Bayesian Curve Fitting using RNested

Bojan Nikolic

Abstract

This document presents some simple examples of fitting curves using the RNested package.

Keywords: Bayesian inference.

> library(RNested)

1. Setting up the problem

The setup of the problem to be analysed proceeds in the standard fashion for Bayesian analysis: the writing down of the likelihood function, the prior probability and the model to be used. The examples here are based on assumption that the observations being analysed have uncertainties that are normally distributed and independent of each other (in real experiments this often not the case!). Therefore, a generator for the log-likelihood $\log P(D|\theta)$ function can be written as:

```r
> lkl1d <- function(x, yobs, model, err)
+ { 
+ lf <- function(p) {
+ ym <- model(x, p)
+ res <- 0.5 * sum(((ym -yobs)/err)**2) +0.5 * sum(log(2*pi* err**2))
+ return (-res)
+ }
+ return (lf);
+ }
```

The parameters to this function are:

- `x` The abscissa points at which observations are available
- `yobs` The observed values
- `model` The model to fit, which should be a function of one variable which is the point in parameter space at which to generate the predicted observations
- `err` The uncertainty in each observations
Bayesian Curve Fitting using RNested

Note also that:

- This is a function that returns a log-likelihood function. The returned function is *closure* because it captures the values of the variables x, yobs, model and err

- The likelihood function is normalised

- The likelihood function takes one argument only which is the point in parameter space \((\theta)\) at which to compute the likelihood. The other necessary information is remembered through the closure mechanism

For the prior function we can use the `boxp` function from the `RNestd` package. So for example to generate a prior with uniform probability in the range \(-1 < x_1 < 6\) and \(-1 < x_2 < 6\) we can use:

```r
> lpf <- boxp(c(-1,6,-1,6))
```

### 2. Fitting a straight line

This is the simplest example: Bayesian fitting of a straight line in one dimension. The model in this case is simply a linear function:

```r
> linemodel <- function(x, p)
+   {
+     x*p[[2]]+p[[1]]
+   }
```

We can generate some synthetic observation data as follows:

```r
> obsx <- 1:10
> obsy <- linemodel(obsx, c(1,2))+rnorm(length(obsx))
> plot(obsx, obsy)
```
Given these synthetic observations and the model, we can generate a likelihood function as:

```r
> slfn <- lkl1d(obsx, obsy, linemodel, rep(1.0, length(obsx)))
```

The next step is to create a starting live set that uniformly populates the prior space. This can be done using the `sset.box` from the `RNested` package:

```r
> ss <- sset.box(array(c(-1,6,-1,6), dim=c(2,2)), 100, slfn)
```

Finally, the nested sampling can be run. It is usually not obvious \textit{a-priori} how many samples need to be taken before the problem converges. For this reason `RNested` provides convenient re-start functionality so that sampler can be restarted if it is determined that convergence has not yet been achieved. This is demonstrated below, where the first run of the sampler is for only 500 samples:

```r
> r <- nested.sample(ss, slfn, lpf, mkCovarianceSampler(), N=500)
> nested.summary(r$cout)
```

```r
*** Evidence: 3.722114e-08
```
As can be seen in the summary plot/evidence growth curve above, the evidence is still clearly increasing after 500 samples, showing that it is that sampler has not converged. Therefore, it is necessary to restart the sampler which can be done as:

```r
> r <- nested.sample(r$cs, slfn, lpf, mkCovarianceSampler(), N=500, r$cout)
> nested.summary(r$cout)

*** Evidence: 1.313578e-07
The evidence curve now shows a turnover indicating that convergence has been achieved.

Function `nested.summary` displays the total evidence while the posterior distribution can be viewed using the `nested.hist2` function:

\[
\text{> nested.hist2(r$cout)}
\]
Finally a fan-diagram can be plotted with:

```r
> nested.fan(r$cout, linemodel,
+ 0, 10, 0,30)
> plotrix::plotCI(obsx, obsy, 1.0, add=T)
```
Affiliation:
Bojan Nikolic
E-mail: bojan@bnikolic.co.uk