STA 250: Statistics
Lab 6

This lab looks at sampling from a pdf that is known only up to a normalizing constant.

Background

We will consider sampling from a pdf $h(\theta) = e^{q(\theta)}/\text{const}$, where $q(\theta)$ is known in closed form, but the normalizing constant is unknown (but it is assumed finite, so the $h(\theta)$ is a proper pdf). We will continue working with the example:

$$q(\theta) = \begin{cases} 20 \log \theta + 20 \log(\theta + 1) - 5.59\theta, & \theta > 0 \\ -\infty & \theta \leq 0 \end{cases}$$

that we worked with in class on Tuesday. The following code translates definition of $q(\theta)$ to the language of `R` :

```
q <- function(theta){
   theta <- pmax(theta, 0)
   return(20 * log(theta) + 20 * log(theta + 1) - 5.59 * theta)
}
```

Graphing $h(\theta)$

Because this is a univariate pdf, we can compute the normalizing constant by numerical quadrature methods. `R` function for this is `integrate()`. Here is how to use it:

```
unnorm.h <- function(theta) return(exp(q(theta)))
norm.const <- integrate(unnorm.h, 0, Inf)$value
print(norm.const)
```

which produces `2.743055e+18`. In the call to `integrate`, the second and third arguments give the range of integration. We can now define the normalized pdf and make a plot of it over $\theta \in (0, 20)$.

```
h <- function(theta) return(exp(q(theta)) / norm.const)
theta.grid <- seq(0, 20, .1)
plot(theta.grid, h(theta.grid), ty = "l", ann = FALSE)
title(xlab = expression(theta), ylab = expression(h(theta)))
```

Task 1. Run all the code pieces above and make a nice plot $h(\theta)$ over $\theta \in (0, 20)$.

Normal approximation with Laplace's technique

Laplace's technique leads to approximating $h(\theta)$ with $\mathsf{Normal}(m, s^2)$ where $m = \hat{\theta}$ = the maxima of $q(\theta)$ and $s = 1/\sqrt{\{-\ddot{q}(\hat{\theta})\}}$. The maxima $\hat{\theta}$ could be calculated by using Newton's method of iteration. Start with an initial guess θ_0 and then iterate $\theta_{t+1} \leftarrow \theta_t - \dot{q}(\theta_t)/\ddot{q}(\theta_t)$, $t = 1, 2, \cdots$ until you see the sequence stabilizes.

TASK 2. Start with an initial guess of $\theta = 10$ and run Newton's recursion until you see stability in the iterate values up to 3 decimal places. How many iterations did you need? Use the following code snippets to get $\dot{q}(\theta)$ and $\ddot{q}(\theta)$.

```
q.dot <- function(theta) return(20 / theta + 20 / (theta + 1) - 5.59)
q.ddot <- function(theta) return(-20 * (1 / theta^2 + 1 / (theta + 1)^2))
```

TASK 3. Verify that Laplace's approximation for $h(\theta)$ is $\mathsf{Normal}(6.69, 1.128^2)$. Overlay a plot of this pdf on the original plot of $h(\theta)$ you made above. How good is the match? Draw $M = 100,000$ samples of θ from the approximating normal pdf and give its 5-quantile summary (i.e., at 2.5%, 25%, 50% 75% and 97.5%).

Rejection sampling

In Notes 11 (next lecture) we will see that if we can find a pdf $g(\theta) = e^{u(\theta)}/\text{const}$, so that

1. We know how to sample from $g(\theta)$
2. For every θ, $q(\theta) \le u(\theta)$

then sampling from $h(\theta)$ can be done as follows:

1. Sample $\theta \sim g(\theta)$ and set $p = \exp\{q(\theta) - u(\theta)\}$
2. Samplie $U \sim \mathsf{Uniform}(0, 1)$. If $U < p$ then stop and return θ. Go back to step 1 otherwise.

The validity of the procedure depends on having $q(\theta) \le u(\theta)$ for all θ. But in order for it to be efficient (i.e., few "rejection" at step 2) $u(\theta)$ should be fairly close to $q(\theta)$.

By concavity of the log function, it turns out that our $q(\theta)$ is always less than or equal to $u(\theta) = 40\log(\theta + 1/2) - 5.59 \cdot (\theta + 1/2) + 5.59/2$. We can create the "u" function in `R` as:

```
u <- function(theta) return(40 * log(theta + 1/2) - 5.59 * (theta + 1/2) + 5.59/2)
```

TASK 4. Make a plot of $q(\theta)$ and $u(\theta)$ together to verify that $q(\theta) \le u(\theta)$ over $\theta \in (0, 20)$. How close are the two functions over this range?

With $u(\theta)$ as above, the corresponding pdf

$$g(\theta) = \frac{e^{u(\theta)}}{\text{const}} = \frac{(\theta + 1/2)^{40} e^{-5.59(\theta+1/2)}}{\text{const}}$$

is the $\mathsf{Gamma}(41, 5.59)$ pdf shifted by 0.5. That is to sample a θ from $g(\theta)$ we first sample a $\theta^* \sim \mathsf{Gamma}(41, 5.59)$ and then set $\theta = \theta^* - 0.5$. Below I use this scheme to construct the rejection sampler.

```
rej.samp <- function(){
  flag <- TRUE
  while(flag){
    theta <- rgamma(1, 41, 5.59) - .5
    p <- exp(q(theta) - u(theta))
    flag <- (runif(1) > p)
  }
  return(theta)
}
```

Task 5. Sample $M = 100,000$ samples of θ from $h(\theta)$ by the rejection sampling method by:

```
theta.rej <- replicate(1e5, rej.samp())
```

Make a histogram of these sampled values through the code:

```
hist(theta.rej, freq = FALSE, breaks = -.125 + seq(0,20,.25))
```

and overlay a graph of $h(\theta)$ on it. How good is the match? Next overlay the normal approximation due to Laplace's technique. Also get 5-quantile summary of these sampled draws.