3.1 1-d Linear data

Consider data sampled from a linear model.

$$z_i = 1 + 2x_i + \epsilon, \quad \text{where} \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, 0.25^2) \tag{13}$$

The following `R` code takes a sample $\{\mathbf{X}, \mathbf{Z}\}$ of size $N = 50$ from (13). It also chooses $N' = 99$ evenly spaced predictive locations $\tilde{\mathbf{X}} =$ `XX`.

```
> X <- seq(0, 1, length = 50)
> XX <- seq(0, 1, length = 99)
> Z <- 1 + 2 * X + rnorm(length(X), sd = 0.25)
```

Using `tgp` on this data with a Bayesian hierarchical linear model goes as follows:

```
> lin.blm <- blm(X = X, XX = XX, Z = Z)

state = 357 144 97 ignored, using R RNG
n=50, d=1, nn=99, BTE=(1000,4000,3), 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]=[0,0], 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:
r=1000 corr=[0] : n = 50

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0] : mh=1 n = 50
r=2000 corr=[0] : mh=1 n = 50
r=3000 corr=[0] : mh=1 n = 50

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

The first group of text printed to `stdout` is a summary of the prior parameterization. Then, MCMC progress indicators are printed every 1,000 rounds. The linear model is indicated by `cor=[0]`.The GUI versions of `R`, on `Windows` or

```
> plot(lin.blm, main = "Linear Model,")
> abline(1, 2, lty = 3, col = "blue")
```

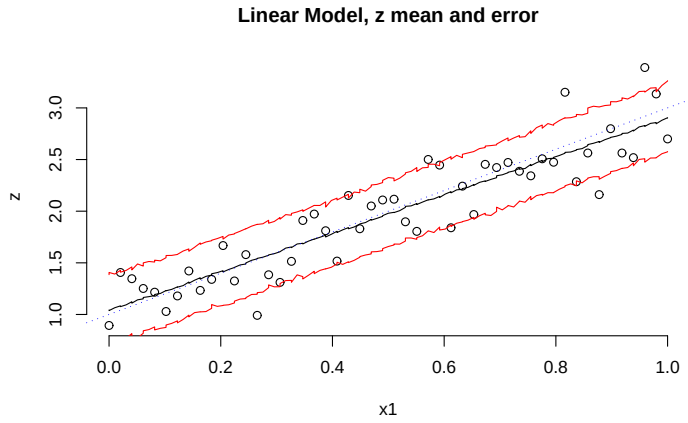

Figure 3: Posterior predictive distribution using `blm` on synthetic linear data: mean and 90% credible interval. The actual generating lines are shown as blue-dotted.

`MacOS X`, usually buffers `stdout`, rendering this feature essentially useless. In terminal versions, e.g. `Unix`, the progress indicators can give a sense of when the code will finish. Also note that a user cannot interact while the `C` code is running. This will be changed in future versions.

The generic `plot` method can be used to visualize the fitted posterior predictive surface in terms of means and credible intervals. Figure 3 shows how to do it, and what you get.

If, say, you were unsure about the dubious "linearness" of this data, you might try a GP LLM (using `btgpllm`) and let a more flexible model speak as to the linearity of the process.

```
> lin.gpllm <- bgpllm(X = X, XX = XX, Z = Z)

state = 474 363 977 ignored, using R RNG
n=50, 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]=[0,0], 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:
r=1000 corr=[0] : n = 50
r=2000 corr=[0] : n = 50

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0.788014] : mh=1 n = 50
r=2000 corr=[0] : mh=1 n = 50
r=3000 corr=[0] : mh=1 n = 50
r=4000 corr=[0.982654] : mh=1 n = 50
r=5000 corr=[0] : mh=1 n = 50

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

```
> plot(lin.gpllm, main = "GP LLM,")
> abline(1, 2, lty = 4, col = "blue")
```

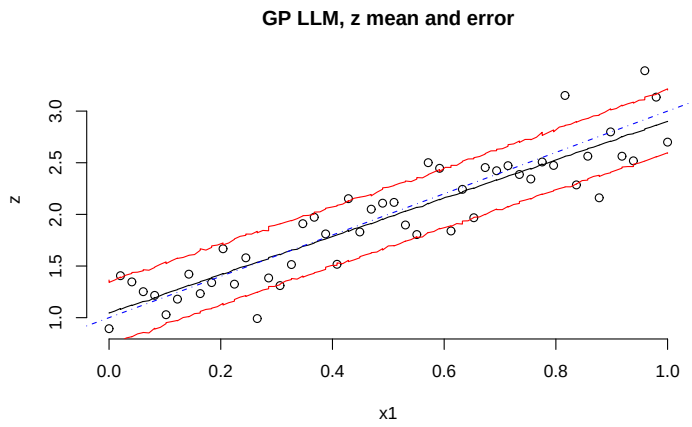

Figure 4: Posterior predictive distribution using `bgpllm` on synthetic linear data: mean and 90% credible interval. The actual generating lines are shown as blue-dotted.

Whenever the progress indicators show `corr[0]` the process is under the LLM in that round, and the GP otherwise. A plot of the resulting surface is shown in Figure 4 for comparison. Since the data is linear, the resulting predictive surfaces should look strikingly similar to one another. On occasion, the GP LLM may find some bendy-ness in the surface. This happens rarely with samples as large as $N = 50$, but is quite a bit more common for $N < 20$.