```
predicting at data locations
correlation: separable power exponential
linear prior: flat
starting d=0.5, nug=0.1, s2=1, tau2=1
starting beta = 0 0
tree[alpha,beta]=[0.25,2], minpart=10
s2[a0,g0]=[5,10]
d[a,b][0,1]=[1,20],[10,10]
nug[a,b][0,1]=[1,1],[1,1]
gamlin = [10,0.2,0.7]
fixing d prior
fixing nug prior
s2 lambda[a0,g0]=[0.2,10]

burn in:
**GROW(1,1)<-1** @depth 0: [0,0.494949], n=(50,50)
**GROW(0,1)<-1** @depth 1: [0,0.242424], n=(25,24)
**PRUNE(1,0)->1** @depth 1: [0,0.292929]
r=1000 corr=[0.00380686] [0] : n = 48 52
r=2000 corr=[0.00433483] [0] : n = 50 50

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0.00240425] [0.0652247] : mh=2 n = 48 52
r=2000 corr=[0.00723703] [0.0488049] : mh=2 n = 49 51
r=3000 corr=[0.00604759] [1.76362] : mh=2 n = 49 51
r=4000 corr=[0.00425486] [0] : mh=2 n = 51 49
r=5000 corr=[0.00351977] [1.00793] : mh=2 n = 48 52
Grow: 0.006006%, Prune: 0.00303%, Change: 0.5336%, Swap: 1%

finished repetition 1 0f 1
removed 2 leaves from the tree
```

The progress indicators show successful *grow* and *prune* operations, and every 1,000 rounds the partitions under the LLM show `corr=[0]`. Figure 8 shows the resulting posterior predictive surface and MAP partition ($\hat{\mathcal{T}}$).

3.3 Synthetic 2-d Exponential Data

The next example involves a two-dimensional input space in $[-2, 6] \times [-2, 6]$. The true response is given by

$$z(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2). \tag{15}$$

A small amount of Gaussian noise (with sd = 0.001) is added. Besides its dimensionality, a key difference between this data set and the last one is that it is not defined using step functions; this smooth function does not have any artificial breaks between regions. The `tgp` package provides a function for data

```
> plot(sin.btgpllm, main = "treed GP LLM,")
```

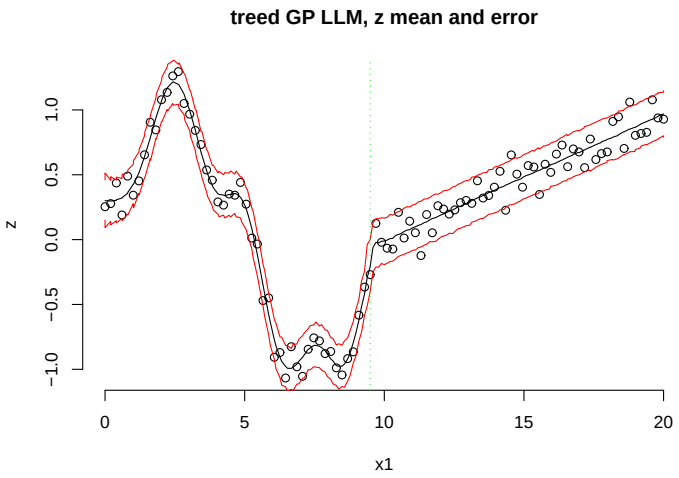


Figure 8: Posterior predictive distribution using `btgpllm` on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition $\hat{\mathcal{T}}$.

subsampled from a grid of inputs and outputs described by (15) which concentrates inputs (`X`) more heavily in the first quadrant where the response is more interesting. Predictive locations (`XX`) are the remaining grid locations.

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

CART is clearly just as inappropriate for this data as it was for the sinusoidal data in the previous section. However, a stationary GP fits this data just fine. After all, the process is quite well behaved. In two dimensions one has a choice between the isotropic and separable correlation functions. Separable is the default in the `tgp` package. For illustrative purposes here, I shall use the isotropic power family.

```
> exp.bgp <- bgp(X = X, Z = Z, XX = XX, corr = "exp")
```

Progress indicators are suppressed. Figure 9 shows the resulting posterior predictive surface under the GP in terms of means (*left*) and variances (*right*). The sampled locations (`X`) are shown as dots on the *right* image plot. Predictive locations (`XX`) are circles. Predictive uncertainty for the stationary GP model is highest where sampling is lowest, despite that the process is very uninteresting there. If any of the surface or perspective plots in the figure have white spaces, or holes, this is becuase of the `akima` bug mentioned in Section 1.1.1. This is not a bug in `tgp`.

```
> plot(exp.bgp, main = "GP,")
```

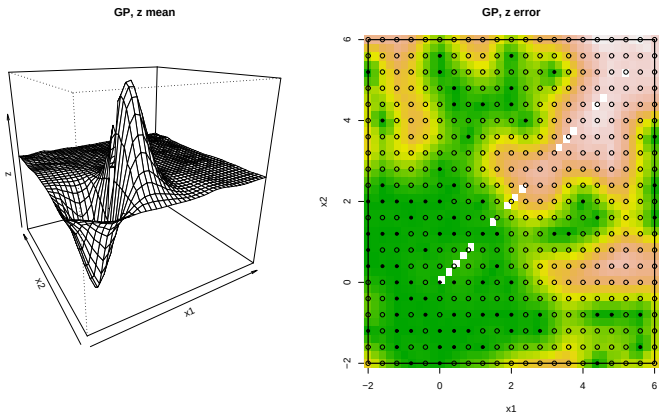


Figure 9: *Left:* posterior predictive mean using `bgp` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

A treed GP seems more appropriate for this data. It can separate out the large uninteresting, essentially zero-response, part of the input space from the interesting part. The result is speedier inference, and region-specific estimates of predictive uncertainty. Chipman et al. recommend restarting the Markov chain a few times in order to better explore the marginal posterior for $\mathcal{T}$ [4]. This becomes more important for higher dimensional inputs, requiring deeper trees. The `tgp` default is `R = 1`, i.e., one chain with no restarts. Here two chains, with one restarts, are obtained using `R=2`.

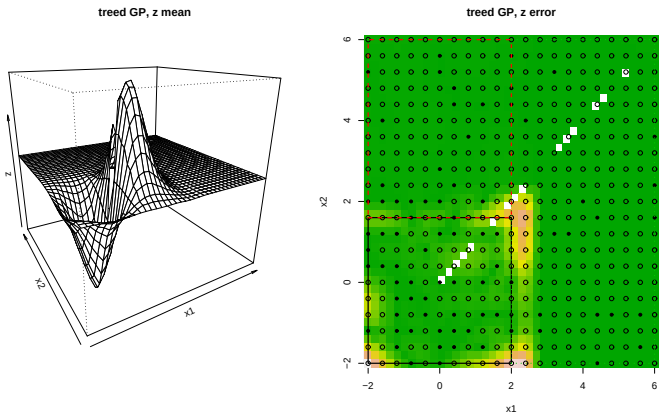
```
> exp.btgp <- btgp(X = X, Z = Z, XX = XX, corr = "exp",
+     R = 2)
```

Figure 6 shows the resulting posterior predictive surface (*top*) and trees (*bottom*). Typical runs of the treed GP on this data find two, and if lucky three, partitions. As might be expected, jumping to the LLM for the uninteresting, zero-response, part of the input space can yield even further speedups [10].

```
> exp.btgpllm <- btgpllm(X = X, Z = Z, XX = XX, corr = "exp",
+     R = 2)

state = 201 588 370 ignored, using R RNG
n=80, d=2, nn=361, BTE=(2000,7000,2), R=2, linburn=0
predicting at data locations
correlation: isotropic power exponential
linear prior: flat
starting d = 0.5 0.5
starting nug=0.1, s2=1, tau2=1
starting beta = 0 0 0
```

```
> plot(exp.btgp, main = "treed GP,")
```




```
> tgp.trees(exp.btgp)
```

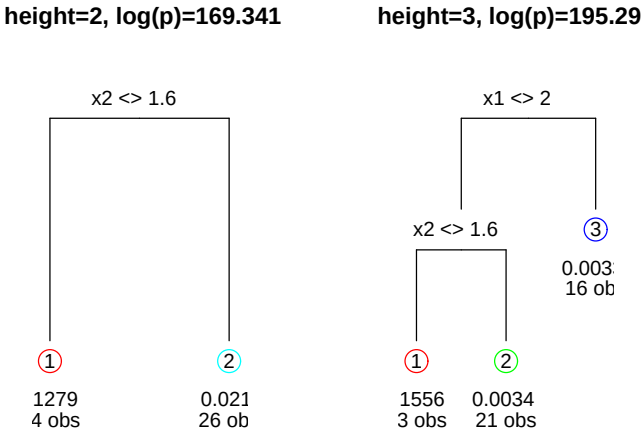


Figure 10: *Left:* posterior predictive mean using `btgp` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

```
tree[alpha,beta]=[0.25,2], minpart=10
s2[a0,g0]=[5,10]
d[a,b][0,1]=[1,20],[10,10]
nug[a,b][0,1]=[1,1],[1,1]
gamlin = [10,0.2,0.7]
fixing d prior
```

```
fixing nug prior
s2 lambda[a0,g0]=[0.2,10]

burn in:
**GROW(0,1)<-0** @depth 0: [1,0.4], n=(50,30)
**PRUNE(0,1)->1** @depth 0: [1,0.35]
r=1000 corr=0.019141 : n = 80
r=2000 corr=0.0208016 : n = 80

Obtaining samples (nn=361 predictive locations):
**GROW(1,1)<-1** @depth 0: [1,0.5], n=(62,18)
r=1000 corr=0.0178545 0.904364 : mh=2 n = 62 18
**GROW(1,0)<-1** @depth 1: [0,0.5], n=(43,11)
r=2000 corr=0.0224447 1.1124 0.0198283 : mh=3 n = 43 11 26
r=3000 corr=0.0270723 0(1.66816) 0.061071 : mh=3 n = 50 14 16
r=4000 corr=0.0236126 0.199734 0(1.13711) : mh=3 n = 50 12 18
r=5000 corr=0.0194921 0(1.39435) 0(0.0761597) : mh=3 n = 50 15 15
Grow: 0.008264%, Prune: 0.004292%, Change: 0.2199%, Swap: 0.2895%

finished repetition 1 0f 2
removed 3 leaves from the tree

burn in:
**GROW(0,0)<-0** @depth 0: [0,0.5], n=(64,16)
**GROW(0,0)<-0** @depth 1: [1,0.5], n=(50,14)
**GROW(1,0)<-0** @depth 2: [0,0.1], n=(16,34)
**PRUNE(1,0)->1** @depth 2: [0,0.1]
r=1000 corr=0.0219221 0(0.791974) 0(0.764177) : mh=3 n = 50 12 18
r=2000 corr=0.0226169 0(1.60567) 0(0.656538) : mh=3 n = 50 12 18

Obtaining samples (nn=361 predictive locations):
r=1000 corr=0.0253895 0.0437966 1.78918 : mh=3 n = 43 21 16
r=2000 corr=0.0176578 0.0134262 0.714019 : mh=3 n = 43 21 16
r=3000 corr=0.021045 0.0128268 0.769922 : mh=3 n = 43 21 16
r=4000 corr=0.0215844 0.02203 0(1.34104) : mh=3 n = 43 21 16
r=5000 corr=0.0196245 0.00991285 0.0674148 : mh=3 n = 43 22 15
Grow: 0.008511%, Prune: 0.003431%, Change: 0.2139%, Swap: 0.2624%

finished repetition 2 0f 2
removed 3 leaves from the tree
```

Progress indicators show where the LLM (`corr=0(`d`)`) or the GP is active. Figure 11 show how similar the resulting posterior predictive surfaces are.

```
> plot(exp.btgpllm, main = "treed GP, LLM")
```

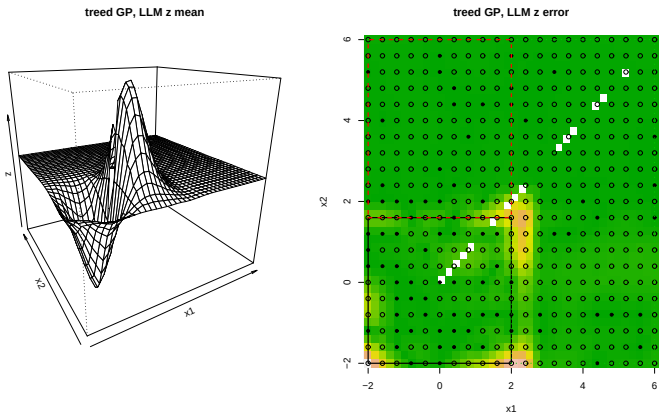


Figure 11: *Left:* posterior predictive mean using `btgpllm` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

3.4 Motorcycle Accident Data

The Motorcycle Accident Dataset [21] is a classic nonstationary data set used in recent literature [18] to demonstrate the success of nonstationary models. The data consists of measurements of acceleration of the head of a motorcycle rider as a function of time in the first moments after an impact. In addition to being nonstationary, the data has input-dependent noise, which makes it useful for illustrating how the treed GP model handles this nuance. There are at least two, and perhaps three regions where the response exhibits different behavior both in terms of the correlation structure and noise level.

The data is included as part of the `MASS` library in `R`.

```
> library(MASS)
```

Figure 12 shows how a stationary GP is able to capture the nonlinearity in the response but fails to capture the input dependent noise, and increased smoothness (perhaps linearity) in parts of the input space.

```
> moto.bgp <- bgp(X = mcycle[, 1], Z = mcycle[, 2], m0r1 = TRUE)
```

Since the responses in this data have a wide range, it helps to translate and rescale them so that they have a mean of zero, and a range of one. The `m0r1` argument to `b*` and `tgp` functions automates this procedure. All progress indicators are surpressed for this example.

A Bayesian Linear CART model is able to capture the input dependent noise but fails to capture the waviness of the "whiplash"—center— segment of th the response.

```
> moto.btlm <- btlm(X = mcycle[, 1], Z = mcycle[, 2], m0r1 = TRUE)
```