nag

NP3667 (TR3/08)

Introduction to Bayesian Analysis Using the NAG Library

September 25, 2008

The Numerical Algorithms Group Ltd
Wilkinson House
Jordan Hill
Oxford
OX2 8DR
UK

Web Site: www.nag.co.uk
General Inquires: info@nag.co.uk
Technical Support: support@nag.co.uk

1 Bayesian Statistics

Statistical methodology can be split roughly, into two approaches; the frequentist (or traditional) approach and the Bayesian approach.

In many situations there is the concept of a "true" effect, which one is interested in estimating and an "observed" effect, the effect one observes in a set of data.

In a frequentist framework the "true" effect is considered fixed and probability statements are made about the "observed" effect. These probability statements are based on the concept of repeatability.

In a Bayesian framework the "observed" effect is considered fixed and probability statements are made about the "true" effect. These probability statements, often expressed in terms of a posterior distribution, can be viewed as a measure of personal belief; they are a combination of prior belief and the information contained within the observed data.

For a set of observed data, y, and a vector of m parameters, θ, Bayes theorem [1] states

$$P(\theta|y) \propto P(y|\theta)P(\theta)$$

that is, the probability of the parameters given the observed data is proportional to the product of the probability of the data given the parameters and the probability of the parameters. The probability $P(\theta|y)$ is usually refered to as the posterior probability and $P(\theta)$ as the prior probability.

If the parameters θ are continuous, this corresponds to

$$\int p(\theta|y) \propto \int L(y|\theta)p(\theta) \tag{1}$$

where $p(.)$ indicates the density function and $L(y|\theta)$ is the likelihood of the data given the parameters. In many cases, the constant of integration, which converts equation (1) from "proportional to" to "equals" is unknown.

In addition to the posterior distribution, the predictive distribution is also of interest. Given a new set of data, $\bar{y}$, the predictive distribution, $p(\bar{y}|y)$, i.e. the probability of the new observation, given the old observations is given by

$$p(\bar{y}|y) = \int_\theta L(\bar{y}|\theta)p(\theta|y)\mathrm{d}\theta$$

In this paper we illustrate a number of ways of making inferences about the posterior and predictive distributions using routines from the NAG library. The routines and code snippets refered to in this document are from the NAG Fortran library, but similar routines are available in the NAG C library as well.

2 Example

In order to illustrate some of the techniques, that can be easily implemented using the NAG library, we will fit a simple generalised linear model to some example data in a Bayesian manner.

The data for this particular example were taken from page 87 of Bayesian Methods in Finance [6] and consists of a binary response y, taking values; default (1) or not default (0), for 46 companies and four independent variables. The independent variables are the ratios; cash flow to total debt, net income to total assets, current assets to current liabilities and current assets to net sales, taken over a period of one year which was two years prior to 21 of the companies defaulting. In addition data for three hypothetical companies are used to illustrate how the model can be used for prediction.

In brief, a generalised linear model (GLM) consists of the following three elements:

- A suitable distribution for the dependent variable y.
- A linear model, with linear predictor $\eta = X\beta$, where X is a matrix of independent variables and β a column vector of m parameters.
- A link function $g(.)$ between the expected value of y and the linear predictor, that is $\mathrm{E}(y) = \eta = g(\mu)$.

In order to fit a GLM in a Bayesian framework one must also consider a prior distribution for the parameters, β.

In this example, we have binary data and therefore logistic regression will be used. The distribution for the dependent variable is a Bernoulli distribution and we will use a logit link, that is

$$\eta_i = \log\left(\frac{\mu_i}{1-\mu_i}\right)$$

hence the likelihood is given by

$$L(y|\beta) = \prod_{i=1}^{n} \mu_i^{y_i}(1-\mu_i)^{1-y_i}$$

and the log likelihood, $l(y|\beta)$ by

$$l(y|\beta) \propto \sum_{i=1}^{n} y_i x_i \beta - \log(1+\exp(x_i\beta))$$

where x_i denotes the ith row of the design matrix X. There are four ratios and we also fit an intercept so in total there are five independent parameters ($m = 5$). Our vector of parameters, $\theta = \{\beta_0, \beta_1, \beta_2, \beta_3, \beta_4\}$, where β_0 is the parameter associated with the intercept.

For ease, we use an (improper) uniform prior for the β, with $p(\beta) \propto 1$, however, it is relatively straightforward to use a more informative prior if required.

Given a new set of data for the independent variables, $\bar{X}$, draws from the predictive distribution can therefore be made as follows:

1. Draw a sample, $\bar{\beta}$ from the posterior distribution, $p(\theta|y)$.
2. Calculate $\bar{\eta} = \bar{X}\bar{\beta}$
3. Set $\bar{y} = g^{-1}(\bar{\eta})$, which in the case of the above logistic model means

$$\bar{y} = \frac{\exp \bar{\eta}}{1 + \exp \bar{\eta}}$$

Once a sample from the predictive distribution has been obtained, standard monte carlo methodology can be used to make inferences about the predictive distribution, i.e. the mean of the distribution can be approximated by the mean of the sample etc.

3 Approximating the Posterior Distribution

In this section, we briefly describe two approximations to a posterior distribution, the Normal approximation and the Laplace approximation, both of which are based on a Taylor expansion around the posterior mode.

3.1 A Taylor Expansion Around the Posterior Mode

Let $p(\theta|y)$ denote a generic posterior density function, upto the constant of integration, for a parameter set θ (i.e. $p(\theta|y)$ has not been normalised to integrate to one). Under certain regularity conditions, the log of the posterior distribution can be approximated by its Taylor expansion around its mode, $\hat{\theta}$, with

$$\begin{aligned}
\log(p(\theta|y)) &\approx \log(p(\hat{\theta}|y)) \\
&+ \left.\frac{d \log(p(\theta|y))}{d\theta}\right|_{\theta=\hat{\theta}} (\theta - \hat{\theta}) \\
&+ \frac{1}{2}(\theta - \hat{\theta})^T \left(\left.\frac{d^2 \log(p(\theta|y))}{d\theta\theta^T}\right|_{\theta=\hat{\theta}} \right) (\theta - \hat{\theta}) \\
&+ O(3)
\end{aligned}$$

As $\hat{\theta}$ is the mode of the posterior distribution, hence a maximum of the log posterior, the second term on the right hand side evaluates to zero. The log-posterior can therefore be approximated by

$$\log(p(\theta|y)) \approx \text{const} + \frac{1}{2}(\theta - \hat{\theta})^T \left(\left.\frac{d^2 \log(p(\theta|y))}{d\theta\theta^T}\right|_{\theta=\hat{\theta}} \right) (\theta - \hat{\theta}) \tag{2}$$

where the term

$$\left(\frac{d^2 \log(p(\theta|y))}{d\theta\theta^T}\bigg|_{\theta=\hat{\theta}} \right) = H$$

is the Hessian matrix of the log posterior distribution evaluated at $\hat{\theta}$.

A discussion of the accuracy of this approximation, and the necessary regularity conditions, can be found in Tierney and Kadane [8] and the references contained therein.

3.2 Obtaining the Posterior Mode and Resulting Hessian

The mode of the posterior distribution is located at the maximum of the posterior density function. The location of this maximum is invariant under scaling, hence the maximum of the normalised posterior density function is in the same location as the maximum of the unnormalised density function. Therefore estimating the posterior mode boils down to finding the maximum of

$$p(y|\theta)p(\theta)$$

which is equivalent to finding the minimum of

$$-\log(p(y|\theta)p(\theta))$$

There are a number of routines in the NAG library which can be used for this, including E04JYF, E04UCA, E04UFA and E04WDF. Which routine is best will depend on the form of the likelihood, prior distributions and whether the first and second derivatives are available.

In this example, we are using the routine `E04UCA`. It is recommended that the derivatives of the objective function are supplied, where possible. In this example, in order to keep the code as simple as possible we are letting `E04UCA` approximate the derivatives. It should be noted, that if the derivatives are supplied, the optimisation will run significantly quicker. The improvement in speed is particularly noticeable when using the Laplace approximation where a number of optimisations are performed.

As we are not supplying derivatives we need to set the optional `'Derivative Level'` argument to zero:

```
CALL E04UEA('Derivative Level = 0',LWSAV,IWSAV,RWSAV,INFO)
```

`E04UCA` allows for constraints to be put on the parameters being optimised. However, when estimating the posterior mode we have no constraints, so use:

```
NCLIN = 0
NCNLN = 0
BL(1:M) = -INFBND
BU(1:M) = INFBND
```

where INFBND is a constant set to a large value (10^{20} in this case).

We require the Hessian of the log-posterior, evaluated at the posterior mode, so set the optional 'Hessian' argument to 'Yes'

```
CALL E04UEA('Hessian = Yes',LWSAV,IWSAV,RWSAV,INFO)
```

The posterior mode can then be estimated via:

```
CALL E04UCA(M,NCLIN,NCNLN,LDA,LDCJ,M,A,BL,BU,E04UDM,OBJFUN,ITER, &
     INFO,CCON,CJAC,CLAMBDA,OBJF,GRAD,H,BETA,IWORK,LIWORK,WORK,LWORK, &
     IUSER,RUSER,LWSAV,IWSAV,RWSAV,IFAIL)
```

where E04UDM is a dummy constraint function supplied with the NAG library and OBJFUN is a routine that returns $-l(y|\theta) - \log p(\theta)$. For the example described in section (2) we are using a uniform prior so

```
LOGPRIOR = ZERO
```

and the log likelihood is given by

```
! Calculate XB = X* (BETA) (DGEMV = F06PAF)
CALL DGEMV('NoTranspose',N,M,ONE,X,N,BETA,1,ZERO,XB,1)

! Calculate XYB = X * Y * B (DDOT = F06EAF)
XYB = DDOT(M,XY,1,BETA,1)

! Calculate the log likelihood
LOGLIKE = ZERO
DO I = 1, N
  LOGLIKE = LOGLIKE + LOG(ONE + EXP(XB(I)))
END DO
LOGLIKE = LOGLIKE - XYB
```

Once the optimisation has been completed, the value held in the parameter IFAIL should be checked. A list of the possible values for IFAIL can be found in the documentation for E04UCA at the NAG web site.

With the correct options turned on the optimisation routine E04UCA returns an estimate of the Cholesky factorisation of the Hessian matrix evaluated at the minimum value, and in many situations this approximation is sufficient. However, there are some instances when a more accurate estimate is required. If the Hessian cannot be calculated explicitly then E04XAA can be used instead.

```
! Re-calculate the Hessian using E04XAA
MSGLVL = 0
EPSRF = ZERO
HFORW(1:M) = ZERO
CALL E04XAA(MSGLVL,M,EPSRF,BETA,MODE,OBJFUN,M,HFORW,OBJF,GRAD, &
            HCNTRL,H,IWARN,WORK,IUSER,RUSER,INFO,LWSAV,IWSAV,  &
            RWSAV,IFAIL)
```

One of the advantages of using `E04UCA` in the above optimisation is that the form of the objective function required by `E04UCA` is identical to that required by `E04XAA`, and so no further coding is required.

3.3 Normal Approximation

The Normal approximation is a quick and dirty way of approximating a unimodel posterior distribution using the Taylor expansion around the mode given in equation (2).

If we take exponentials of both sides of equation (2), we get

$$p(\theta|y) \propto \exp\left(-\frac{1}{2}(\theta-\hat{\theta})^T(-H)(\theta-\hat{\theta})\right)$$

which is the kernel of a multivariate Normal distribution, with mean $\hat{\theta}$ and covariance matrix $\Sigma = -H^{-1}$. We can therefore use a multivariate Normal distribution, MVN($\hat{\theta}$,$-H^{-1}$) as an approximation to the posterior distribution, $p(\theta|y)$, allowing us to make inferences about the posterior distribution by making inferences about the multivariate Normal distribution. For example, credible intervals for each of the parameters can be obtained by calling `G01FAF`, as in

```
QUANT = G01FAF('C',0.05E0_WP,IFAIL)
DO J = 1, M
  TMP = QUANT*SQRT(SIGMA(J,J))
  WRITE (NOUT,'(I5,3(E12.5,1X))') J, BETA(J) - TMP, BETA(J),   &
        BETA(J) + TMP
END DO
```

Where the vector `BETA` contains the posterior modes and the matrix `SIGMA` holds the covariance matrix obtained from the Hessian matrix.

Given a series of `NPRED` new observations, draws from the predictive distributions can be obtained by making draws from a multivariate Normal distribution and then inverting the link function, for example:

```
! Generate NGEN N(BETA,SIGMA) values
```

```
IFAIL = INFAIL
CALL G05RZF(MODE,NGEN,M,BETA,SIGMA,M,R,LR,STATE,Q,LDQ,IFAIL)

! Get the "eta" from the logistic regression (i.e. the linear predictor)
! Calculate QB = PX * Q (give or take a few transposes) (DGEMM = F06YAF)
CALL DGEMM('NoTranspose','Transpose',NPRED,NGEN,M,ONE,PX,NPRED,Q, &
           LDQ,ZERO,QB,NPRED)

DO I = 1, NGEN
  ! Invert the link function to get the probability
  DO J = 1, NPRED
    EQB = EXP(QB(J,I))
    QB(J,I) = EQB/(ONE+EQB)
  END DO

  ! Display the simulated values
  WRITE (NOUT,*) I, QB(1:NPRED,I)
END DO
```

Prior to calling any of the NAG random number generators, an initialisation routine must first be called. This can either be G05KFF or G05KGF. G05KFF initialises the generators to a repeatable sequence, that is, if you run the program again you will get the same answer. For a non-repeatable sequence the routine G05KGF should be called instead. Within a program, only a single call should be made to one of the initialisation routines as opposed to continually re-initialising the sequence.

3.4 Laplace Approximation

An alternative to the Normal approximation is the Laplace approximation. As with the Normal approximation, the Laplace approximation uses the Taylor expansion given in equation (2). Unlike the Normal approximation, the Laplace approximation uses this expansion twice, once to approximate the marginal distribution and once to approximate the constant of integration. The Laplace approximation is generally better than the Normal approximation, especially for small sample sizes. Both approximations are likely to be inappropriate for more complicated posterior distributions, for example, multimodal distributions.

A description of using the Laplace approximation to estimate both the moments of the posterior distribution and a marginal density is given in Tierney and Kadane [8].

In order to approximate a marginal distribution, a number of calls to the optimisation routine must be made. Firstly the posterior mode, $\hat{\beta}$, and associated Hessian, H, must be obtained in a similar manner to described in section 3.3. Then, for the marginal distribution of β_j, the posterior mode and associated Hessian of $p(\beta_j = k, \beta_{-j}|y)$ is required, where k is a constant and β_{-j} is all the parameters, excluding β_j. If the mode and Hessian for $p(\beta_j = k, \beta_{-j}|y)$ are

denoted $\hat{\beta}_{-j}$ and H_{-j} respectively, then the posterior distribution at $\beta_j = k$ is proportional to

$$\left(\frac{|H_{-j}|}{|H|}\right)^{1/2} \frac{p(\beta_j = k, \hat{\beta}_{-j}) L(y|\beta_j = k, \hat{\beta}_{-j})}{p(\hat{\beta}) L(y|\hat{\beta})}$$

The main additional coding requirement for the Laplace approximation over the Normal approximation, is to write the objective function in such a way that you can fix the parameters, one at a time, to different values.

Estimating the marginal posterior distribution using the Laplace approximation is one of the situations where it is better to use the more accurate Hessian supplied by `E04XAA`, rather than the one used in the optimisation by `E04UCA`.

When using a Laplace approximation to approximate the marginal density function, Tierney and Kadane recommend that the estimated values are renormalised. This can be done via a call to `D01GAF`.

```
DO J = 1, M
  ! Get and approximation to the integral of the marginal
  CALL D01GAF(XVALS(1,J),STORE(1,J),NPOINTS,ANS,ER,IFAIL)

  ! Renormalise the values
  STORE(1:NPOINTS,J) = STORE(1:NPOINTS,J)/ANS
END DO
```

where `XVALS(I,J)` holds a value in the domain of the jth marginal and `STORE(I,J)` holds the Laplace approximation to the jth marginal distribution evaluated at `XVALS(I,J)`.

4 Iterative Methods for Sampling from the Posterior Distribution

Rather than using an approximation, one could use an iterative sampling method in order to explore the posterior distribution.

Two approaches will be briefly mentioned here, one based on a Metropolis-Hastings sampler and the other based on importance sampling.

4.1 Metropolis-Hasting Independence Sampler

Markov Chain Monte Carlo (MCMC) methodology is one way of drawing samples from a specified distribution. As its name suggests, an MCMC algorithm uses the stationary

distribution of a Markov chain to sample from the distribution and Monte Carlo integration to make inferences about the distribution.

In order to sample using a Markov chain one must first construct a chain whose stationary distribution is the distribution of interest. The Metropolis-Hastings algorithm [5] is one way of doing this. This algorithm consists of four steps:

1. Generate an initial set of values for the parameters θ and label it $\theta^{(0)}$. Set the counter $t = 1$.
2. Generate a new set of values, θ', from some arbitary conditional distribution $q(\theta'|\theta^{(t)})$, often called the proposal distribution.
3. Accept the new values, θ', with probability

$$\alpha(\theta^t, \theta') = \min\left(1, \frac{p(y|\theta')p(\theta')}{p(y|\theta^{(t)})p(\theta^{(t)})}\frac{q(\theta^{(t)}|\theta')}{q(\theta'|\theta^{(t)})}\right)$$

   and set $\theta^{(t+1)} = \theta'$, otherwise keep the old values and set $\theta^{(t+1)} = \theta^{(t)}$.
4. Increment the counter t and repeat steps (2) and (3).

The choice of proposal distribution q is paramount when using an MCMC algorithm. There have been a number of standard types of proposal distribution suggested, including the Metropolis sampler, the independence sampler and the Gibbs sampler. Gilks et al. [3] give a good description of MCMC methodology, including a discussion of these proposal distributions amongst others.

In this example we will take a look at one particular example of an independence sampler. The independence sampler uses a proposal distribution that is independent of the current parameter values, that is $q(\theta'|\theta^{(t)}) = q(\theta')$. It is usually good practice to choose a proposal distribution that has a similar location and scale to the posterior distribution. One such choice is a Normal distribution, with mean equal to the posterior mode and covariance matrix $-H^{-1}$ where H is the Hessian of the log-posterior distribution evaluated at the posterior mode, $\hat{\theta}$, i.e.

$$q(\theta') \sim N(\hat{\theta}, -H^{-1})$$

It is common practice to scale the covariance matrix by some factor `SCALE`. This can be done by scaling `H`, the matrix returned by `E04UCA` during the estimation of the posterior mode (see section 3.2) as follows:

```
SSCALE = ONE/SQRT(SCALE)
DO J = 1, M
  ! Scale the upper part of H (DSCAL = F06EDF)
  CALL DSCAL(J,SSCALE,H(1,J),1)
END DO
```

prior to explicitly calculating the covariance matrix Σ:

```
DETS = ONE
DO I = 1, M
  ! Copy upper part of H (DCOPY = F06EFF)
  CALL DCOPY(I,H(1,I),1,SIGMA(1,I),1)
  DETS = DETS*H(I,I)
END DO
LDETS = -TWO*LOG(DETS)

! Calculate SIGMA = (H^T H)^{-1} (DPOTRI = F07FJF)
CALL DPOTRI('Upper',M,SIGMA,M,INFO)
```

Here we first copy the upper Cholesky decomposition of the Hessian which is held in `H` into `SIGMA`, before calculating the inverse of the Hessian with a call to `DPOTRI`. We also calculate the log of the determinant of Σ, `LDETS`, at the same time.

In order to carry out the analysis using an MCMC algorithm an array of proposed values is required, which in this case come from a multivariate Normal distribution

```
CALL G05RZF(MODE,NSIM,M,BETA,SIGMA,M,R,LR,STATE,Q,LDQ,IFAIL)
```

along with a vector of values from a standard uniform distribution to test for acceptance / rejection of the proposed values.

```
CALL G05SAF(NSIM,STATE,U,IFAIL)
```

As mentioned before, prior to calling any of the NAG random number generators, one of the initialisation routines, G05KFF or G05KGF must have been called first. Now all we do is loop over each of these generated values.

```
SIM_LOOP:DO I = 1, NSIM
  ! Copy a single variate from the array of multivariate normal
  ! variates
  CQ(1:M) = Q(I,1:M)

  ! Get probability of the proposed value from the proposal distribution
  PQ2 = MVN_PDF(M,CQ,BETA,SH,M,SLDETS,D)

  ! Get posterior, P(Q | X, Y), upto the normalising constant
  LX2 = EXP(-LOGPOSTERIOR(N,M,X,N,CQ,XY,XB))

  ! Acceptance probability
  ACCEPT = MIN(ONE,(LX2*PQ1)/(LX1*PQ2))
```

```
  IF (ACCEPT>=U(I)) THEN
    ! Accept the new values, so save LX2 and PQ2 and the accepted values
    PQ1 = PQ2
    LX1 = LX2
    SCQ(1:M) = CQ(1:M)
  END IF

  ! Display the simulated values
  WRITE(NOUT,*) I, SCQ(1:M)
END DO SIM_LOOP
```

Here `LOGPOSTERIOR` returns minus the log of the posterior density function (up to the normalising constant) and is the same function used when we estimated the mode of the posterior distribution in section 3.2. The call to `MVN_PDF` returns the probability density function for the multivariate Normal distribution evaluated at `CQ`.

For a large number of simulations it may be necessary to split the simulation into smaller blocks of values, i.e. generate `NBLK` blocks of `NGEN` values, where `NBLK = NSIM / NGEN`, rather than all `NSIM` in one go. It should be noted that it is more efficient to make multiple draws from the proposal and uniform distributions in one go, as done above, rather than putting `G05RZF` and `G05SAF` into the `SIM_LOOP` loop and generating a single value from each at each iteration.

Drawing a sample from the predictive distribution for a new set of data proceeds in a manner similar to that described in section 3.3, but with the values stored in `SCQ` above being substituted for the draws from the Normal distribution.

When drawing inferences about a posterior or predictive distribution using any MCMC technique, care must be taken that the underlying markov chain has converged, see Gilks et al. [3] for a discussion of this.

4.2 Importance Sampling

Importance sampling (see Ripley [7]) involves estimating the expected value of a function of the parameters, given the data, $E(f(\theta|y)$ by the following:

$$E(f(\theta)|y) \approx \frac{\sum_t f(\theta^{(t)})w(\theta^{(t)})}{\sum_t w(\theta^{(t)})}$$

where the $\theta^{(t)}$ are draws from some arbitary distribution, with density function $h(\theta)$, that closely approximates the posterior distribution and

$$w(\theta) = \frac{L(y|\theta)p(\theta)}{h(\theta)}$$

Implementing the importance sampler using the NAG library is very similar to implementing the independent Metropolis-Hastings sampler, discussed briefly in section 4.1. The random number generators must be initialised in the same way and it is usual to scale the covariance matrix in a similar manner.

For this particular implementation of the importance sampler, we are using a multivariate distribution, centred on the posterior mode, as the sampling distribution, $h(\theta)$. In such cases Geweke [2] suggests scaling the covariance matrix (H^{-1}) by a factor of $1.44(1.2^2)$. Unlike with the M-H independence sampler, we do not have an acceptance probability and so there is no need to call `G05SAF`. The main generating loop for this importance sampler is

```
SIM_LOOP:DO I = 1, NSIM
  ! Copy a single variate from the array of multivariate normal variates
  CQ(1:M) = Q(I,1:M)

  ! Get probability of variate from the sampling distribution
  PQ = MVN_PDF(M,CQ,BETA,SH,M,SLDETS,D)

  ! Get posterior, P(Q | X, Y), upto the normalising constant
  LX = EXP(LOGPOSTERIOR(N,M,X,N,CQ,XY,XB))

  ! Calculate the importance weighting
  WT = LX / PQ

  ! Display the simulated values and the associated weights
  WRITE(NOUT,*) I, WT, CQ(1:M)
END DO SIM_LOOP
```

Drawing a sample from the predictive distribution for a new set of data proceeds in a manner similar to that described in section 3.3, but with the values stored in `CQ` above being substituted for the draws from the Normal distribution. These draws from the predictive distribution are weighted using the `WT` calculated above.

4.3 Summarising a Series of Draws from a Distribution

One method for making inferences about a distribution is via Monte Carlo integration. Monte Carlo integration involves estimating the expected value of a function, f, of the distribution by using random draws $X^{(t)}$ from that distribution, such that

$$E(f) \approx \frac{1}{T} \sum_{t=1}^{T} f(X^{(t)})$$

The upshot of this is that you can make inferences about the distribution of interest by making inferences about a sample from that distribution.

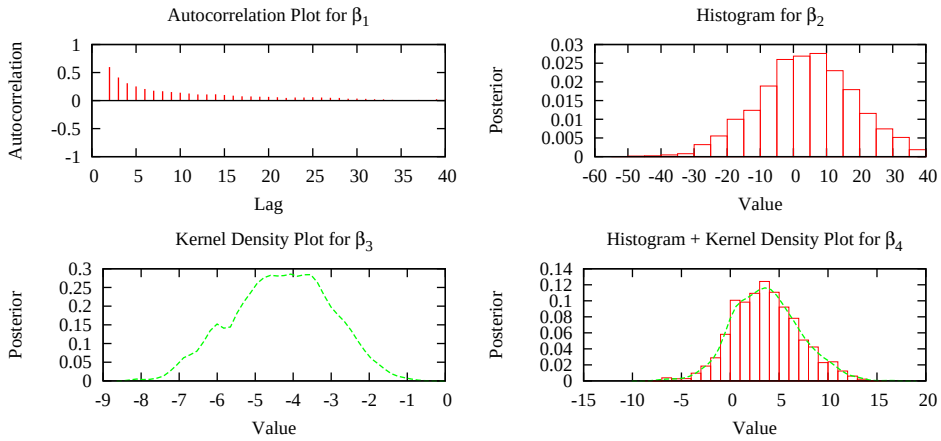

Figure 1: Series of Example Plots
The data for the various plots was taken from a series of NAG routines. The plots themselves were generated using version 4.2 of gnuplot [4]

Many of the methods discussed in the previous sections result in samples. The Normal approximation gives rise to samples from the predictive distribution. The independence sampler and importance sampling algorithms give rise to samples from both the posterior and predictive distributions. In the case of the importance sampling algorithms, these samples are weighted.

There are a number of routines in the NAG library that are useful for summarising such samples.

4.3.1 Summarising an Unweighted Sample

Assume that `Q(I,J)` holds the ith sample for the jth parameter. Estimates of a number of summary statistics, including the mean, variance etc, can be obtained using G01AAF.

```
DO J = 1, M
  IWT = 0
  CALL G01AAF(NSIM,Q(1,J),IWT,WT,XMEAN(J),S2,S3,S4,XMIN(J),XMAX(J), &
              SCAT(J),IFAIL)
END DO
```

We have an unweighted sample, so `IWT = 0` and the vector `WT` need not be assigned.

Median values and credible intervals can be calculated using G01AMF as follows:

```
! Returning the 2.5%, 50.0% and 97.5% quantiles
RQUANT(1:3) = (/ 0.025E0_WP, 0.5E0_WP, 0.975E0_WP/)

DO J = 1, M
  ! Estimate of quantiles can be obtained from G01AMF, without sorting
  CALL G01AMF(NSIM,Q(1,J),3,RQUANT,QUANT(1,J),IFAIL)
END DO
```

where the lower and upper limits for a 95% credible interval will be held in the first and third rows of `QUANT`. It should be noted that `G01AMF` overwrites the input data with a partially sorted version. Therefore `G01AMF` should be the last summary routine run, unless the data is copied first.

A contingency table of the sample, and hence the values required to construct a histogram, can be calculated via a call to G01AEF

```
ICLASS = 0
CALL G01AEF(NSIM,NCAT(J) + 1,Q(1,J),ICLASS,CB(1,J),IFREQ(1,J), &
            TXMIN,TXMAX,IFAIL)
```

If `ICLASS` is not one, then each column of the array `CB` must contain a set of `NCAT(J)` suitable break points for the histogram, i.e. the data for the jth parameter is grouped into values that fall between `CB(1,J)` and `CB(2,J)`, `CB(2,J)` and `CB(3,J)` etc. If `ICLASS` is set to one, then `G01AEF` calculates its own break points. However, these break points might not be best suited to producing a histogram. Once constructed, the resulting frequencies can be scaled by `NSIM*STEP(J)`, where `STEP(J) = CB(2,J) - CB(1,J)` (assuming equally spaced break points), so that the area under the chart sums to one.

Rather than a histogram, a kernel density plot of the posterior distribution can be constructed. The values required for this are obtained from G10BAF as follows:

```
USEFFT = .FALSE.
DO J = 1, M

  ! Get the WINDOW for the density estimation
  WINDOW = STEP(J)*KDSMOOTH

  ! Values of SLO and SHI are taken from recommendation in
  ! the documentation for G10BAF
  SLO = XMIN(J) - THREE*WINDOW
  SHI = XMAX(J) + THREE*WINDOW

  ! Do the kernel density estimation
  IFAIL = INFAIL
  CALL G10BAF(NSIM,Q(1,J),WINDOW,SLO,SHI,NPOINTS,SMOOTH(1,J),  &
```

```
       T(1,J),USEFFT,FFT,IFAIL)
END DO
```

Other summaries are possible, for example, when using an MCMC sampler, it is often useful to be able to plot the autocorrelation between successive samples. The required information can be obtained by a call to G13ABF:

```
DO J = 1, M
 IFAIL = INFAIL
 CALL G13ABF(Q(1,J),NSIM,NAUTO,XM,XV,R(1,J),STAT,IFAIL)
END DO
```

where NAUTO is the maximum lag required and R holds the correlation on exit.

4.3.2 Summarising a Weighted Sample

There are less routines that are directly usable on a weighted sample.

As with the unweighted sample, various summary statistics can be obtained using G01AAF. In the weighted example, the variable IWT needs to be set to 1, and the vector WT will hold the (importance sampling) weights.

Credible intervals can be obtained by a combination of the sorting routines, M01DAF and M01ZAF and, if required, the interpolation routines E01BEF and E01BFF. Rather than using the E01 routines, simple linear interpolation could be used instead.

For each margin, J, rank the sample in ascending order:

```
ORDER = 'A'
CALL M01DAF(Q(1,J),1,NSIM,ORDER,IRANK,IFAIL)
```

the convert the ranks into indices:

```
 CALL M01ZAF(IRANK,1,NSIM,IFAIL)
```

i.e. now IRANK(I) will hold the row of the original column of Q that would sit in the ith position in a sorted list. Calculate an (unnormalised) empirical CDF for the jth margin:

```
! Get an empirical CDF (not normalised)
CDF(1) = WT(IRANK(1))
XVAR(1) = Q(IRANK(1),J)
K = 1
DO I = 2, NSIM
```

```
  S2 = CDF(K) + WT(IRANK(I))
  IF (S2 > CDF(K)) THEN
    ! Ensure that the values of CDF are strictly increasing
    ! if two or more values are the same, then this takes the smallest
    ! XVAR value
    K = K + 1
    CDF(K) = S2
    XVAR(K) = Q(IRANK(I),J)
  END IF
END DO
```

Preprocess the empirical CDF to allow for interpolation:

```
CALL E01BEF(K,CDF,XVAR,D,IFAIL)
```

Interpolate the required quantiles:

```
! Returning the 2.5%, 50.0% and 97.5% quantiles
RQUANT(1:3) = (/ 0.025E0_WP, 0.5E0_WP, 0.975E0_WP/)

! Interpolate the required values
IFAIL = QUIET
CALL E01BFF(K,CDF,XVAR,D,3,RQUANT,QUANT(1,J),IFAIL)
! Check the IFAIL by hand, as E01BFF warns if the interpolant
! is outside the data range and this isn't cause to stop
! the program
IF (IFAIL/=0) THEN
  IF (IFAIL/=4 .AND. INFAIL/=QUIET) THEN
    WRITE (*,*) 'Error in E04XAF, IFAIL = ', IFAIL
    IF (INFAIL==HARD) STOP
   END IF
 END IF
```

5 Results for Example

This section gives a brief summary of the various results obtained from fitting the model described in section 2, using the methods described in this document.

Figure 2 gives density plots for the marginal posterior distributions for each of the five parameters from the logistic regression described in section 2. These plots were obtained using the Laplace approximation.

As can be seen from figure 3, similar results were obtained using the importance sampling and Metropolis-Hasting independence sampler. The Normal approximation gives a good

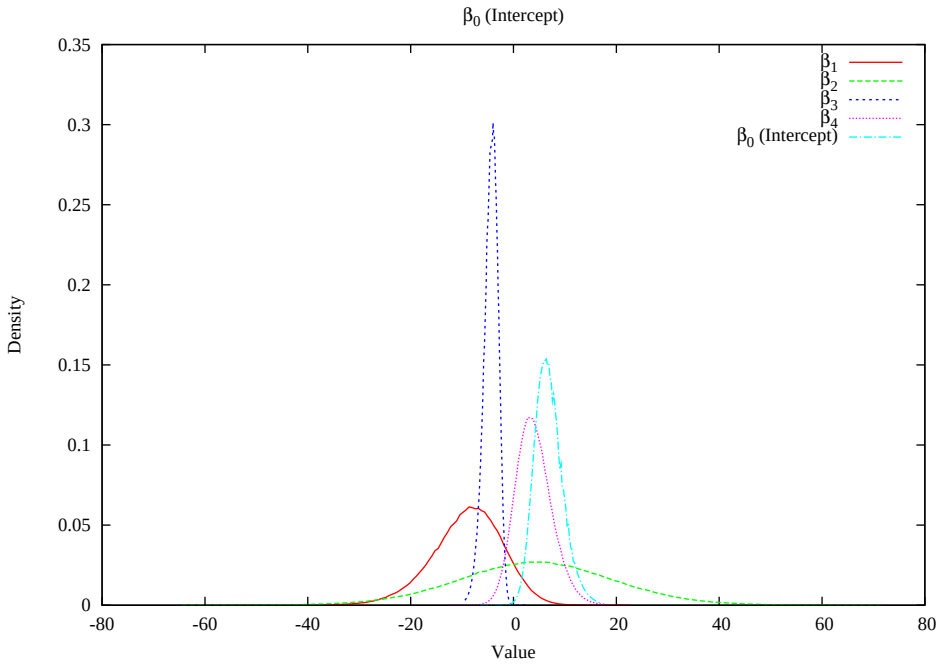

Figure 2: Marginal Posterior Distributions (Laplace Approximation)

approximation for β_2, but is out for the other parameters. It should be noted that in all cases the Laplace approximation is as good as, or better than the Normal approximation.

This pattern is also seen when examining the 95% credible intervals obtained from each method. The importance sampling and Metropolis-Hasting independence sampler give similar results and the Normal approximation differs. The values obtained from the importance sampling and the Normal approximation are given in tables 1 and 2 respectively.

95% credible intervals and the associated density plots for the predictive distribution for the three hypothetical companies are given in table 3 and figure 4 respectively.

As with figure 1, the data for all the figures in this section were taken from the various pieces of Fortran code. The plots themselves were produced using gnuplot version 4.2 [4].

Margin	2.5% Quantile	Median	Mean	97.5% Quantile
β_0	2.1093	6.6086	6.7720	12.882
β_1	-22.443	-8.6305	-8.9313	2.8983
β_2	-25.527	4.0536	3.7635	32.560
β_3	-7.2141	-4.2463	-4.3418	-1.9365
β_4	-2.3767	3.6311	3.8798	11.253

Table 1: Posterior Distribution Credible Intervals (Importance Sampling)
Mean, median and 95% credible intervals for marginal distributions of the model parameters. Calculated using importance sampling, with a sample of 30,000.

Margin	2.5% Quantile	Median	Mean	97.5% Quantile
β_0	5.1721	5.3197	5.3197	5.46739
β_1	-7.4795	-7.1050	-7.1050	-6.73044
β_2	2.7649	3.6190	3.6190	4.47308
β_3	-3.4907	-3.4152	-3.4152	-3.33965
β_4	2.7761	2.9703	2.9703	3.16457

Table 2: Posterior Distribution Credible Intervals (Normal Approximation)
Mean, median and 95% credible intervals for marginal distributions of the model parameters. Calculated using a Normal approximation.

Margin	2.5% Quantile	Median	Mean	97.5% Quantile
Company 1	0.2660	0.6503	0.6307	0.9274
Company 2	0.8457	0.9995	0.9847	1.000
Company 3	$0.5426e^{-10}$	$0.5723e^{-3}$	0.1719	0.9991

Table 3: Prediction Distribution Credible Intervals
Mean, median and 95% credible intervals for the prediction distributions. Calculated using importance sampling, and a sample size of 30,000.

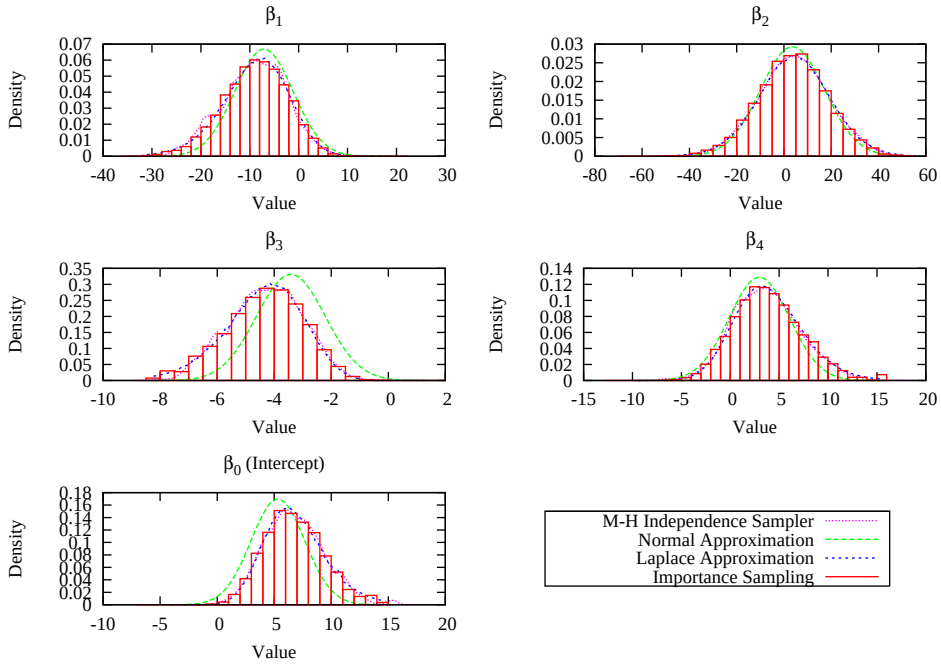

Figure 3: Marginal Posterior Distributions (Comparisons of Four Different Methods)

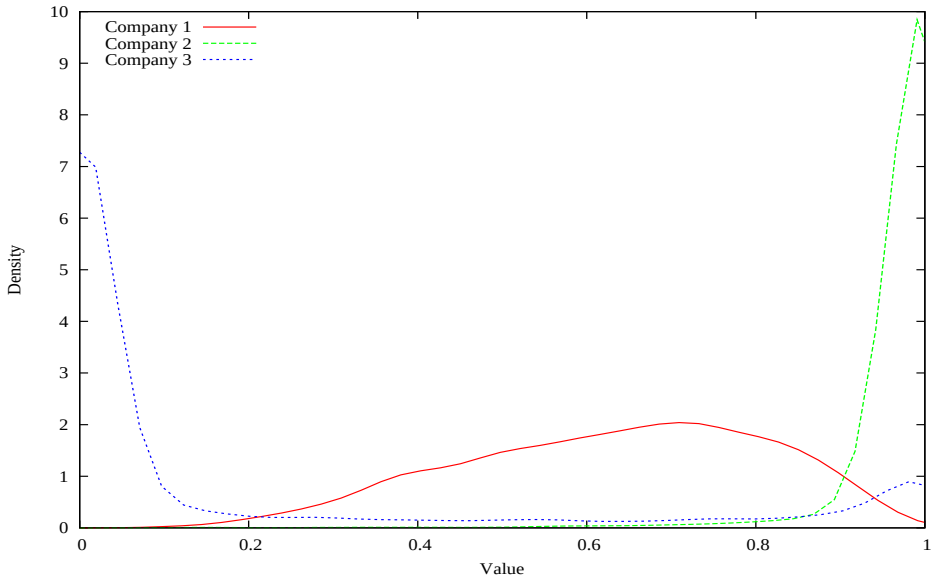

Figure 4: Prediction Distributions (Independence Sampler)
Prediction Distributions for Three Hypothetical Customers. This plots shows the predictive distribution for the probability of defaulting for each of three hypothetical customers. The distributions were estimated using the Metropolis-Hastings Independence Sampler

Summary of NAG Routines Referenced

The Fortran code supplied with this document has been written with Mark 22 of the Fortran library in mind. Six routines referenced in the code and this document are new at Mark 22, these are; `G05KFF`, `G05KGF`, `G05SAF`, `G05RZF`, `G01AMF` and `M01NAF`. Of these, four have equivalent routines in Mark 21:

`G05KGF`

Mark 22 code:

```
G05KGF(GENID,SUBID,STATE,LSTATE,IFAIL)
```

Mark 21 code:

```
G05KCF(IGEN,ISEED)
```

`G05KFF`

Mark 22 code:

```
G05KFF(GENID,SUBID,ISEED,LSEED,STATE,LSTATE,IFAIL)
```

Mark 21 code:

```
G05KBF(IGEN,ISEED)
```

`G05RZF`

Mark 22 code:

```
G05RZF(MODE,N,M,XMU,C,LDC,R,LR,STATE,X,LDX,IFAIL)
```

Mark 21 code:

```
G05LYF(MODE,M,XMU,C,LDC,N,X,LDX,IGEN,ISEED,R,LR,IFAIL)
```

`G05SAF`

Mark 22 code:

```
G05SAF(N,STATE,X,IFAIL)
```

Mark 21 code:

```
G05LGF(ZERO,ONE,N,X,IGEN,ISEED,IFAIL)
```

In the example code supplied we are using the Mersenne Twister as the base generator as denoted by the use of `GENID=3` when calling the initialisation routine `G05KFF`. This base generator is not in Mark 21 of the library. If using the Mark 21 routines it is recommended that one of the Wichmann-Hill generators be used instead, i.e `IGEN = 1` will use the first of the 273 Wichmann-Hill generators.

The Mark 22 routine G01AMF, which calculates quantiles of an unsorted vector, has no equivalent in the Mark 21 library. Rather than using G01AMF when summarising an un-weighted sample you can use the same code as for a weighted sample replacing the weight WT, with a value of 1.0.

The Mark 22 routine M01NAF, which performs a binary search of a vector, has no equivalent in the Mark 21 library. The call to M01NAF:

```
ID = M01NAF(.FALSE.,CB(1,J),1,NCAT(J),Q(I,J),IFAIL) + 1
```

can be replaced by a inear search along the lines of:

```
ID = NCAT(J) + 1
DO K = 1, NCAT(J)
  IF (Q(I,J) .LT. CB(K,J)) THEN
    ID = K
    EXIT
  END IF
END DO
```

Referenced NAG Routines:

G01AAF	Mean, variance, skewness, kurtosis, etc, one variable, raw data
G01AEF	Frequency table from raw data
G01AMF	Specified quantiles from a vector of unsorted data (Mark 22 only)
G01FAF	Computes quantiles (deviates) for the standard Normal distribution
G05KBF	Initialize random number generating routines to give a repeatable sequence
G05KCF	Initialize random number generating routines to give a non-repeatable sequence
G05KFF	Initialize random number generating routines to give a repeatable sequence (Mark 22 only)
G05KGF	Initialize random number generating routines to give a non-repeatable sequence (Mark 22 only)
G05LGF	Generate a vector of random numbers from a uniform distribution
G05LYF	Generate a matrix of random numbers from a multivariate Normal distribution
G05SAF	Generate a vector of random numbers from a standard uniform distribution (Mark 22 only)
G05RZF	Generate a matrix of random numbers from a multivariate Normal distribution (Mark 22 only)
G10BAF	Kernel density estimate using Gaussian kernel
G13ABF	Univariate time series, sample autocorrelation function
E01BFF	Interpolate values, interpolant calculated by E01BEF
E01BEF	Interpolating functions, monotonicity-preserving, piecewise cubic Hermite

E04JYF	Minimum, function of several variables, quasi-Newton algorithm, simple bounds, using function values only (easy-to-use)
E04UCA	Minimum, function of several variables, sequential QP method, nonlinear constraints, using function values and optionally first derivatives (forward communication, comprehensive)
E04UEA	Suppy optional parameters to E04UCA or E04UFA
E04UFA	Minimum, function of several variables, sequential QP method, nonlinear constraints, using function values and optionally first derivatives (reverse communication, comprehensive)
E04WDF	Solves the nonlinear programming (NP) problem
E04XAF	Estimate (using numerical differentation) gradient and/or Hessian of a function
F06EAF	Dot product two real vectors
F06EDF	Multiply real vector by scalar
F06EFF	Copy real vector
F06PAF	Matrix-vector product, real rectangular matrices
F06YAF	Matrix-matrix product, two real rectangular matrices
F07FJF	Inverse of a real symmetric positive-definite matrix, matrix already factorised
M01DAF	Rank a vector
M01NAF	Binary search (Mark 22 only)
M01ZAF	Invert a permutation

Data

The data were taken from page 87 of Bayesian Methods in Finance [6] and consists of a binary response y, taking values default (1) or not default (0), for 46 companies, and four independent variables. The independent variables are the ratios, cash flow to total debt (X_{i1}), net income to total assets (X_{i2}), current assets to current liabilities (X_{i3}) and current assets to net sales (X_{i4}), taken over a period of one year, two years prior to 21 of the companies defaulting.

X_{i1}	X_{i2}	X_{i3}	X_{i4}	y_i	X_{i1}	X_{i2}	X_{i3}	X_{i4}	y_i
-0.45	-0.41	1.09	0.45	1	0.51	0.10	2.49	0.54	0
-0.56	-0.31	1.51	0.16	1	0.08	0.02	2.01	0.53	0
0.06	0.02	1.01	0.40	1	0.38	0.11	3.27	0.35	0
-0.07	-0.09	1.45	0.26	1	0.19	0.05	2.25	0.33	0
-0.10	-0.09	1.56	0.67	1	0.32	0.07	4.24	0.63	0
-0.14	-0.07	0.71	0.28	1	0.31	0.05	4.45	0.69	0
0.04	0.01	1.50	0.71	1	0.12	0.05	2.52	0.69	0
-0.06	-0.06	1.37	0.40	1	0.02	0.02	2.05	0.35	0
0.07	-0.01	1.37	0.34	1	0.22	0.08	2.35	0.40	0
-0.14	-0.14	1.42	0.43	1	0.17	0.07	1.80	0.52	0
-0.23	-0.30	0.33	0.18	1	0.15	0.05	2.17	0.55	0
0.07	0.02	1.31	0.25	1	0.10	-0.01	2.50	0.58	0
0.01	0.00	2.15	0.70	1	0.14	-0.03	0.46	0.26	0
-0.28	-0.23	1.19	0.66	1	0.14	0.07	2.61	0.52	0
0.15	0.05	1.88	0.27	1	0.15	0.06	2.23	0.56	0
0.37	0.11	1.99	0.38	1	0.16	0.05	2.31	0.20	0
-0.08	-0.08	1.51	0.42	1	0.29	0.06	1.84	0.38	0
0.05	0.03	1.68	0.95	1	0.54	0.11	2.33	0.48	0
0.01	0.00	1.26	0.60	1	0.33	-0.09	3.01	0.47	0
0.12	0.11	1.14	0.17	1	0.48	0.09	1.24	0.18	0
-0.28	-0.27	1.27	0.51	1	0.56	0.11	4.29	0.44	0
					0.20	0.08	1.99	0.30	0
					0.47	0.14	2.92	0.45	0
					0.17	0.04	2.45	0.14	0
					0.58	0.04	5.06	0.13	0

Data for three hypothetical companies used to illustrate prediction.

X_{i1}	X_{i2}	X_{i3}	X_{i4}
0.05	0.05	1.8	0.5
0.1	0.1	0.5	1.0
2.0	1.0	0.1	0.0

References

[1] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society*, pages 330–418, 1763.

[2] J Geweke. Bayesian comparison of econometric models. *Working Paper 532. Federal Reserve Bank of Minneapolis*, 1994.

[3] W R Gilks, S Richardson, and D J Spiegelhalter. *Markov Chain Monte Carlo in Practice.* Chapman and Hall, 1996.

[4] gnuplot. http://www.gnuplot.info/.

[5] W Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[6] Svetlozar T Rachev, John S J Hsu, Biliana S Bagasheva, and Frank J Fabozzi. *Bayesian Methods in Finance.* John Wiley and Sons, 2008.

[7] Brian D. Ripley. *Stochastic Simulation.* John Wiley and Sons, 1987.

[8] Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81:82–86, 1986.