Directional Metropolis Hastings

Abhirup Mallik

February 26, 2017

The R package "dirmcmc" implements the directional Metropolis Hastings Algorithm. This is a variant of usual Metropolis Hastings algorithm, that is so frequently used for many MCMC problems. The usual variant of Metropolis Hastings is Random Walk Metropolis algorithm, which uses a multivariate normal kernel with identity matrix as the covarinace matrix. Directional Metropolis Hastings generalizes this and uses a kernel with covariance matrix dependant on state. The kernel used in DMH is

$$\mathrm{N}_d(x + h\nabla \log f(x), tI_d + (s-1)G(x))$$

Where, $G(x)$ is constructed using $\nabla \log f(x)$ and is used to produce a multivariate normal density contoured towards the gradient of the target density at the current state. The transition mechanism is exactly same as usual Metropolis Hastings algorithm. Here we show an example of using this algorithm.

We choose a density with irregular shape for this example. The following distribution is proposed by Haario 1999. The target density is given by:

$$f_B(x_1, \cdots, x_d) \propto \exp[-x_1^2/200 - \frac{1}{2}(x_2 + Bx_1^2 - 100B)^2 - \frac{1}{2}(x_3^2 + \cdots + x_d^2)]$$

The parameter B controls how twisted the contout is. To use this library to sample from this density:

```
> library(dirmcmc)
> lupost.banana <- function(x,B){
+   -x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2
+ }
```

```
> ##Banana Gradient
> gr.banana <- function(x,B){
+   g1 <- -x[1]/100 - 2*B*(x[2]+B*x[1]^2-100*B)
+   g2 <- -(x[2]+B*x[1]^2-100*B)
+   g <- c(g1,g2)
+   return(g)
+ }
> B <- 0.03
> x1 <- seq(-25, 25, length=120)
> x2 <- seq(-25, 25, length=120)
> d.banana <- matrix(apply(expand.grid(x1, x2), 1,
+                          lupost.banana, B=B), nrow=120)
> image(x1, x2, exp(d.banana), col=cm.colors(60), asp=1, main="MH")
> contour(x1, x2, exp(d.banana), add=TRUE, col=gray(0.6))
> out.metdir.banana <- metropdir(obj = lupost.banana, dobj = gr.banana,
+                                initial = c(0,1),lchain = 2000,
+                                sd.prop=1.5,
+                                steplen=0.01,s=1.1,B=B)
> plot(out.metdir.banana$batch,xlab="x1",ylab = "x2",cex=0.5)
```

The output produces a list with acceptance rate and the chain.

```
> out.metdir.banana$accept

[1] 0.549

> apply(out.metdir.banana$batch,2,mean)

[1] -0.3518116  0.4721628
```

There is also an adaptive version of the DMH algorithm available, with very similar interface.

```
> out.metdir.adapt.banana <- metropdir.adapt(obj = lupost.banana,
+                                            dobj = gr.banana,
+                                initial = c(0,1),lchain = 2000,
+                                sd.prop=1.5,
+                                steplen=0.01,s=1.1,targetacc = 0.44,B=B)
> image(x1, x2, exp(d.banana), col=cm.colors(60), asp=1, main="MH")
> contour(x1, x2, exp(d.banana), add=TRUE, col=gray(0.6))
> plot(out.metdir.adapt.banana$batch,xlab="x1",ylab = "x2",cex=0.5)
```

There are functions for diagnostics of the resulting chain. The function mcmcdiag() takes the chain and produces multivariate and marginal Effective sample size, Integrated auto correlation times, mean square jump distance. Any one of these specific functions also can be called by using respective functions.

```
> mcmcdiag(out.metdir.adapt.banana$batch)

$MEss
[1] 53.06916

$ess
[1] 47.38742 48.45614

$iact
[1] 228.5288 226.7962

$msjd
[1] 15.97856
```