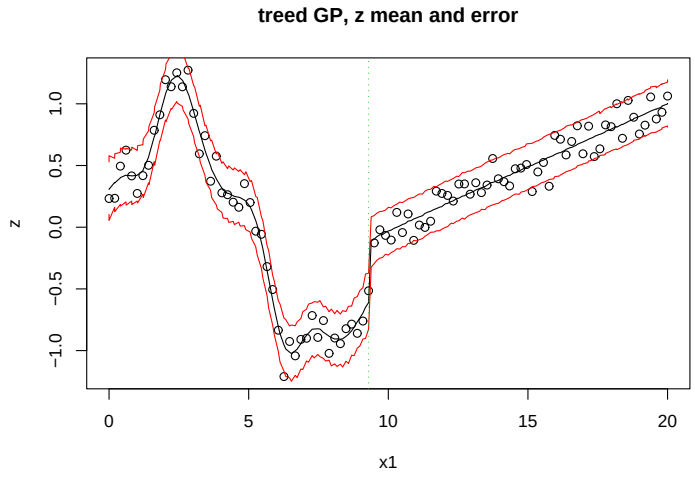



```
> tgp.trees(sin.btgp)
```

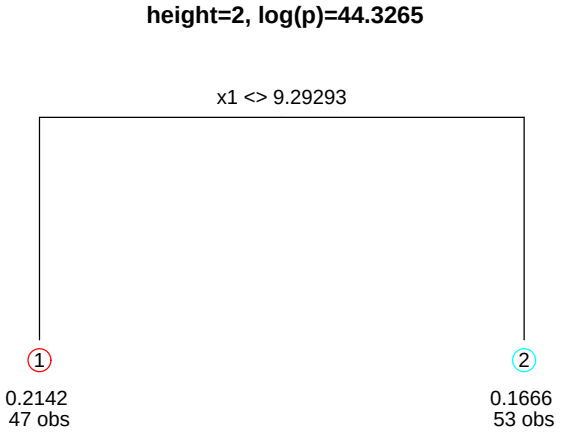


Figure 7: *Top:* Posterior predictive distribution using `btgp` on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition ($\hat{\mathcal{T}}$); *Bottom* MAP trees for each height encountered in the Markov chain.

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 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
```

Linear 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")
```

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

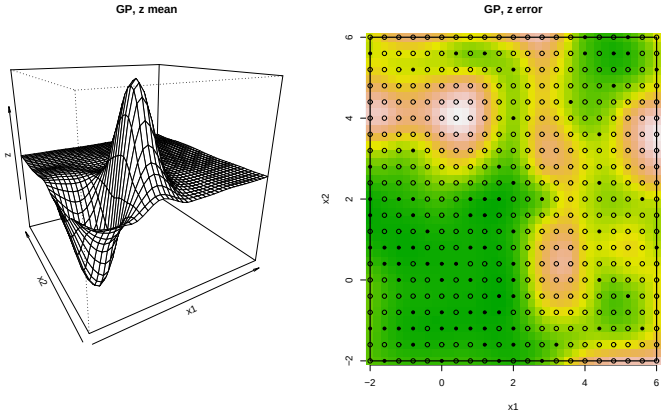


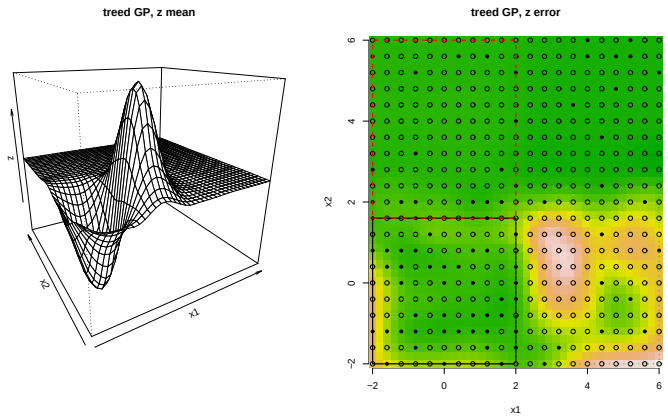
Figure 8: *Left:* posterior predictive mean using `bgp` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

Progress indicators are suppressed. Figure 8 shows the resulting posterior predictive surface under the GP in terms of means (*left*) and variances (*right*) in the default layout. 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.

A treed GP seems more appropriate for this data. It can separate out the large uninteresting oart of the input space from the interesting part. The result is speedier inference and region-specific estimates of predictive uncertainty.

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

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




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

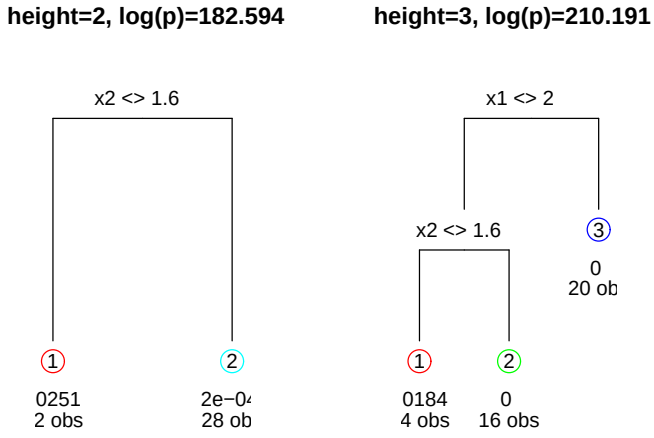


Figure 9: *Top-Left:* posterior predictive mean using `btgp` on synthetic exponential data; *top-right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles). `Bottom:` MAP trees of each height encountered in the Markov chain with $\hat{\sigma}^2$ and the number of observations n at the leaves.

Figure 9 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]. Also, Chipman et al. recommend restarting the Markov chain a few times in order to better explore the marginal posterior for $\mathcal{T}$ [4]. This can be important for higher dimensional inputs requiring deeper trees. The `tgp` default is `R = 1`, i.e., one chain with no restarts. Here two chains—one restart—are obtained using `R = 2`.

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

n=80, d=2, nn=361
BTE=(2000,7000,2), R=2, linburn=0
preds: data
tree[alpha,beta,minpart]=[0.25,2,10]
linear prior: flat
s2[a0,g0]=[5,10]
s2 lambda[a0,g0]=[0.2,10]
corr prior: isotropic power
nug[a,b][0,1]=[1,1],[1,1]
nug prior fixed
gamlin=[10,0.2,0.7]
d[a,b][0,1]=[1,20],[10,10]
d prior fixed

burn in:
**GROW** @depth 0: [1,0.35], n=(45,35)
**PRUNE** @depth 0: [1,0.4]
r=1000 corr=0.018638 : n = 80
**GROW** @depth 0: [0,0.5], n=(60,20)
r=2000 corr=0.0206168 0.114274 : n = 60 20

Obtaining samples (nn=361 predictive locations):
**GROW** @depth 1: [1,0.5], n=(50,10)
r=1000 corr=0.0226481 0(0.84922) 0(0.602632) : mh=3 n = 50 10 20
r=2000 corr=0.0230381 0(0.770479) 0.965604 : mh=3 n = 44 16 20
r=3000 corr=0.0209883 0(1.25422) 0(0.681911) : mh=3 n = 50 10 20
r=4000 corr=0.019928 0.785746 0.0729928 : mh=3 n = 44 16 20
r=5000 corr=0.0198103 0.813851 0.854068 : mh=3 n = 44 16 20
Grow: 0.008333%, Prune: 0.003597%, Change: 0.04887%, Swap: 0.178%

finished repetition 1 of 2
removed 3 leaves from the tree

burn in:
**GROW** @depth 0: [0,0.5], n=(60,20)
```

```
**GROW** @depth 1: [0,0.1], n=(15,36)
**PRUNE** @depth 1: [0,0.05]
**GROW** @depth 1: [1,0.5], n=(50,10)
r=1000 corr=0.0210184 0.0147621 0.94543 : mh=3 n = 44 16 20
r=2000 corr=0.0247164 0.0838776 1.39391 : mh=3 n = 44 16 20

Obtaining samples (nn=361 predictive locations):
r=1000 corr=0.0211024 1.32905 1.41035 : mh=3 n = 44 16 20
r=2000 corr=0.0203233 0(1.2382) 0(0.0643748) : mh=3 n = 50 10 20
r=3000 corr=0.021733 0(0.625732) 0.0239015 : mh=3 n = 50 10 20
r=4000 corr=0.0213785 0.0360728 0.108831 : mh=3 n = 44 16 20
r=5000 corr=0.0201093 0.0127475 0.0738182 : mh=3 n = 44 16 20
Grow: 0.007979%, Prune: 0.003356%, Change: 0.04622%, Swap: 0.1706%

finished repetition 2 of 2
removed 3 leaves from the tree


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

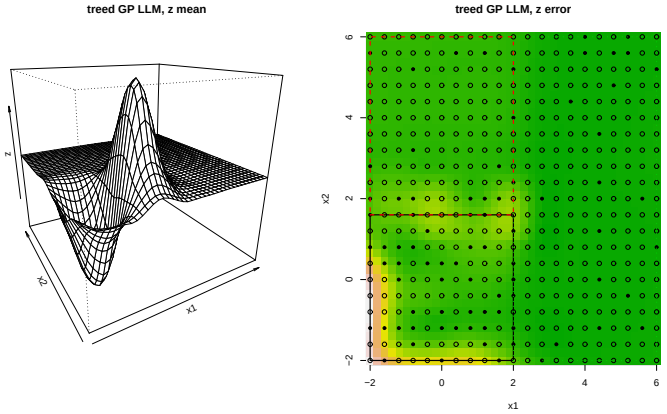
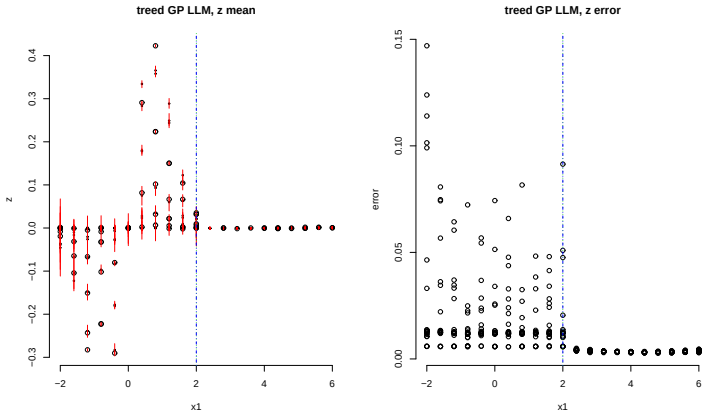


Figure 10: *Left:* posterior predictive mean using `btgpllm` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

Progress indicators show where the LLM (`corr=0(`d`)`) or the GP is active. Figure 10 show how similar the resulting posterior predictive surfaces are compared to the treed GP (without LLM).

Finally, viewing 1-d projections of `tgp`-class output is possible by supplying a 1-vector `proj` argument to the `plot.tgp`. Figure 11 shows the two projections for `exp.btgpllm`. In the *left* surface plots the open circles indicate the mean of posterior priedictive distribution. Red lines show the 90% intervals, the norm of which are shown on the *right*.

```
> plot(exp.btgpllm, main = "treed GP LLM,", proj = c(1))
```




```
> plot(exp.btgpllm, main = "treed GP LLM,", proj = c(2))
```

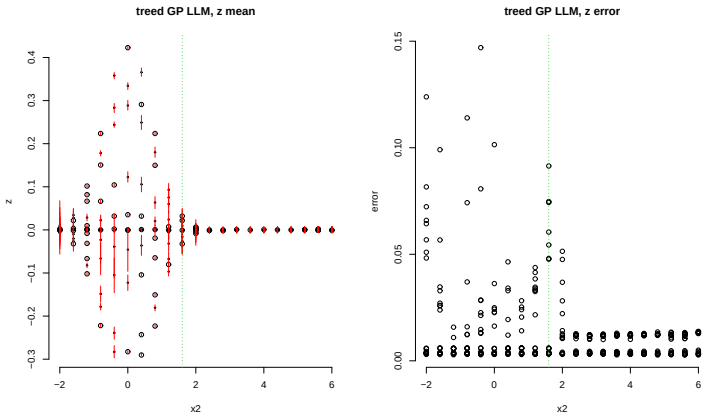


Figure 11: 1-d projections of the posterior predictive surface (*left*) and normed predictive intervals (*right*) of the 1-d tree GP LLM analysis of the synthetic exponential data. The *top* plots show projection onto the first input, and the *bottom* ones show the second.

3.4 Motorcycle Accident Data

The Motorcycle Accident Dataset [22] is a classic nonstationary data set used in recent literature [19] to demonstrate the success of nonstationary models. The data consists of measurements of the 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—perhaps three—three regions where the response exhibits different