Tests of Nested Sampling

John Skilling

October 19, 2008

Abstract

Nicolas Chopin and Christian P. Robert[1] have proposed tests of nested sampling's response to dimensionality, response to shape, and convergence. The tests are well chosen but, regrettably, their results are wrong in every case.

They report a bias with dimension, which cannot and does not occur. They report a bias with shape, which likewise cannot and does not occur. They find a failure of convergence which again cannot and does not occur. They do not offer explanations, but invite the reader to believe their programming rather than the mathematical properties of nested sampling.

I have programmed the same tests, and find none of the reported biasses. My programs and results are in the accompanying files.

1 Example 1: Dimensionality

The first example has a Gaussian likelihood in various dimensions, de-centred with respect to the Gaussian prior. The prior is $\theta_i = 0 \pm 1$ for $i = 1, 2, \ldots, d$, explicitly the Gaussian

$$\Pr(\theta) = \prod_{i=1}^{d} \frac{\exp(-\theta_i^2/2)}{\sqrt{2\pi}}$$

The data are $y_i = \theta_i \pm 1$ with all data values $y_i = 3$, so the likelihood is

$$\Pr(\texttt{Data} \mid \theta) = \prod_{i=1}^{d} \frac{\exp\big(-(\theta_i - 3)^2/2\big)}{\sqrt{2\pi}}$$

Analytically, the evidence for this problem is

$$Z = \Pr(\texttt{Data}) = \left(\frac{\exp(-9/4)}{2\sqrt{\pi}}\right)^d$$

and the information is

$$H = (7/8 + \log\sqrt{2})\, d$$

The nested-sampling program "`Example1.c`" codes this problem. The dimension d is hardwired as `DIM` in line 8, and the main program uses `R` runs of `N` objects, terminating after a number `M` of iterates known analytically to be

[1] Chopin N. and Robert C.P. (October 2, 2007) *Contemplating Evidence: properties, extensions of, and alternatives to Nested Sampling*, copied to `Nestea.pdf` from Chopin's website `http://www.crest.fr/ses.php?user=3017`.

more than adequate. (Practical programs have to guess M, but I can avoid that guesswork here by knowing H and setting $M \gg NH$.) I usually use my own random-number library `ranlib.[ch]` in my codes, and the program starts by seeding this generator with the dimension, which avoids any question of correlation between runs at different dimension. (For user convenience, this generator can also be seeded with a negative number, in which case it uses the positive current time as seed, to enable different but reproducible random sequences.)

The main program continues with standard nested sampling, with exploration within the likelihood constraint delegated to the `Explore` procedure. As programmed, the form of likelihood used internally is the proxy

$$\log L(\theta) = -\sum_{i=1}^{d} (\theta_i - 3)^2$$

which is only modulated to $\Pr(\texttt{Data} \mid \theta)$ when the sequence is sent to store. `Explore` starts by translating the log-likelihood bound `Lstar` to the radius-squared $\texttt{RR} = -\log L$ of the spherical domain of currently allowed θ. It then runs 10 cycles of Gibbs exploration, each time taking the `DIM` coordinates in random order, as controlled by the `Ranperm` random permutation procedure. At each step, only one coordinate `u` is changed, inside the range (`u1`, `u2`) that lies within the constraining radius. Procedure `sampleu` returns a random sample from the prior, within this range. On completion of an iterate, the worst (lowest likelihood) object is written to its stored sequence.

When each run is finished, procedure `Evidence` calculates its Z and H. Actually, Z is calculated and printed as the difference $\Delta \log Z$ between $\log Z$ and the known, analytic truth. It is printed along with its uncertainty, as calculated "properly" by simulating the sequence of enclosed volumes X that presumably accompanied the given likelihood values L. Because the X's are known statistically but not exactly, I allow 1000 simulations to make (almost) sure that the average and standard deviation of $\log Z$ reliably represent the correct inference from the likelihood sequence. Using 1000 simulations is usually ridiculously many, but this is a demonstration that nested sampling works, so I play safe. The main program accumulates the statistics of the deviation $\Delta \log Z$ for final output.

Results for dimensions `DIM` of $5, 10, 15, \ldots, 50$ are given in files `gibbs5.txt` to `gibbs50.txt`. Being the results of random floating-point exploration, such results may differ in detail when compiled and run on different hardware.

- The mean $\Delta \log Z$ should be close to zero, within a standard deviation or so of the inferred average. *Correctly, it is.*
- The individual standard deviation should be typical of the variation computed for each individual run. *Correctly, it is.*
- Because this is a simple problem, those individual standard deviations should also be close to $\sqrt{H/N}$. *Correctly, they are.*
- The 10%, 25%, 50%, 75% and 90% quantiles of $\Delta \log Z$ are also calculated and printed, for display in box-and-whisker form. The median should show no bias with dimension. *Correctly, it doesn't.*

These results are plotted in figure 1. Nested sampling works in this example just as mathematics insists that it must. How could it possibly be otherwise?

Chopin & Robert (their figure 1) report an upward bias of about $d/5$ in $\log Z$. Upward bias cannot be due to inadequate exploration, which necessarily biasses likelihoods and hence evidence values downwards because the high-likelihood region hasn't been found. It can only be programming error, either keyboard or conceptual.

2 Example 2: Shape

In the second example, the likelihood of each datum is averaged over two alternative models. There are six data

$$\mathbf{y} = \{0.25,\ 0.88,\ 2.16,\ 2.45,\ 2.84,\ 3.50\}$$

Each is explicable as a sample from either $\mathcal{N}(0,1)$ or $\mathcal{N}(\mu,\sigma)$, where the mean μ could be anywhere uniformly distributed from -2 to 6, and the variance σ^2 could be anywhere from 0.001 to 16, uniformly distributed in the logarithm (*i.e.* as Jeffreys). The mixture likelihood is thus

$$L(\mu,\sigma) = \Pr(\mathbf{y} \mid \mu,\sigma) = \prod_{i=1}^{6} \left(p \, \frac{\exp(-y_i^2/2)}{\sqrt{2\pi}} + (1-p) \, \frac{\exp\big(-(y_i-\mu)^2/2\sigma^2\big)}{\sqrt{2\pi}\sigma} \right)$$

in which $p = \frac{1}{2}$. This can be re-written as

$$L = Z_0 \, e^{\lambda}$$

where

$$Z_0 = \prod_{i=1}^{6} \frac{1}{2} \, \frac{\exp(-y_i^2/2)}{\sqrt{2\pi}} = 7.75398 \times 10^{-12}$$

is a constant, and

$$\lambda = \sum_{i=1}^{6} \log(1+t_i)\,, \quad t_i = \frac{\big(\exp(-(y_i-\mu)^2/2\sigma^2)\big) \,/\, \sqrt{2\pi}\sigma}{\big(\exp(-y_i^2/2)\big) \,/\, \sqrt{2\pi}}$$

can be used as a proxy for L.

The advantage of the latter form is that ordering can be retained, in the sense that different (μ,σ) almost always lead to arithmetically-different values of t_i, which (with suitably careful coding of $\log(1+t)$ as t when t is small), give the arithmetically-different values of L that nested sampling requires. Otherwise, there is a significant part of the domain in which the latter "$(1-p)$" terms all submerge beneath the former "p" terms to 53-bit double-precision accuracy, leading to apparently-equal likelihoods. With the natural coding of nested sampling, that would bias the evidence value Z. Even so, 2% of the domain has all the t's *extremely* small, less than 2^{-1024} and underflowing the usual double-precision arithmetical range. To retain proper representation even in these corners where λ would appear to be 0, I compute it as `loglogL` representing $\log\lambda$. My apologies for the infelicity — problems aren't usually this sensitive to coding.

Program "`Example2.c`" runs nested sampling for $\mathtt{N} = 10000$ objects, with $\mathtt{M} = 100000$ iterations being more than adequate. Exploration is by very simple Gibbs sampling, allowing the controlling variables

$$x = (\mu + 2)\,/\,8\,, \qquad y = \log(1000\,\sigma^2)\,/\log(16000)$$

to move through the unit square $(0,1)^2$ under uniform prior. The only random generator that's needed is for $\mathtt{Uniform}(0,1)$, so I just use the language's built-in generator (which may give less portable results than my own).

Lazily using the average compression $X_i = e^{-i/N}$ with the likelihood sequence L_i gives an evidence $\log Z = -12.9006 \pm 0.0154$. For comparison, the true value, as evaluated on an adequate 1000 × 1000 grid, is $\log Z = -12.8894$. Nested sampling's estimate is correct.

The program then outputs 1000 histogram-equalised posterior samples, in which point j (from 1 to 1000) is at a likelihood value that excludes a fraction $(j - \frac{1}{2})/1000$ of the posterior, as estimated by nested sampling. This plot gives a more uniform sampling over the posterior than is obtained by selecting iid random samples. The samples are printed out in file `mixture.txt` and plotted in figure 2. Here the ordinate is logarithmic in σ, to ensure that the prior is uniform so that the visual patterns show the effect of likelihood alone. Contours exclude, respectively, 0.0001%, 0.01%, 1%, 10%, 50% of the posterior, as calculated on the 1000 × 1000 grid. The nested-sampling points are coded as

0 to 1% = *black*, 1 to 10% = *blue*, 10 to 50% = *green*, 50 to 100% = *red.*

Because I used a large number of objects, the 1000 plotted points lie almost perfectly within the exact contours.

Chopin & Robert claim that the narrow funnels leading down to the abscissa data points (arrowed) are "attractors for nested sampling" which exhibit "a fatal attraction" for the samples. Er, "**fatal**"??! They plot such a picture in their figure 7 (confusingly using a linear scale in σ, on which the prior is non-uniform). But the picture is quite wrong, and the appearance of the attractive effect merely proves their program to be in error. Nested sampling has no way of detecting the shape of the likelihood contours. It responds to volumes, but is invariant to the shapes so it cannot possibly discover their corners. Indeed, it doesn't. Nested sampling's distribution of points is correct.

Figure 3 plots the eight basins of attraction into which steepest-descent algorithms for $-L$ would fall. If nested sampling did show an attraction into such basins, it might perhaps also show some such bias into the light-blue basin around ($\mu = 0.6$, $\sigma = 0.3$), as well as into the funnels related to individual data. Of course, it doesn't.

3 Example 3: Convergence

In the third example (a study of well usage in Bangladesh), the setting of a switch is modelled as a function of seven variables. I find it difficult to take this study seriously, but the dataset `wells.txt` downloaded from website `http://www.stat.columbia.edu/~gelman/arm/examples/arsenic` has 3020 records, each including precursors for the following variables:

$x_1 =$ distance to nearest well, in units of 100 metres, relative to mean,
$x_2 = \log_e$(arsenic concentration in water, in mg/l), relative to mean,
$x_3 =$ education level from 0 to 17, divided by 4, relative to mean,
$x_4 = x_1 x_2 =$ correlation between distance and arsenic,
$x_5 = x_1 x_3 =$ correlation between distance and education,
$x_6 = x_2 x_3 =$ correlation between arsenic and education,
$x_7 = 1$, to compensate for offsets relative to the means;

$y \ = 0/1$ switch for whether the individual has changed wells recently.

The aim seems to be to find a linear combination

$$s = \sum_{i=1}^{7} x_i \theta_i$$

of these x's which can be used to predict the corresponding y, by using a likelihood defined as

$$L(\theta) = \prod_{k=1}^{3020} \begin{cases} \Phi(s_k) & \text{if } y_k = 1 \\ 1 - \Phi(s_k) & \text{if } y_k = 0 \end{cases}$$

Arsenic poisoning is a serious problem, though I have doubts about the utility of this model. Anyway, Φ is the usual normal cumulant between 0 and 1, and the prior is the conservatively wide Gaussian $\theta_i = 0 \pm 10$ for $i = 1, 2, \ldots, 7$.

My program `readwells.c` translates the given data to the 3020 records of (y, x_1, x_2, x_3) that are needed, writing them to `wells.dat`. It's tedious but possible to evaluate the 7-dimensional θ-integrals by unambiguous brute force to obtain

$$\log Z = -1969.552\,, \qquad H = 34.208\,.$$

Program `Example3.c` generates the nested-sampling sequence of likelihood values, using `N` objects. Exploration is simplest when set up over a prior that is uniform over the unsigned-integer hypercube $[0, 2^{32})^7$, in which each coordinate u_i ranges from 0 to $2^{32}-1$. These u coordinates are transformed to θ values by a large 2^{25}-element lookup table `THETA` of normal cumulants, the lowest 7 bits being ignored.

$$u = 2^{32} \int_{-\infty}^{\theta} \frac{\exp(-x^2/2)}{\sqrt{2\pi}}\, dx$$

Then, with s evaluated for a record, its contribution Φ or $1-\Phi$ (according to the setting of y) is obtained by another lookup table `LOGPHI`. The range of s is in principle unbounded, so I squeeze it into $(-1,\ 1)$ by defining $x = s/\sqrt{1+s^2}$ before digitising x into the lookup table. These tables considerably accelerate the computation, while being big enough to retain better than 1-in-10^6 accuracy without interpolation.

Procedure `Explore` proposes trial moves in which all the u's change simultaneously, through randomisation of their low-order bits. At first, only one high-order bit is preserved, but if that fails two such bits are preserved, then three, and so on until a move is eventually accepted. This integer-based slice-sampling procedure obeys detailed balance, and about ten successful moves are enough to avoid detectable bias in $\log Z$ from 100 runs.

The main program controls 100 runs, each of which produces a sequence of likelihood values L_i, to be accompanied by enclosed volumes X_i. As before, I allow 1000 simulations of the X's to construct $\log Z$, its uncertainty, and H. These estimates are printed out as the result of the program. File `arsenic1.txt` tabulates 100 runs with `N` = 1 object, file `arsenic10.txt` tabulates 100 runs with `N` = 10 objects, file `arsenic100.txt` tabulates 100 runs with `N` = 100 objects, and file `arsenic1000.txt` tabulates 100 runs with `N` = 1000 objects.

Figure 4 displays these as box-and-whisker plots. Specifically, these 100 runs with different numbers of objects gave the following results for $\log Z$:

# objects	1	10	100	1000
mean	−1970.40	−1969.72	−1969.57	−1969.55
±std.dev.	±6.12	±1.64	±0.63	±0.18
90% quantile	−1962.78	−1967.24	−1968.83	−1969.36
75% quantile	−1965.76	−1968.61	−1969.13	−1969.44
50% quantile	−1970.58	−1969.70	−1969.55	−1969.55
25% quantile	−1974.83	−1970.85	−1970.01	−1969.66
10% quantile	−1978.59	−1971.77	−1970.45	−1969.82

These results are straightforwardly consistent with the standard $\log Z \pm \sqrt{H/N} = -1969.552 \pm \sqrt{34.208/N}$ prediction of nested sampling.

Chopin & Robert obtain a bias of +1.1 upwards in $\log Z$, as shown in their figure 8. They call it "small", but it's an obvious and unacceptable 3-sigma-significant systematic over-estimate of Z by a factor of three. Once again, their result is wrong. As they themselves try to prove, nested sampling converges to the truth as the number of objects increases, so even on their own view there can be no such bias.

4 Nested sampling's convergence

The nested sampling estimate of $\log Z$ converges in mean square to the true value as the number of objects $N \to \infty$, provided only that Z and H are both bounded. A proof is in Skilling (2007)[2], reproduced here as `Convergence.pdf` in a manuscript that also improves the presentation of nested sampling. Chopin & Robert attempted a related proof under more restrictive conditions including differentiability which (because the algorithm is independent of the shape and even the topology of the likelihood function) can not be and are not needed.

[2] Skilling J. (2007) *Nested Sampling's Convergence*, comprehensively rejected by Biometrika as inappropriate, unclear, ambiguous, too difficult, unmathematical, actually mistaken, and lacking utility in statistics. It's odd, sometimes, how other people's opinions can mirror one's own...

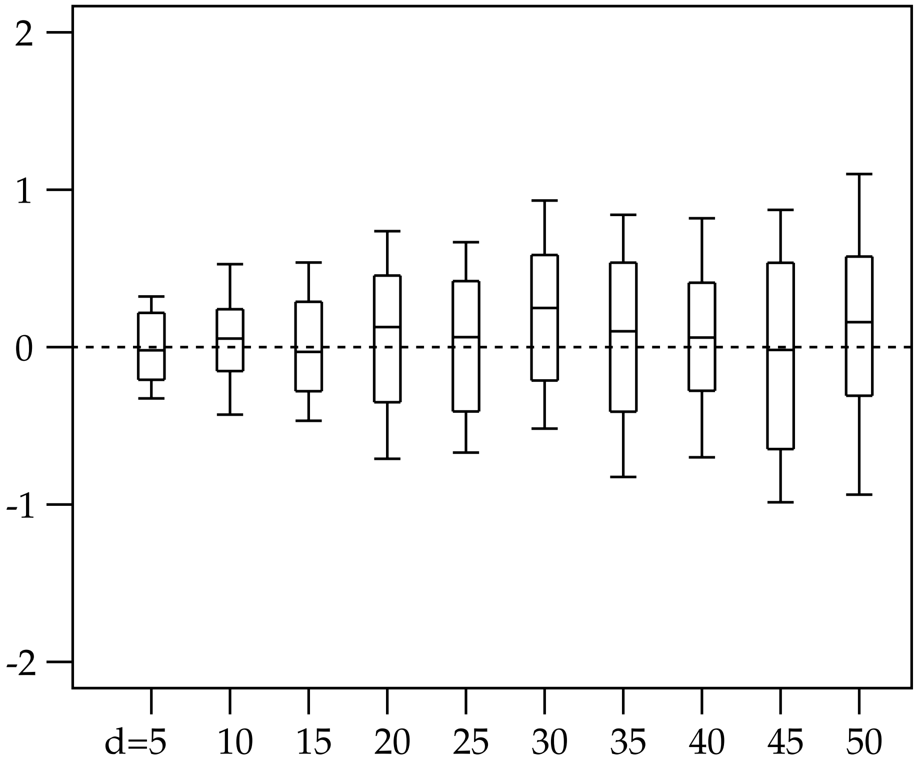

Figure 1:
EXAMPLE 1: Quantiles of $\log Z$ excess at 10%, 25%, 50%, 75%, 90% for 100 runs of 100 objects, for various dimensions up to 50. Sampling is 10 cycles of Gibbs.

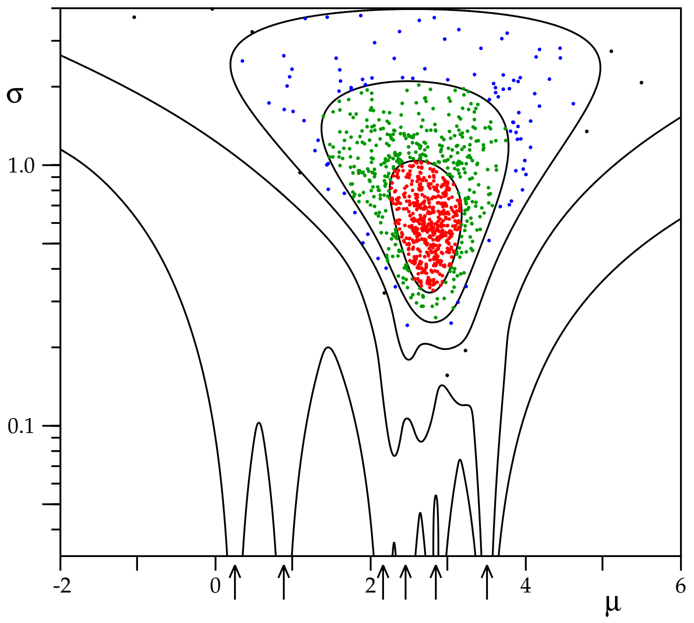

Figure 2:
EXAMPLE 2: 1000 histogram-equalised sample points, superposed on contours enclosing 0.0001%, 0.01%, 1%, 10%, 50% of the posterior. The lowest 1% of points are black, then 1–10% blue, then 10–50% green, then the top 50% red.

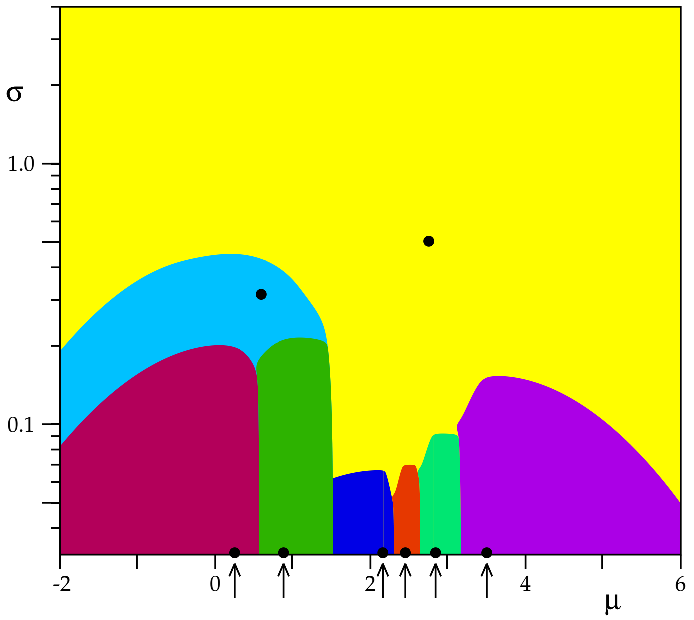

Figure 3:
EXAMPLE 2: The eight basins of attraction, with local maxima shown as dots. Six funnel down to the data points (arrowed), one (light blue) is minor, while the major basin (yellow) covers most of the posterior.

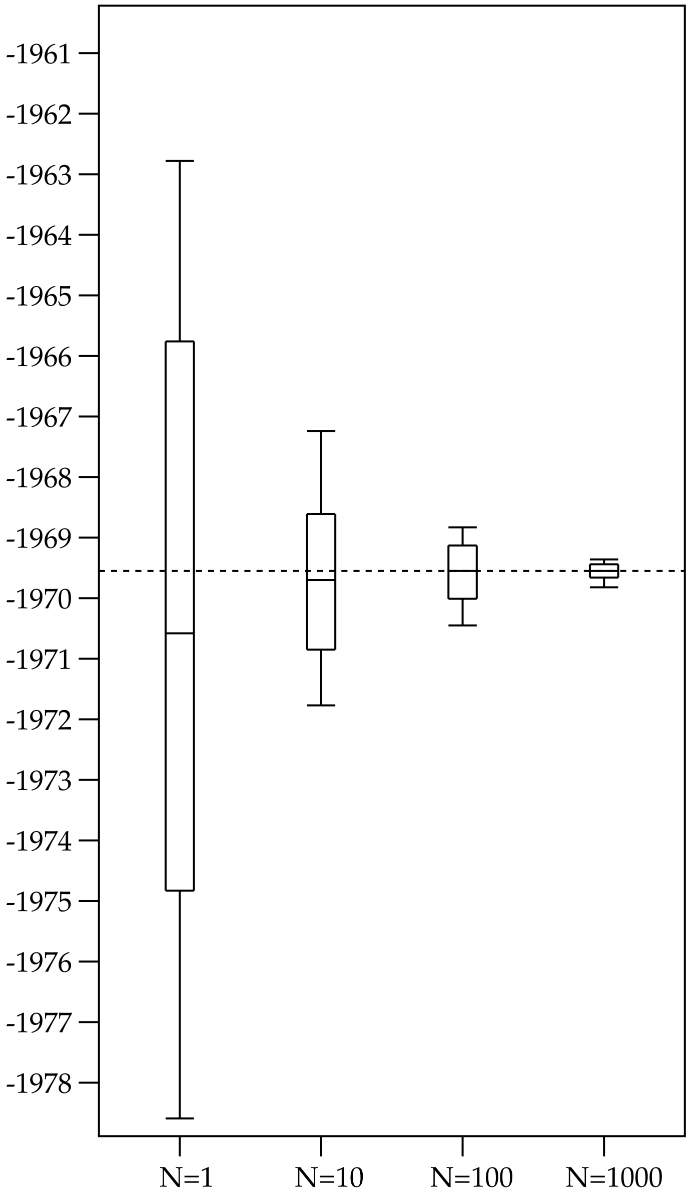

Figure 4:
EXAMPLE 3: Quantiles of $\log Z$ at 10%, 25%, 50%, 75%, 90% for 100 runs of N objects. The dashed line shows the truth. The variation agrees with the predicted standard deviation $\sqrt{34.208/N}$.