```
r=2000 corr=0.0233133 0(0.67959) 0.0167081 : mh=3 n = 44 17 19
r=3000 corr=0.021841 0(0.561704) 0.0322474 : mh=3 n = 44 17 19
r=4000 corr=0.0203374 0(1.06896) 0(0.0606924) : mh=3 n = 44 17 19
r=5000 corr=0.0175408 0(1.26111) 0(1.00789) : mh=3 n = 44 17 19
Grow: 0.003979%, Prune: 0%, Change: 0.006608%, Swap: 0.01258%

finished repetition 2 of 2
removed 3 leaves from the tree
```

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

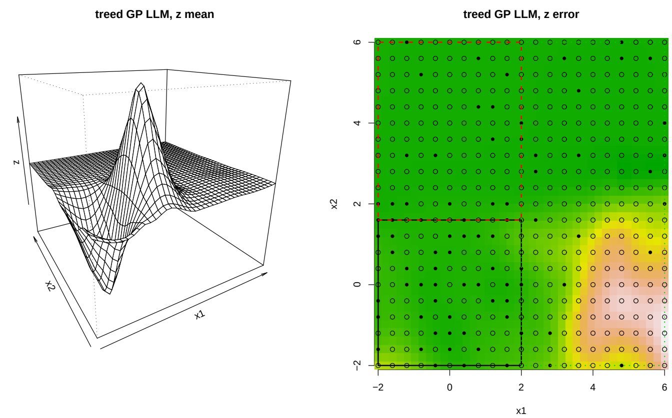


Figure 10: *Left:* posterior predictive mean using `btgpllm` on synthetic exponential data; *right* image plot of posterior predictive variance with data locations `X` (dots) and predictive locations `XX` (circles).

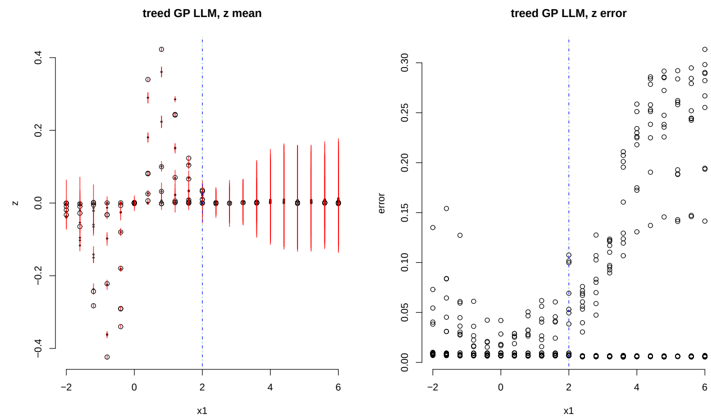
Progress indicators show where the LLM (`corr=0(`d`)`) or the GP is active. Figure 10 show how similar the resulting posterior predictive surfaces are compared to the treed GP (without LLM).

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*.

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 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

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




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

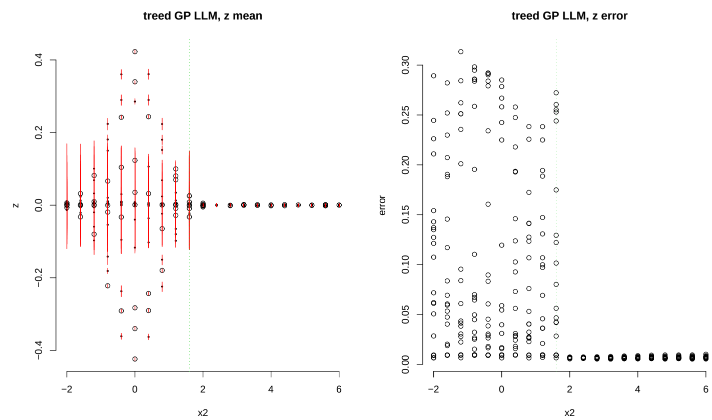


Figure 11: 1-d projections of the posterior predictive surface (*left*) and normed predictive intervals (*right*) of the 1-d tree GP LLM analysis of the synthetic exponential data. The *top* plots show projection onto the first input, and the *bottom* ones show the second.

least two—perhaps three—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. Progress indicators are surpressed.

```
> plot(moto.bgp, main = "GP,", layout = "surf")
```

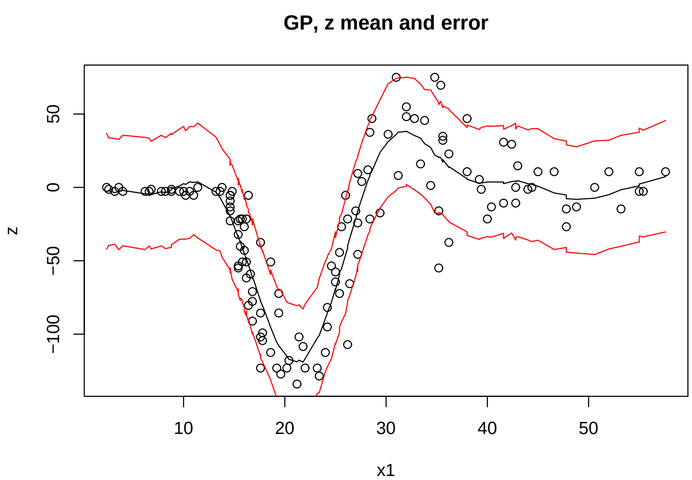


Figure 12: Posterior predictive distribution using `bgp` on the motorcycle accident data: mean and 90% credible interval

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)
```

Figure 13 shows the resulting piecewise linear predictive surface and MAP partition ($\hat{\mathcal{T}}$).

A treed GP model seems appropriate because it can model input dependent smoothness *and* noise. A treed GP LLM is probably most appropriate since the left-hand part of the input space is likely linear. One might further hypothesize that the right-hand region is also linear, perhaps with the same mean as the left-hand region, only with higher noise. The `b*` and `tgp` functions can force an i.i.d. hierarchical linear model by setting `bprior=b0`. Moreover, instead of rescaling the responses with `m0r1`, one might try encoding a mixture prior for the nugget in order to explicitly model region-specific noise. This requires direct usage of `tgp`.