tgp: an R package for Bayesian treed Gaussian process models

Robert B. Gramacy[†] (bobby@statslab.cam.ac.uk) • The Statistical Laboratory • University of Cambridge, UK

[†] most of this work was done at UC Santa Cruz, under the supervision of Herbie Lee

Abstract

The tgp package is a tool for fully Bayesian semiparametric and nonstationary regression by treed Gaussian processes with jumps to the limiting linear model.

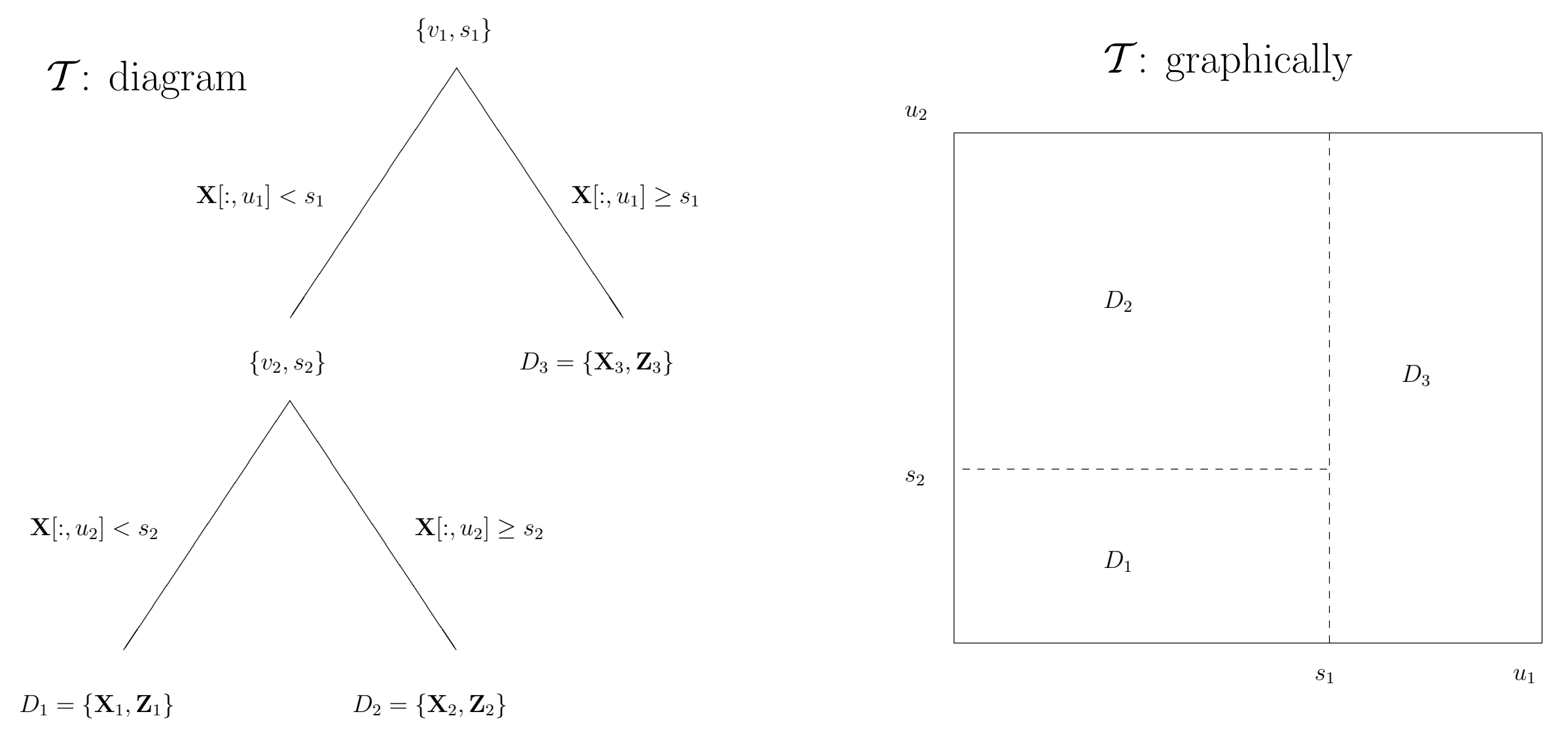
Special cases also implemented include the Bayesian linear model, linear CART, and the stationary separable or isotropic Gaussian process (GP).

In addition to inference and posterior prediction, the package supports the (sequential) design of experiments under these models paired with several objective criteria.

1-d and 2-d plotting, with higher dimension projection and slice capabilities, and tree drawing functions are provided.

1. Treed GP hierarchical model

A tree $\mathcal{T}$ makes recursive binary splits, partitioning the design matrix and responses $\{\mathbf{X}, \mathbf{y}\}$ into independent regions $\{D_\nu\}_{\nu=1}^{D}$.

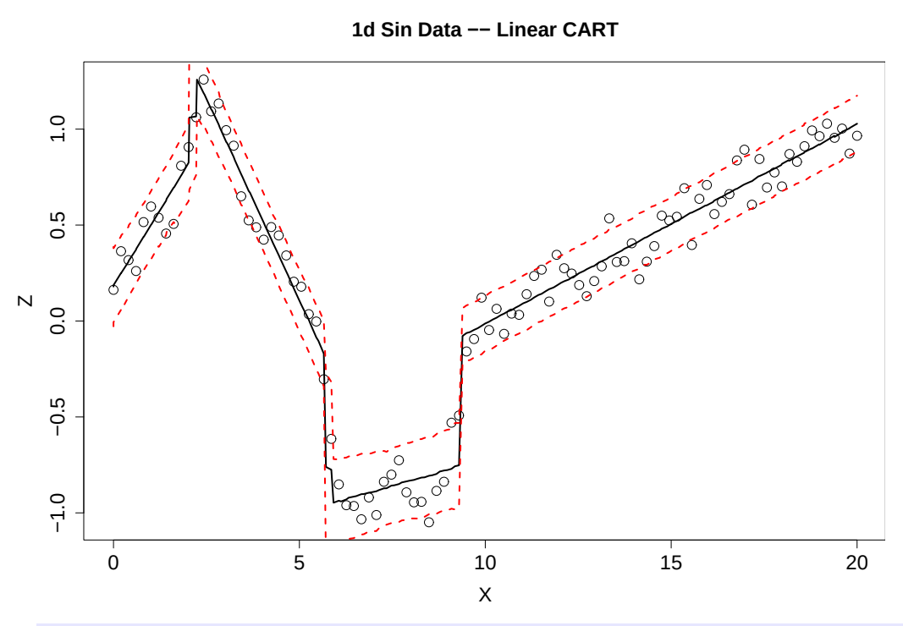

$\mathcal{T}$: diagram
$\{v_1, s_1\}$
$\mathbf{X}[:, u_1] < s_1$
$\mathbf{X}[:, u_1] \geq s_1$
$\{v_2, s_2\}$
$D_3 = \{\mathbf{X}_3, \mathbf{Z}_3\}$
$\mathbf{X}[:, u_2] < s_2$
$\mathbf{X}[:, u_2] \geq s_2$
$D_1 = \{\mathbf{X}_1, \mathbf{Z}_1\}$
$D_2 = \{\mathbf{X}_2, \mathbf{Z}_2\}$
$\mathcal{T}$: graphically
u_2
D_2
D_3
s_2
D_1
s_1
u_1

- Treed GPs extend the Bayesian Linear CART model of Chipman et al. by using GPs at leaves $\{D_\nu\}_{\nu=1}^{D}$ instead of LMs
- The predicted value of $y(\mathbf{x} \in D_\nu)$ follows typical *kriging* equations
- A process prior is placed on $\mathcal{T}$, and joint sampling $(\mathcal{T}, \boldsymbol{\theta})$ obtained via RJ-MCMC with *grow, prune, change*, and *swap* tree operations
- A special case: treed GP with jumps to the limiting linear model (LLM) implements Linear CART
- Either a GP or the LLM can govern the marginal process in each dimension
- Bayesian model averaging yields an adaptively semiparametric nonstationary regression model which is fast, parsimonious, and numerically stable

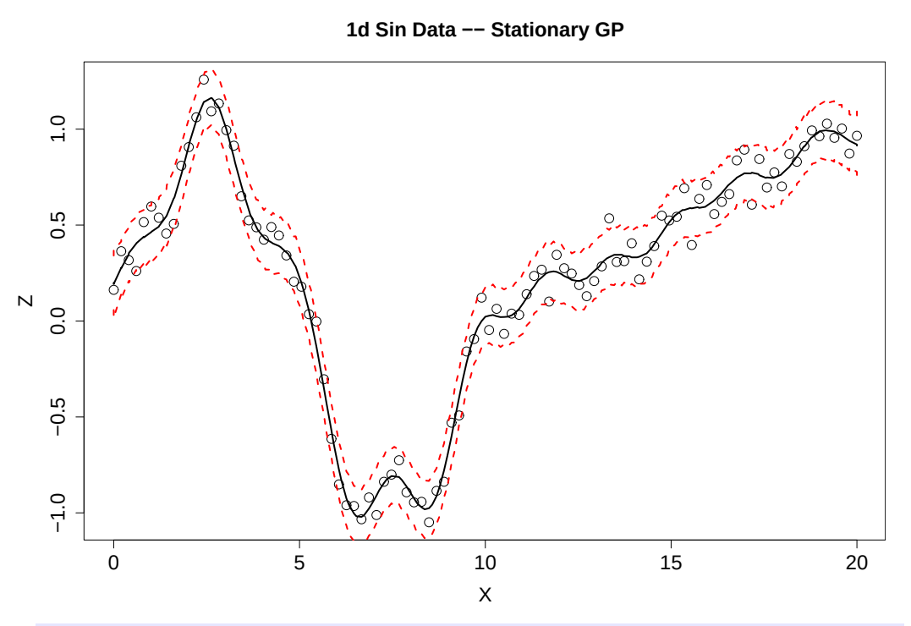
2. Model Comparison

Simple synthetic data from a mixture of sines, cosines, and LM:

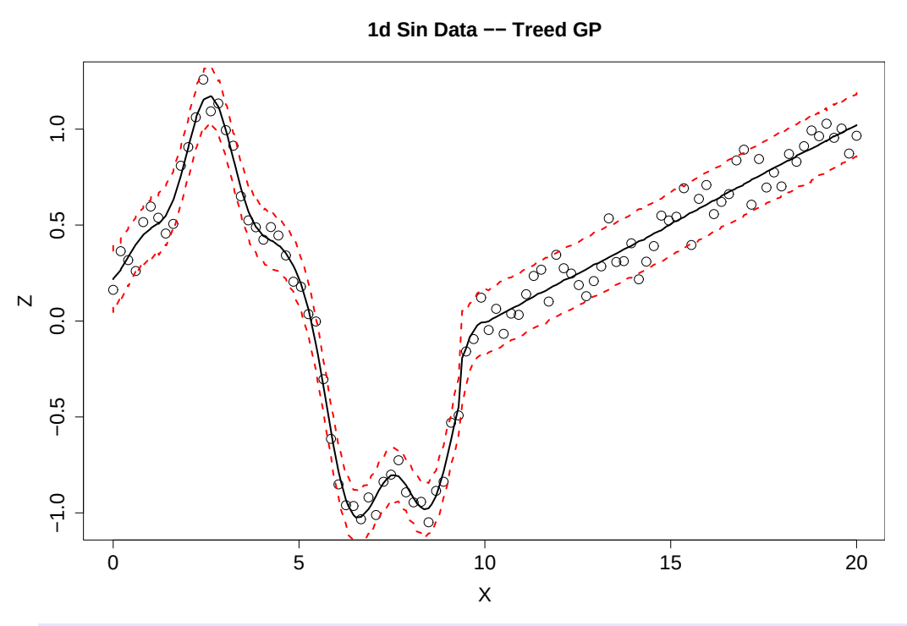
```
X <- seq(0,20,length=100); XX <- seq(0,20,length=99)
Z <- (sin(pi*X/5) + 0.2*cos(4*pi*X/5)) * (X <= 9.6)
lin <- X>9.6; Z[lin] <- -1 + X[lin]/10
Z <- Z + rnorm(length(Z), sd=0.1)
```



```
sin.btlm <- btlm(X=X, Z=Z)
plot(sin.btlm, layout="surf")
```



```
sin.bgp <- bgp(X=X, Z=Z)
plot(sin.btlm, layout="surf")
```

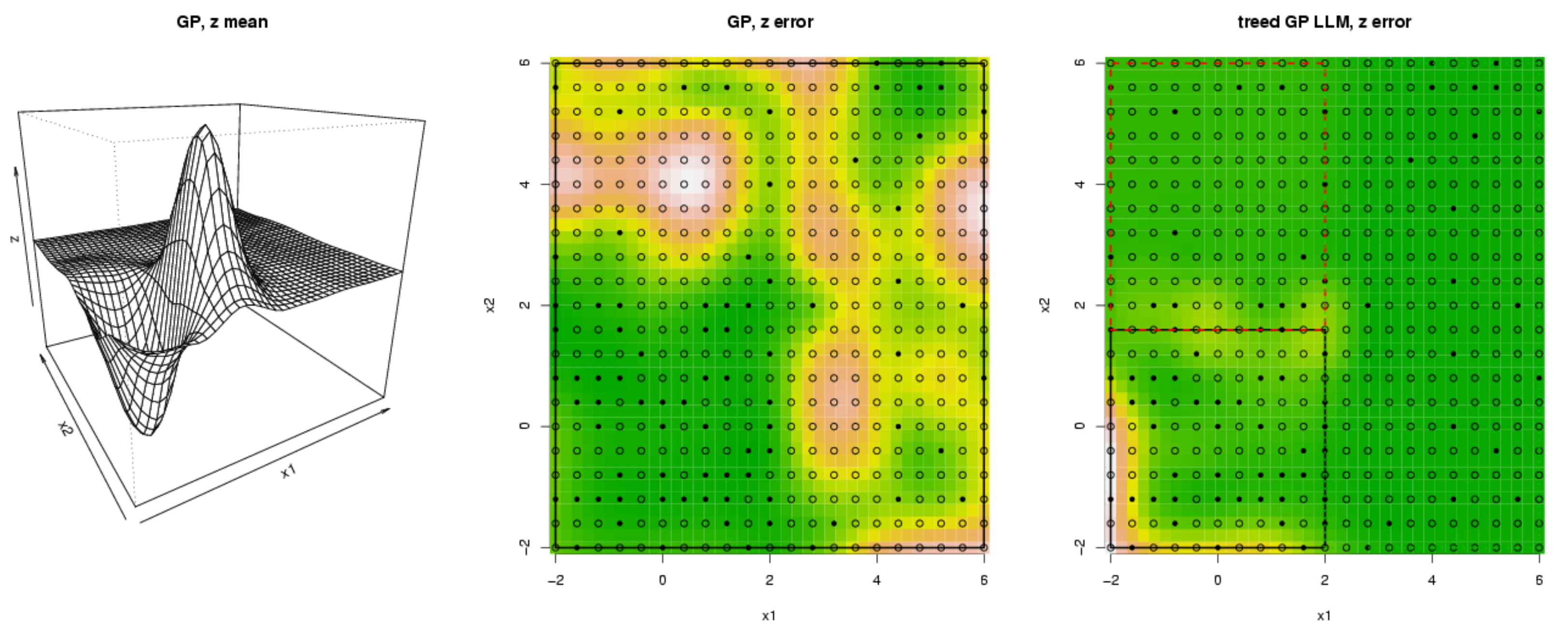


```
sin.btgp <- btgp(X=X, Z=Z)
plot(sin.btgp, layout="surf")
```

3. Nonstationary predictive uncertainty

- Predictive uncertainty of stationary GPs is always largest where sampling is lowest. Partitioning (e.g., treed GP) allows for more reasonable estimates.

```
exp2d.data <- exp2d.rand()
X <- exp2d.data$X; XX <- exp2d.data$XX
Z <- exp2d.data$Z
```

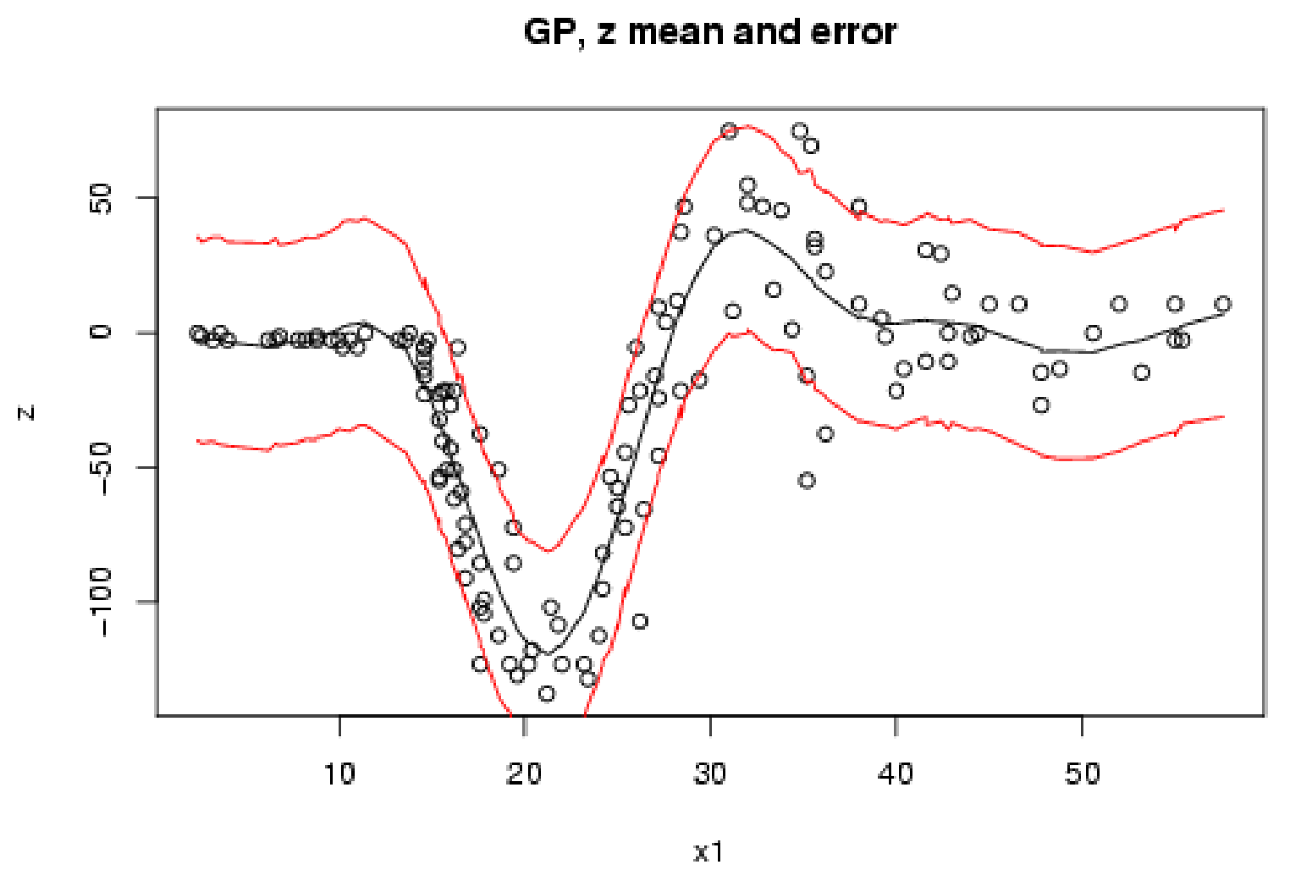


```
exp.bgp <- bgp(X=X, Z=Z, XX=XX); plot(exp.bgp)
```

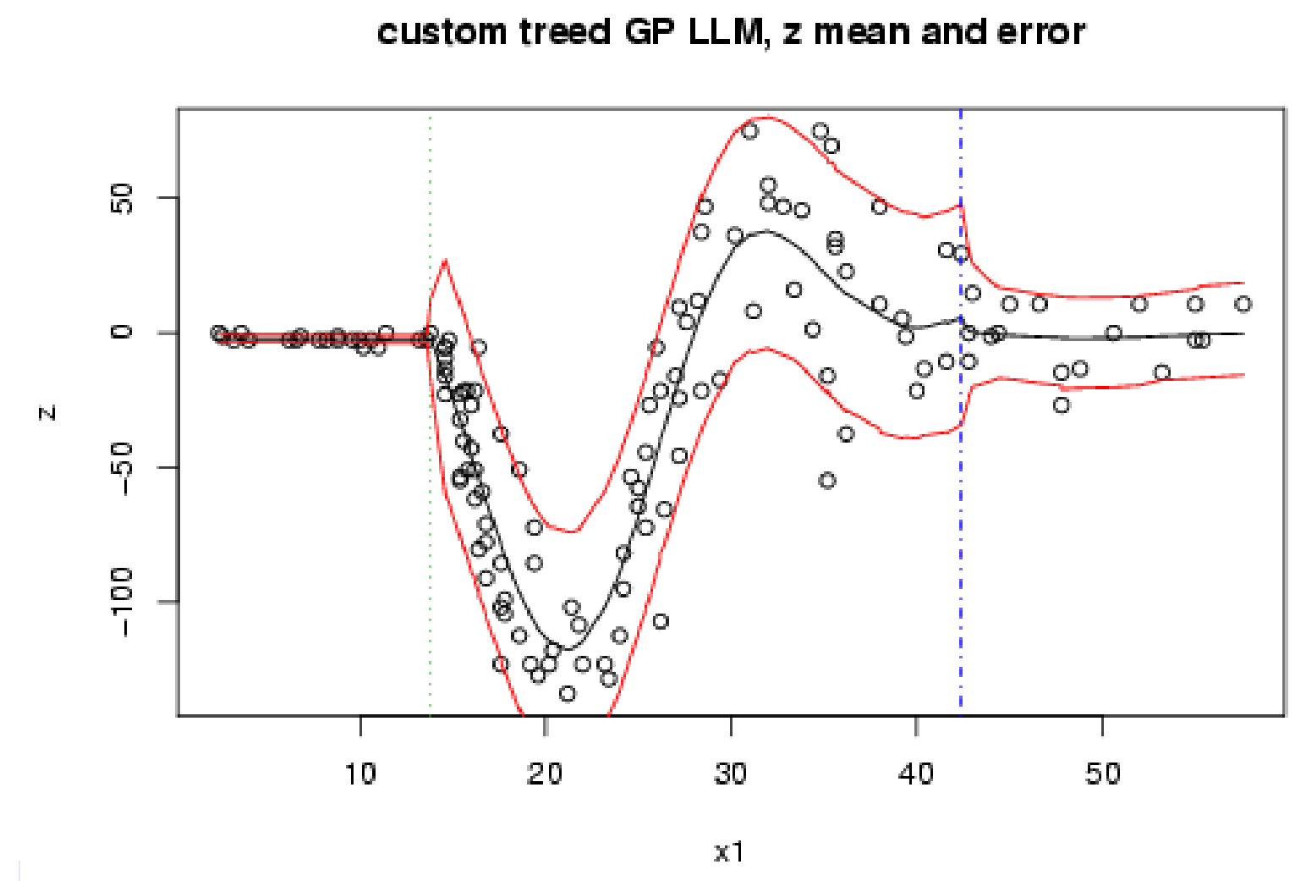
```
exp.bgpllm <- btgpllm(X=X, Z=Z, XX=XX)
plot(exp.btgpllm, layout="as")
```

- A key feature of the treed GP is its ability to learn about input-dependent noise. A classic example is the Motorcycle data, in R.

```
library(MASS); X <- mcycle[,1]; Z <- mcycle[,2]
```



```
moto.bgp <- bgp(X=X, Z=Z)
plot(moto.bgp, layout="surf")
```



```
p <- tgp.default.params(2)
p$bprior <- "b0"; p$nug.p <- c(1,.1,10,.1)
moto.tgp <- tgp(X=X, Z=Z, params=p
                BTE=c(2000,22000,2))
```

4. Semiparametric Modeling

- The 1st Friedman data set has 10 covariates, but the response $E(Z|\mathbf{x}) = \mu = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$ depends only on the 1st 5, combining nonlinear, linear, and irrelevant effects
- The following R code compares Bayesian linear CART to the (non-treed) GP LLM in terms of RMSE

```
f <- friedman.1.data(200); X <- f[,1:10]; Z <- f$Y
ff <- friedman.1.data(1000); XX <- ff[,1:10]
fr.btlm <- btlm(X=X, Z=Z, XX=XX, tree=c(0.95,2,10), m0r1=TRUE)
fr.bgpllm <- bgpllm(X=X, Z=Z, XX=XX, m0r1=TRUE)
sqrt(mean((fr.btlm$ZZ.mean - ff$Ytrue)^2)) #> 2.191923
sqrt(mean((fr.bgpllm$ZZ.mean - ff$Ytrue)^2)) #> 0.4157481
```

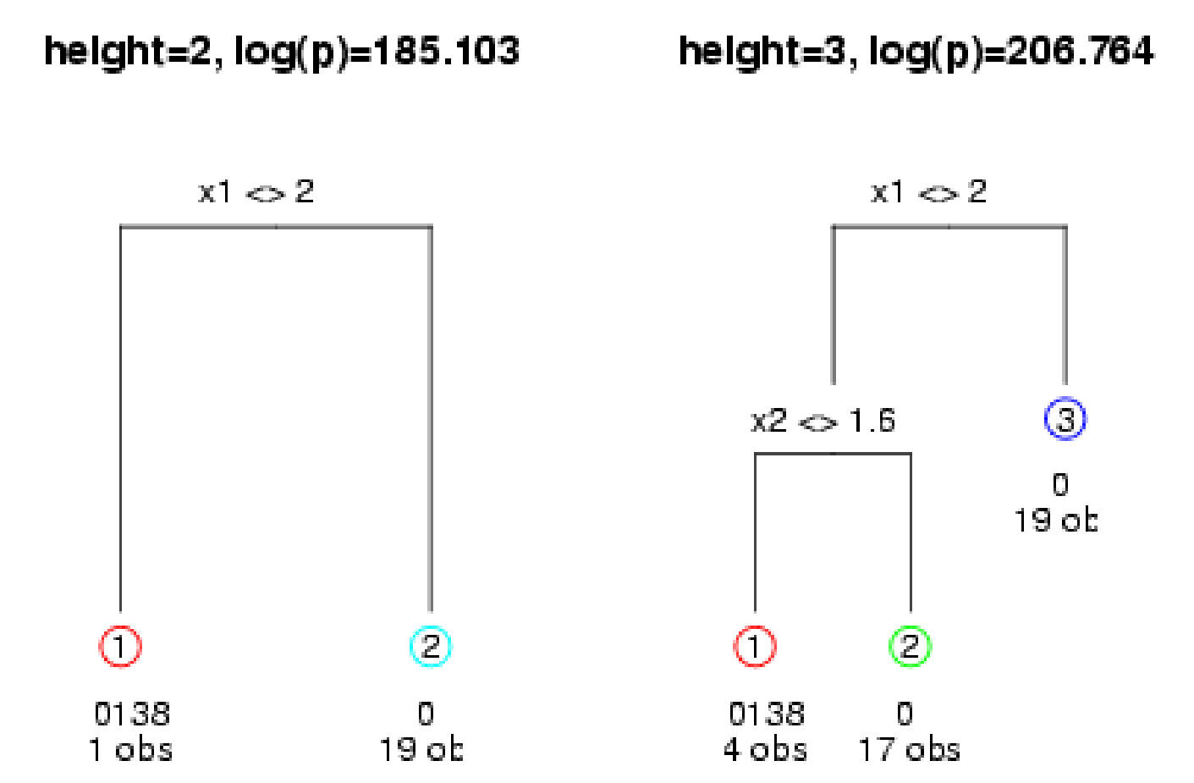
5. Sequential Design

Consider sequentially designing an experiment for the exp data (*left*), by subsampling from candidates:

```
Xcand <- exp2d.data$XX
```

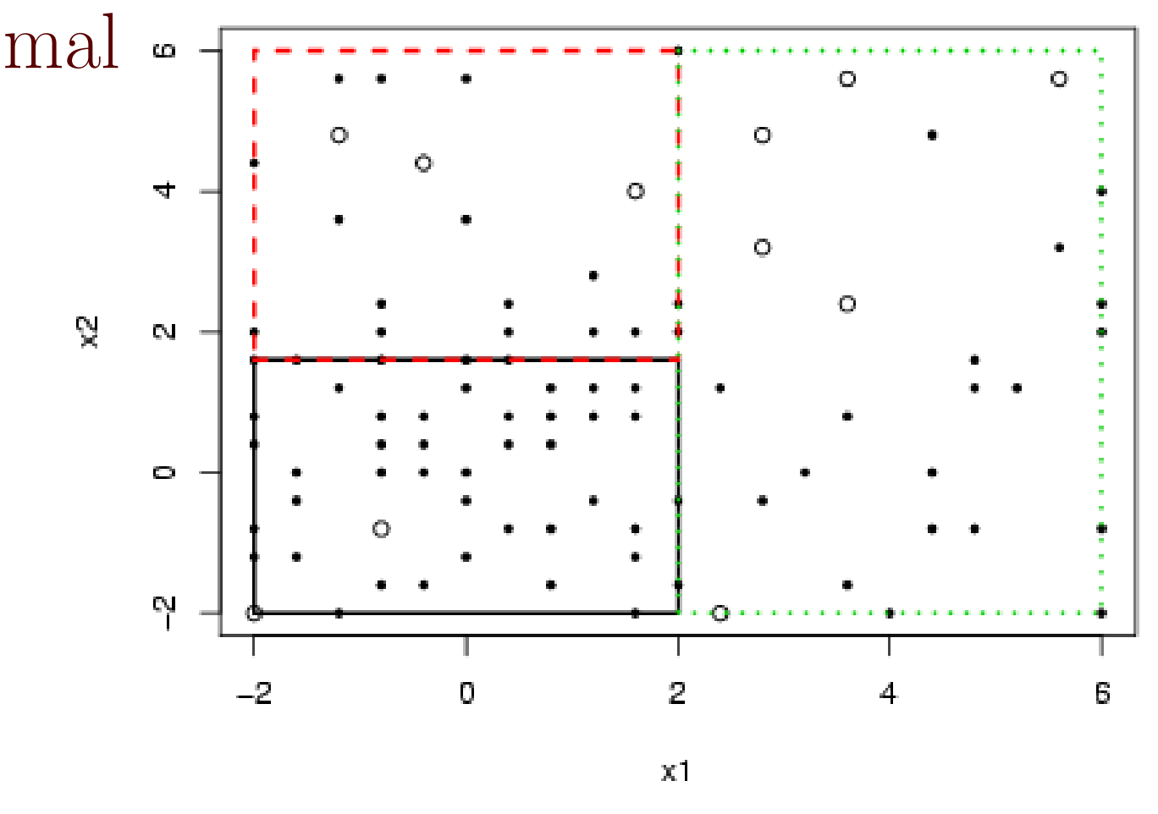
- Start by finding the MAP tree $\hat{\mathcal{T}}$

```
exp1 <- btgpllm(X=X, Z=Z,
                pred.n=FALSE,
                corr="exp")
tgp.trees(exp1)
```



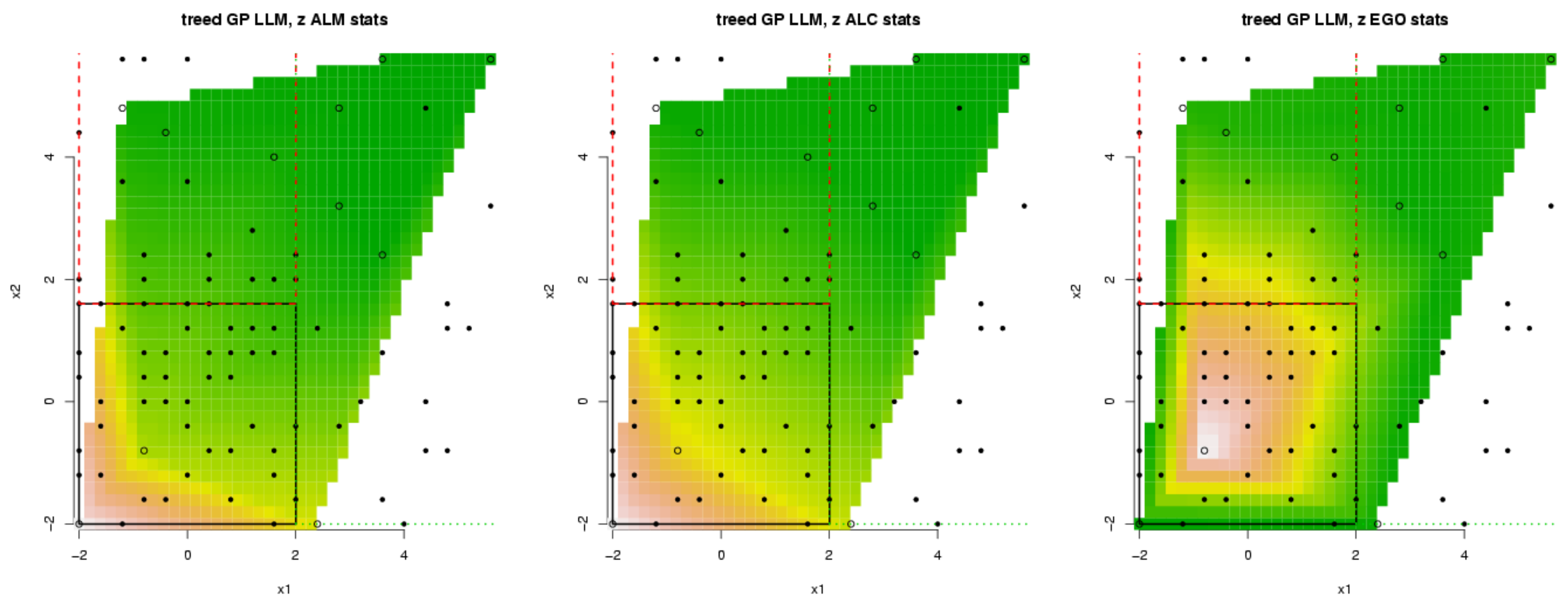
- Then, obtain a sequential treed D-optimal design, i.e., for each GP in $\hat{\mathcal{T}}$

```
XX <- tgp.design(10, Xcand, exp1)
plot(exp1$X, pch=19, cex=0.5)
tgp.plot.parts.2d(exp1$parts)
points(XX)
```



- Use the treed GP model, again and gather "adaptive sampling" statistics:
  - ALM: pred. var $\sigma^2(\tilde{\mathbf{x}})$
  - ALC: expected reduction in pred. var $\Delta\sigma^2(\tilde{\mathbf{x}}) = \int_{\mathbf{y}} \hat{\sigma}_{\mathbf{y}}^2 - \hat{\sigma}_{\mathbf{y}}^2(\tilde{\mathbf{x}})$
  - EGO: expected global optimization (min) $E[\max(f_{\min} - Z(\tilde{\mathbf{x}}), 0)]$

```
exp1.btgpllm <- btgpllm(X=X, Z=Z, XX=XX, corr="exp", ego=TRUE, ds2x=TRUE)
```



```
par(mfrow=c(1,3), bty="n"); main="treed GP LLM"
plot(exp1.btgpllm, main=main, method="akima", layout="as", as="alm")
plot(exp1.btgpllm, main=main, method="akima", layout="as", as="alc")
plot(exp1.btgpllm, main=main, method="akima", layout="as", as="ego")
```

- ALM & ALC agree, whereas EGO recommends sampling closer to the min

6. Closing thoughts

- tgp is licensed under the LGPL, and available from CRAN

  http://www.cran.r-project.org/src/contrib/Descriptions/tgp.html
  http://www.ams.ucsc.edu/˜rbgramacy/tgp.html

- Parallelization via Pthreads can be enabled at compile time
- For a tutorial, from R simply enter vignette("tgp")