Using C to do Bayesian inference via the Metropolis algorithm

First, you need to go to my website `www.biostat.umn.edu/~cavanr` and obtain the files `bayesProgram.cpp` (this holds the code), `ydata.dat` (this holds some simulated data), `initVal.dat` (this holds some initial values), `initValu.dat` (this holds 3 sets of intial values for simulating from 3 chains), and `cov.dat` (this holds the covariance matrix used in the metropolis algorithm).

To compile the program, at the UNIX directory you type

```
g++ bayesProgram.cpp
```

Then to run the program you type (there should be a file called `a.out` in your directory after compiling)

```
a.out
```

Here is a simple description of use of the program. We will suppose that we have 40 measurements that are iid $N(\mu, \sigma^2)$ and try to estimate these parameters using the standard non-informative priors for location and scale parameters. So, first we simulate some data from a N(3,1) distribution (this uses S-plus syntax)

```
> ySim1 <- rnorm(40,mean=3,sd=1)
> var(ySim1)
[1] 0.9633639
> mean(ySim1)
[1] 2.846493
```

and so a confidence interval for mu is

```
> mean(ySim1)+1.96*sqrt(var(ySim1)/40)
[1] 3.150667
> mean(ySim1)-1.96*sqrt(var(ySim1)/40)
[1] 2.54232
```

So the interval is (2.54, 3.15).

Then we wrote the following initial values to file

```
> write(c(2,3),"initVal.dat")
```

I have also set up a file that holds cov matrix (this needs to be done with some compilers even if we won't use this file right away)

```
> write(matrix(c(1,0,0,1),2,2),"cov.dat")
```

Next we run program to get the posterior mode (set MAXX=1 and use comments to select a method), the results are as follows:

direction set: 3.0 0.867 (gives 19.3573)

conjugate gradient: 2.846 0.939 (gives 18.7472)

simulated annealing (set SIMANL=1): 2.847 0.939 (gives 18.7472)

So now examine covariance matrix at mode (set GTVR=1)

```
> v1 <- matrix(scan("asymCov.dat"),byrow=T,ncol=2)
> v1
          [,1]        [,2]
[1,] 0.0625382 0.019941
[2,] 0.0199410 0.129903
```

and so we can get the standard errors (but note that the standard error for the second parameter is on the log scale because of the parameterization of the likelihood)

```
> sqrt(diag(v2))
[1] 0.2500764 0.3604206
```

giving the confidence interval for μ

```
> 2.85+1.96*.25
[1] 3.34
> 2.85-1.96*.25
[1] 2.36
```

or just (2.36, 3.34). Next, write this cov matrix to file

```
> write(v1,"cov.dat")
```

then set up initial values

```
> write(c(2,3),"initValu.dat")
```

and run metropolis algorithm (set SMPL=1), burn in 1000 and save next 1000 using one chain. Then read the results into S-plus

```
> s1 <- matrix(scan("postSmpl.dat"),byrow=T,ncol=2)
> quantile(s1[,1],c(.025,.975))
2.5 97.5
2.51548 3.163595
> quantile(s1[,2],c(.025,.975))
2.5 97.5
0.650265 1.531391
```

Now set up 3 chains so that they are overdispersed (use the 3 means 1, 5, 3)

```
> write(c(1,2,5,2,3,.1),"initValu.dat")
```

and the results are similar. We also observe that the chain converges very quickly.