbol{\beta}) + c(y)].$$

- $\mu = E[y|\boldsymbol{x}] = -b'(\boldsymbol{x}'\boldsymbol{\beta})/a'(\boldsymbol{x}'\boldsymbol{\beta})$.
- This formalism includes regression models for responses that are binary, Poisson and Gaussian (with known error variance), and can be easily extended to the cases with a dispersion parameter.
- $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)'$ is the vector of inclusion indicators.
- Let $P(\gamma_j = 1) = \lambda$, $\boldsymbol{\beta}_{\boldsymbol{\gamma}} \sim N(\mathbf{0}, \boldsymbol{V}_{\boldsymbol{\gamma}})$.

- $\boldsymbol{U}$ is a matrix having independent standard normal entries.
- $\boldsymbol{M} + \mathcal{N}(\boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ will stand for a matrix variate normal distribution of $\boldsymbol{V} = \boldsymbol{M} + \boldsymbol{A}'\boldsymbol{U}\boldsymbol{B}$, where $\boldsymbol{M}$, $\boldsymbol{A}$, $\boldsymbol{B}$ are fixed matrices satisfying $\boldsymbol{A}'\boldsymbol{A} = \boldsymbol{\Gamma}$, and $\boldsymbol{B}'\boldsymbol{B} = \boldsymbol{\Sigma}$.
- Thus $\boldsymbol{M}$ is the matrix mean of $\boldsymbol{V}$.
- $\gamma_{ii}\boldsymbol{\Sigma}$ and $\sigma_{jj}\boldsymbol{\Gamma}$ are the covariance matrices of the ith row and jth column respectively of $\boldsymbol{V}$.
- If $\boldsymbol{U}$ is of the order $n \times p$, $n \geq p$, the notation $IW(\delta, \boldsymbol{\Sigma})$ with $\delta = n - p + 1$ will stand for the distribution of $\boldsymbol{B}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{B}$, an inverse Wishart distribution.

The Model

$$\boldsymbol{y} - \mathbf{1}\boldsymbol{\alpha}' - \boldsymbol{X}\boldsymbol{B} \sim \mathcal{N}(\boldsymbol{I}_n, \boldsymbol{\Sigma})$$

- $\boldsymbol{y}$ is an $n \times q$ random response matrix.
- $\boldsymbol{X}$ is the predictor matrix of dimension $n \times p$.
- $\boldsymbol{B}$ is a $p \times q$ matrix of predictor coefficients.

Prior

$$\boldsymbol{\alpha}' - \boldsymbol{\alpha}_0' \sim N(\boldsymbol{h}, \boldsymbol{\Sigma}), \boldsymbol{B}_{\boldsymbol{\gamma}} - \boldsymbol{B}_{0\boldsymbol{\gamma}} \sim \mathcal{N}(\boldsymbol{H}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma})$$

- $\boldsymbol{H}_{\boldsymbol{\gamma}} = \boldsymbol{D}_{\boldsymbol{\gamma}} \boldsymbol{R}_{\boldsymbol{\gamma}} \boldsymbol{D}_{\boldsymbol{\gamma}}$.
- $\boldsymbol{D}_{\boldsymbol{\gamma}}$ is a diagonal matrix with jth diagonal entry v_{0j} if $\gamma_j = 0$ and v_{1j} if $\gamma_j = 1$.
- $\boldsymbol{R}_{\boldsymbol{\gamma}}$ is a correlation matrix.
- $\gamma_j \sim Ber(w_j)$, $\boldsymbol{\Sigma} \sim IW(\delta, \boldsymbol{Q})$.

Classification and Regression Trees

- What happens when the relationship between the response and the predictors is not linear?
- The idea is to split the predictor space into a large number of sub-domains.
- Let $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$ be the training sample with $y_i \in \mathbb{R}$ and $\boldsymbol{x}_i \in \mathbb{R}^p$.
- The $\mathbb{R}^p$ dimensional predictor space is split into sub-domains with binary splitting.
- When prediction of a response at the predictor value $\boldsymbol{z} \in \mathbb{R}^p$ needs to be done, first we recognize in which subdomain c the predictor value belongs to.
- Next, predict the response at $\boldsymbol{z}$ by $\frac{1}{n_c} \sum_{i \in c} y_i$, i.e. by the average of the responses corresponding to the predictors residing in that sub-domain.
- In the case of classification, we decide with majority votes.

Classification and Regression Trees Significance

- One of the most comprehensible non-parametric methods used to be the k-nearest-neighbors: find the points which are most similar to you, and do what, on average, they do.
- **Drawback 1:** First, you are defining "similar" in terms of the inputs, not the response.
- **Drawback 2:** Second, k is constant everywhere, when some points just might have more very-similar neighbors than others.
- Trees get around both problems.
- Leaves correspond to regions of the input space (a neighborhood), but one where the responses are similar, as well as the inputs being nearby.
- Additionally, their size can vary arbitrarily.
- Prediction trees are adaptive nearest-neighbor methods.

Constructing the trees

- Every recursive algorithm needs to know when it is done, a stopping criterion.
- Here this means when to stop trying to split nodes.
- Obviously nodes which contain only one data point cannot be split, but giving each observations its own leaf is unlikely to generalize well.
- A more typical criterion is something like, halt when each child would contain less than five data points, or when splitting increases the information by less than some threshold. Picking the criterion is important to get a good tree.
- The sum of squared errors for a tree $\mathscr{T}$ is

$$S = \sum_{c \in leaves(\mathscr{T})} \sum_{i \in c} (y_i - m_c)^2,$$

  where $m_c = \frac{1}{n_c} \sum_{i \in c} y_i$, the prediction for leaf c.
- We will minimize S while constructing a tree.

Constructing a regression tree

The basic regression-tree-growing algorithm then is as follows

- Start with a single node containing all points. Calculate m_c and S.
- If all the points in the node have the same value for all the input variables, stop. Otherwise, search over all binary splits of all variables for the one which will reduce S as much as possible.
- If the largest decrease in S would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop. Otherwise, take that split, creating two new nodes.
- In each new node, go back to step 1.
- The choice of q, δ are arbitrary.
- In a more sophisticated algorithm, we use pruning of the trees based on the test sample.

Bagging (Breiman, 1994)

- The test and training data are $\mathscr{T}$ and $\mathscr{L}$.
- A classification (or regression tree) is constructed using the training data $\mathscr{L}$ and prediction is done on $\mathscr{T}$ based on the constructed tree.
- Bootstrap sample $\mathscr{L}_B$ is selected from $\mathscr{L}$. This is used to construct the tree and carry out prediction of $\mathscr{T}$.
- This is repeated multiple times.
- Average prediction is provided over all the bootstrap samples.

Random Forest

- CART may provide inaccurate prediction when there is noise in the response or predictors.
- Rather than constructing one single tree, random forest constructs a large number of trees, i.e. a forest.
- The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners.
- Let the training predictors be $\mathscr{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ with the corresponding responses $\mathscr{Y} = \{y_1, ..., y_n\}$.

`For` $b = 1, ..., B$:

- Sample, with replacement, n training samples $\mathscr{X}_b, \mathscr{Y}_b$ from $\mathscr{X}, \mathscr{Y}$.
- Train a classification or regression tree h_b on $\mathscr{X}_b, \mathscr{Y}_b$.
- After training, predictions for unseen samples $\boldsymbol{z}$ can be made by averaging the predictions from all the individual regression trees on $\boldsymbol{z}$: $\hat{h}(\boldsymbol{z}) = \frac{1}{B} \sum_{b=1}^{B} h_b(\boldsymbol{z})$.

Random Forest

- For classification, we take the majority vote.
- The construction of tree is little different in random forest compared to CART.
- In random forest, in each tree, we only allow $m << p$ randomly selected predictors to undergo binary split.
- As shown by Breiman, random forest decreases the variance of the model, without increasing the bias.
- Thus, while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not.
- Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

Bayesian CART

- Place CART within a Bayesian framework by specifying a prior on tree space.
- Can get multiple tree realizations by using tree-changing proposal distribution: birth/death/change/swap.
- Get multiple realizations of 1 tree, average over posterior to form predictions.
- A regression tree models this data as

$$y = g(x, T, M) + \epsilon,$$

  where $g(x, T, M)$ represents the regression tree.
- Bayesian Framework

$$\pi(T, M) = \pi(M|T)\pi(T).$$

Bayesian CART

- Assuming that there are b terminal nodes for a tree T, $M = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_b\}$ is the set of parameters corresponding to the terminal nodes of the tree T.
- $g(x, T, M)$ is $\boldsymbol{\theta}_j$ if x belongs to the jth terminal node of the tree.
- Let $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_b)'$ be the set of responses in the b terminal nodes of the tree.
- $\boldsymbol{y}_j = (y_{j1}, ..., y_{jn_j})'$ are the observations in the jth terminal node.
- $p(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{X}, T) = \prod_{i=1}^{b} \prod_{j=1}^{n_i} f(y_{ij}|\boldsymbol{\theta}_i)$
- $f(y_{ij}|\boldsymbol{\theta}_i) = N(y_{ij}|\mu_i, \sigma_i^2)$, $\boldsymbol{\theta}_i = (\mu_i, \sigma_i^2)$ for regression.
- When y_{ij} belongs to one of the K categories $C_1, ..., C_K$, $f(y_{ij}|\boldsymbol{\theta}_i) = \prod_{k=1}^{K} p_{ik}^{I(y_{ij} \in C_k)}$, where $\boldsymbol{\theta}_i = (p_{i1}, ..., p_{iK})'$.

Parameter Prior

- $\mu_i|\sigma_i^2, T \sim N(\bar{\mu}, \sigma_i^2/a)$, $\sigma_i^2|T \sim IG(\nu/2, \nu\lambda/2)$.
- Integrating over the parameters $p(\boldsymbol{y}|\boldsymbol{X}, T)$ comes in closed form.
- In the case of classification, $(p_{i1}, ..., p_{iK})|T \sim Dir(\alpha_1, ..., \alpha_K)$.
- Again, in the case of classification $p(\boldsymbol{y}|\boldsymbol{X}, T)$ comes in closed form.

Tree prior

- Instead of directly specifying a prior on T, we specify $p(T)$ implicitly by a tree-generating stochastic process.
- Each realization of such a process can simply be considered as a random draw from this prior.
- The generating process is determined by the specification of two functions, $p_{SPLIT}(\eta, T)$ and $p_{RULE}(\rho|\eta, T)$.
- $p_{SPLIT}(\eta, T)$ is the probability that terminal node η is split and $p_{RULE}(\rho|\eta, T)$ is the probability of assigning splitting rule ρ to η given that η will undergo split.

Tree prior

The stochastic process for drawing a tree from this prior can be described in the following recursive manner:

- Begin by setting T to be the trivial tree consisting of a single root (and terminal) node denoted η.
- Split the terminal node η with probability $p_{SPLIT}(\eta, T)$.
- If the node splits, assign it a splitting rule ρ according to the distribution $p_{RULE}(\rho|\eta, T)$, and create the left and right children nodes. Let T denote the newly created tree, and apply steps 2 and 3 with T equal to the new left and the right children (if nontrivial splitting rules are available).

Choice of $p_{SPLIT}(\eta, T)$ and $p_{RULE}(\rho|\eta, T)$

- $p_{SPLIT}(\eta, T) = \alpha < 1$. Setting α small will tend to yield smaller trees and is a simple convenient way to control the size of trees generated by growing process.
- It is somewhat limited because it assigns equal probability to all binary trees with b terminal nodes regardless of their shape.
- $p_{SPLIT}(\eta, T) = \alpha(1 + d_\eta)^{-\beta}$, where $\alpha \in (0, 1)$, $\beta \in [0, \infty)$, d_η is the depth of the node η.
- $p_{RULE}(\rho|\eta, T)$ is set by first randomly picking one of the available predictors (say x_i) which can undergo split. Then choose a cut point s uniformly from the available observed values of x_i if x_i is quantitative, or choose C uniformly from the set of available subsets if x_i is qualitative.
- This is called the *uniform splitting rule*.
- There are non-uniform choices as well.

Stochastic Search of the Posterior

- Once you integrate out the parameters, $p(T|\boldsymbol{y}, \boldsymbol{X}) \propto p(\boldsymbol{y}|T, \boldsymbol{X})p(T)$.
- Thus we will run MCMC to create a sequence of trees $T^0, T^1, ...$
- MH algorithm will be used.
- Step 1: Generate a candidate tree T^* with probability distribution $q(T^i, T^*)$.
- Step 2: Set $T^{i+1} = T^*$ with probability

$$\alpha(T^i, T^*) = min\left\{\frac{q(T^*, T^i)p(\boldsymbol{y}|T^*, \boldsymbol{X})p(T^*)}{q(T^i, T^*)p(\boldsymbol{y}|T^i, \boldsymbol{X})p(T^i)}, 1\right\}$$

Otherwise set $T^{i+1} = T^i$.

Generating candidate trees

- *GROW*: Randomly pick a terminal node. Split it into two new ones by randomly assigning it a splitting rule according to p_{RULE} used in the prior.
- *PRUNE*: Randomly pick a parent of two terminal nodes and turn it into a terminal node by collapsing the nodes below it.
- *CHANGE*: Randomly pick an internal node, and randomly reassign it a splitting rule according to p_{RULE} used in the prior.
- *SWAP*: Randomly pick a parent-child pair which are both internal nodes. Swap their splitting rules.

Bayesian Additive Regression Tree

- Let T denote a binary tree consisting of a set of interior node decision rules and a set of terminal nodes.
- $M = \{\mu_1, ..., \mu_b\}$ denote a set of parameter values associated with each of the b terminal nodes of T.
- For a given T and M, we use $g(x, T, M)$ to denote the function which assigns $\mu_i \in M$ to x.
- Thus a single tree model looks like

$$y = g(x, T, M) + \epsilon.$$

- For the sum of tree model

$$y = \sum_{j=1}^{m} g(x, T_j, M_j) + \epsilon, \;\; \epsilon \sim N(0, \sigma^2)$$

- $\mu_{ij} \in M_j$ are the terminal node specific parameter.

Prior on parameters

- Assign independent prior on trees.

$$p((T_1, M_1), ..., (T_m, M_m), \sigma) = p(\sigma) \prod_{j=1}^{m} p(T_j, M_j)$$
$$= p(\sigma) \prod_{j=1}^{m} p(M_j | T_j) p(T_j)$$

- $p(M_j | T_j) = \prod_i p(\mu_{ij} | T_j)$, where $\mu_{ij} \in M_j$.
- Under such priors, the tree components (T_j, M_j) are independent of each other and of σ, and the terminal node parameters of every tree are independent.
- To simplify further, assume identical prior forms for all $p(T_j)$ and also identical forms for all $p(\mu_{ij} | T_j)$.

Prior on $p(\mu_{ij}|T_j)$

- $p(\mu_{ij}|T_j) = N(\mu_{ij}|\mu_\mu, \sigma^2_\mu)$.
- The induced prior on $E[y|x]$ is $N(m\mu_\mu, m\sigma^2_\mu)$.
- It is highly probable that $E[y|x]$ is between y_{min} and y_{max}, the observed minimum and maximum of response in the data.
- This can be conveniently done by choosing μ_μ and σ_μ so that $m\mu_\mu - k\sqrt{m}\sigma_\mu = y_{min}$, $m\mu_\mu + k\sqrt{m}\sigma_\mu = y_{max}$. One takes $k = 2$.
- $\sigma^2 \sim \nu\lambda/\chi^2_\nu$.
- Two natural choices for $\hat{\sigma}$ are a) the "naive" specification, in which we take $\hat{\sigma}$ to be the sample standard deviation of y, or b) the "linear model" specification, in which we take $\hat{\sigma}$ as the residual standard deviation from a least squares linear regression of y on the original x's.
- Pick a value of ν between 3 and 10 to get an appropriate shape, and a value of λ so that the qth quantile of the prior on σ is located at $\hat{\sigma}$, i.e. $P(\sigma < \hat{\sigma}) = q$.
- One uses $q = 0.75, 0.90, 0.99$.

Prior on T_j

- The probability that a node at depth d is nonterminal, given by $\alpha(1+d)^{-\beta}$, where $\alpha \in (0,1)$, $\beta \in [0,\infty)$.
- The distribution on the splitting variable assignments at each interior node is given by the uniform prior on available variables.
- The distribution on the splitting rule assignment in each interior node, conditional on splitting variable is the uniform prior on the discrete set of available splitting values.
- Depending on α and β, we put apriori belief of how deep the tree will be.
- For $\alpha = 0.95$, $\beta = 2$, only trees with upto 5 terminal nodes have non-negligible probability.

Choice of m

- One can put prior on m and implement a full Bayes implementation of BART.
- Alternatively, one can choose the "best" m by some cross validation technique.
- Both are computationally cumbersome, hence m is kept fixed with a default value of 200.

Model Fitting

- Model fitting is done through MCMC.
- Need to update $(T_j, M_j)|(T_{(j)}, M_{(j)}), \sigma$ for $j = 1, ..., m$ and $\sigma|\, T_1, ..., T_m, M_1, ..., M_m$.
- $T_{(j)} = \{T_1, ..., T_{j-1}, T_{j+1}, ..., T_m\}$ and $M_{(j)} = \{M_1, ..., M_{j-1}, M_{j+1}, ..., M_m\}$.
- Again it will involve reversible jump, hence the inference will be done after marginalizing over M_js.
- Note that the full conditional distribution of σ^2 follows an Inverse Gamma distribution.

Model Fitting

- Observe that the conditional distribution $(T_j, M_j)|(T_{(j)}, M_{(j)}), \sigma, \boldsymbol{y}$ depends on $(T_{(j)}, M_{(j)}), \boldsymbol{y}$ with $R_j = y - \sum_{k \neq j} g(x, T_k, M_k)$.
- $(T_j, M_j)|(T_{(j)}, M_{(j)}), \sigma, \boldsymbol{y}$ is equivalent to $(T_j, M_j)|R_j, \sigma$.
- Note that $R_j = g(x, T_j, M_j) + \epsilon$, $\epsilon \sim N(0, \sigma^2)$.
- $p(T_j|R_j, \sigma) \propto p(T_j) \int p(R_j|M_j, T_j, \sigma) p(M_j|T_j, \sigma) dM_j$.
- Similar to the Bayesian CART, $\int p(R_j|M_j, T_j, \sigma) p(M_j|T_j, \sigma) dM_j$ comes in a closed form.
- The mixing is dramatically better compared to the Bayesian CART.

Prediction from MCMC Iterates

- Draw $M_j|R_j, T_j, \sigma$ from a normal distribution.
- Let $\{T_j^{*(l)}, M_j^{*(l)}, \sigma^{*(l)}\}_{j=1}^{L}$ be the post burn-in L MCMC samples.
- The posterior predictive samples of the response at $\boldsymbol{x}$ are $y_1^*, ..., y_L^*$, where $y_l^* \sim N(\sum_{j=1}^{m} g(x, T_j^{*(l)}, M_j^{*(l)}), \sigma^{2*(l)})$.

Modeling Nonlinear Relationship using Gaussian processes

- Gaussian process is a special type of stochastic process.
- It is one of the most widely used stochastic processes.
- A stochastic process is a collection of random variables $\{X_t\}_{t\in\mathscr{T}}$, where $\mathscr{T}$ is a subset of $[0,\infty)$.
- $\mathscr{T}$ can be a discrete set in which case the sequence of $\{X_t\}_{t\in\mathscr{T}}$ is countably infinite.
- Discrete stochastic processes include discrete Markov chains, birth-death processes and so on.
- When $\mathscr{T}$ is continuous, the sequence of $\{X_t\}_{t\in\mathscr{T}}$ is countably infinite.

More on Stochastic Processes

- Why stochastic processes are important?
- Stochastic processes are used to specify distribution of an unknown function.
- In the context of Bayesian statistic, it can be used as a prior distribution on unknown functions.
- For example, if y is the response and $\boldsymbol{x} \in \mathscr{R}^p$ is a p-dimensional predictor, one typically models $y = f(\boldsymbol{x}) + \epsilon$ to capture the non-linear relationship between $\boldsymbol{x}$ and y.
- $f(\cdot)$ is an unknown function which is modeled with a stochastic process.

Defining a Stochastic Process

- In contrast to the case of random vectors or random variables, it is not easy to define a notion of a density (or a probability mass function) for a stochastic process.
- Without going into details why exactly this is a problem, let me just mention that the main culprit is the infinity.
- One usually defines a family of *finite-dimensional distributions*, i.e., the joint distributions of random vectors $(X_{t_1}, ..., X_{t_n})$ for all $n \in \mathbb{N}$ and $t_1, ..., t_n \in \mathscr{T}$.
- One needs to ensure that these final dimensional distributions lead to a consistent stochastic process.
- That is ensured by the Kolmogorov consistency theorem.
- When each random variable $X_t : \Omega \to \mathbb{R}$, for every $\omega \in \Omega$, the function $t \to X_t(\omega)$ is called a sample path.

Gaussian Processes

- A Gaussian process is a stochastic process such that any finite dimensional distribution is Gaussian.
- Thus $(X_{t_1}, ..., X_{t_n})' \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, for any $t_1, ..., t_n \in \mathscr{T}$ and any n.
- How to determine the $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ so that given any $t_1, ..., t_n$ they are automatically determined?
- One defines a Gaussian process through a mean function and a covariance function.

- $\{X_t : t \in \mathscr{T}\} \sim GP(\mu(\cdot), C_{\boldsymbol{\theta}}(\cdot,\cdot))$ implies

$$\boldsymbol{X} = (X_{t_1}, ..., X_{t_n})' \sim N(\boldsymbol{\mu}, \boldsymbol{C}_{\boldsymbol{\theta}})$$

  for any finite set of locations $t_1, ..., t_n$.
- $\boldsymbol{C}_{\boldsymbol{\theta}} = (C_{\boldsymbol{\theta}}(t_i, t_j))$ is the $n \times n$ covariance matrix.
- $\boldsymbol{\mu} = (\mu(t_1), ..., \mu(t_n))'$.
- Note that $C_{\boldsymbol{\theta}}(\cdot,\cdot)$ should be such that the matrix $\boldsymbol{C}_{\boldsymbol{\theta}}$ is positive definite for any n and any $t_1, ..., t_n$.
- Thus any function can't be chosen as a candidate of $C_{\boldsymbol{\theta}}(\cdot,\cdot)$.