

Rejection Sampling

Jim Albert

June 29, 2007

To illustrate rejection sampling, suppose one observes y_1, \dots, y_n from a Cauchy density with median μ and scale parameter 1. If one assigns a uniform prior to μ , the posterior density for μ is given by

$$g(\mu) \propto \prod_{i=1}^n \frac{1}{1 + (y_i - \mu)^2}.$$

Suppose we observe the data values 1, 2, 3, 2, 5.2, 6, 7, 8 that we place in the vector `data`.

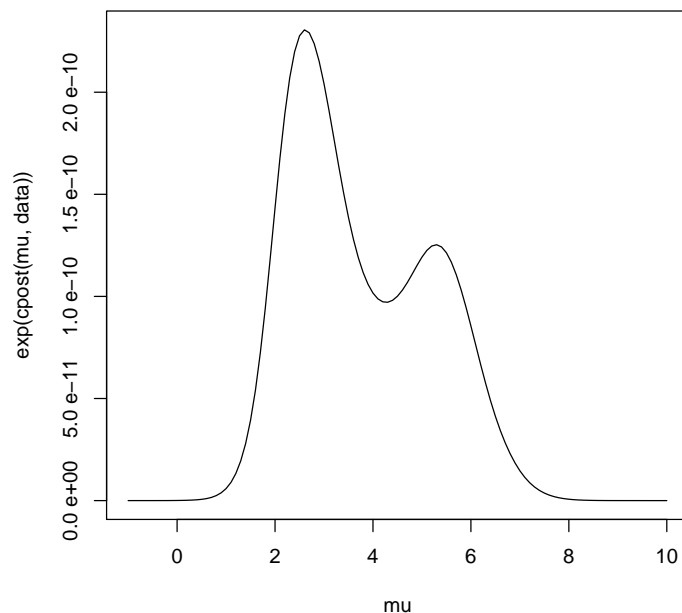
```
> data = c(1, 2, 3, 2, 5.2, 6, 7, 8)
```

We define the log posterior of μ using the function `cpost`.

```
> cpost = function(mu, data) {  
+   val = 0 * mu  
+   sigma = 1  
+   for (i in 1:length(data)) {  
+     val = val + log(dt((data[i] - mu)/sigma, df = 1)/sigma)  
+   }  
+   return(val)  
+ }
```

To graph the posterior density, we set up a grid of values in the vector `mu` and use the `plot` command.

```
> mu = seq(-1, 10, by = 0.1)  
  
> plot(mu, exp(cpost(mu, data)), type = "l")
```



To get a quick approximation to the posterior, we use the `laplace` command to fit a normal curve. We use the starting values $\mu = 2$ to find the left mode, and apply 10 iterations of Newton's method.

```
> fit = laplace(cpost, 2, 10, data)
```

Based on this output, we decide on using a t covering density with mean 4, “variance” parameter 2.25 and 4 degrees of freedom. We place this parameter values in the list `tpar` and define a list `datapar` that contains both the parameter values and the data.

```
> tpar = list(m = 4, var = 2.25, df = 4)
> datapar = list(data = data, par = tpar)
```

To use rejection sampling, we need to find a constant c such that $g(\mu) \leq cg_T(\mu)$, for all μ , where g_T is the covering density. Equivalently, we find a constant d such that $\log g(\mu) \leq \log g_T(\mu) + d$ for all μ . We define a function `cpost` that computes the function $\log g(\mu) - \log g_T(\mu)$.

```
> cpostT = function(mu, datapar) {
+   val = 0 * mu
+   data = datapar$data
+   tpar = datapar$par
+   mn = tpar$m
```

```

+   sigma = sqrt(tpar$var)
+   df = tpar$df
+   val = cpost(mu, data) - log(dt((mu - mn)/sigma, df = df)/sigma)
+   return(val)
+ }

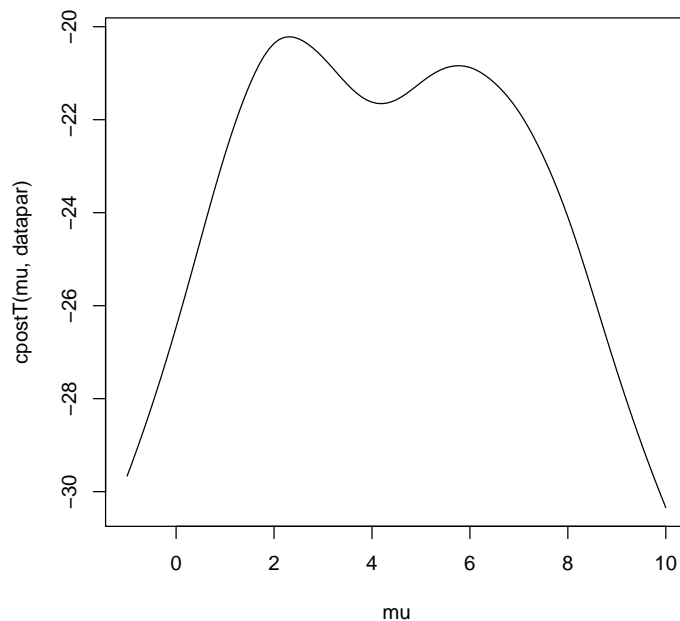
```

We graph this difference in logs function and note that it appears to be maximized at a value of μ close to 2.

```

> plot(mu, cpostT(mu, datapar), type = "l")

```



We use the function `laplace` to find the value of μ where this difference in logs function is maximized. The constant d is defined to be the difference in logs function evaluated at this optimal value.

```

> fit2 = laplace(cpostT, 2, 10, datapar)
> d = cpostT(fit2$mode, datapar)

```

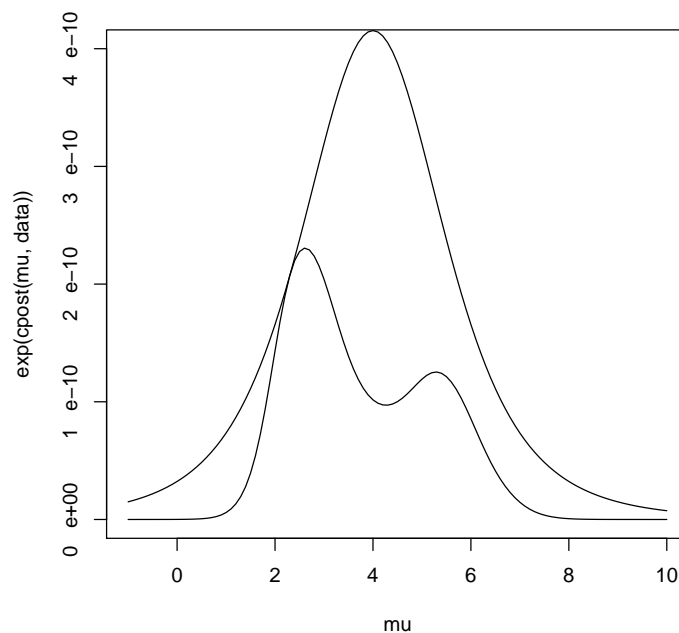
We show this covering graphically. Below we graph the posterior density $g(\mu)$ and the covering density $cg_T(\mu)$.

```

> plot(mu, exp(cpost(mu, data)), type = "l", ylim = c(0, 4e-10))
> m = tpar$m
> sigma = sqrt(tpar$var)

```

```
> df = tpar$df
> lines(mu, 1/sigma * dt((mu - m)/sigma, df) * exp(d))
```



We are now ready to use the `rejectsampling` function. The inputs are the density of interest, defined in `cpost`, the parameters of the t covering function `tpar`, the constant `d`, the number of draws from the covering density, and the `data` vector. The output is a vector of simulated draws that are accepted in the algorithm. Here the length of this vector is approximately 20000 which indicates that the acceptance rate of this algorithm is approximately 40%.

```
> M = rejectsampling(cpost, tpar, d, 50000, data)
```

To illustrate that we have indeed obtained a simulated sample from the posterior density, we plot the actual posterior density in red and a density estimate of the simulated draws from `rejectsampling` in blue. The two curves match up very well.

```
> f = exp(cpost(mu, data))
> f = f/sum(f)/0.1

> plot(mu, f, type = "l", col = "red")
> lines(density(M), col = "blue")
> legend(6, 0.3, legend = c("Exact", "Simulated"), col = c("red",
+      "blue"), lty = c(1, 1))
```

