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

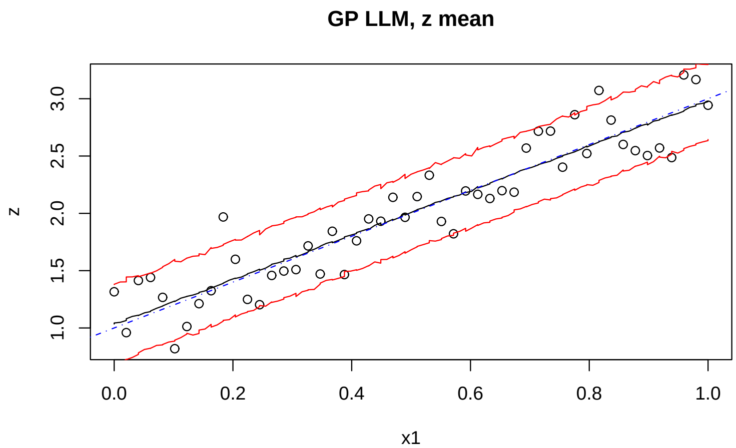

Figure 5: Posterior predictive distribution using `bgpllm` on synthetic linear data: mean and 90% credible interval. The actual generating lines are shown as blue-dotted.

To see the proportion of time the Markov chain spent in the LLM requires the gathering of traces (Appendix B.1). For example

```
> lin.gpllm.tr <- bgpllm(X = X, XX = 0.5, Z = Z, pred.n = FALSE,
+     trace = TRUE, verb = 0)
> mla <- mean(lin.gpllm.tr$trace$linarea$la)
> mla

[1] 0.96
```

shows that the average area under the LLM is 0.96. Progress indicators are suppressed with `verb=0`. Alternatively, the probability that input location `xx` = 0.5 is under the LLM is given by

```
> 1 - mean(lin.gpllm.tr$trace$XX[[1]]$b1)

[1] 0.96
```

This is the same value as the area under the LLM since the process is stationary (i.e., there is no treed partitioning).

3.2 1-d Synthetic Sine Data

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

$$z(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right) & x < 10 \\ x/10 - 1 & \text{otherwise} \end{cases} \tag{16}$$

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. Some evenly spaced predictive locations XX are also created.

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

By design, the data is clearly nonstationary in its mean. Perhaps not knowing this, a good first model choice for this data might be a GP.

```
> sin.bgp <- bgp(X = X, Z = Z, XX = XX, verb = 0)
```

```
> plot(sin.bgp, main = "GP,", layout = "surf")
> lines(X, Ztrue, col = 4, lty = 2, lwd = 2)
```

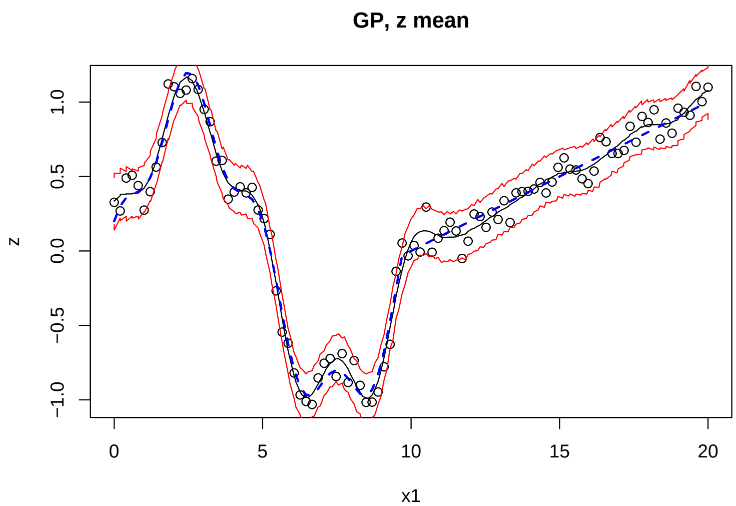

Figure 6: Posterior predictive distribution using bgp on synthetic sinusoidal data: mean and 90% pointwise credible interval. The true mean is overlayed with a dashed line.

Figure 6 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. The true mean (16) is overlayed with a dashed line.

So one might consider a Bayesian treed linear model (LM) instead.

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

```
burn in:
**GROW** @depth 0: [0,0.424242], n=(43,57)
**GROW** @depth 1: [0,0.252525], n=(26,19)
**GROW** @depth 2: [0,0.131313], n=(14,13)
r=1000 d=[0] [0] [0] [0]; n=(11,19,17,53)
r=2000 d=[0] [0] [0] [0]; n=(11,17,19,53)

Sampling @ nn=99 pred locs:
r=1000 d=[0] [0] [0] [0]; mh=3 n=(15,14,18,53)
r=2000 d=[0] [0] [0] [0]; mh=4 n=(14,14,19,53)
r=3000 d=[0] [0] [0] [0]; mh=4 n=(12,16,19,53)
r=4000 d=[0] [0] [0] [0]; mh=4 n=(13,16,18,53)
r=5000 d=[0] [0] [0] [0]; mh=4 n=(13,15,19,53)
Grow: 0.8403%, Prune: 0%, Change: 36.3%, Swap: 84%
```

MCMC progress indicators show successful *grow* and *prune* operations as they happen, and region sizes n every 1,000 rounds. Specifying `verb=3`, or higher will show echo more successful tree operations, i.e., *change*, *swap*, and *rotate*.

Figure 7 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 treed LM 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 because it can be both smooth and wiggly.

```
> sin.btgp <- btgp(X = X, Z = Z, XX = XX, verb = 0)
```

Figure 8 shows the resulting posterior predictive surface (*top*) and MAP $\hat{\mathcal{T}}$ with height=2.

Finally, speedups can be obtained if the GP is allowed to jump to the LLM [15], since half of the response surface is *very* smooth, or linear. This is not shown here since the results are very similar to those above, replacing `btgp` with `btgpllm`. Each of the models fit in this section is a special case of the treed GP LLM, so a model comparison is facilitated by fitting this more general model. The example in the next subsection offers such a comparison for 2-d data. A followup in Appendix B.1 shows how to use parameter traces to extract the posterior probability of linearity in regions of the input space.

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

$$z(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2). \tag{17}$$

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