Bayesian Treed Gaussian Processes

STAT8810, Fall 2017

M.T. Pratola

October 22, 2017

Today

Bayesian Treed Gaussian Processes

Bayesian Treed Gaussian Process Model

- Gramacy and Lee† propose to use GP models in the terminal nodes of a Bayesian tree.

† R. B. Gramacy and H. K. H. Lee: *Bayesian Treed Gaussian Process Models With an Application to Computer Modeling*, Journal of the American Statistical Association, vol.103:, pp.1119–1130 (2008).

Bayesian Treed Gaussian Process Model

- Gramacy and Lee† propose to use GP models in the terminal nodes of a Bayesian tree.
- Idea is to gain additional flexibility of the GP model in different areas of predictor space.

† R. B. Gramacy and H. K. H. Lee: *Bayesian Treed Gaussian Process Models With an Application to Computer Modeling*, Journal of the American Statistical Association, vol.103:, pp.1119–1130 (2008).

Bayesian Treed Gaussian Process Model

- Gramacy and Lee† propose to use GP models in the terminal nodes of a Bayesian tree.
- Idea is to gain additional flexibility of the GP model in different areas of predictor space.
- And reduce the computationally challenges of inverting large correlation matrices due to the localization effect of the treed GP approach.

† R. B. Gramacy and H. K. H. Lee: *Bayesian Treed Gaussian Process Models With an Application to Computer Modeling*, Journal of the American Statistical Association, vol.103:, pp.1119–1130 (2008).

Bayesian Treed Gaussian Process Model

- Gramacy and Lee† propose to use GP models in the terminal nodes of a Bayesian tree.
- Idea is to gain additional flexibility of the GP model in different areas of predictor space.
- And reduce the computationally challenges of inverting large correlation matrices due to the localization effect of the treed GP approach.
- Basically combines the Bayesian scalar-terminal-node single tree model we have seen with the Bayesian GP model we have seen.

† R. B. Gramacy and H. K. H. Lee: *Bayesian Treed Gaussian Process Models With an Application to Computer Modeling*, Journal of the American Statistical Association, vol.103:, pp.1119–1130 (2008).

Bayesian Treed Gaussian Process Model

- Gramacy and Lee† propose to use GP models in the terminal nodes of a Bayesian tree.
- Idea is to gain additional flexibility of the GP model in different areas of predictor space.
- And reduce the computationally challenges of inverting large correlation matrices due to the localization effect of the treed GP approach.
- Basically combines the Bayesian scalar-terminal-node single tree model we have seen with the Bayesian GP model we have seen.
    - But their formulation has some differences, and since there is more than one GP there are now a lot more parameters to deal with – increased complexity of sampling algorithm.

† R. B. Gramacy and H. K. H. Lee: *Bayesian Treed Gaussian Process Models With an Application to Computer Modeling*, Journal of the American Statistical Association, vol.103:, pp.1119–1130 (2008).

Bayesian Single Tree Model

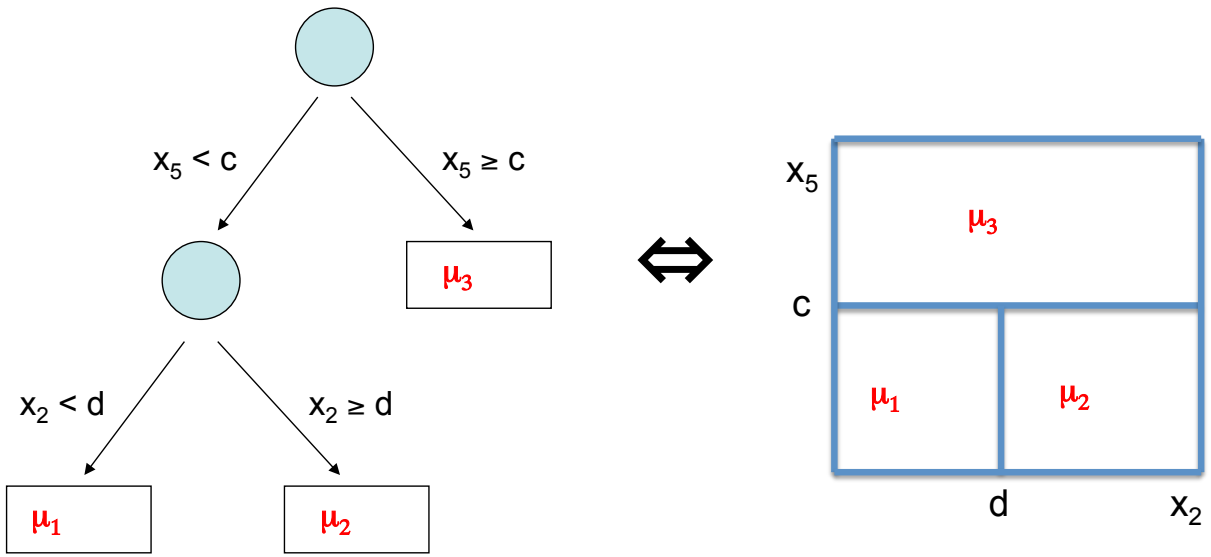


Figure 1: A Single Tree with Scalar Terminal Nodes

Bayesian Treed GP Model

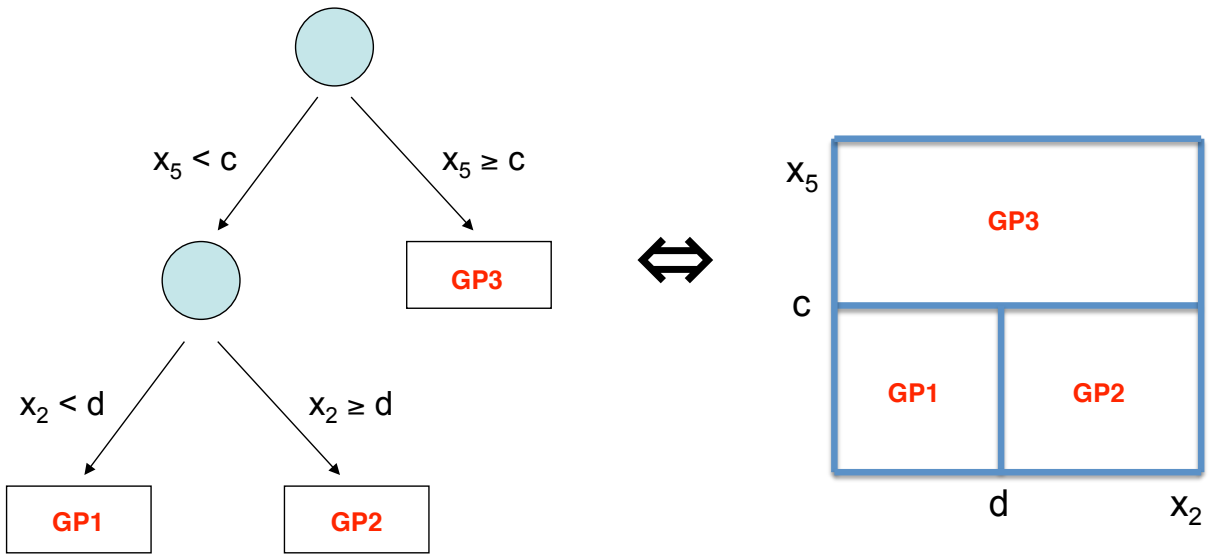


Figure 2: A Single Tree with GP Terminal Nodes

Bayesian Treed Gaussian Process Model

- Suppose our tree $\mathcal{T}$ divides the predictor space into R regions, labeled r_ν for $\nu = 1, \ldots, R$.

Bayesian Treed Gaussian Process Model

- Suppose our tree $\mathcal{T}$ divides the predictor space into R regions, labeled r_ν for $\nu = 1, \ldots, R$.
- Each region has data $D_\nu = [\mathbf{X}_\nu, \mathbf{Z}_\nu]$ of n_ν observations.

Bayesian Treed Gaussian Process Model

- Suppose our tree $\mathcal{T}$ divides the predictor space into R regions, labeled r_ν for $\nu = 1, \ldots, R$.
- Each region has data $D_\nu = [\mathbf{X}_\nu, \mathbf{Z}_\nu]$ of n_ν observations.
- Let m be the total number of predictors plus the intercept.

Bayesian Treed Gaussian Process Model

- Suppose our tree $\mathcal{T}$ divides the predictor space into R regions, labeled r_ν for $\nu = 1, \ldots, R$.
- Each region has data $D_\nu = [\mathbf{X}_\nu, \mathbf{Z}_\nu]$ of n_ν observations.
- Let m be the total number of predictors plus the intercept.
- Their general formulation includes a mean model for the GP (while for simplicity we assumed it was 0).

Bayesian Treed Gaussian Process Model

- Suppose our tree $\mathcal{T}$ divides the predictor space into R regions, labeled r_ν for $\nu = 1, \ldots, R$.
- Each region has data $D_\nu = [\mathbf{X}_\nu, \mathbf{Z}_\nu]$ of n_ν observations.
- Let m be the total number of predictors plus the intercept.
- Their general formulation includes a mean model for the GP (while for simplicity we assumed it was 0).
- The model is specified in multiple hierarchies.

GP Model within a given terminal node ν.

- Given we are in region r_ν (i.e. terminal node ν) the GP model for the data mapping to this node is

$$\mathbf{Z}_\nu|\beta_\nu, \sigma_\nu^2, \mathbf{K}_\nu \sim N_{n_\nu}\left(\mathbf{F}_\nu\beta_\nu, \sigma_\nu^2\mathbf{K}_\nu\right)$$

where β_ν is an $m \times 1$ parameter vector, σ_ν^2 is a scalar parameter,

$$F_\nu = [\mathbf{1}, \mathbf{X}_\nu]$$

and the correlation is specified as

$$\mathbf{K}_\nu(\mathbf{x}_j, \mathbf{x}_k) = exp\left(\sum_i \frac{|x_{ji} - x_{ki}|^2}{d_i}\right) + g\delta_{\mathbf{x}_j=\mathbf{x}_k}$$

where $d_i > 0$ is a correlation length scale parameter for each dimension.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the mean prior's mean is

$$\beta_0 \sim N_m(\boldsymbol{\mu}, \mathbf{B})$$

where $\boldsymbol{\mu}$ and $\mathbf{B}$ are treated as fixed, known hyperparameters.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the scalar marginal variance is

$$\sigma_\nu^2 \sim \text{InverseGamma} \left(\frac{\alpha_\sigma}{2}, \frac{q_\sigma}{2} \right)$$

where α_σ, q_σ are treated as fixed, known hyperparameters.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the scalar marginal variance is

$$\sigma_\nu^2 \sim \text{InverseGamma} \left(\frac{\alpha_\sigma}{2}, \frac{q_\sigma}{2} \right)$$

  where α_σ, q_σ are treated as fixed, known hyperparameters.
- Note that if $A \sim \chi^{-2}(a, b^2)$ then $A \sim \text{InverseGamma}(\frac{a}{2}, \frac{ab^2}{2})$. So their formulation is relatively similar to the scaled-inverse-chisquared formulation we had in our scalar single tree model.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the scalar marginal variance is

$$\sigma_\nu^2 \sim \text{InverseGamma} \left(\frac{\alpha_\sigma}{2}, \frac{q_\sigma}{2} \right)$$

  where α_σ, q_σ are treated as fixed, known hyperparameters.
- Note that if $A \sim \chi^{-2}(a, b^2)$ then $A \sim \text{InverseGamma}(\frac{a}{2}, \frac{ab^2}{2})$. So their formulation is relatively similar to the scaled-inverse-chisquared formulation we had in our scalar single tree model.
  - Think of q_σ as $\alpha_\sigma \times$ scale.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the scalar marginal node-specific variance parameter is

$$\tau_\nu^2 \sim \text{InverseGamma} \left(\frac{\alpha_\tau}{2}, \frac{q_\tau}{2} \right)$$

where α_τ, q_τ are treated as fixed, known hyperparameters.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the mean priors precision is

$$\mathbf{W}^{-1} \sim \text{Wishart} \left((\rho \mathbf{V})^{-1}, \rho \right)$$

where ρ and $\mathbf{V}$ are treated as fixed, known hyperparameters.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the mean priors precision is

$$\mathbf{W}^{-1} \sim \text{Wishart} \left((\rho \mathbf{V})^{-1}, \rho \right)$$

  where ρ and $\mathbf{V}$ are treated as fixed, known hyperparameters.
- We can think of $\mathbf{V}$ as some a-priori information about the relatedness of the regression coefficients. Note that this is a common parameter across all the terminal nodes.

Prior on GP regression coefficient, β_ν.

- The prior on the regression coeffficient is

$$\beta_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \beta_0 \sim N_m \left(\beta_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W} \right).$$

- The prior on the mean priors precision is

$$\mathbf{W}^{-1} \sim \text{Wishart} \left((\rho \mathbf{V})^{-1}, \rho \right)$$

  where ρ and $\mathbf{V}$ are treated as fixed, known hyperparameters.
- We can think of $\mathbf{V}$ as some a-priori information about the relatedness of the regression coefficients. Note that this is a common parameter across all the terminal nodes.
- ρ is a degrees of freedom parameter. A common weakly informative choice is to take $\rho = m$.

Prior on correlation parameters

- For the correlation length scale parameters d_i and "nugget" parameter g,

$$\pi(\mathbf{d}_\nu, g_\nu) = \pi(g_\nu) \prod_i \pi(d_{\nu,i})$$

Prior on correlation parameters

- For the correlation length scale parameters d_i and "nugget" parameter g,

$$\pi(\mathbf{d}_\nu, g_\nu) = \pi(g_\nu) \prod_i \pi(d_{\nu,i})$$

  - Note that these parameters are unique in each region r_ν, so for instance the correlation behavior of the response can be different in each region.

Prior on correlation parameters

- For the correlation length scale parameters d_i and "nugget" parameter g,

$$\pi(\mathbf{d}_\nu, g_\nu) = \pi(g_\nu) \prod_i \pi(d_{\nu,i})$$

  - Note that these parameters are unique in each region r_ν, so for instance the correlation behavior of the response can be different in each region.

- The specific priors used are

$$g_\nu \sim \text{Exponential}(\lambda)$$

where λ is a user-specified hyperparameter, and

$$d_{\nu,i} \sim \frac{1}{2}\left[\text{Gamma}(\alpha = 1, \beta = 20) + \text{Gamma}(\alpha = 10, \beta = 10)\right].$$

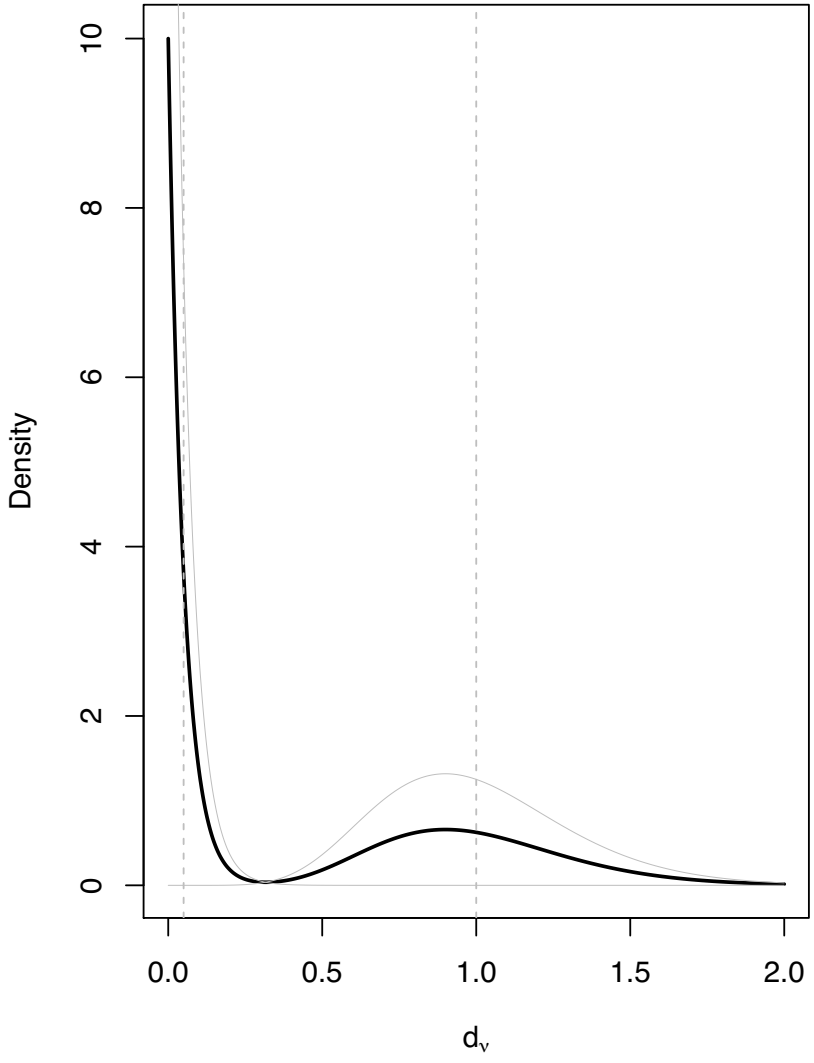
Mixture prior on correlation parameters

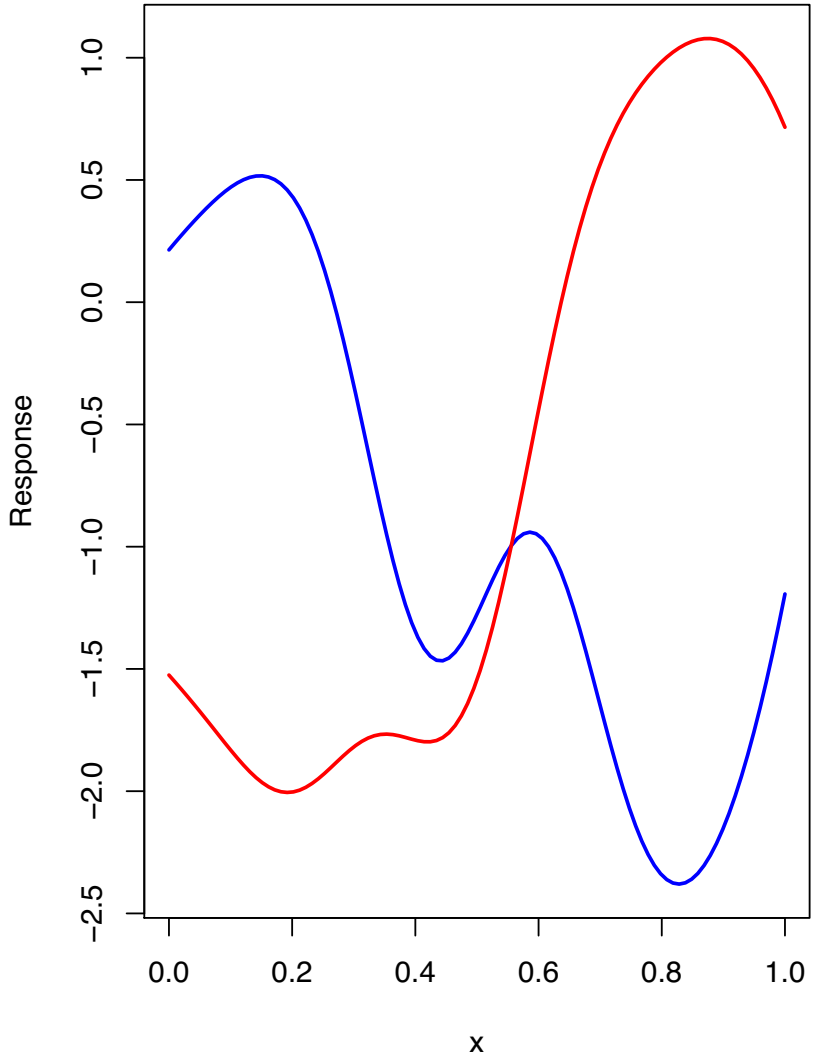
```
x=seq(0,2,length=1000)
da=dgamma(x,shape=1,rate=20)
db=dgamma(x,shape=10,rate=10)
d=0.5*da+0.5*db
par(mfrow=c(1,2))
plot(x,d,type='l',lwd=2,xlab=expression(d[nu]),
     ylab="Density")
lines(x,da,lwd=0.5,col="grey")
lines(x,db,lwd=0.5,col="grey")
abline(v=1/20,lty=2,col="grey")
abline(v=10/10,lty=2,col="grey")
```

Mixture prior on correlation parameters

```
set.seed(99)
x=seq(0,1,length=100)
D=abs(outer(x,x,"-"))
Ra=exp(-D^2/(1/20))  # like rho=2e-9
Rb=exp(-D^2/(10/10)) # like rho=0.37
La=t(chol(Ra+diag(100)*1e-10))
Lb=t(chol(Ra+diag(100)*1e-10))
Za=La%*%rnorm(100)
Zb=Lb%*%rnorm(100)
plot(x,Za,type='l',lwd=2,col="blue",
     ylim=range(c(Za,Zb)),ylab="Response")
lines(x,Zb,lwd=2,col="red")
```

Mixture prior on correlation parameters





Summary of parameters

- So in total we have overall parameteres

$$\boldsymbol{\theta}_0 = \{\mathbf{W}, \boldsymbol{\beta}_0\}\,.$$

Summary of parameters

- So in total we have overall parameteres

$$\boldsymbol{\theta}_0 = \{\mathbf{W}, \boldsymbol{\beta}_0\}.$$

- And terminal-node specific parameters

$$\boldsymbol{\theta}_\nu = \left\{\boldsymbol{\beta}_\nu, \sigma^2_\nu, \mathbf{d}_\nu, g_\nu, \tau^2_\nu\right\}.$$

Summary of parameters

- So in total we have overall parameteres

$$\boldsymbol{\theta}_0 = \{\mathbf{W}, \beta_0\}\,.$$

- And terminal-node specific parameters

$$\boldsymbol{\theta}_\nu = \left\{\beta_\nu, \sigma^2_\nu, \mathbf{d}_\nu, g_\nu, \tau^2_\nu\right\}.$$

- Overall,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 \cup \{\cup_{\nu=1}^{R} \boldsymbol{\theta}_\nu\}.$$

Summary of parameters

- So in total we have overall parameteres

$$\boldsymbol{\theta}_0 = \{\mathbf{W}, \beta_0\}.$$

- And terminal-node specific parameters

$$\boldsymbol{\theta}_\nu = \left\{\beta_\nu, \sigma^2_\nu, \mathbf{d}_\nu, g_\nu, \tau^2_\nu\right\}.$$

- Overall,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 \cup \{\cup_{\nu=1}^{R} \boldsymbol{\theta}_\nu\}.$$

- And we have user-specified hyperparameters

$$\boldsymbol{\mu}, \mathbf{B}, \mathbf{V}, \rho, \alpha_\sigma, q_\sigma, \alpha_\tau, q_\tau.$$

Summary of parameters

- So in total we have overall parameteres

$$\boldsymbol{\theta}_0 = \{\mathbf{W}, \boldsymbol{\beta}_0\}.$$

- And terminal-node specific parameters

$$\boldsymbol{\theta}_\nu = \left\{\boldsymbol{\beta}_\nu, \sigma^2_\nu, \mathbf{d}_\nu, g_\nu, \tau^2_\nu\right\}.$$

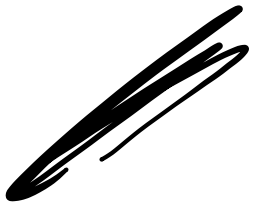
- Overall,

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 \cup \{\cup_{\nu=1}^{R} \boldsymbol{\theta}_\nu\}.$$

- And we have user-specified hyperparameters

$$\boldsymbol{\mu}, \mathbf{B}, \mathbf{V}, \rho, \alpha_\sigma, q_\sigma, \alpha_\tau, q_\tau.$$

- A complicated model!



In terminal node v,

$$K_v(x_i, x_{i'}) = e^{-\sum_l \frac{|x_{il} - x_{i'l}|^2}{d_{wl}}} + q_v^2 \delta_{ii'}$$

$$\pi(\cdot \mid Z_v) \propto L(\cdot \mid Z_v)\,\pi(\cdot)$$

$$= \frac{1}{(2\pi)^{n_v/2} |\sigma_v^2 K_v|^{1/2}} e^{-\frac{1}{2}(Z_v - F_v\beta_v)^T (\sigma_v^2 K_v)^{-1} (Z_v - F_v\beta_v)} \times \underbrace{\frac{1}{(2\pi)^{p/2} |\sigma_v^2 \tau_v^2 W|^{1/2}} e^{-\frac{1}{2}(\beta_v - \beta_0)^T (\sigma_v^2\tau_v^2 W)^{-1} (\beta_v - \beta_0)}}_{\pi(\beta_v \mid \cdot)}$$

$$\times \underbrace{\frac{1}{(2\pi)^{p/2} |B|^{1/2}} e^{-\frac{1}{2}(\beta_0 - \mu)^T B^{-1} (\beta_0 - \mu)}}_{\pi(\beta_0)} \times \underbrace{\frac{\left(\frac{q_\sigma}{2}\right)^{\frac{q_\sigma}{2}}}{\Gamma\left(\frac{q_\sigma}{2}\right)} (\sigma_v^2)^{-\frac{q_\sigma}{2}-1} e^{-\frac{q_\sigma/2}{\sigma_v^2}}}_{\pi(\sigma_v^2)} \times \underbrace{\frac{\left(\frac{q_\tau}{2}\right)^{\frac{q_\tau}{2}}}{\Gamma\left(\frac{q_\tau}{2}\right)} (\tau_v^2)^{-\frac{q_\tau}{2}-1} e^{-\frac{q_\tau/2}{\tau_v^2}}}_{\pi(\tau_v^2)} \times \underbrace{\frac{|W^{-1}|^{\frac{p-m-1}{2}} e^{-\frac{1}{2}\mathrm{tr}(\rho V W^{-1})}}{2^{\frac{pq_\rho}{2}} |(\rho V)^{-1}|^{\frac{q_\rho}{2}} \Gamma_p\left(\frac{q_\rho}{2}\right)}}_{\pi(W^{-1})}$$

$$\times \underbrace{\lambda e^{-\lambda q_v}}_{\pi(q_v)} \times \prod_i \frac{1}{2}\left[\frac{20}{\Gamma(1)} e^{-20 d_{wi}} + \frac{10^{10}}{\Gamma(10)} d_{wi}^{10-1} e^{-10 d_{wi}}\right]$$

$\pi(d_{w1}) \cdots$

$Z \mid \beta, \rho \sim GP(\cdot, \cdot \mid R(\rho))$

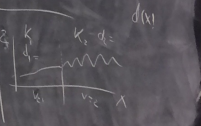
$\pi(\lambda) \sim \cdot$

$\pi(\rho) \sim$

$Z_v \mid \beta_v, \beta_0, \sigma_v^2, K_v$

$\beta_v \mid \sigma_v^2, \tau_v^2, \beta_0, W$

$\beta_0 \mid \cdots$



MCMC Algorithm

- The algorithm to draw samples from the posterior of this model proceeds as follows:

MCMC Algorithm

- The algorithm to draw samples from the posterior of this model proceeds as follows:

1. Draw $\boldsymbol{\theta}|\mathcal{T}, \mathbf{Z}$

MCMC Algorithm

- The algorithm to draw samples from the posterior of this model proceeds as follows:

1. Draw $\boldsymbol{\theta}|\mathcal{T}, \mathbf{Z}$
   - Draw $\boldsymbol{\theta}_\nu|\boldsymbol{\theta}_0, \mathbf{Z}_\nu$ for $\nu = 1, \ldots, R$.

MCMC Algorithm

- The algorithm to draw samples from the posterior of this model proceeds as follows:

1. Draw $\boldsymbol{\theta}|\mathcal{T}, \mathbf{Z}$
   - Draw $\boldsymbol{\theta}_\nu|\boldsymbol{\theta}_0, \mathbf{Z}_\nu$ for $\nu = 1, \ldots, R$.
   - Draw $\boldsymbol{\theta}_0|\cup_{\nu=1}^{R} \boldsymbol{\theta}_\nu, \mathbf{Z}$.

MCMC Algorithm

- The algorithm to draw samples from the posterior of this model proceeds as follows:

1. Draw $\boldsymbol{\theta}|\mathcal{T}, \mathbf{Z}$
   - Draw $\boldsymbol{\theta}_\nu|\boldsymbol{\theta}_0, \mathbf{Z}_\nu$ for $\nu = 1, \ldots, R$.
   - Draw $\boldsymbol{\theta}_0|\cup_{\nu=1}^{R} \boldsymbol{\theta}_\nu, \mathbf{Z}$.

2. Draw $\mathcal{T}|\boldsymbol{\theta}, \mathbf{Z}$

Draw $\theta|\mathcal{T}, \mathbf{Z}$

- I will use the symbol "$\cdot$" to mean "everything else" to reduce notation overload.

1a. Draw

$$\beta_\nu|\cdot \sim N_m\left(\tilde{\beta}_\nu, \sigma_\nu^2 V_{\tilde{\beta}_\nu}\right)$$

where

$$V_{\tilde{\beta}_\nu} = \left(\mathbf{F}_\nu^T \mathbf{K}_\nu^{-1} \mathbf{F}_\nu + \mathbf{W}^{-1}/\tau_\nu^2\right)^{-1}$$

and

$$\tilde{\beta}_\nu = V_{\tilde{\beta}_\nu}\left(\mathbf{F}_\nu^T \mathbf{K}_\nu^{-1} \mathbf{Z}_\nu + \mathbf{W}^{-1}\beta_0/\tau_\nu^2\right).$$

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1b. Draw

$$\beta_0|\cdot \sim N_m\left(\tilde{\beta}_0, V_{\tilde{\beta}_0}\right)$$

where

$$V_{\tilde{\beta}_0} = \left(\mathbf{B}^{-1} + \mathbf{W}^{-1}\sum_{\nu=1}^{R}(\sigma_\nu\tau_\nu)^{-2}\right)^{-1}$$

and

$$\tilde{\beta}_0 = V_{\tilde{\beta}_0}\left(\mathbf{B}^{-1}\boldsymbol{\mu} + \mathbf{W}^{-1}\sum_{\nu=1}^{R}\beta_\nu(\sigma_\nu\tau_\nu)^{-2}\right).$$

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1c. Draw

$$\tau_\nu^2|\cdot \sim \text{InverseGamma}\left((\alpha_\tau + m)/2, (q_\tau + b_\nu)/2\right)$$

where

$$b_\nu = (\beta_\nu - \beta_0)^T \mathbf{W}^{-1} (\beta_\nu - \beta_0)/\sigma_n^2$$

and m is the number of predictor variables including intercept.

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1d. Draw

$$\mathbf{W}^{-1}|\cdot \sim \text{Wishart}_m\left(\left(\rho\mathbf{V} + V_{\widehat{\mathbf{W}}}\right)^{-1}, \rho + R\right)$$

where

$$\mathbf{V}_{\widehat{\mathbf{W}}} = \sum_{\nu=1}^{R} \frac{1}{(\sigma_\nu \tau_\nu)^2}(\beta_\nu - \beta_0)(\beta_\nu - \beta_0)^T.$$

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1e. Draw $d_{\nu,1}, \ldots,$ for $\nu = 1, \ldots, R$ and g_ν for $\nu = 1, \ldots, R$.

These draws are performed using Metropolis-Hastings steps. Similar to how we integrated some parameters out of our single-tree model, they integrate out β_ν and σ^2_ν giving

$$\begin{aligned}\pi(\mathbf{K}_\nu|\mathbf{Z}_\nu, \beta_0, \mathbf{W}, \tau^2, \mathbf{Z}_\nu) &= \left(\frac{|\mathbf{V}_{\beta_\nu}|(2\pi)^{-n_\nu}}{|\mathbf{K}_\nu||\mathbf{W}|\tau^{2m}}\right)^{1/2} \\ &\times \frac{(q_\sigma/2)^{\alpha_\sigma/2}\Gamma\left((1/2)(\alpha_\sigma + n_\nu)\right)}{\left((1/2)(q_\sigma + \Psi_\nu)\right)^{(\alpha_\sigma+n_\nu)/2}\Gamma(\alpha_\sigma/2)} \\ &\times \pi(\mathbf{K}_\nu)\end{aligned} \tag{1}$$

where $\Psi_\nu = \mathbf{Z}_\nu^T\mathbf{K}_\nu^{-1}\mathbf{Z}_\nu + \beta_0^T\mathbf{W}^{-1}\beta_0/\tau^2 - \tilde{\beta}_\nu^T\mathbf{V}_{\tilde{\beta}_\nu}^{-1}\tilde{\beta}_\nu$.

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1e. Draw $d_{\nu,1}, \ldots,$ for $\nu = 1, \ldots, R$ and g_ν for $\nu = 1, \ldots, R$.

- Using (1) one can perform MH steps for the $d_{\nu,i}$'s and the g_ν's similar to how we did for our Bayesian GP model. (The authors here don't expand on how they actually implement this).

Draw $\theta|\mathcal{T}, \mathbf{Z}$

1f. Draw

$$\sigma_\nu^2|\cdot \sim \text{InverseGamma}\left((\alpha_\sigma + n_\nu)/2, (q_\sigma + \Psi_\nu)/2\right).$$

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- Similar to our Bayesian single-tree model, here the tree space will be explored using birth/death proposals as well as change/swap moves for updating the internal node decision rules.

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- Similar to our Bayesian single-tree model, here the tree space will be explored using birth/death proposals as well as change/swap moves for updating the internal node decision rules.
- We will look at the Birth proposal. Similar to our earlier approach, the authors integrate out continuous parameters to make these dimension-changing proposals easier to implement by using Equation (1).

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- Similar to our Bayesian single-tree model, here the tree space will be explored using birth/death proposals as well as change/swap moves for updating the internal node decision rules.
- We will look at the Birth proposal. Similar to our earlier approach, the authors integrate out continuous parameters to make these dimension-changing proposals easier to implement by using Equation (1).
- However, there are some continuous parameters that cannot be integrated in closed form, namely the $d_{\nu,i}$'s and g_ν's.

A 1-slide crash course on RJ-MCMC

- Reversible-Jump MCMC (RJ-MCMC) is needed when the dimension of continuous parameters will change from one iteration of the MCMC to the next.

† P. J. Green: *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*, Biometrika, vol.82, pp.711–732 (1995).

A 1-slide crash course on RJ-MCMC

- Reversible-Jump MCMC (RJ-MCMC) is needed when the dimension of continuous parameters will change from one iteration of the MCMC to the next.
- A seminal paper by Peter Green† derives the appropriate acceptance probability as

$$\alpha = min\left\{1, \frac{\pi(\theta')q(\theta' \to \theta)}{\pi(\theta)q(\theta \to \theta')q(u)}\left|\frac{\partial\theta'}{\partial(\theta, u)}\right|\right\}.$$

† P. J. Green: *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*, Biometrika, vol.82, pp.711–732 (1995).

A 1-slide crash course on RJ-MCMC

- Reversible-Jump MCMC (RJ-MCMC) is needed when the dimension of continuous parameters will change from one iteration of the MCMC to the next.
- A seminal paper by Peter Green† derives the appropriate acceptance probability as

$$\alpha = min\left\{1, \frac{\pi(\theta')q(\theta' \to \theta)}{\pi(\theta)q(\theta \to \theta')q(u)}\left|\frac{\partial\theta'}{\partial(\theta, u)}\right|\right\}.$$

- Here, u is the augmentation of the continuous parameters of the existing state to match dimensions with the proposed state after a birth.

† P. J. Green: *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*, Biometrika, vol.82, pp.711–732 (1995).

A 1-slide crash course on RJ-MCMC

- Reversible-Jump MCMC (RJ-MCMC) is needed when the dimension of continuous parameters will change from one iteration of the MCMC to the next.
- A seminal paper by Peter Green† derives the appropriate acceptance probability as

$$\alpha = min\left\{1, \frac{\pi(\theta')q(\theta' \to \theta)}{\pi(\theta)q(\theta \to \theta')q(u)}\left|\frac{\partial\theta'}{\partial(\theta, u)}\right|\right\}.$$

- Here, u is the augmentation of the continuous parameters of the existing state to match dimensions with the proposed state after a birth.
- The expression at the right denotes the determinant of the Jacobian matrix describing the deterministic maps between the lower-dimensional (existing) state to the higher-dimensional proposed state resulting from birth.

† P. J. Green: *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*, Biometrika, vol.82, pp.711–732 (1995).

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- In TGP, the authors use simple maps for the dimension-changing moves so that the determinant of the Jacobian matrix is 1.

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- In TGP, the authors use simple maps for the dimension-changing moves so that the determinant of the Jacobian matrix is 1.
- For example, in birth, one child node is randomly selected to have the d_ν, g_ν's from the parent node and the other child node randomly draws these parameters from the prior.

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- In TGP, the authors use simple maps for the dimension-changing moves so that the determinant of the Jacobian matrix is 1.
- For example, in birth, one child node is randomly selected to have the d_ν, g_ν's from the parent node and the other child node randomly draws these parameters from the prior.
- A similar approach applies for death proposals.

Draw $\mathcal{T}|\theta, \mathbf{Z}$

- The resulting MH ratio for birth is calculated as

$$\frac{|\mathcal{G}|}{|\mathcal{P}|}\frac{\pi(\eta \text{ splits})pi(\eta_{(l)} \text{ terminal})\pi(\eta_{(r)} \text{ terminal})}{\pi(\eta \text{ terminal})}$$

$$\times\frac{\pi(\mathbf{K}_{(l)}|\mathbf{Z}_{(l)}\beta_0\tau^2_{(l)}, \mathbf{W})\pi(\mathbf{K}_{(r)}|\mathbf{Z}_{(r)}\beta_0\tau^2_{(r)}, \mathbf{W})}{\pi(\mathbf{K}_{\nu}|\mathbf{Z}_{\nu}\beta_0\tau^2_{\nu}, \mathbf{W})}$$

where $\pi(\eta \text{ splits}) = a(1 + d_\eta)^{-b}$ and $|\mathcal{P}|$ is the number of nodes in $\mathcal{T}$ where a death proposal can occur and $|\mathcal{G}|$ is the number of nodes where a birth proposal can occur.

Prediction

- Similar to earlier, first write down the (conditional) predictive distribution, then marginalize with respect to the posterior to arrive at the posterior predictive.

Prediction

- Similar to earlier, first write down the (conditional) predictive distribution, then marginalize with respect to the posterior to arrive at the posterior predictive.
- The conditional distribution at a new input $\mathbf{x}$ mapping to terminal node ν is Normal with mean

$$E[Z(\mathbf{x})|\cdot, \mathbf{x} \in \nu] = \mathbf{f}^T(\mathbf{x})\tilde{\beta}_\nu + \mathbf{k}_\nu(\mathbf{x})^T\mathbf{K}_\nu^{-1}(\mathbf{Z}_\nu - \mathbf{F}_\nu\tilde{\beta}_\nu)$$

and variance

$$\text{Var}(Z(\mathbf{x})|\cdot, \mathbf{x} \in \nu) = \sigma_\nu^2\left(k_\nu(\mathbf{x},\mathbf{x}) - \mathbf{q}_\nu^T(\mathbf{x})\mathbf{C}_\nu^{-1}\mathbf{q}_\nu(\mathbf{x})\right)$$

where $\mathbf{C}_\nu^{-1} = (\mathbf{K}_\nu + \tau_\nu^2\mathbf{F}_\nu\mathbf{W}\mathbf{F}_\nu^T)^{-1}$,
$\mathbf{q}_\nu(\mathbf{x}) = \mathbf{k}_\nu(\mathbf{x}) + \tau_\nu^2\mathbf{F}_\nu\mathbf{W}_\nu\mathbf{f}(\mathbf{x})$ and
$k_\nu(\mathbf{x},\mathbf{x}') = K_\nu(\mathbf{x},\mathbf{x}') + \tau_\nu^2\mathbf{f}^T(\mathbf{x})\mathbf{W}f(\mathbf{x}')$.

Software

- The model is available in the `R` package `tgp` on `CRAN`. Lots of built-in demos.

Software

- The model is available in the `R` package `tgp` on `CRAN`. Lots of built-in demos.
- There is also a vignette and publication in JSS describing more practical aspects.

Software

- The model is available in the `R` package `tgp` on `CRAN`. Lots of built-in demos.
- There is also a vignette and publication in JSS describing more practical aspects.
- Among other things, the software can take advantage of the tree-induced conditional independence to sample the $\boldsymbol{\theta}_\nu | \mathcal{T}, \mathbf{Z}$ in parallel.

Software

- The model is available in the `R` package `tgp` on `CRAN`. Lots of built-in demos.
- There is also a vignette and publication in JSS describing more practical aspects.
- Among other things, the software can take advantage of the tree-induced conditional independence to sample the $\boldsymbol{\theta}_\nu | \mathcal{T}, \mathbf{Z}$ in parallel.
- Although conditional on a tree the model will have sharp discontinuities at the splits, posterior averaging tends to smooth these out.

Software

- The model is available in the `R` package `tgp` on `CRAN`. Lots of built-in demos.
- There is also a vignette and publication in JSS describing more practical aspects.
- Among other things, the software can take advantage of the tree-induced conditional independence to sample the $\boldsymbol{\theta}_\nu | \mathcal{T}, \mathbf{Z}$ in parallel.
- Although conditional on a tree the model will have sharp discontinuities at the splits, posterior averaging tends to smooth these out.
- An advantage of this model is the ability to model heteroscedasticity and non-stationarity to some degree. Some also use this model as a means for learning where in predictor space the behaviour of a response changes.

Example

```
library(tgp)
demo(package="tgp")
```

Example

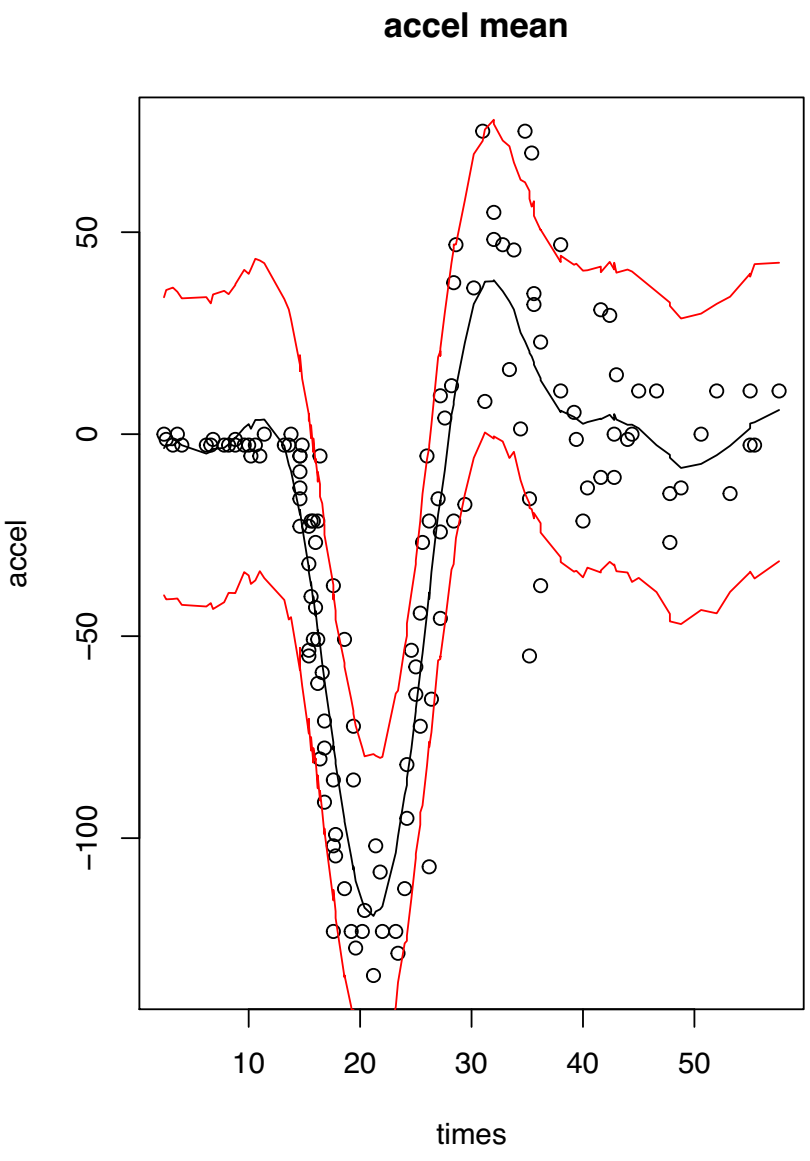
- Main function is `btgp()`. Lets look at the `moto` data.

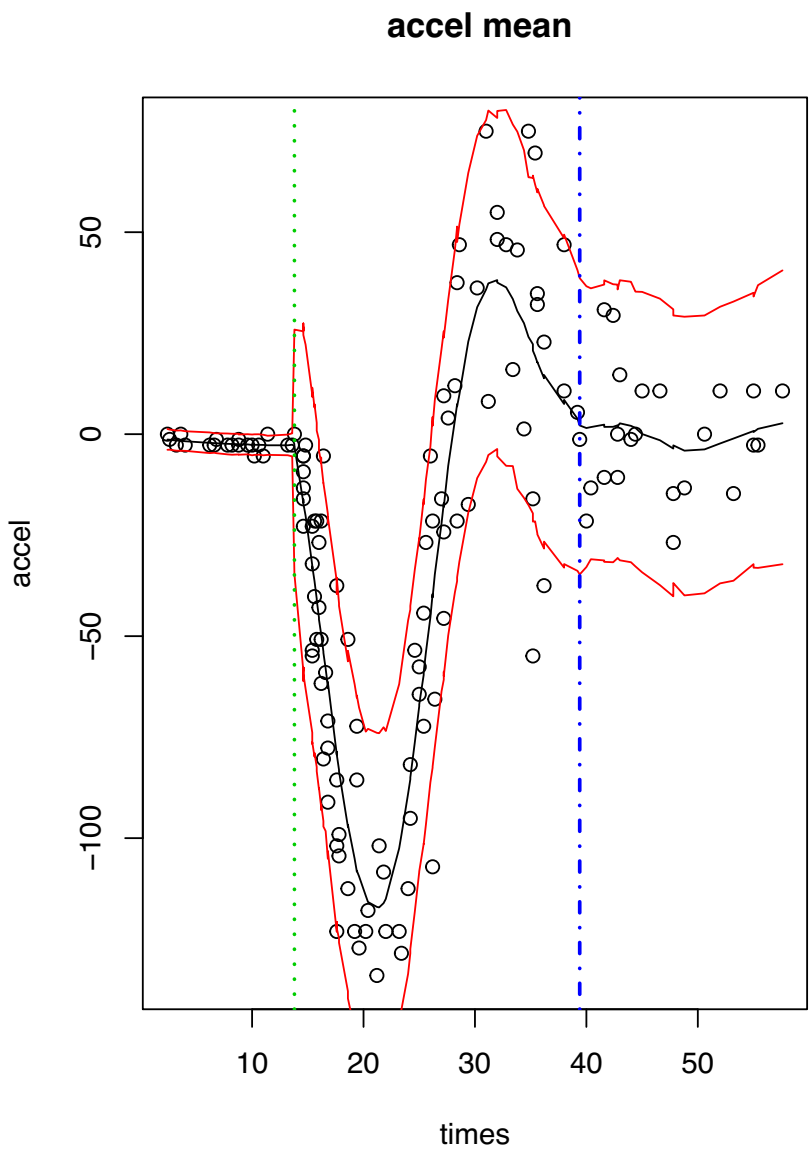
```
set.seed(88)
library(MASS)
X=data.frame(times=mcycle[,1])
Z=data.frame(accel=mcycle[,2])
fit.gp=bgp(X=X,Z=Z,verb=0) # Regular GP fit (no tree)
fit.tgp=btgp(X=X,Z=Z,bprior="b0",verb=0) # Treed GP
```

Example

```
# Plot both fits (posterior mean predictions) side by side
par(mfrow=c(1,2))
plot(fit.gp,layout='surf')
plot(fit.tgp,layout='surf')
```

Example


accel mean
accel
50
0
−50
−100
10
20
30
40
50
times


accel mean
accel
50
0
−50
−100
10
20
30
40
50
times

Example

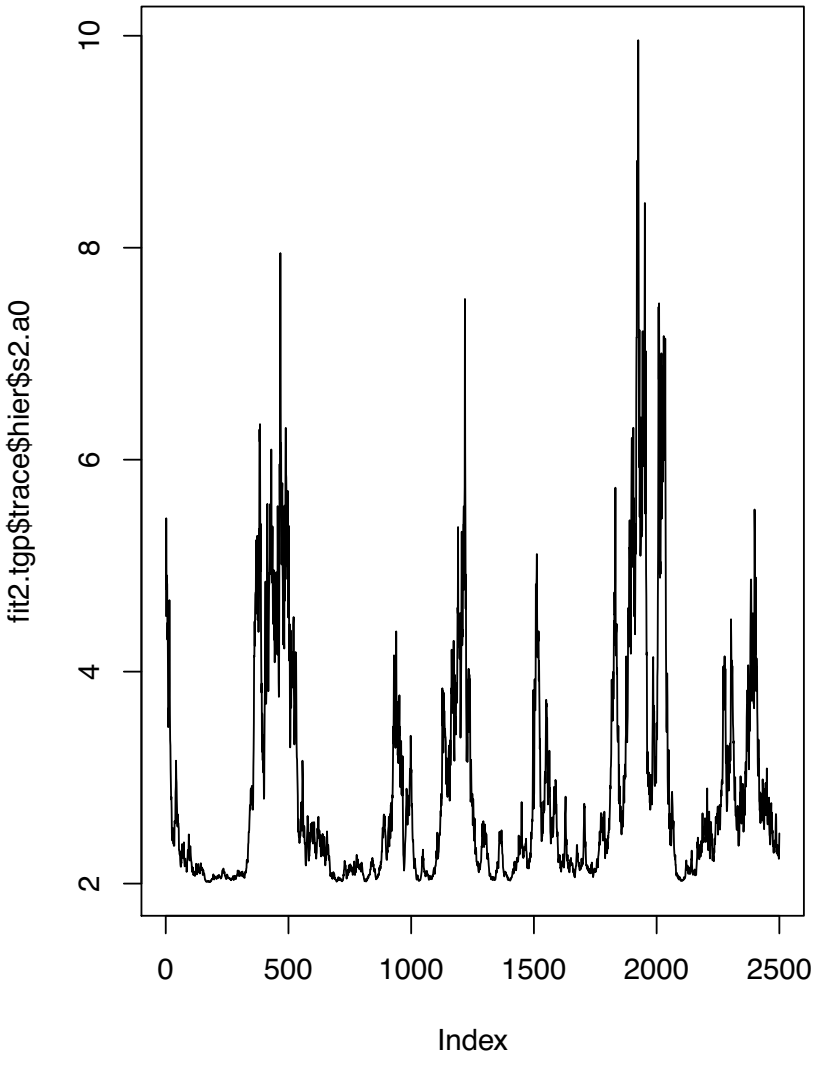
```
# Model details:
str(fit.tgp)
```

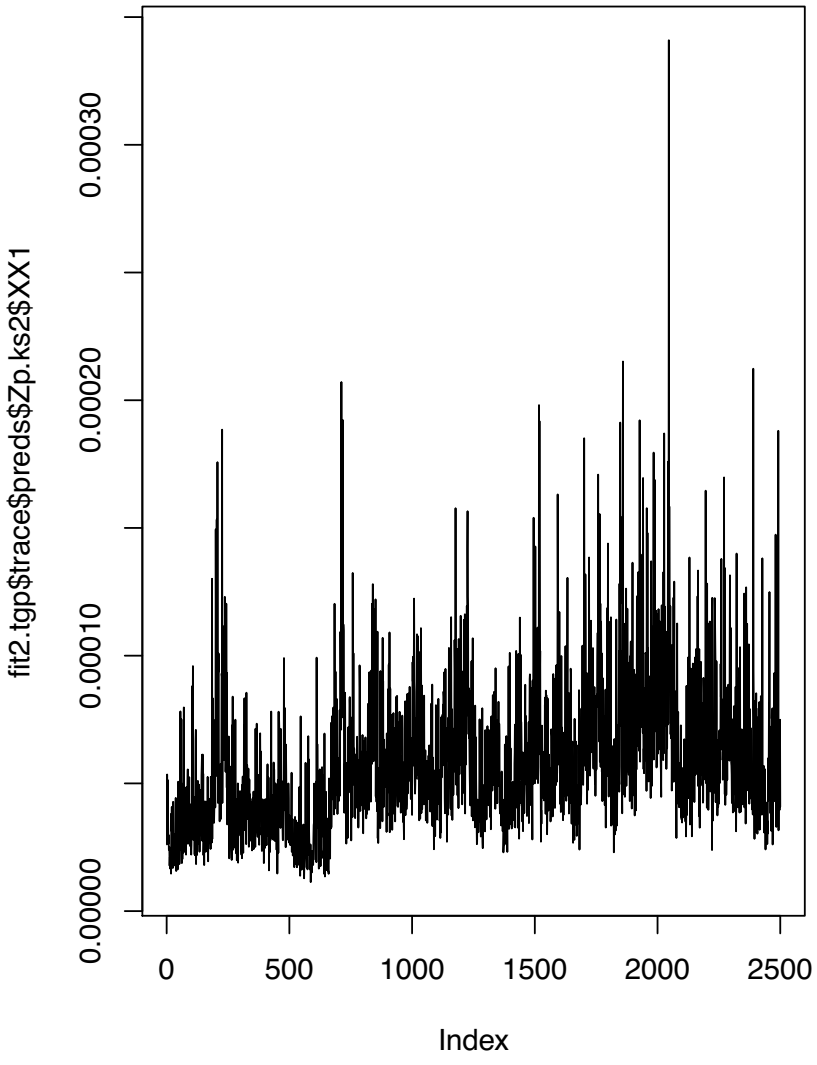
```
## List of 31
##  $ X        :'data.frame':    133 obs. of  1 variable:
##   ..$ times: num [1:133] 2.4 2.6 3.2 3.6 4 6.2 6.6 6.8 7
##  $ n        : int 133
##  $ d        : int 1
##  $ Z        : num [1:133] 0 -1.3 -2.7 0 -2.7 -2.7 -2.7 -1
##  $ nn       : int 0
##  $ Xsplit   :'data.frame':    133 obs. of  1 variable:
##   ..$ times: num [1:133] 2.4 2.6 3.2 3.6 4 6.2 6.6 6.8 7
##  $ BTE      : int [1:3] 2000 7000 2
##  $ R        : int 1
##  $ linburn  : logi FALSE
##  $ g        : int [1:2] 0 0
##  $ dparams  : num [1:45] 0.5 2 10 1 1 0 0 0 0 1 ...
```

Example

```
# By default samples are not saved from the posterior,
# only the posterior quantities we want are recorded.
# Use trace=TRUE to save more information.
# However, storage may be an issue.
fit2.tgp=btgp(X=X,Z=Z,bprior="b0",verb=0,trace=TRUE)
par(mfrow=c(1,2))
plot(fit2.tgp$trace$hier$s2.a0,type='l')
plot(fit2.tgp$trace$preds$Zp.ks2$XX1,type='l')
```

Example

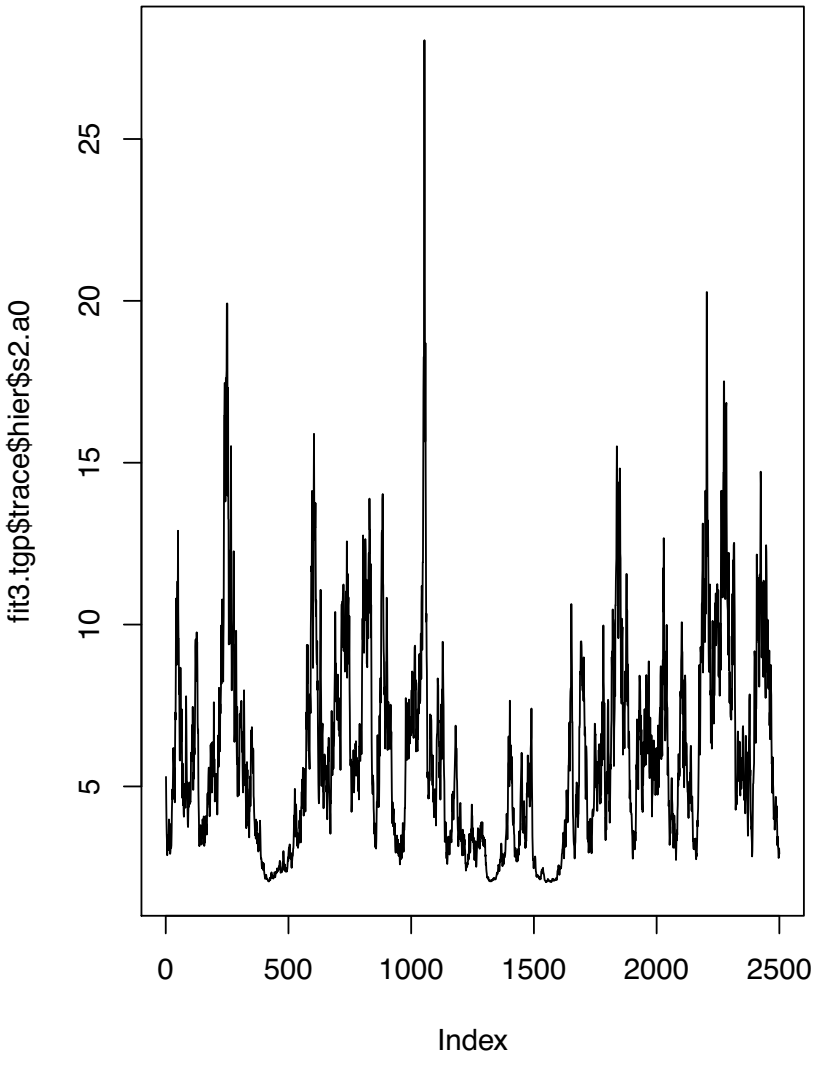

fit2.tgp$trace$hier$s2.a0
2
4
6
8
10
0
500
1000
1500
2000
2500
Index

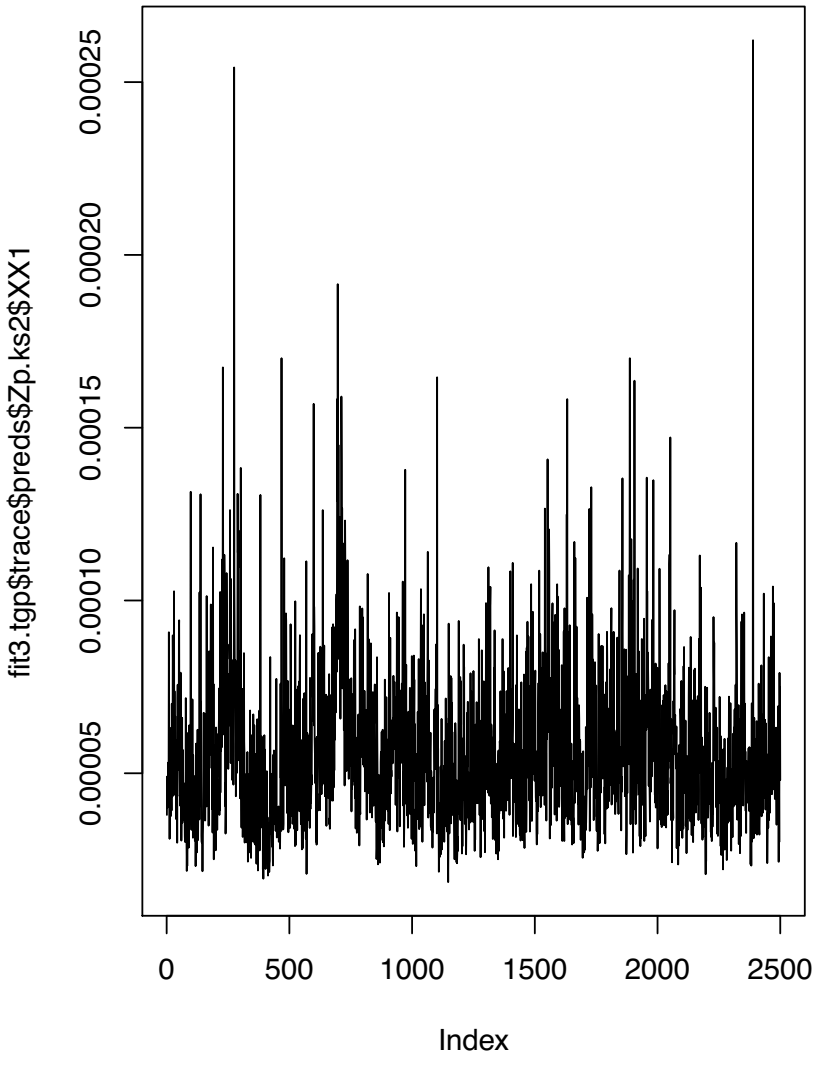

fit2.tgp$trace$preds$Zp.ks2$XX1
0.00000
0.00010
0.00020
0.00030
0
500
1000
1500
2000
2500
Index

Example

```
# Run for more iterations.
# BTE=(burn,total,every)
# Default is BTE=(2000,7000,2)
fit3.tgp=btgp(X=X,Z=Z,bprior="b0",verb=0,trace=TRUE,BTE=c(
par(mfrow=c(1,2))
plot(fit3.tgp$trace$hier$s2.a0,type='l')
plot(fit3.tgp$trace$preds$Zp.ks2$XX1,type='l')
```

Example


fit3.tgp$trace$hier$s2.a0
5
10
15
20
25
0
500
1000
1500
2000
2500
Index

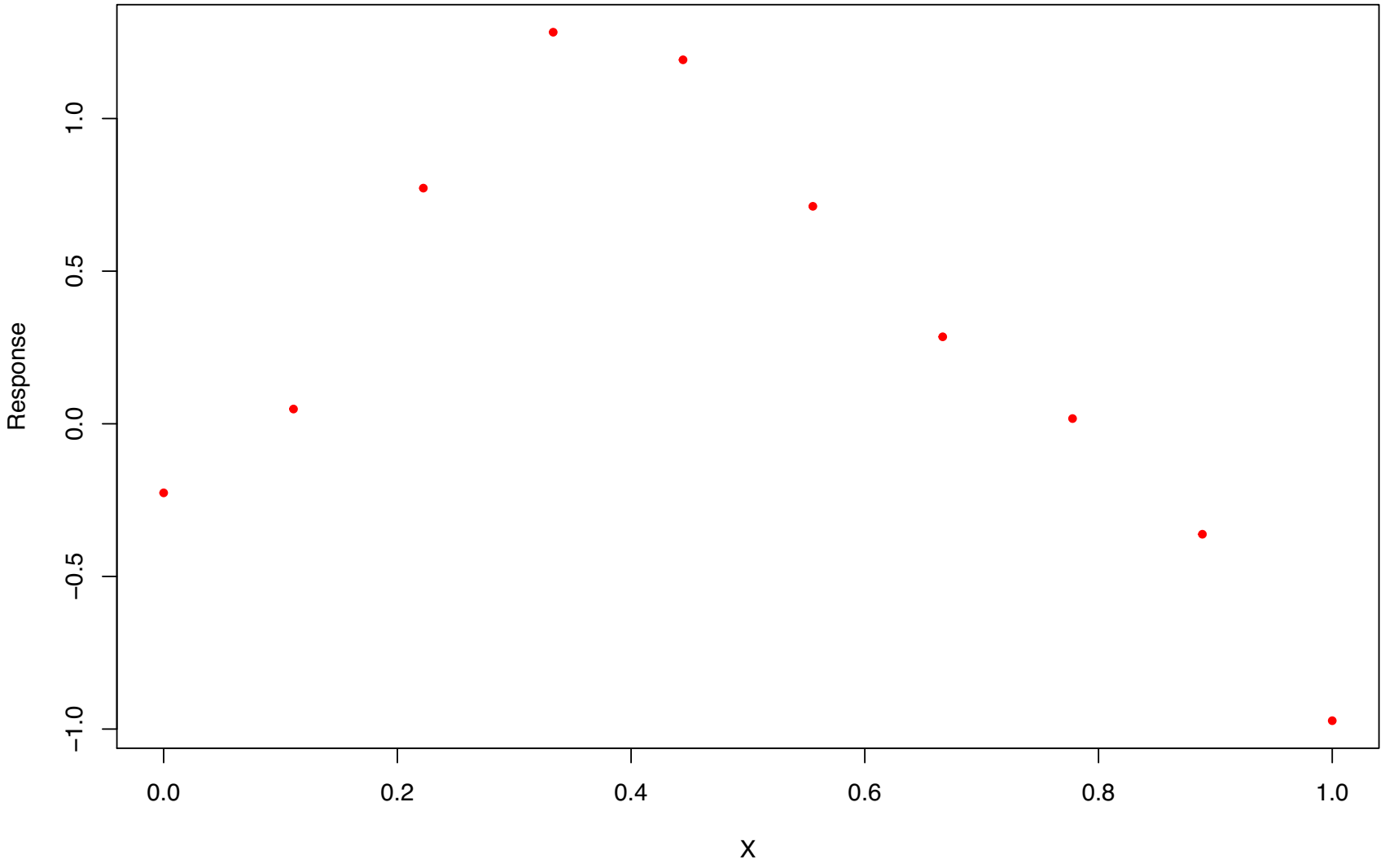

fit3.tgp$trace$preds$Zp.ks2$XX1
0.00005
0.00010
0.00015
0.00020
0.00025
0
500
1000
1500
2000
2500
Index

Example

- What happens if we try our stationary example from earlier?

```
set.seed(88)
x=seq(0,1,length=10)
D=abs(outer(x,x,"-"))
R=0.001^(D^2)
L=t(chol(R))
Z=L%*%rnorm(10)
X=data.frame(x)
Z=data.frame(Z)
plot(X,Z,pch=20,col="red",xlab="X",ylab="Response")
```

Example

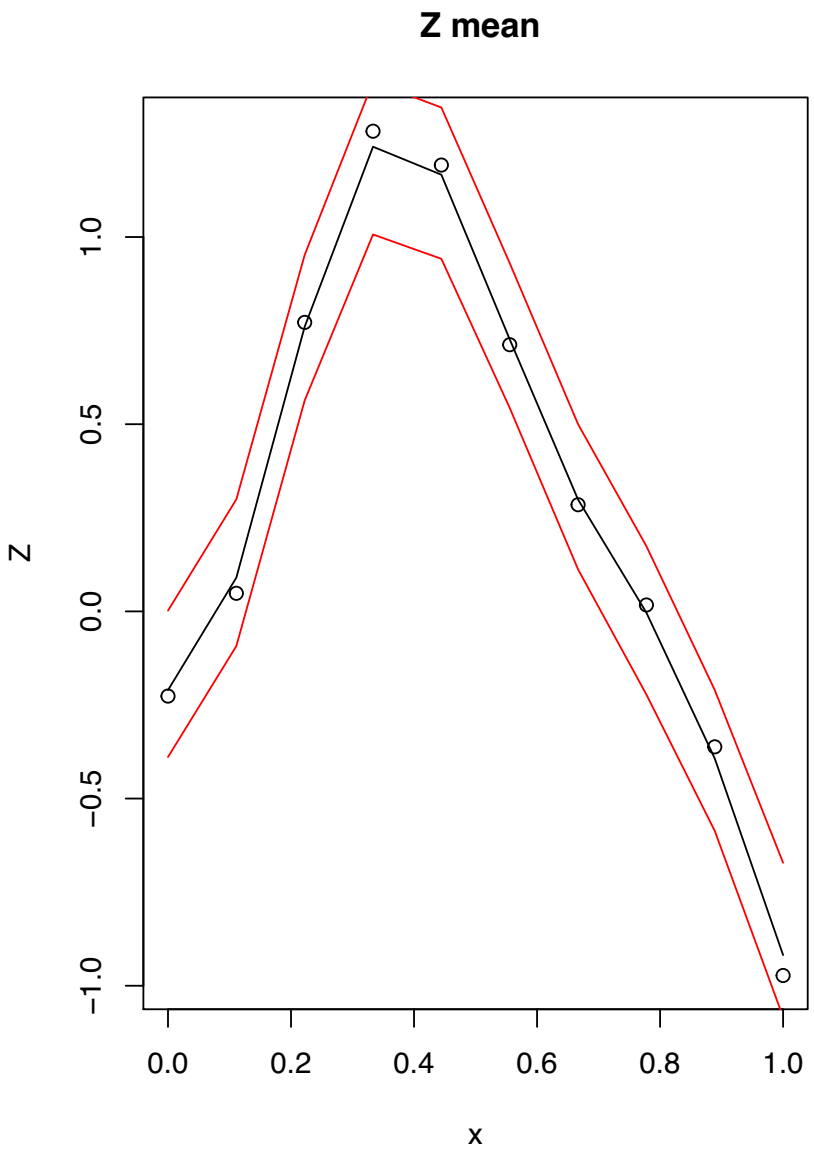


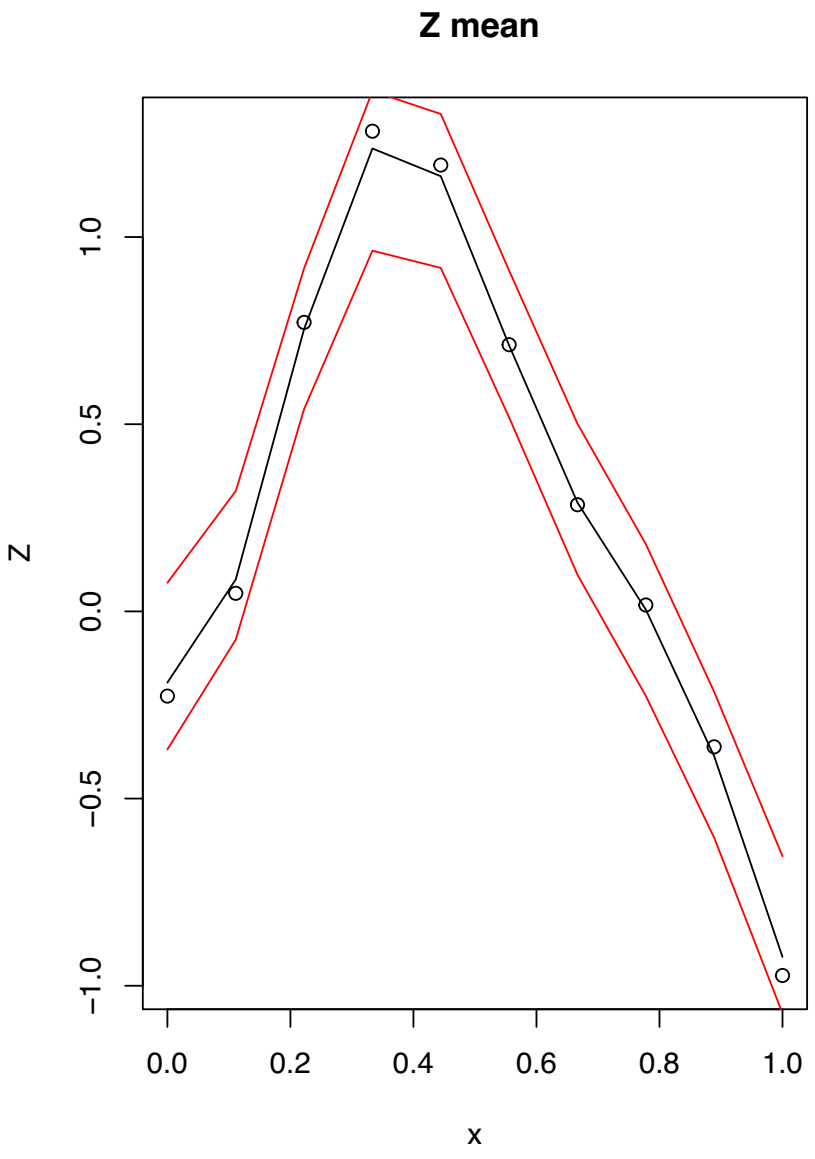
Example

```
fit.gp=bgp(X=X,Z=Z,verb=0) # Regular GP fit (no tree)
fit.tgp=btgp(X=X,Z=Z,bprior="b0",verb=0) # Treed GP

# Plot both fits (posterior mean predictions) side by side
par(mfrow=c(1,2))
plot(fit.gp,layout='surf')
plot(fit.tgp,layout='surf')
```

Example


Z mean
Z
x
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0


Z mean
Z
x
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0