Parallel Monte Carlo

Colin Gillespie

Fri Nov 7 12:18:38 2014

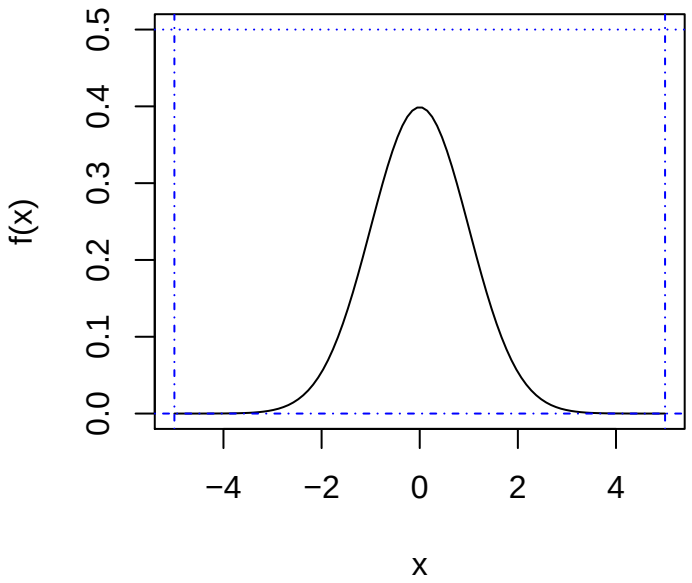
We want to estimate the integral

$$\int_{-5}^{5} \frac{\exp(-u^2/2)}{\sqrt{2\pi}} du$$

using a simple Monte Carlo rejection algorithm. The function is the Standard normal distribution, so the integral is approximately equal to 1. Plot the function and the sampling region

```
f = function(u) exp(-u^2/2)/sqrt(2*pi)

x = seq(-5, 5, length.out = 100)
plot(x, f(x), type="l", ylim=c(0, 0.5))
abline(h=0, lty=4, col=4);abline(h=0.5, lty=3, col=4);
abline(v=c(-5, 5), lty=4, col=4)
```



The standard C/Fortran approach is to use a for loop. This of course is very bad R code.

```
# Dummy argument - used in parSapply below
simulate_pt = function(i) {
  x = runif(1, -5, 5); y = runif(1, 0, 0.5)
  y < f(x)
}
N = 10^5; hits = 0
for(i in 1:N)
  hits = hits + simulate_pt()
(estimate = hits/N*(0.5*10))
```

```
## [1] 0.9993
```

As an aside, the (optimal?) R way is to use a vectorised functions

```
sum(runif(N, 0, 0.5) < f(runif(N, -5, 5)))/N*(0.5*10)
```

```
## [1] 1.003
```

Moving to parallel, we change the for loop to sapply

```
hits = sum(sapply(1:N, simulate_pt))
(estimate = hits/N*(0.5*10))
```

```
## [1] 1.008
```

Then it's a straightforward to run in parallel.

```
# Load the package
library("parallel")
cl = makeCluster(4)
clusterExport(cl, "f")
hits = sum(parSapply(cl, 1:N, simulate_pt))
(estimate = hits/N*(0.5*10))
```

```
## [1] 0.9913
```

```
stopCluster(cl)
```