Computing a posterior distribution

Introduction

Given a prior density $\pi(\theta)$, the posterior density of θ is

$$\pi(\theta|x_1, \cdots, x_n) = k(x_1, \cdots, x_n)\pi(\theta)\mathcal{L}(\theta)$$

where $\mathcal{L}(\theta)$ is the likelihood function

$$k(x_1, \cdots, x_n) = \left\{ \int_\Theta \pi(\theta)\mathcal{L}(\theta)\, d\theta \right\}^{-1}.$$

Typically, we must evaluate $k(x_1, \cdots, x_n)$ numerically, which in theory is not difficult but in practice may be problematic. In particular, the integral defining $k(x_1, \cdots, x_n)$ may be difficult to evaluate as $\mathcal{L}(\theta)$ will often be very small for certain values of θ and in extreme cases may not even be computable (due to potential underflow problems).

Pre-normalization

To facilitate the computation of the posterior density, we can "pre-normalize" by dividing $\pi(\theta)\mathcal{L}(\theta)$ by some constant $k'(x_1, \cdots, x_n)$ so that

$$\mathcal{U}(\theta) = \frac{\pi(\theta)\mathcal{L}(\theta)}{k'(x_1, \cdots, x_n)}$$

with, for example,

$$\max_\theta \mathcal{U}(\theta) = 1.$$

(Setting $\max_\theta \mathcal{U}(\theta) = 1$ is very arbitrary – the key point is to divide $\pi(\theta)\mathcal{L}(\theta)$ by some constant to avoid problems with the numerical integration.)

The pre-normalized $\mathcal{U}(\theta)$ is quite easy to compute. Even if $\mathcal{L}(\theta)$ is very small (and therefore difficult to compute), its logarithm $\ln \mathcal{L}(\theta)$ is much more manageable (and computable); this $\mathcal{U}(\theta)$ can be computed as

$$\mathcal{U}(\theta) = \exp\left[\ln \pi(\theta) + \ln \mathcal{L}(\theta) - \max_\theta \{\ln \pi(\theta) + \ln \mathcal{L}(\theta)\}\right]$$

and so

$$\pi(\theta|x_1, \cdots, x_n) = \frac{\mathcal{U}(\theta)}{\displaystyle\int_\Theta \mathcal{U}(s)\, ds}.$$

In the next section, we will discuss how to evaluate the denominator above.

Numerical integration

To evaluate the integral $\int_\Theta \mathcal{U}(s)\,ds$ numerically, we typically use an approximation of the form

$$\sum_{k=1}^{N} w_k \mathcal{U}(\theta_k)$$

where $\{w_k\}$ are some weights and $\{\theta_k\}$ are either some fixed points in Θ (numerical quadrature) or chosen at random from some probability distribution (Monte Carlo integration). In the latter case, if $\{\theta_k\}$ are generated from some probability density function g on Θ then we have

$$\frac{1}{N}\sum_{k=1}^{N} \frac{\mathcal{U}(\theta_k)}{g(\theta_k)} \approx \int_\Theta \mathcal{U}(s)\,ds$$

for sufficiently large N (by the Weak Law of Large Numbers); this procedure is known as **importance sampling**. Importance sampling can be particularly useful when Θ is higher dimensional.

If Θ is a bounded subset of the real line then choosing $\{\theta_k\}$ to be an equally spaced set of points leads to simple approximations for $\int_\Theta \mathcal{U}(s)\,ds$. In fact, the assumption that Θ is bounded is often unnecessary: In practice, $\mathcal{U}(\theta) \to 0$ as $\theta \to \pm\infty$ and so we can find a and b such that $\mathcal{U}(\theta) \approx 0$ for $\theta < a$ and $\theta > b$ (a and b can be determined graphically or analytically); thus we can approximate $\int_\Theta \mathcal{U}(s)\,ds$ by $\int_a^b \mathcal{U}(s)\,ds$.

Numerical quadrature works by approximating an integrand (in this case $\mathcal{U}(s)$) over short intervals $[\theta, \theta + h]$ by low degree polynomials, whose integrals are easy to compute. Perhaps the simplest quadrature method is the **trapezoidal rule**, which approximates the integrand by a piecewise linear function. To be more precise, suppose that $\Theta = [a, b]$ and define $\theta_0 = a$ and $\theta_N = b$ with $\theta_k = \theta_0 + kh$ where $h = (b-a)/N$; if N is large enough (so that h is small enough) then on the interval $[\theta_{k-1}, \theta_k]$,

$$\mathcal{U}(s) \approx \mathcal{U}(\theta_{k-1}) + \frac{\mathcal{U}(\theta_k) - \mathcal{U}(\theta_{k-1})}{\theta_k - \theta_{k-1}}(s - \theta_{k-1})$$

and so

$$\begin{aligned}\int_{\theta_{k-1}}^{\theta_k} \mathcal{U}(s)\,ds &\approx \int_{\theta_{k-1}}^{\theta_k} \left\{\mathcal{U}(\theta_{k-1}) + \frac{\mathcal{U}(\theta_k) - \mathcal{U}(\theta_{k-1})}{\theta_k - \theta_{k-1}}(s - \theta_{k-1})\right\} ds \\ &= \frac{h}{2}\left\{\mathcal{U}(\theta_k) + \mathcal{U}(\theta_{k-1})\right\}.\end{aligned}$$

Then we can approximate $\int_a^b \mathcal{U}(s)\,ds$ as follows:

$$\begin{aligned}\int_a^b \mathcal{U}(s)\,ds &= \sum_{k=1}^{N} \int_{\theta_{k-1}}^{\theta_k} \mathcal{U}(s)\,ds \\ &\approx \frac{h}{2}\sum_{k=1}^{N}\left\{\mathcal{U}(\theta_k) + \mathcal{U}(\theta_{k-1})\right\} \\ &= \frac{h}{2}\mathcal{U}(\theta_0) + h\sum_{k=1}^{N-1}\mathcal{U}(\theta_k) + \frac{h}{2}\mathcal{U}(\theta_N).\end{aligned}$$

If the second derivative of $\mathcal{U}$ is bounded over $[a, b]$ then the absolute value of approximation error is less than constant $\times\ (b-a) \times h^2$.

More sophisticated quadrature methods (based on higher order polynomial approximations) can also be used. For example, Simpson's rule uses a quadratic approximation over an interval $[\theta_{k-2}, \theta_k]$ where the approximation equals $\mathcal{U}$ at $\theta = \theta_{k-2}$, $\theta = \theta_{k-1}$ and $\theta = \theta_k$. Then, defining $\{\theta_k\}$ as above and taking N even, we get

$$\int_a^b \mathcal{U}(s)\, ds \approx \frac{h}{3} \sum_{k=1}^{N/2} \left\{ \mathcal{U}(\theta_{2k-2}) + 4\mathcal{U}(\theta_{2k-1}) + \mathcal{U}(\theta_{2k}) \right\}.$$

Quadrature methods can also be combined; for example, we may be able to get a better approximation of $\int_a^b \mathcal{U}(s)\, ds$ by taking an average of the approximations given by the trapezoidal rule and Simpson's rule.

Example: The Zipf distribution

Suppose that a language consists of N words, which occur with probabilities $p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_N$ where $p_1 + \cdots + p_N = 1$. A common model for $\{p_k\}$ is the Zipf distribution[1]

$$p_k = \frac{k^{-\theta}}{H(\theta, N)} \quad \text{for } k = 1, \cdots, N$$

where $\theta > 0$ and

$$H(\theta, N) = \sum_{j=1}^{N} j^{-\theta}.$$

To estimate θ, we can look at the frequency of the m most common words where m is typically much smaller than N. It is easy to show that the probability of the k-th most popular word in the m most common words also follows a Zipf distribution with the parameters θ and m. The data given below are the frequencies of the 20 most common passwords from a list of roughly 10000 phished Hotmail passwords:

64 18 11 10 9 9 9 9 8 7 7 7 7 7 6 6 6 6 5 5

The likelihood function for these data is

$$\mathcal{L}(\theta) = \prod_{k=1}^{20} \left\{ \frac{k^{-\theta}}{H(\theta, 20)} \right\}^{x_k}$$

where $x_1 = 64$, $x_2 = 18$, $x_3 = 11$ and so on; we will assume the prior density $\pi(\theta) = \exp(-\theta)$ for $\theta > 0$.

The two R functions given below compute the log-likelihood function $\ln \mathcal{L}(\theta)$ and pre-normalized $\mathcal{U}(\theta)$ for values of θ contained in the vector `theta`.

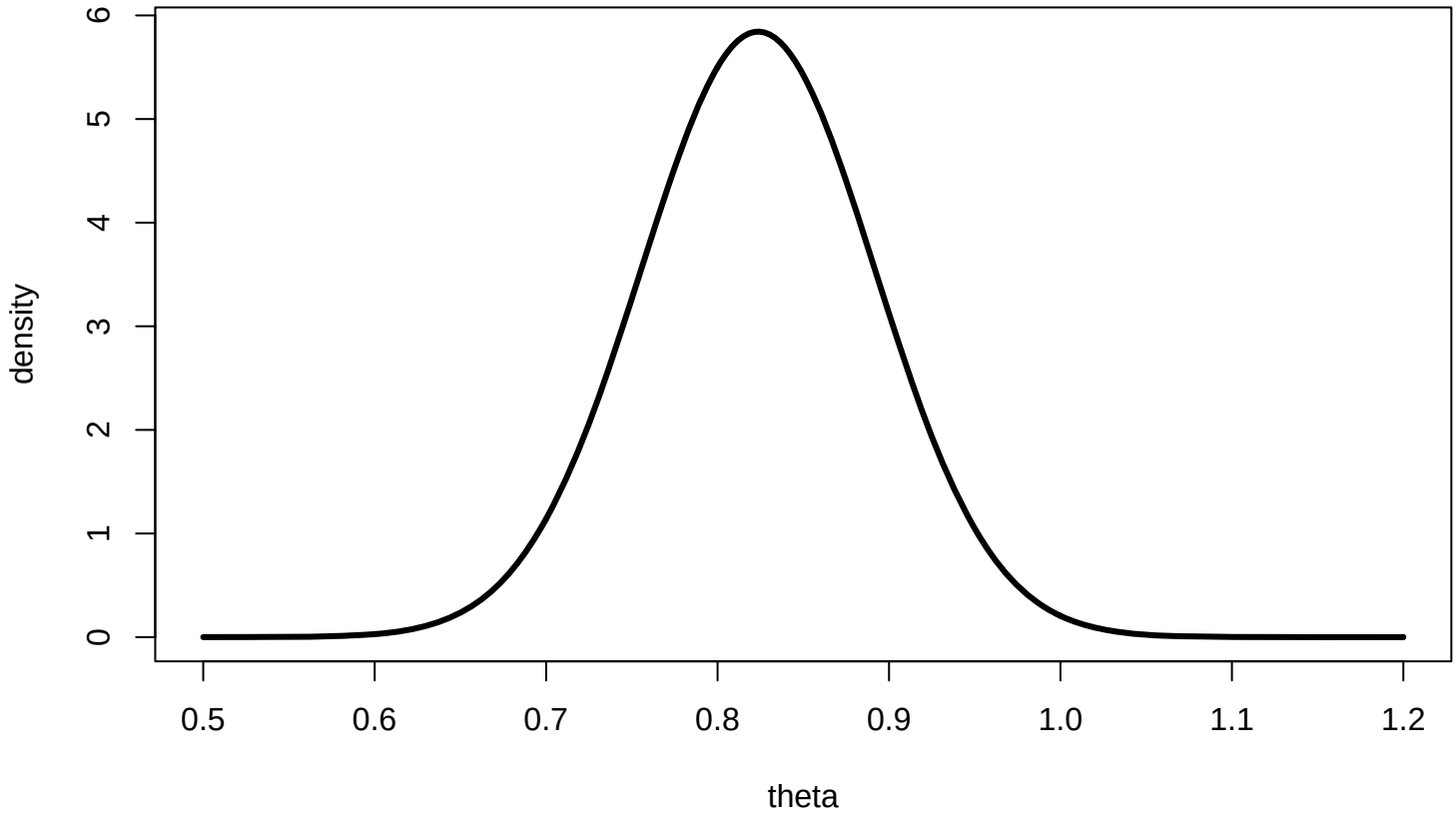
[1]The Zipf distribution arises from Zipf's law, which was proposed by the linguist George Kingsley Zipf, who theorised that given a large body of language (for example, a long book) the frequency of each word is close to inversely proportional to its rank.

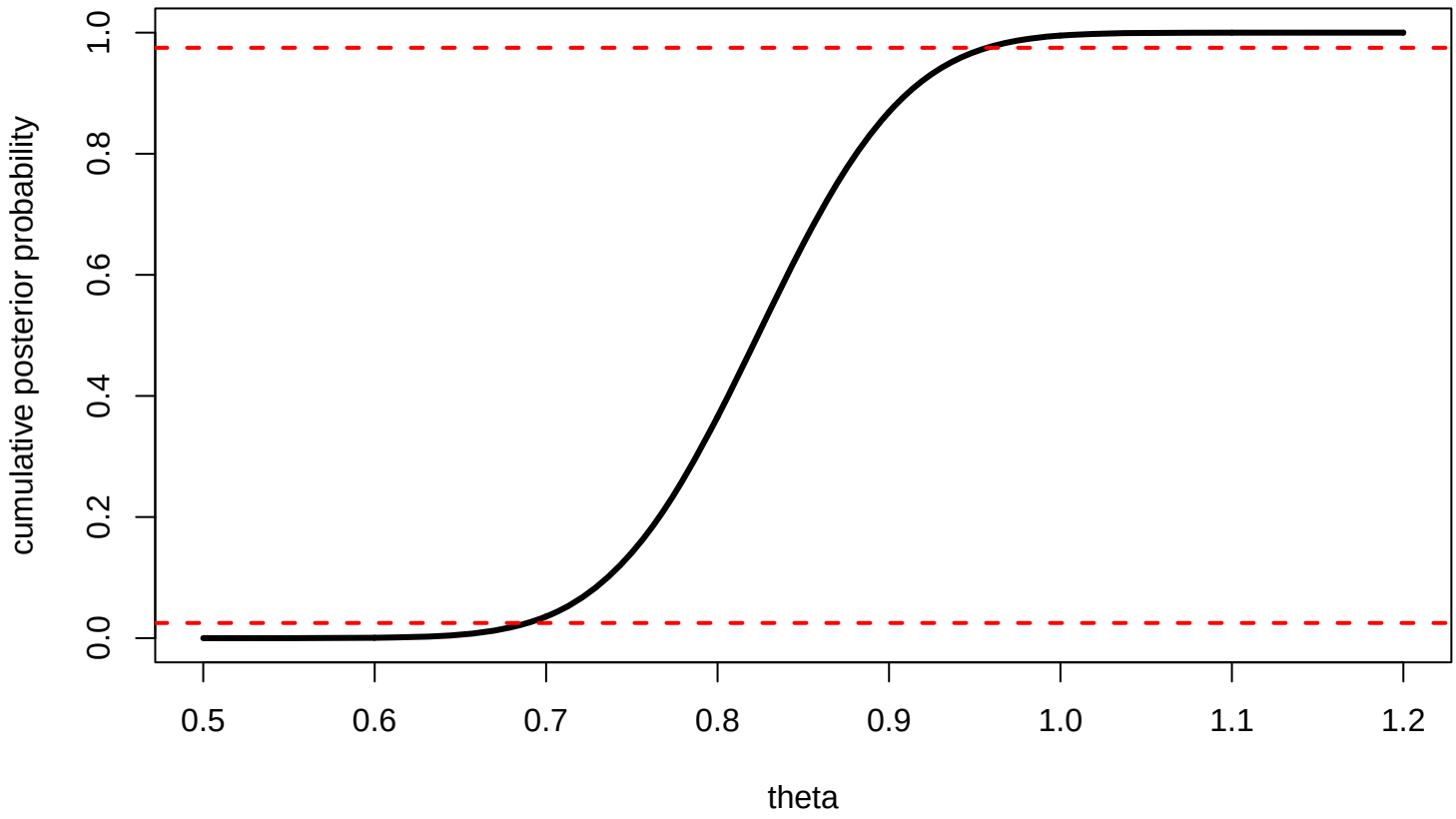
```
loglikelihood <- function(x,theta) {
            m <- length(x)
            k <- c(1:m)
            H <- NULL
            L <- -theta*sum(x*log(k))
            for (a in theta) H <- c(H,log(sum(k^(-a))))
            loglike <- L - sum(x)*H
            loglike
            }

prenorm <- function(x,theta) {
            r <- loglikelihood(x,theta)
            r <- r - theta # add log-prior
            r <- r - max(r) # subtract maximum
            pre <- exp(r) # pre-normalized
            pre
            }
```

Using trial and error, we can determine that the posterior is "significantly non-zero" for $a = 0.5 \leq \theta \leq 1.2 = b$. The R code below implements the trapezoidal rule using $h = 1/10000$ and $N = 7000$.

```
> x <- c(64,18,11,10,9,9,9,9,8,7,7,7,7,7,6,6,6,6,5,5)
> theta <- c(5000:12000)/10000  # 7001 values of theta
> post <- prenorm(x,theta) # compute pre-normalized posterior
> mult <- c(1/2,rep(1,6999),1/2) # multipliers for trapezoidal rule
> norm <- sum(mult*post)/10000 # integral evaluated using trapezoidal rule
> post <- post/norm # normalized posterior
> plot(theta,post,type="l",ylab="posterior density"
> # now compute 95% credible interval
> post.cdf <- cumsum(mult*post)/10000 # compute the posterior cdf
> plot(theta,post.cdf,type="l",ylab="cumulative posterior probability")
> abline(h=c(0.025,0.975),lty=2)
> lower <- max(theta[post.cdf<0.025]) # lower limit for credible interval
> upper <- min(theta[post.cdf>0.975]) # upper limit for credible interval
> c(lower,upper)
[1] 0.6891 0.9571
```

Thus a 95% credible interval is $[0.689, 0.957]$. Plots of the posterior density and posterior distribution function of θ are given on the following page.


density
0
1
2
3
4
5
6
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
theta


cumulative posterior probability
0.0
0.2
0.4
0.6
0.8
1.0
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
theta