Bayesian Computation: Nested Sampling Algorithm

Ehsan Karim
ehsan@stat.ubc.ca

April 24, 2008

Abstract

This report is basically discussion on Chopin and Robert [2007] and others on their work on the Nested Sampling, its properties and extension.

1 Motivation of Estimating $\mathfrak{Z}$

Inspired by a quote of MacKay [2003], Skilling [2006] intended to evaluate $\mathfrak{Z}$ by the way of numerical approximation, where

$$\mathfrak{Z} = \int_0^1 \varphi(x)dx. \tag{1.1}$$

The term $\mathfrak{Z}$ is introduced by the name 'evidence'. From the standard Bayesian terminology, $\mathfrak{Z}$ can be expressed as follows:

$$\mathfrak{Z} = \frac{L(\theta|X) \times \pi(\theta)}{P(\theta|X)}, \tag{1.2}$$

where $L(\theta|X)$ is the Likelihood function, $\pi(\theta)$ is the prior distribution. Thus $\mathfrak{Z}$ is basically

$$\mathfrak{Z} = \int L(\theta|X) \times \pi(\theta)d\theta, \tag{1.3}$$

i.e., the predictive or the marginal density. Usually what is done is in usual Bayesian analysis or MCMC algorithms is that estimating θ is the primary interest - hence we ignore $\mathfrak{Z}$, and manage to do that by normalization. Therefore, it might be a valid question why in the first place we would be interested about $\mathfrak{Z}$.

If we look carefully at the current model comparison Bayesian procedures, its obviously Bayes factor, defined as

$$\mathfrak{BF}_{12} = \frac{\int L_1(\theta_1|X) \times \pi_1(\theta_1)d\theta_1}{\int L(\theta_2|X) \times \pi_2(\theta_2)d\theta_2}, \tag{1.4}$$

which is just the ratio of two $\mathfrak{Z}$s, and the nested sampling method just gives us an approximation of such component. That might indicate the importance of estimating $\mathfrak{Z}$, even more than the posterior distribution $P(\theta|X)$ while assessing the models in question. It is thus of interest to see the emerge of a novel proposal for the approximative computation.

2 Nested Sampling: Basic Principle

2.1 Formal Representation

Nested sampling is essentially a new way to solve difficult integration problem. Here we get to evaluate $\mathfrak{Z} = E^{\pi}[L(\theta|X)]$ (for simplicity at first we are considering without any condition imposed on L or π). The basic principle being summing up the use of a step-function with points generated from the prior restricted to higher and higher regions of the likelihood. One simple approach could be to sample θ_1, θ_2, ..., θ_N from the prior π and compute $\frac{1}{N}\sum_{i=1}^{N} L(\theta_i|X)$. One important aspect of nested sampling is that it resorts to simulating θ_i from the prior $\pi(\theta)$ constrained to θ having a larger likelihood value than some increasing threshold l. In large dimension space, simulating from the prior till the constraint is satisfied can be unrealistic. However, lets begin with the simple case:

Let

- $\varphi^{-1}(l) = P^{\pi}[L(\theta|X) > l]$ so that $\varphi^{-1}(l)$ is the complementary cumulative density function (in other words, survival function) of the random variable $L(\theta)$ where $\theta \sim \pi$.
- $\varphi(x) = \sup\{l : \varphi^{-1}(l) > x\}$

Then if $x \sim U(0,1)$, we have $\varphi(x) \sim L(\theta|X)$ in general from the inverse transform sampling principle.

Thus,

$$\begin{aligned} \mathfrak{Z} &= \int_0^{\varphi_{max}} l d\varphi^{-1}(l) \\ &= \int_0^1 \varphi(x)dx, \end{aligned} \tag{2.1}$$

where the integrand is $\varphi(x)$ is a decreasing function.

Since equation (2.1) is one-dimensional, this can be approximated by standard quadratic methods, so that –

$$\begin{aligned} \hat{\mathfrak{Z}} &\cong \sum_{i=1}^{j} \varphi(x_i)(x_{i-1} - x_i) \\ &\cong \sum_{i=1}^{j} \varphi(x_i)w_i, \end{aligned} \tag{2.2}$$

where $0 = x_j < x_{j-1} < ... < x_1 < x_0 = 1$ is an arbitrary grid over $[0,1]$ and $w_i = (x_{i-1} - x_i)$. However, with equation (2.2), the problem now reduces to estimating $\varphi(x_i)$, which is the $(1 - x_i)$-th quantile of $L(\theta|X)$, where $\theta \sim \pi(\theta)$. However, the function $\varphi(x_i)$

is usually not tractable, and hence have to be approximated by an iterative mechanism, that avoids the need to evaluate $\varphi(x_i)$ directly.

Among the other approaches, a convenient one is as follows:

- Assume that x_i, $i = 1, 2, .., j$ are randomly generated
- Let $x_1 = t_1$, $x_2 = t_1 \times t_2$, ... , $x_j = t_1 \times t_2 \times ... \times t_j$, that is, in general, $x_i = t_1 \times t_2 \times ... \times t_i$ or $x_i = t_i \times x_{i-1}$ (with $x_0 = 1$) where t_i is distributed as the largest order statistics in a sample on N from the $U(0, 1)$ distribution - that makes $f(t_i) = Nt_i^{N-1}$, i.e., $t_i \sim Beta(N, 1)$ independently (using properties of order statistics).
- The properties of the $\log t_i$ is as follows:
  1. We evaluate $E(\log t_i)$:

$$\begin{aligned} E(\log t_i) &= \int_0^1 \log t_i N t_i^{N-1} dt \\ &= \int_{-\infty}^0 sNe^{(N-1)s} de^s \\ &= \frac{1}{N} \int_{-\infty}^0 ve^v dv \\ &= -\frac{1}{N} \int_0^\infty u^{2-1} e^{-u} du \\ &= -\frac{1}{N}\Gamma(2) \\ &= -\frac{1}{N}, \end{aligned} \tag{2.3}$$

  by using variable transformations $s = \log t$, $v = Ns$ and $u = -v$; and applying gamma function.

  2. We evaluate $E((\log t_i)^2)$:

$$\begin{aligned} E((\log t_i)^2) &= \int_0^1 (\log t_i)^2 N t_i^{N-1} dt \\ &= 2\frac{1}{N^2}, \end{aligned} \tag{2.4}$$

  by using variable transformations and applying gamma functions just in the same way as equation (2.3).

  3. We evaluate $Var(\log t_i)$:

$$\begin{aligned} Var(\log t_i) &= E((\log t_i)^2) - [E(\log t_i)]^2 \\ &= \frac{1}{N^2}, \end{aligned} \tag{2.5}$$

- As each individual $\log t_i$ are independent, therefore, after i-th iteration $\log x_i \approx -(i \pm \sqrt{i})/N$. To simplify the calculation, its approximated as follows (and hence some noise is expected):

$$\begin{aligned} \log x_i &\cong -(i \pm \sqrt{i})/N \\ \Rightarrow \log x_i &\cong -i/N \\ \Rightarrow x_i &\cong \exp\{-i/N\} \end{aligned} \tag{2.6}$$

  However, one should know that this approximation is rather crude, unless N is large. And the above relation does not come from expectation, although the difference can be defined like that for large N.

- Thus, $E(w_i) = E(x_{i-1} - x_i) \simeq \exp(-\{i-1\}/N) - \exp(-i/N)$ as $N \to \infty$

- Also we know $\varphi(x_i)$ has the same distribution of $L(\theta_i|X)$ as mentioned before. Thus, our estimate becomes

$$\hat{\mathfrak{z}} \cong \sum_{i=1}^{j} L(\theta_i|X)\big[\exp(-\{i-1\}/N) - \exp(-i/N)\big] \tag{2.7}$$

  However, to make this even rigorous, we need to show that equation (2.7) converges to equation (2.1).

We will discuss about how to estimate that $\varphi(x_i)$ in § 2.1.1, with a simple illustration in § A.

2.1.1 Algorithm

1. Initialization:

   a. We start with drawing N independent points $\boldsymbol{\theta} = \{\ \theta_1,\ \theta_2,\ ...,\ \theta_N\ \}$ from prior distribution $\pi(\theta)$

   b. Set $\mathfrak{z} = 0$

   c. Set $x_0 = 1$

2. Repeat for $i = 1, 2, ..., j$: where the number of iterations j is chosen by guesswork (j is chosen such that very small changes in $\mathfrak{z}$ is observed, or reached the maximal value of $L(\theta|X)$, for the bounded cases when the maximum is known):

   a. Obtain the Likelihoods with respect to the drawn θ_1, θ_2, ..., θ_N

   b. Find the minimum of the Likelihood values, and name it $\varphi_i = \min\limits_i L(\theta_i|X)$

c. Find the θ_i value that is responsible for that minimum value of the Likelihood, and name it as $\theta_i = \underset{i}{\operatorname{argmin}} \; L(\theta_i|X)$

d. Set $x_i = \exp(-i/N)$

e. Set $w_i = x_{i-1} - x_i$

f. Increment $\mathfrak{Z}$ such that $\mathfrak{Z} = \mathfrak{Z} + \varphi_i \times w_i$

g. Remove the θ_i used above from the $\boldsymbol{\theta}$ from the i-th position, and replace the i-th position by another one parameter value θ_i' which is drawn from the same prior distribution $\pi(\theta)$: subject to condition that $L(\theta_i'|X) > \varphi_i$ as was defined in the current iteration

3. The output of the algorithm is then the approximation of $\mathfrak{Z}$ given by equation (2.7) which estimates equation (2.1).

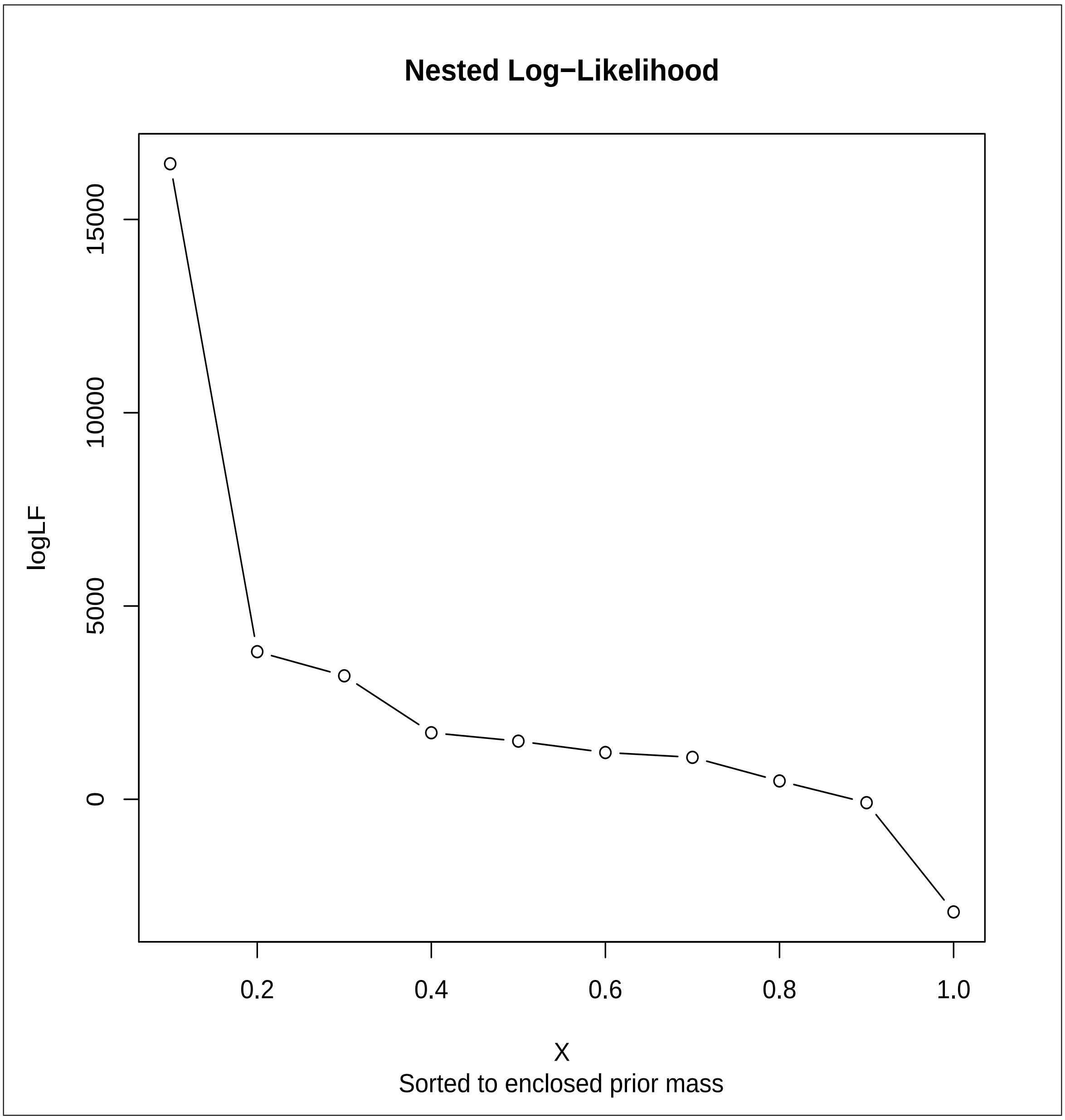
2.1.2 The Probit Example

The illustration of Nested sampling is given by an example where we use probit model on the arsenic data set as is provided in
`http://www.stat.columbia.edu/ gelman/arm/examples/arsenic/wells.dat`
In the data set,

- the dependent variable y_i is 'whether or not the surveyed individual changed the well he/she drinks from in the past three years' and
- the covariates x_i being
  1. distance to nearest safe well in 100 meter unit
  2. educational level for the head of the house
  3. log of arsenic level of the nearest well
  4. interaction between distance to nearest safe well in 100 meter unit and educational level for the head of the house
  5. interaction between distance to nearest safe well in 100 meter unit and log of arsenic level of the nearest well
  6. interaction between educational level for the head of the house and log of arsenic level of the nearest well
  7. and the intercept

The considered characteristics of the probit model is

Figure 2.1: Nested Log-Likelihood sorted to enclosed prior mass from the Probit model example



- The coefficient of the model is a vector θ of size d, where d is the dimension. In particular, for this example, $d = 7$
- Likelihood function is

  $$L(\theta) = \prod_{i=1}^{n} \{\Phi(x_i'\theta)\}^{y_i} \{1 - \Phi(x_i'\theta)\}^{1-y_i} \tag{2.8}$$

  for $n = 3020$
- Prior distribution is $\aleph_d(0, 10^2 I_d)$
- Instrumental Prior distribution is $\aleph_d(\hat{\theta}, \tau^2 I_d)$ for the tuning parameters being $\hat{\theta} =$ posterior modes (shown in figure 2.1) which are obtained using Gibbs sampling algorithm as described in Albert and Chib [1993] (in the paper it says that such posterior mode was obtained numerically, but does not say explicitly which particular method) and τ is set to a large number, say 100 times the identity matrix. The tuning of $\hat{\theta}$ could have values other than the posterior modes. For example, it could be mean of the posterior distribution. Finding such values for $\hat{\theta}$ might not be very easy for some cases. Good news is we do not need to know exact shape of the posterior distribution.
- In the calculation, very small values return: therefore, each step is done on a logarithmic scale.

Figure 2.1 shows nested Log-Likelihood sorted to enclosed prior mass in one single run of the R code presented in § B. The following summary is obtained after putting N = 10, j = 5 in 1000 simulations which shows more variability than expected. Probably it has something to do with log scale, which we needed to consider as the value returns from the written program were so small that R could not manage to keep track of such tiny amount of numbers in usual scale. Even the minimum $\mathfrak{Z}$ value, when converted in usual scale is returned as 0. Including logarithmic version of the analysis, made it very complex anyway.

Minimum	1st Quartile	Median	Mean	3rd Quartile	Max	$\frac{SD}{Mean}$
-3610000	-863000	-543000	-656000	-323000	-2910	-0.705

This is also evident from figure 3.1, although they show the tendency to be more concentrated at some place.

2.2 Variations of the Estimates

Another suggestion for equation (2.2) could be

$$\begin{aligned} \widehat{\hat{\mathfrak{Z}}} &\cong \sum_{i=1}^{j}(x_{i-1}-x_i)\left(\frac{\varphi(x_i)+\varphi(x_{i-1})}{2}\right) \\ &\cong \sum_{i=1}^{j} w_i\left(\frac{(\varphi(x_i)+\varphi(x_{i-1}))}{2}\right), \end{aligned} \tag{2.9}$$

with a higher order quadrature approximation, assuming $\varphi_0 = 0$ (which could be applied in part f of step 2 on the algorithm as discussed in § 2.1.1). However, the gain in using $\widehat{\hat{\mathfrak{Z}}}$ relative to $\hat{\mathfrak{Z}}$ is negligible to the approximation error as the approximation error of $\hat{\mathfrak{Z}}$ is dominated by a stochastic term that has a limiting Gaussian distribution.

3 Subject Matters of the Paper

The paper of Chopin and Robert [2007] is concerned about the following factors:

1. **Universality**: Simulation from s constrained distribution can cause a lot of practical difficulties. There are lot of cases where such techniques are not straightforward. One of the suggestions were to use MCMC: i.e., sample values of θ by iterating, say k MCMC steps (treating the truncated prior as the invariant distribution where the starting values would be the point selected at the previous iteration). But such solution is not free from problems:

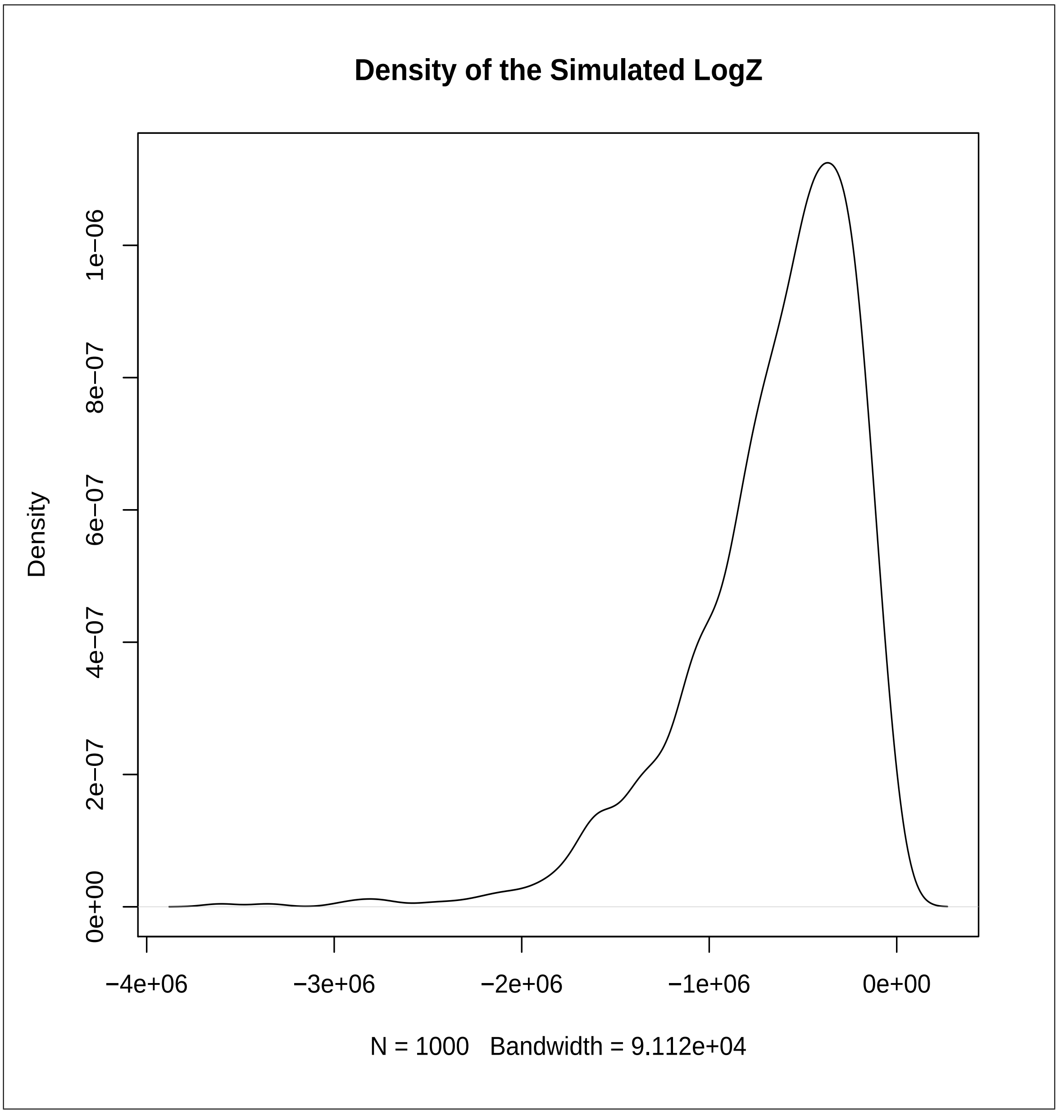
   - As the k-th iterate of a MCMC chain is not distributed according to the constrained prior, such procedure introduces bias
   - In some cases, implementing such MCMC is not straightforward at all.

   Therefore, this paper recommends an extension of this nested sampling to avoid use of MCMC, which we will call 'Nested Importance Sampling' (discussed in § 5) for obvious reasons.

2. **Convergence Properties**: This paper shows that the approximate error has nice properties, such as having a limiting Gaussian distribution. However, the variance estimates grow linearly with dimension, although the integral in equation (2.1) is always uni-dimensional.

However, this paper does not discuss about the vague priors - that usually induces a lack of efficiency when likelihood is quite concentrated within the support of prior (i.e.,

Figure 3.1: Density Plot of the LogZ for N = 10, j = 5, seed = 1 to 1000



unusually low evidence value), thus not enough to provide a well-defined Bayes factor. Although the method has some similarity with annealing (which works by softly compressing the domain), this paper does not address it properly. Also the issue of multi-modality is not addressed in details.

4 Posterior Simulation

Nested sampling can provide simulations from the posterior distribution at no extra cost as the existing sequence of the parameter points (of θ) gives a vector of simulations from the posterior already (subject to the condition that i-th one is the appropriate importance weight $w_i \times L_i$ for L_i being equal to φ_i). That is, at each iterations of computing evidence, a few optional steps can calculate the posterior. It is evident from the computation of the posterior expectation of a given function f as follows:

$$\mu(f) \quad = \quad \int \frac{\pi(\theta)L(\theta)f(\theta)}{\int \pi(\theta)L(\theta)d\theta}d\theta \tag{4.1}$$

which can be estimated as follows:

$$\hat{\mu(f)} \quad \simeq \quad \frac{\sum_{i=1}^{j}(x_{i-1}-x_i)\varphi(x_i)f(\theta_i)}{\hat{\mathfrak{Z}}} \tag{4.2}$$

where numerator $\sum_{i=1}^{j}(x_{i-1}-x_i)\varphi(x_i)f(\theta_i)$ is an approximation to $\sum_{i=1}^{j}(x_{i-1}-x_i)\varphi(x_i)\tilde{f}(\varphi_i)$ which is obviously an estimate of $\int_0^1 \varphi(x)\tilde{f}\{\varphi(x)\}$. Also, instead of using $\hat{\mathfrak{Z}}$, it is possible to use $\hat{\hat{\mathfrak{Z}}}$ in equation (4.2). Here $\tilde{f}(l)$ is defined as the prior expectation of $f(\theta)$ conditional on $L(\theta) = l$. If we can prove that $\int_0^1 \varphi(x)\tilde{f}\{\varphi(x)\}$ is equal to the numerator of equation (4.1), that is $\int \pi(\theta)L(\theta)f(\theta)d\theta$, it will justify all these calculations / approximation. The proof is given in Theorem 1.

Theorem 1. *Let $\tilde{f(l)} = E^{\pi}[f(\theta)|L(\theta) = l]$ for $l > 0$, then $\tilde{f}$ is absolutely continuous, $\int_0^1 \varphi(x)\tilde{f}\{\varphi(x)\} = \int \pi(\theta)L(\theta)f(\theta)d\theta$*

Proof: *As $\tilde{f}$ is absolutely continuous, there exists increasing and decreasing parts as well by definition. Let us consider the increasing part only. Also, due to the absolutely continuity, adding arbitrary constant will return positive function. Thus $\tilde{f}$ is real valued as well.*

Given, $l > 0$ and $\tilde{f}$ is assumed to be positive. For any positive random variable $Y > 0$, $\int_0^{\infty} P(Y > t)dt = E(Y)$. Also, as $\psi : l \rightarrow l\tilde{f}(l)$, this implies $\psi(l) = l\tilde{f}(l) > 0$. Hence, *using these, we get*

$$\begin{aligned}\int \pi(\theta)L(\theta)f(\theta)d\theta &= E^{\pi}[\psi\{L(\theta)\}]\\ &= \int_0^{\infty} P^{\pi}[\psi\{L(\theta)\} > t]dt\\ &= \int_0^{\infty} P^{\pi}[L(\theta) > \psi^{-1}(t)]dt\end{aligned} \tag{4.3}$$

And as $\varphi^{-1} = l \rightarrow P^{\pi}[L(\theta|X) > l]$ this implies $\varphi^{-1}(l) \rightarrow P^{\pi}[L(\theta|X) > l]$. Using that, from equation (4.3) we get -

$$\int \pi(\theta)L(\theta)f(\theta)d\theta = \int_0^{\infty} \varphi^{-1}(\psi^{-1}(t))dt \tag{4.4}$$

Now let

$$\begin{aligned}x = \varphi^{-1}(\psi^{-1}(t))\\ \Rightarrow \varphi(x) = \psi^{-1}(t)\\ \Rightarrow t = \psi\{\varphi(x)\}\end{aligned} \tag{4.5}$$

Therefore from equation (4.4) and (4.5), we get,

$$\begin{aligned}\int \pi(\theta)L(\theta)f(\theta)d\theta &= \int_0^{\infty} \varphi^{-1}(\psi^{-1}(t))dt\\ &= \int_0^{\infty} xdt\\ &= xt|_0^{\infty} - \int_1^0 tdx\\ &= \int_0^1 tdx\\ &= \int_0^1 \psi\{\varphi(x)\}dx\end{aligned} \tag{4.6}$$

where as $x = \varphi^{-1}(\psi^{-1}(t)) = P^{\pi}(L(\theta) > \psi^{-1}(t))$, then while determining limits, if $t = 0$ will give $P^{\pi}(L(\theta) > \psi^{-1}(0)) = P^{\pi}(\psi(L(\theta)) > 0) = 1$ and if $t = \infty$ will give $P^{\pi}(L(\theta) > \psi^{-1}(\infty)) = P^{\pi}(\psi(L(\theta)) > \infty) = 0$.

5 Extension: Nested Importance Sampling

Let

1. $\tilde{\pi(\theta)}$ = an instrumental prior (assuming support of π is included in the support of $\tilde{\pi}$)

2. $L(\tilde{\theta|}X)$ = an instrumental Likelihood

3. We define $\varpi(\theta)$ such that

$$\tilde{\pi}(\theta)\tilde{L}(\theta|X)\varpi(\theta) = \pi(\theta)L(\theta|X) \tag{5.1}$$

Then we can approximate $\mathfrak{Z}$ by implementing Nested sampling algorithm to simulate iteratively from $\tilde{\pi(\theta)}$ subject to the constraint $\tilde{L(\theta|X)} > l$, and computing the generalized Nested sampling estimator as follows (from equation (4.2) which does not require MCMC steps):

$$\begin{aligned}
\hat{\mu(\varpi)} &\simeq \frac{\sum_{i=1}^{j}(x_{i-1} - x_i) \times \varphi(x_i)\varpi(\theta_i)}{\hat{\mathfrak{Z}}} \\
&\simeq \frac{\sum_{i=1}^{j} w_i \times \tilde{L}(\theta_i|X)\varpi(\theta_i)}{\hat{\mathfrak{Z}}} \\
&\simeq \frac{\sum_{i=1}^{j} w_i \times \left(\frac{\pi(\theta_i)L(\theta_i|X)}{\tilde{\pi}(\theta_i)}\right)}{\hat{\mathfrak{Z}}},
\end{aligned} \tag{5.2}$$

utilizing relation in equation (5.1).

For illustration, let $\tilde{\pi} \sim \aleph_d(\hat{\theta}, \tau^2 I_d)$, where d is the dimension of θ. Thus, in equation (5.2), only tuning parameters of the algorithm are the provided hyper-parameters $\hat{\theta}$ and τ of the instrumental prior $\tilde{\pi(\theta)}$. The paper suggests to use the posterior mode as the value for $\hat{\theta}$ and some large values for τ to obtain reasonably good estimates.

The paper also stresses that this extended version of the nested sampling also suffers from same drawbacks as numerical integration - including the curse of dimensionality.

A A Toy Example in R

```
# Nested sampling for normal variance and 1/exponential(1) prior
begin.time<-Sys.time()
begin.times <- format(begin.time, "%X")
# Step 1: initialize N, Z, X, thetas
N=10
thetas=rexp(N) # N points of thetas from prior - 1(a)
Z=0 # initialize for Z - 1(b)
j=10 # select subjectively by guessing until stabilized
X=double(length=j+1) # initialize for X - 1(c)
X[1]=1

# Step 2: The loop

L <- function(y){dnorm(5,sd=1/sqrt(y))} # Likelihood for 2(a)
# record the lowest of the current likelihood values as L[i]
low.position.in.L=order(L(thetas))[1]
low.current.L=(L(thetas))[low.position.in.L]
# lowest value of likelihood - 2(b)
theta.for.low.L=thetas[low.position.in.L]
# theta responsible for lowest value of Likelihood - 2(c)
X[2]=exp(-1/N) # crude x[i] - 2(d)
wi = X[1]-X[2] # simple w[i] - 2(e)
Z=Z+low.current.L*wi # increase Z by L[i]*w[i] - 2(f)

# replacing theta by theta' with condition - 2(g)
for (i in 2:j){
condition = T
     while(condition){
     new.theta = rexp(1)
     condition = L(new.theta)<low.current.L # condition on 2(g)
     }
   thetas[low.position.in.L]=new.theta # the replacement included
   low.position.in.L=order(L(thetas))[1]
   low.current.L=(L(thetas))[low.position.in.L] # 2(b)
   theta.for.low.L=thetas[low.position.in.L] # 2(c)
   X[i]=exp(-i/N) # 2(d)
   wi = (X[i-1]-X[i]) # 2(e)
   Z=Z+low.current.L*wi # 2(f)
   # go back to 2(g) untill j iterations completed

cat("results of iteration", i, "for diagnosis", "\n")
cat("--------------------------------------","\n")
cat("new.theta =",new.theta,"\n")
cat("thetas[low.position.in.L] =",thetas[low.position.in.L],"\n")
cat("low.position.in.L =",low.position.in.L,"\n")
cat("low.current.L =",low.current.L,"\n")
cat("theta.for.low.L =",theta.for.low.L,"\n")
cat("X[",i,"] =",X[i],"\n")
cat("Z =",Z,"\n")
cat("end of iteration", i,"\n")
cat(" ","\n")
}

# Step 3:
end.time<-Sys.time()
end.times <- format(end.time, "%X")
run.time<-difftime(end.time,begin.time,units="secs")

cat("The estimated value of logZ is ",log(Z),"\n")
cat("R Program Run time:", run.time, ’sec.\n’)
```

For evidence, Mean is -10.180, standard deviation is 2.421401 on log scale, calculated for 1000 runs.

B The Probit Example R code

Here is the probit example (for checking purposes, a logit option is also included) on which we implement the nested sampling as follows by writing a simple function:

```
n.s<-function(N =10, j=10, model = c("probit","logit"), seed = 100){

options(digits=4)

# Probit regression for arsenic data, available from
# http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat

wells <- read.table("http://ehsan.karim.googlepages.com/wells.dat",
header=TRUE)
attach(wells)
dist100 <- dist/100
log.arsenic <- log (arsenic)
educ4 <- educ/4
probit.fit <- glm (switch ~ dist100 + log.arsenic + educ4 +
dist100:log.arsenic + dist100:educ4 + log.arsenic:educ4,
family=binomial(link="probit"))
logit.fit <- glm (switch ~ dist100 + log.arsenic + educ4 +
dist100:log.arsenic + dist100:educ4 + log.arsenic:educ4,
family=binomial(link="logit"))
dist100.log.arsenic<-dist100*log.arsenic
dist100.educ4<-dist100*educ4
log.arsenic.educ4<-log.arsenic*educ4
# making suitable data for further calculation
arsenic.data<-cbind(rep(1,length(educ4)),dist100,log.arsenic,
educ4, dist100.log.arsenic, dist100.educ4, log.arsenic.educ4)

# simulate from the posterior distribution by means of the data
# augmentation / Gibbs sampling algorithm of Albert and Chib (1993)

bayes.probit <- function (y, X, m) {
    N = length(y)
    fit = glm(y ~ X - 1, family = binomial(link = probit))
    beta = fit$coef
    p = length(beta)
    beta = array(beta, c(p, 1))
    Mb = array(0, dim = c(m, p))
    aa = chol(solve(t(X) %*% X))
    for (i in 1:m) {
        lp = X %*% beta
        bb = pnorm(-lp)
        tt = (bb * (1 - y) + (1 - bb) * y) * runif(N) + bb *
            y
        z = qnorm(tt) + lp
        mn = solve(t(X) %*% X) %*% (t(X) %*% z)
        beta = t(aa) %*% array(rnorm(p), c(p, 1)) + mn
        Mb[i, ] = t(beta)
    }
    return(Mb)
}

sim.par <- bayes.probit(switch, arsenic.data, 100)
# calculating median
theta.med <- apply(sim.par, 2, quantile, 0.5)
# calculating mode by approximation
mdd1 <- 3*median(sim.par[,1]) - 2*mean(sim.par[,1])
mdd2 <- 3*median(sim.par[,2]) - 2*mean(sim.par[,2])
mdd3 <- 3*median(sim.par[,3]) - 2*mean(sim.par[,3])
mdd4 <- 3*median(sim.par[,4]) - 2*mean(sim.par[,4])
mdd5 <- 3*median(sim.par[,5]) - 2*mean(sim.par[,5])
mdd6 <- 3*median(sim.par[,6]) - 2*mean(sim.par[,6])
mdd7 <- 3*median(sim.par[,7]) - 2*mean(sim.par[,7])
mdd<-c(mdd1,mdd2,mdd3,mdd4,mdd5,mdd6,mdd7)
Switch <- switch
detach()

# useful function to deal with log summation
# for x=log(a) and y=log(b), returns log(a+b)
log.addition.formula <- function(x,y) {
    if(x>y) x+log(1+exp(y-x))
    else    y+log(1+exp(x-y))
```

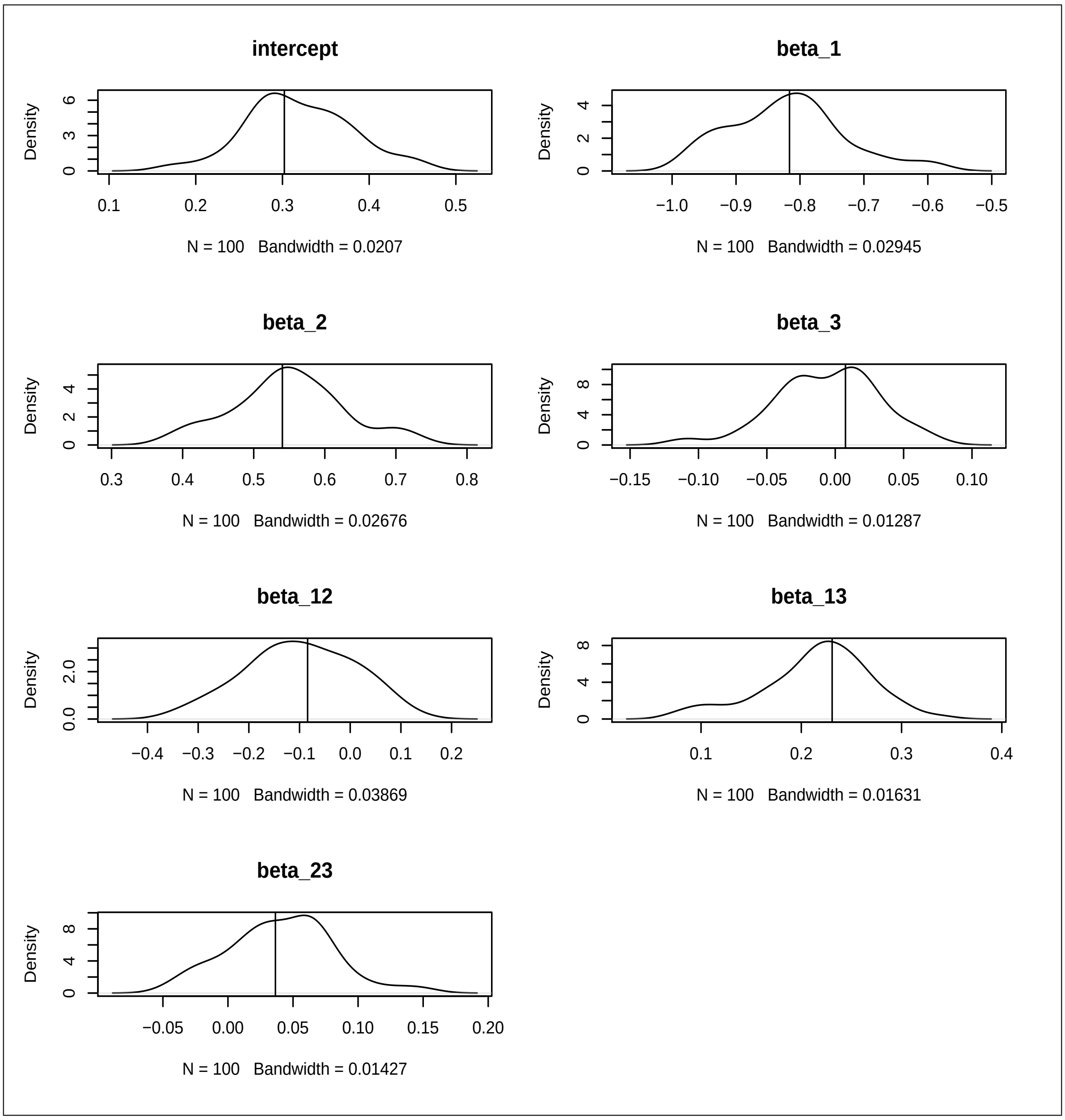
```
}

# Step 1: initialize N, Z, X, thetas
begin.time<-Sys.time()
begin.times <- format(begin.time, "%X")
d= 7 # dimesnsion
logZ<- double(length=j+1)
# ln(Evidence Z, initially 0)
logZ[1] <- -Inf

# Instrumental Prior
require(MASS)
set.seed(seed)
thetas.i = mvrnorm(n = N, mu =mdd , Sigma = 100*diag(d))

# Prior
thetas.v <- mvrnorm(n = N, mu=rep(0,d), Sigma = 100*diag(d))

# LF
# function to calculate the log version of the likelihood

probitll=function(beta,y,X){
lLF=pnorm(X%*%as.vector(beta),mean=0, sd=1, lower.tail = T, log.p = T)
sum(y*(lLF)+(1-y)*(1-lLF))
}
logitll=function(beta,y,X)
{
lF1=plogis(X%*%as.vector(beta),log.p=TRUE)
lF2=plogis(-X%*%as.vector(beta),log.p=TRUE)
sum(y*lF1+(1-y)*lF2)
}

L.i <-double(length=N)
if (model == "logit")
{
for (i in 1:N){
L.i[i] <- logitll(thetas.v[i,],Switch,arsenic.data)
}
}
if (model == "probit")
{
for (i in 1:N){
L.i[i] <- probitll(thetas.v[i,],Switch,arsenic.data)
}
}

cat("Under", model, "model, initial log-LF =",L.i, "\n")
cat(" ","\n")

# Step 2: The loop
# record the lowest of the current likelihood values as L[i]
low.position.in.L=order(L.i)[1]
low.current.L=L.i[low.position.in.L]
# lowest value of likelihood - 2(b)
theta.for.low.L=thetas.v[low.position.in.L,]
# theta responsible for lowest value of Likelihood - 2(c)
logwidth <- double(length=j)
logwidth[1] <- log(1 - exp(-1/N))
log.change <- double(length=j)
log.change[1] <- logwidth[1] + low.current.L
logZnew <- log.addition.formula(logZ[1], log.change[1])
logZ[2] <- logZnew
logZ



cat("results of iteration 1 for diagnosis", "\n")
cat("-------------------------------","\n")
cat("thetas.v[low.position.in.L] =",thetas.v[low.position.in.L],"\n")
cat("low.position.in.L =",low.position.in.L,"\n")
cat("low.current.L =",low.current.L,"\n")
cat("theta.for.low.L =",theta.for.low.L,"\n")
cat("logZ =",logZ[2],"\n")
cat("end of iteration", i,"\n")
cat(" ","\n")
```

```
# replacing theta by theta' with condition - 2(g)
for (i in 2:j){
condition = T
     while(condition){
     new.theta = mvrnorm(n = 1, mu=mdd, Sigma = 100*diag(d))
     condition = probitll(new.theta,Switch,arsenic.data)<low.current.L
     # condition on 2(g)
     }
   thetas.v[low.position.in.L,]=new.theta # the replacement included
   low.position.in.L=order(L.i)[1]
   low.current.L=L.i[low.position.in.L] # 2(b)
   theta.for.low.L=thetas.v[low.position.in.L,] # 2(c)
   logwidth[i] <- log(exp(-(i-1)/N) - exp(-i/N))
   log.change[i] <- logwidth[i] + low.current.L
   logZnew <- log.addition.formula(logZ[i], log.change[i])
   logZ[i+1] <- logZnew
   # go back to 2(g) untill j iterations completed

cat("results of iteration", i, "for diagnosis", "\n")
cat("-------------------------------------","\n")
cat("new.theta =",new.theta,"\n")
cat("thetas.v[low.position.in.L] =",thetas.v[low.position.in.L],"\n")
cat("low.position.in.L =",low.position.in.L,"\n")
cat("low.current.L =",low.current.L,"\n")
cat("theta.for.low.L =",theta.for.low.L,"\n")
cat("logZ =",logZ[i+1],"\n")
cat("end of iteration", i,"\n")
cat(" ","\n")
}

# pdf("c:\\LL.pdf", width = 7, height = 7, version = "1.4")
index<-(1:N)/N
plot(index,sort(L.i,decreasing = T),main = "Nested Log-Likelihood",
sub = "Sorted to enclosed prior mass", xlab="X", ylab ="logLF", type = "b")
# dev.print( postscript, horizontal=FALSE, onefile=FALSE,
# bg="white", width=7, height=7, paper="special", file="c:\\LL.eps" )
# dev.off()

# Step 3:
end.time<-Sys.time()
end.times <- format(end.time, "%X")
run.time<-difftime(end.time,begin.time,units="secs")

cat("The estimated value of log(Z) is ",logZ[j+1],"for N =",N,
"and j =",j, "under",model, "model","\n")
cat("R Program Run time:", run.time, 'sec.\n')

return(list(Sim.par=sim.par, modes = mdd, logz = logZ[-1], LLF = L.i))
}
```

And use the function as follows to get the results:

```
ns<- n.s(N = 10, j = 10, model = "probit", seed =601)

# checking graphically how well the approximation of mode is
x11()
# postscript("c:\\prb.eps", width = 7, height = 7)
# pdf("c:\\prb.pdf", width = 7, height = 7, version = "1.4")
par(mfrow=c(4,2))
plot(density(ns$Sim.par[,1]),main = "intercept")
abline(v=ns$modes[1])
plot(density(ns$Sim.par[,2]),main = "beta_1")
abline(v=ns$modes[2])
plot(density(ns$Sim.par[,3]),main = "beta_2")
abline(v=ns$modes[3])
plot(density(ns$Sim.par[,4]),main = "beta_3")
abline(v=ns$modes[4])
plot(density(ns$Sim.par[,5]),main = "beta_12")
abline(v=ns$modes[5])
plot(density(ns$Sim.par[,6]),main = "beta_13")
abline(v=ns$modes[6])
plot(density(ns$Sim.par[,7]),main = "beta_23")
abline(v=ns$modes[7])
```

Figure B.1: Finding the approximate Modes of the Parameter distribution



```
# dev.off()
```

References

J.H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

N. Chopin and C. Robert. Contemplating Evidence: properties, extensions of, and alternatives to Nested Sampling. Technical report, Technical Report 2007-46, CEREMADE, Universite Paris Dauphine, 2007.

D.J.C. MacKay. Information theory, inference, and learning algorithms. 2003.

J. Skilling. Nested sampling for Bayesian computations. *Proceedings of the Valencia/ISBA 8th World Meeting on Bayesian Statistics*, 2006.