

统计计算 HW4

朱强强 17064001

2020 年 4 月 27 日

假设样本是由如下混合分布生成：

$$\delta N(7, 1^2) + (1 - \delta)N(10, 1^2) \quad (1)$$

其中，参数 $0 \leq \delta \leq 1$ 是第一个正态总体的占比（未知）。如果我们对于 δ 的先验分布是 $\delta \sim Unif(0, 1)$ ，可得到其后验密度

$$p(\delta|x) \propto \prod_{i=1}^n \left(\delta e^{-(x_i-7)^2/2} + (1-\delta)e^{-(x_i-10)^2/2} \right) \quad (2)$$

if $0 \leq \delta \leq 1$

上面公式 (2) 所对应的 δ 的分布可以使用 MCMC 方法进行抽样。本习题的目标是研究 Metropolis-Hastings 算法中推荐转移概率密度的选择对于目标密度 (2) 的抽样效率影响。

- 1) 从混合分布 (1) 中生成 200 个随机数，令 $\delta = 0.7$ 。

```
> set.seed(123)
> delta <- 0.7
> x <- delta * rnorm(200, 7, 1) + (1 - delta) * rnorm(200, 10, 1)
```

- 2) 将任务 1 生成 200 个随机数作为样本，写出似然函数。

```
> likelihood <- function(x, delta) {
+   llk <- prod(delta * dnorm(x, 7, 1) + (1 - delta) * dnorm(x, 10, 1))
+   return(llk)
+ }
> likelihood(x, delta)
```

```
[1] 1.160019e-144
```

- 3) 将任务 1 生成 200 个随机数作为样本，将 independent proposal，例如 $\delta^{[new]} \sim Unif(0, 1)$ ，作为 Metropolis-Hastings 算法转移分布的推荐密度，实现后验密度 (2) 的重复抽样。

```
> mh1 <- function(n, ini=0.5, seed=123) {
+   set.seed(seed)
+   X <- rep(NA, n+1)
```

```

+ X[1] <- ini
+ for (k in 1:n) {
+   Xold <- X[k]
+   Xnew <- runif(1)
+   U <- runif(1)
+   accp.r <- ifelse(likelihhod(x, Xold) != 0,
+                   likelihhod(x, Xnew)/likelihhod(x, Xold), 1)
+   X[k+1] <- ifelse(U < accp.r, Xnew, Xold)
+ }
+ return(X)
+ }

```

- 4) 将任务 1 生成 200 个随机数作为样本,将 random walk proposal,例如 $\delta^{[new]} \sim Unif(\delta^{[old]} - 1, \delta^{[old]} + 1)$, 作为 Metropolis-Hastings 算法转移分布的推荐密度, 实现后验密度 (2) 的重复抽样。

```

> mh2 <- function(n, d=1, ini=0.5, seed=123) {
+   set.seed(seed)
+   X <- rep(NA, n+1)
+   X[1] <- ini
+   for (k in 1:n) {
+     Xold <- X[k]
+     Xnew <- Xold + runif(1, -d, d)
+     U <- runif(1)
+     accp.r <- ifelse(likelihhod(x, Xold) != 0,
+                     likelihhod(x, Xnew)/likelihhod(x, Xold), 1)
+     X[k+1] <- ifelse(U < accp.r, Xnew, Xold)
+   }
+   return(X)
+ }

```

- 5) 比较任务 3 和 4 的 MCMC 算法对 δ 的估计 (提示: 首先分别删去前 500 个抽养值)。

```

> X1 <- mh1(1e4, ini=0.5, seed=123)
> X2 <- mh2(1e4, d=1, ini=0.5, seed=123)
> delta1 <- mean(X1[-(1:500)])
> delta2 <- mean(X2[-(1:500)])
> delta1
> delta2

```

```
[1] 0.9375547
```

```
[1] 17.98105
```

6) 比较任务 3 和 4 的 MCMC 算法的收敛性和有效性。

```
> X1 <- mh1(1e4, ini=1.2, seed=111)
> X2 <- mh1(1e4, ini=0.5, seed=222)
> X3 <- mh1(1e4, ini=-0.5, seed=333)
> Y1 <- mh2(1e4, d=1, ini=1.2, seed=111)
> Y2 <- mh2(1e4, d=1, ini=0.5, seed=222)
> Y3 <- mh2(1e4, d=1, ini=-0.5, seed=333)
> par(mfrow=c(1, 2))
> index <- 1:1000
> plot(X1[index], type="l", ylab="x", main="Trace plot",
+       ylim=c(-1, 1))
> lines(X2[index], col=2)
> lines(X3[index], col=3)
> plot(Y1[index], type="l", ylab="x", main="Trace plot",
+       ylim=c(-10, 10))
> lines(Y2[index], col=2)
> lines(Y3[index], col=3)
```

