3.2 1-d Synthetic Sine Data

Consider 1-dimensional simulated data which is partly a mixture of sines and cosines, and partly linear.

$$z(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right) & x < 10 \\ x/10 - 1 & \text{otherwise} \end{cases} \tag{14}$$

The R code below obtains $N = 100$ evenly spaced samples from this data in the domain $[0, 20]$, with noise added to keep things interesting.

```
> 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)
```

Some evenly spaced predictive locations XX are also created. By design, the data is clearly nonstationary. Not knowing this, good first model choice for this data might be a GP, since it is clearly nonlinear.

```
> sin.bgp <- bgp(X = X, Z = Z, XX = XX)

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

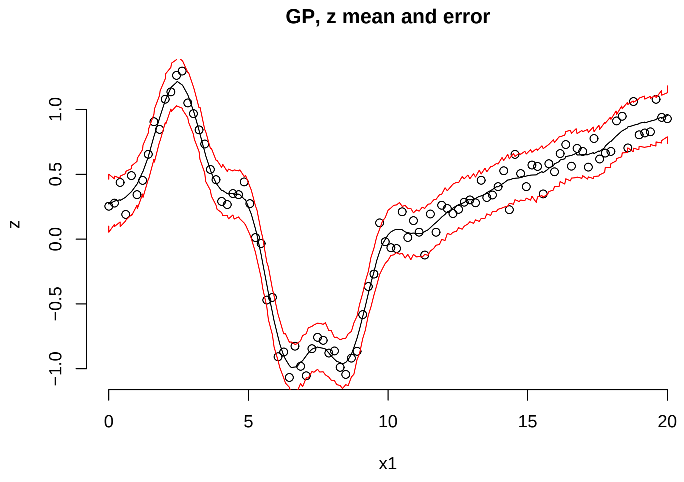


Figure 5: Posterior predictive distribution using bgp on synthetic sinusoidal data: mean and 90% credible interval

Progress indicators have been suppressed. Figure 5 shows the resulting posterior predictive surface under the GP. Notice how the (stationary) GP gets the wiggliness of the sinusoidal region, but fails to capture the smoothness of the linear region. This is becuase the data comes from a process that is nonstationary.

So one might consider a Bayesian CART model instead.

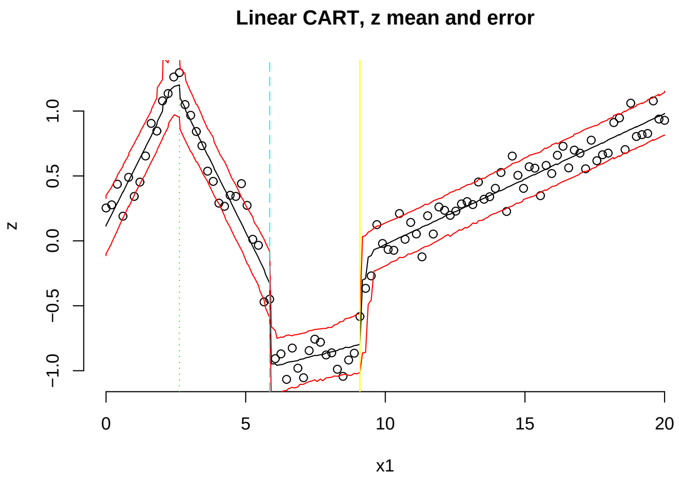
```
> sin.btlm <- btlm(X = X, Z = Z, XX = XX)

state = 888 23 373 ignored, using R RNG
n=100, d=1, nn=99, BTE=(2000,7000,2), R=1, linburn=0
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 = [-1,0.2,0.7]
fixing d prior
fixing nug prior
s2 lambda[a0,g0]=[0.2,10]

burn in:
**GROW(0,0)<-0** @depth 0: [0,0.474747], n=(48,52)
**GROW(0,0)<-0** @depth 1: [0,0.161616], n=(17,29)
**GROW(0,0)<-0** @depth 2: [0,0.272727], n=(11,18)
r=1000 corr=[0] [0] [0] [0] : n = 15 15 16 54
r=2000 corr=[0] [0] [0] [0] : n = 13 17 16 54

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0] [0] [0] [0] : mh=4 n = 10 20 16 54
r=2000 corr=[0] [0] [0] [0] : mh=4 n = 14 16 16 54
r=3000 corr=[0] [0] [0] [0] : mh=4 n = 14 16 16 54
r=4000 corr=[0] [0] [0] [0] : mh=4 n = 14 16 17 53
r=5000 corr=[0] [0] [0] [0] : mh=4 n = 13 17 16 54
Grow: 0.008174%, Prune: 0%, Change: 0.3092%, Swap: 0.7701%

finished repetition 1 Of 1
removed 4 leaves from the tree
```

MCMC progress indicators printed to `stdout` indicate successful *grow* and *prune* operations as they happen, and region sizes n every 1,000 rounds.

Figure 6 shows the resulting posterior predictive surface (*top*) and trees (*bottom*). The MAP partition ($\hat{\mathcal{T}}$) is also drawn onto the surface plot (*top*) in the form of vertical lines. The CART model captures the smoothness of the linear region just fine, but comes up short in the sinusoidal region—doing the best it can with piecewise linear models.

The ideal model for this data is the Bayesian treed GP becuase it can be both smooth and wiggly.

```
> plot(sin.btlm, main = "Linear CART,")
```



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

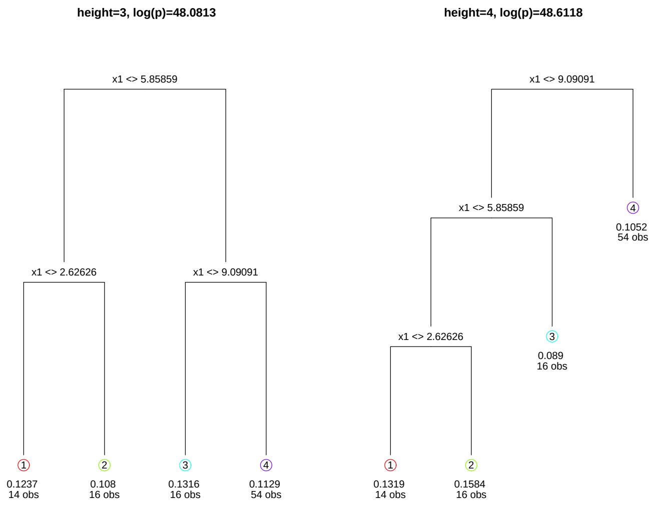
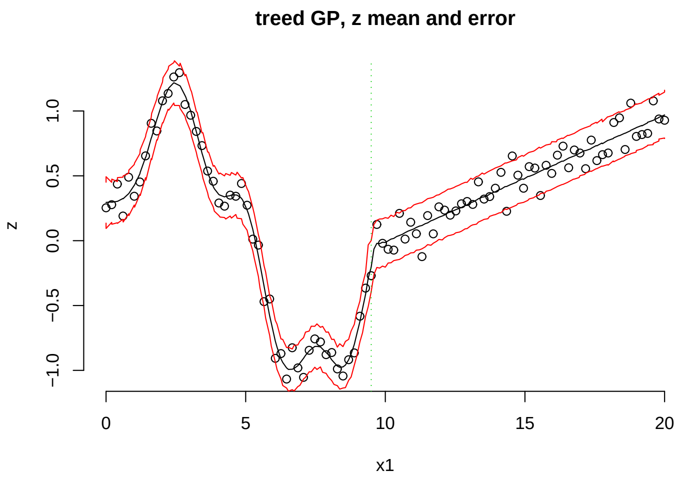


Figure 6: *Top:* Posterior predictive distribution using `btlm` on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition ($\hat{\mathcal{T}}$); *Bottom* MAP trees for each height encountered in the Markov chain.

```
> sin.btgp <- btgp(X = X, Z = Z, XX = XX)
```

Progress indicators have been suppressed. Figure 7 shows the resulting posterior predictive surface (*top*) and trees (*bottom*).



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

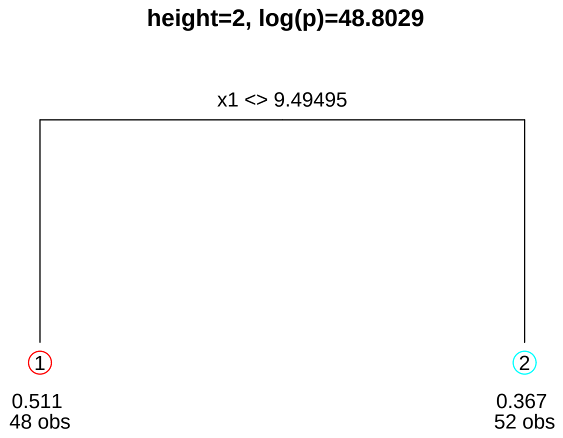


Figure 7: *Top:* Posterior predictive distribution using `btgp` on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition ($\hat{\mathcal{T}}$); *Bottom* MAP trees for each height encountered in the Markov chain.

Finally, speedups can be obtained if the GP is allowed to jump to the LLM [10], since half of the response surface is *very* smooth, or linear.

```
> sin.btgpllm <- btgpllm(X = X, Z = Z, XX = XX)
```

```
state = 477 854 556 ignored, using R RNG
n=100, d=1, nn=99, BTE=(2000,7000,2), R=1, linburn=0
```

```
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 Of 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

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

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