Laplace approximation for estimation moments of posterior distribution

rstats-gsoc/gsoc2015: Test for ”Statistical algorithms in NIMBLE” project

Egorov Evgenii

Abstract

We consider the problem of calculating improper integrals with the Laplace approximation method. Next, we will formulate the problem of calculating the moments of the posterior distribution of parameters in terms of allowing apply the method discussed above.

As result, we will obtain R functions that allow:

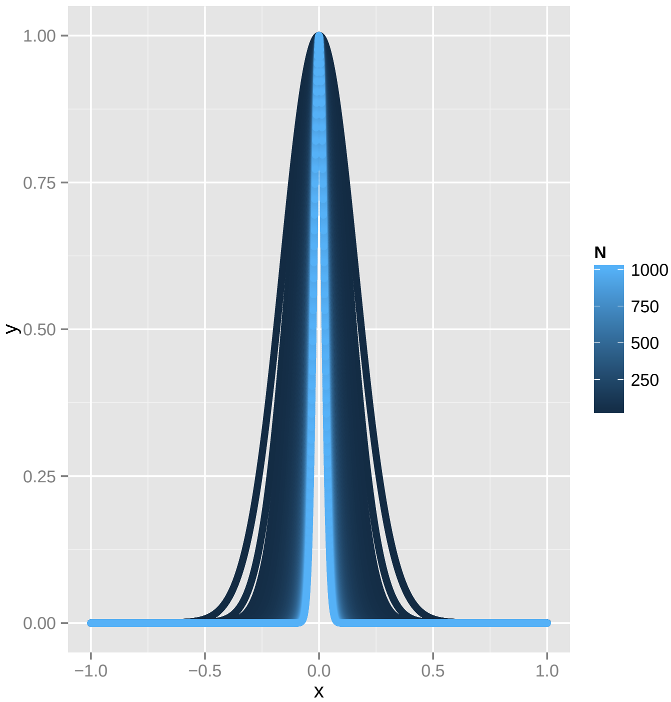
- estimate moment of the posterior distribution, take as input likehood function and prior density
- estimate moment of the posterior distribution, take as input $-1/N$ log of posterior density
- estimate improper integral in ”normal form”

Laplace approximation

Consider a function:

$$I(N) = \int_{-\infty}^{+\infty} e^{-Nl(x)}$$

We are interested in its asymptotic behavior for large values of N. For example, look at the behavior of the family of functions e^{-Nx^2} (just assume for example $l(x) = x^2$) for change N by 10 to 1000:



From the graphs conclusion suggests itself that with increasing N, improper integral values will be better approximated by the behavior of the function $l(x)$ near its extremum. From here we can get a method for computing I for large N (as Laplace did):

1. approximate the behavior of the function $l(x)$ around its modal value $\hat{x}$ using the Taylor series expansion
2. using the terms in the expansion of the second order Gaussian identify and take advantage of its known value

Thus it becomes clear requirements for the function $l(x)$

1. $l(x)$ is unimodal
2. $l(x) \in C^2$
3. $l''(x) > 0$

Implement the proposed plan for the one-dimensional case. Let's start with the approximation of the function $l(x)$ at the extremum $\hat{x}$. We denote the i-th derivative of the function $l(x)$, as $l_i(x)$. Given that $l_1(\hat{x}) = 0$, we have:

$$l(x) = l_0(\hat{x}) + \frac{1}{2!}l_2(\hat{x})(x-\hat{x})^2 + \frac{1}{3!}l_3(\hat{x})(x-\hat{x})^3 + \dots$$

Substituting in the integral (for readability omit the limits of integration):

$$\begin{aligned} I(N) &= \textstyle\int e^{-Nl(x)}dx = \\ &= \int \exp\left\{-N\left(l_0(\hat{x}) + \frac{1}{2!}l_2(\hat{x})(x-\hat{x})^2 + \frac{1}{3!}l_3(\hat{x})(x-\hat{x})^2 + \dots\right)\right\} dx \approx \\ &\approx e^{-Nl(\hat{x})} \int e^{\frac{-N}{2}l_2(\hat{x})u^2} exp\left\{-\frac{N}{6}l_3(\hat{x})u^3\right\} du \qquad (1) \end{aligned}$$

Hence:

$$I(N) \approx e^{-Nl(\hat{x})} \cdot \sqrt{2\pi} \cdot (l_2(\hat{x})N)^{-\frac{1}{2}}$$

If we want to obtain result in the multivariate case, we should go through the same way. Denote x as a d dimentional vector, we will find:

$$I(N) \approx e^{-Nl(\hat{x})} \cdot (2\pi)^{d/2} \cdot (det(H^{-1}(\hat{x})))^{1/2} \cdot N^{-d/2}$$

where H is hessian of $l(x)$ function. Thus, in this section we obtain an algorithm for estimation integrals of form:

$$I(N) = \int e^{-Nl(x)}dx$$

Let's call this form as **normal form**. Our next goal is to follow Tierney,Kadane (Journal of the American Statistical Association (1986) $81 : 82 - 86$) and reformulate the problem of finding the posterior moments in the form of calculation of two integrals in the normal form.

R code: est.normf

R code to estimate integral form in **normal form**. This is core of our algorithm, further progress is associated only with a view to convey the desired function l to the est.normf function.
As input:

- n – constant before l function
- l – l(x) function
- theta0 – initial point to search modal value
- l.b, u.b – bounds for modal value

```
> est.normf <- function(n,l,theta0,l.b,u.b){
+     theta.hat <- as.numeric(optimx(fn = l,
+                                  #function l is function h earlier in the text
+                       par = theta0,
+                       #initial point to find modal
+                       lower = l.b, upper = u.b,
+                       #bounds to find modal
+                       method = 'L-BFGS-B')[1,1:length(theta0)])
+     #result: vector of such x that l(x) modal
+
+     diff2 <- numericHessian(f = l, t0 = theta.hat) #hessian in modal point
+     result <- exp(-n*l(theta.hat)) * (2*pi)^(length(theta.hat)/2) *
+       (1/n)^(length(theta.hat)/2) * sqrt(det(ginv(diff2)))
+     #integral of normal form estimation
+     list(result = result, #integral aprox value
+         modal = theta.hat) #vector of such x that l(x) modal
+   }
```

As output we obtain integral value and modal point. Let's try it and evaluate known improper integral $\int\limits_{-\infty}^{+\infty}\int\limits_{-\infty}^{+\infty} e^{-(x^2+y^2)}dxdy = \pi$

```
> est.normf <- function(n,l,theta0,l.b,u.b){
+   theta.hat <- as.numeric(optimx(fn = l,
+                                 par = theta0,
+                                 lower = l.b, upper = u.b,
+                                 method = 'L-BFGS-B')[1,1:length(theta0)])
+
+   diff2 <- numericHessian(f = l, t0 = theta.hat)
+   result <- exp(-n*l(theta.hat)) * (2*pi)^(length(theta.hat)/2) *
+  (1/n)^(length(theta.hat)/2) * sqrt(det(ginv(diff2)))
+   list(result = result,
+        modal = theta.hat)
+ }
> l <- function(theta){
+   sum(theta^2)
+ }
> est.normf(1,l,c(1,1),l.b=c(-1,-1),u.b=c(1,1))
```

```
$result
[1] 3.141593

$modal
[1] -8.735256e-21 -8.735256e-21
```

Close enough!

Approximation of conditional probabilities

Assume:

- $\pi(\theta)$ prior θ density
- $L(\theta)$ likehood (not log-likehood!) function given the data $Y^{(N)}$
- $h(\theta)$ strictly positive function

Then θ posterior density:

$$p(\theta|Y^{(N)}) = \frac{\pi(\theta)L(\theta)d\theta}{\int \pi(\theta)L(\theta)d\theta}$$

Thus:

$$E_n[h] = E[h(\theta)|Y^{(N)}] = \frac{\int h(\theta)\pi(\theta)L(\theta)d\theta}{\int \pi(\theta)L(\theta)d\theta}$$

Now we want to estimate $E[h(\theta)|Y^{(N)}]$. We can face with two possibilities:

1. we alredy have θ posterior density as function $post.den(\theta)$ and $h(\theta)$
2. we have $\pi(\theta)$, $L(\theta)$ and $h(\theta)$ functions

Thus, I found it necessary to make function, that can directly estimate integral in normal form or take as input $\pi(\theta)$, $L(\theta)$ and $h(\theta)$ functions and generate disered l function itself.

```
> laplace_estimate <- function(type = 0,
+                               # 0 for estimate diretly "normal from" with function l,
+                               # 1 for assume known g, density of theta and likehod
+                               n,
+                               theta0, l.b, u.b,
+                               l,
+                               h,theta.density,like){
+   if (type == 1){
+     l <- function(theta){
+      -1*(log(h(theta) * theta.density(theta) * like(theta)))/n
+     }
+   }
+
+   est.normf <- function(n,l,theta0,l.b,u.b){
+     ...
+   }
+   return(est.normf(n,l,theta0,l.b,u.b))
+ }
```

In the first case, assume $p(\theta|Y^{(N)}) = post.den(\theta)$, thus:

$$E_n[h] = E[h(\theta)|Y^{(N)} = \int post.den(\theta) \cdot h(\theta) d\theta$$

To bring integral to the normal form:

$$l(\theta) = -(\log(post.den(\theta)) + \log(h(\theta)))/N$$

```
> l <- function(theta){
+   -1*(log(post.den(theta))+log(h(theta)))/N
+ }
> laplace_estimate(n,l,theta0,l.b,u.b)
```

In the second case, form the function $l(\theta)$ to bring the integrals in the numerator and the denominator to the normal form which we have learned to compute previously.
For numerator:

$$l(\theta) = -(\log(h(\theta) + \log(\pi(\theta)) + \log(L(\theta)))$$

For denumerartor:

$$l(\theta) = -(\log(h(\theta) + \log(\pi(\theta)) + \log(L(\theta)))$$

Thus, we just need to take a ratio. As initial point to search modal value $\hat{\theta_n}$ in numerator we will use modal value $\hat{\theta_d}$ from denumerator. Mixing it all together, we obtain "expect" function.

```
> expect <- function(n,
+                    theta0,l.b,u.b,
+                    h,theta.density,like){
+   one <- function(theta){
+     return(1)
+   }
+
+   denum <- laplace_estimate(n = n, theta0 = theta0, l.b = l.b, u.b = u.b, type = 1,
+                              h = one,
+                              theta.density = theta.density,like = like)
+   num <- laplace_estimate(n = n, theta0 = denum$modal, l.b = l.b, u.b = u.b,
+                            type = 1, h = h, theta.density = theta.density,like = like)
+   return(list(
+     expect = num$result/denum$result,
+     norm.constant = denum$result
+   ))
+ }
```