STAT7400 HW11, 2017

Yiheng Liu

Apr 23, 2017

Problem 1

(a) The basic function that produces estimated bias and standard error for `gam` estimator is given below.

```
> library(mgcv)
> m <- function(x) {
+     1 - sin(5 * x)^2 * exp(-4 * x)
+ }
> #R is simulation replications
> #n is sample size
> #s is error standard deviation
> gamsim <- function(R, n, s) {
+     x <- (1 : n) / (n + 1)
+     yhat <- sapply(seq_len(R), function(i) {
+         y <- rnorm(n, m(x), s)
+         gam(y ~ s(x))$fitted
+     })
+     eb <- rowMeans(yhat) - m(x)
+     ese <- apply(yhat, 1, sd)
+     cbind(eb, ese)
+ }
```

(b) A parallelized version that runs `gamsim` on the nodes in the cluster and merges the results.

```
> library(parallel)
> mergeBiases <- function(b, n, RR) rowMeans(matrix(b, n, length(RR)))
> mergeSDs <- function(s, n, RR) {
+     sqrt(rowSums(matrix(s^2, n, length(RR))*(RR[1]-1))/(sum(RR) - 1))
+ }
> pgamsim <- function(cl, R, n, s) {
+     nw <- length(cl)
+     RR <- rep(R / nw, nw)
```

```
+     val <- do.call(rbind, parLapply(cl, RR, gamsim, n, s))
+     eb <- mergeBiases(val[ ,"eb"], n, RR)
+     ese <- mergeSDs(val[ ,"ese"], n, RR)
+     cbind(eb, ese)
+ }
```

(c) Run `pgamsim` for $n = 50$, $\sigma = 0.2$, and $R = 10,000$ on a cluster of 2 workers. Compared to `gamsim`, the parallel approach significantly reduces the elapsed time. Besides, identical results are acquired with the same seed.

```
> cl <- makeCluster(2)
> clusterExport(cl, c("m"))
> clusterEvalQ(cl, library(mgcv))

[[1]]
[1] "mgcv"      "nlme"      "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

[[2]]
[1] "mgcv"      "nlme"      "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

> clusterSetRNGStream(cl, 123)
> system.time(val1 <- pgamsim(cl, 10000, 50, 0.2))

   user  system elapsed
  0.001   0.001  37.534

> system.time(val <- gamsim(10000, 50, 0.2))

   user  system elapsed
 67.831   0.184  68.479

> clusterSetRNGStream(cl, 123)
> val2 <- pgamsim(cl, 10000, 50, 0.2)
> identical(val1, val2)

[1] TRUE

> stopCluster(cl)
```