and that the rest are much closer to zero.

```
> apply(trace[, 11:15], 2, mean)
```

```
      beta6       beta7       beta8       beta9      beta10 
-0.23968561  0.37046946  0.13081722 -0.07842566  0.11911203 
```

## 3.6 Adaptive Sampling

In this section, sequential design of experiments, a.k.a. *adaptive sampling*, is demonstrated on the exponential data of Section 3.3. Gathering, again, the data:

```
> exp2d.data <- exp2d.rand(lh = 0, dopt = 10)
> X <- exp2d.data$X
> Z <- exp2d.data$Z
> Xcand <- lhs(1000, rbind(c(-2, 6), c(-2, 6)))
```

In contrast with the data from Section 3.3, which was based on a grid, the above code generates a randomly subsampled $D$–optimal design $\mathbf{X}$ from LH candidates, and random responses $\mathbf{Z}$. As before, design configurations are more densely packed in the interesting region. Candidates $\tilde{\mathbf{X}}$ are from a large LH–sample.

Given some data $\{\mathbf{X}, \mathbf{Z}\}$, the first step in sequential design using `tgp` is to fit a treed GP LLM model to the data, without prediction, in order to infer the MAP tree $\hat{\mathcal{T}}$.

```
> exp1 <- btgpllm(X = X, Z = Z, pred.n = FALSE, corr = "exp",
+     verb = 0)
```

The trees are shown in Figure 16. Then, use the `tgp.design` function to create $D$–optimal candidate designs in each region of $\hat{\mathcal{T}}$. For the purposes of illustrating the `improv` statistic, I have manually added the known (from calculus) global minimum to `XX`.

```
> XX <- tgp.design(200, Xcand, exp1)
```

```
sequential treed D-Optimal design in 3 partitions
dopt.gp (1) choosing 55 new inputs from 272 candidates
dopt.gp (2) choosing 53 new inputs from 263 candidates
dopt.gp (3) choosing 93 new inputs from 465 candidates
```

```
> XX <- rbind(XX, c(-sqrt(1/2), 0))
```

Figure 17 shows the sampled `XX` locations (circles) amongst the input locations `X` (dots) and MAP partition ($\hat{\mathcal{T}}$). Notice how the candidates `XX` are spaced out relative to themselves, and relative to the inputs `X`, unless they are near partition boundaries. The placing of configurations near region boundaries is

```
> tgp.trees(exp1)

NOTICE: skipped plotting tree of height 1, with lpost = 120.941
```

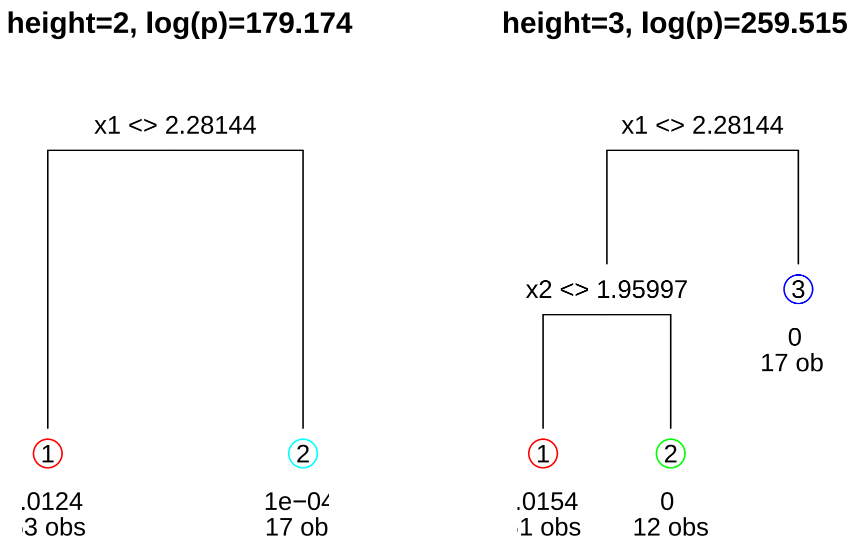

Figure 16: MAP trees of each height encountered in the Markov chain for the exponential data, showing $\hat{\sigma}^2$ and the number of observations $n$ at the leaves. $\hat{\mathcal{T}}$ is the one with the maximum $\log(p)$ above.

a symptom particular to $D$–optimal designs. This is desirable for experiments with `tgp` models, as model uncertainty is usually high there [3].

Now, the idea is to fit the treed GP LLM model, again, in order to assess uncertainty in the predictive surface at those new candidate design points. The following code gathers all three adaptive sampling statistics: ALM, ALC, & EI.

```
> exp.as <- btgpllm(X = X, Z = Z, XX = XX, corr = "exp",
+     improv = TRUE, Ds2x = TRUE, verb = 0)
```

Figure 18 shows the posterior predictive estimates of the adaptive sampling statistics. The error surface, on the *left*, summarizes posterior predictive uncertainty by a norm of quantiles.

In accordance with the ALM algorithm, candidate locations `XX` with largest predictive error would be sampled (added into the design) next. These are most likely to be in the interesting region, i.e., the first quadrant. However, these results depend heavily on the clumping of the original design in the uninteresting areas, and on the estimate of $\hat{\mathcal{T}}$. Adaptive sampling via the ALC, or EI (or both) algorithms proceeds similarly, following the surfaces shown in *center* and *right* panels of Figure 18.

```
> plot(exp1$X, pch = 19, cex = 0.5)
> points(XX)
> mapT(exp1, add = TRUE)
```

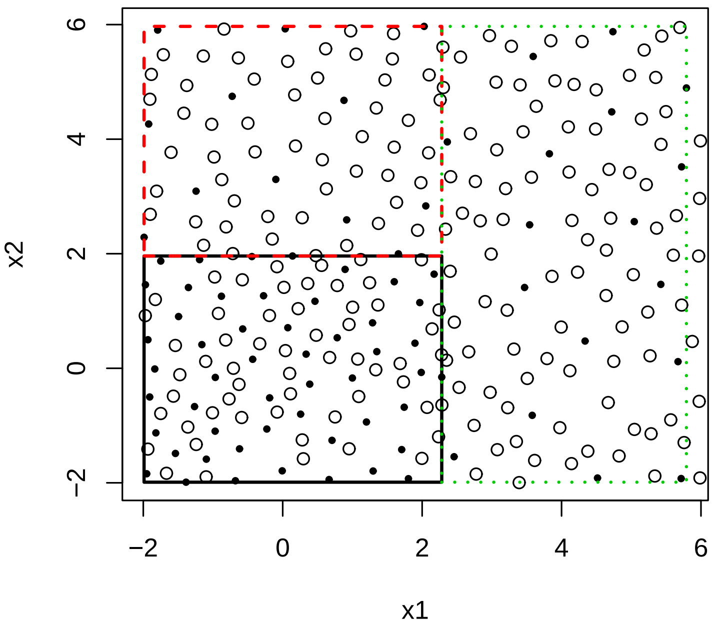

Figure 17: Treed $D$–optimal candidate locations XX (circles), input locations X (dots), and MAP tree $\hat{\mathcal{T}}$

```
> par(mfrow = c(1, 3), bty = "n")
> plot(exp.as, main = "tgpllm,", layout = "as", as = "alm")
> plot(exp.as, main = "tgpllm,", layout = "as", as = "alc")
> plot(exp.as, main = "tgpllm,", layout = "as", as = "improv")
```

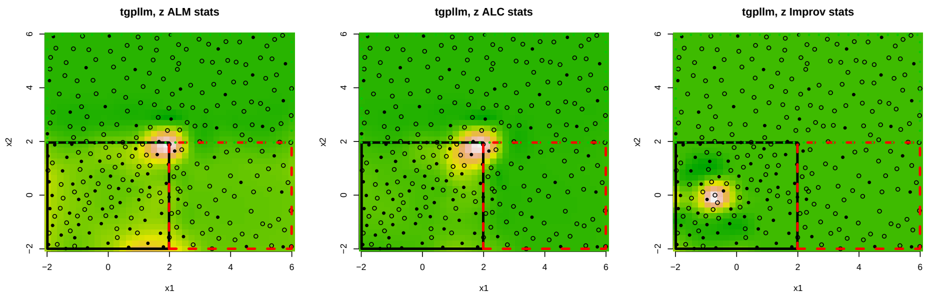

Figure 18: *Left*: Image plots of adaptive sampling statistics and MAP trees $\hat{\mathcal{T}}$; *Left*; ALM adaptive sampling image for (only) candidate locations XX (circles); *center*: ALC; and *right:* EI.