Statistical Programming - Individual Project 2

Question 1

MCMC methods scale up directly to higher dimensions. Suppose we have a distribution that generates two dimensional vectors $y = (y_1, y_2)$, such as the bivariate Gaussian distribution. One possible MCMC algorithm is to make joint proposals for y_1 and y_2, i.e. given that we have $y^{i-1} = (y_1^{i-1}, y_2^{i-1})$, then propose $y^* = (y_1^*, y_2^*)$ where:

$$y_1^* = y_1^{i-1} + \epsilon_1 \quad \epsilon \sim N(0, \sigma_1^2)$$
$$y_2^* = y_2^{i-1} + \epsilon_1 \quad \epsilon \sim N(0, \sigma_2^2)$$

Write a MCMC sampler to sample from the bivariate Gaussian distribution with mean vector $\mu = (1, 1)$ and covariance matrix:

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

using proposal variances $\sigma_1^2 = \sigma_2^2 = 1$. Draw 10,000 samples from this distribution and plot the density of your samples as either a contour plot, heatmap, or a 2 dimensional density (the base R function persp can be used for this, and the ggplot2 libary includes functions for heat maps and contour plots. You could look up the stat_density2d() and scale_fill_gradient() functions). Include your plot in your submission file.

Question 2

In the lecture I illustrated MCMC with the example of simulating from a Student-t distribution using a Gaussian random walk proposal. This question will expore whether this is a sensible idea.

1. Use MCMC to draw 10,000 samples from the Student-t(20) (i.e. with $v = 20$) distribution using random walk Metropolis-Hastings and a $N(0, 1)$ proposal distribution. If these samples are genuine draws from the correct Student-t distribution, then the variance of the sampled values should be close to the theoretical variance of the Student-t(20) distribution (up to some small sampling error). Verify that this is indeed the case. Include your sample variance in your submission file.

2. Use the same procedue to draw samples from the Student-t(3) distribution. Again compare the variance of your samples to the theoretical variance of the Student-t(3) (i.e. with $v = 3$) distribution. Are they equal? If not, then give an explanation why. Include your sample variance in your submission file.

Question 3

This question compares frequentist and Bayesian approaches to parameter estimation and prediction. It will hopefully give you an insight into the difference between these two ways of viewing statistical inference.

We will focus on modelling time-between-event data, i.e. the time that elapses between the occurrence of events like hurricanes or earthquakes. For $t \in \{1, 2, \ldots, n+1\}$ let z_t denote the time at which the z^{th} event occurred at, so that $y_t = z_{t+1} - z_t$ are the time between events, also known as the **inter-event times**.

The most basic statistical model for this sort of data is to assume that the event times follow a Poisson process, in which case the inter-event times have a $y_t \sim Exponential(\lambda)$ distribution. However this model is often not good for real world data, since it only has one parameter (λ) and hence the mean and variance of the distribution cannot be modelled separately. An alternative distribution that is often used is the Weibull distribution, which has two parameters (k, λ) and density function:

$$p(y|k, \lambda) = \frac{k}{\lambda}\left(\frac{y}{\lambda}\right)^{k-1} e^{-\left(\frac{y}{\lambda}\right)^k}, \quad y > 0, k > 0, \lambda > 0$$

Frequentist Inference

Download the file eventtimes.csv from the course webpage and load it into R. This is a vector $y = (y_1, \ldots, y_{100})$ containing 100 inter-event times which correspond to the number of weeks which occurred between rainfall days in a particualr city (e.g. $y_1 = 2.146$ so there were 2.146 weeks between the time of the first rainfall, and the second). This data is to be modelled with a Weibull distribution. You are told that the value of λ is $\lambda = 2$, and hence you only need to estimate k.

1. Derive the log-likelihood function of the Weibull distribution with n observations as a function of k, and implement this in R.

2. Use optim() to find the maximum likelihood estimate of k for the eventtimes.csv data. Note that we can use optim() to maximise rather than minimise a function by including the argument control=list(fnscale= -1) as an argument to this function. Your function call should hence look something like this:

```
optim(initval, fn, lambda=lambda, y=y, control=list(fnscale = -1))
```

3. Suppose that it rained today. Compute the probability of there being more than 1.5 weeks until the next time that rain occurs, assuming that k is equal to its maximum likelihood estimate (i.e. if $\tilde{y}$ is the number of weeks until the next rainfall, then compute $p(\tilde{y} > 1.5|k = \hat{k}, \lambda = 2)$ where $\hat{k}$ is the MLE.

Bayesian Inference

In the previous section, we predicted the future by assuming that k was equal to its maximum likelihood estimate. However our prediction will be misleadingly confident since it ignore the uncertainty in the estimate of k – the true value of k will not be exactly equal to the MLE! Bootstrapping can help this to some degree, since it incorporates uncertainty about k. Another approach is to use a Bayesian approach where we work with the whole posterior distribution for k. Again, we suppose that $\lambda = 2$ is known.

4. We will use a $Gamma(1, 1)$ prior for k. Write an R function to implement the resulting log posterior distribution (i.e. the log of the Weibull likelihood multiplied by a Gamma(1,1) distribution).

When we want to make prediction aobut future event times, the Bayesian approach is to average over the posterior. In other words given observed data y (which here consists of the 100 inter-event times), then if $\tilde{y}$ is a future inter-event time, then we would predict its value using:

$$p(\tilde{y}|y) = \int p(\tilde{y}|k)p(k|y)dk$$

Since we do not have a conjugate prior, it will not be possible to solve this analytically. The usual approach is to instead sample values $k^{(1)}, k^{(2)}, \ldots, k^{(M)}$ from the posterior $p(k|y)$ and use these to approximate the integral:

$$p(\tilde{y}|y) = \int p(\tilde{y}|k)p(k|y)dk \approx \frac{1}{M}\sum_{i=1}^{M} p(\tilde{y}|k^{(i)}), \quad k^{(i)} \sim p(k|y)$$

So for example for some value z we can write:

$$p(\tilde{y} > z|y) \approx \frac{1}{M}\sum_{i=1}^{M} p(\tilde{y} > z|k^{(i)}), \quad k^{(i)} \sim p(k|y)$$

Note how this compares to the frequentist approach above: the frequentist predicts by using $p(\tilde{y}|\hat{k})$ where $\hat{k}$ is the maximum likelihood estimate, whereas the Bayesian predicts by averaging $p(\tilde{y}|k^{(i)})$ over values sampled from the posterior.

5. Use MCMC to to draw $10,000$ samples from the posterior distribution $p(k|y)$ where y is the above data. Based on your samples, report the posterior mean, median, and standard deviation (note: these should be close to the MLE!).

6. Using your samples from above, estimate the Bayesian probability that it will be more than 1.5 weeks until the next time that it rains. (i.e. $p(\tilde{y} > 1.5|y)$). Quantify the level of uncertainty in this prediction by also reporting its standard deviation (i.e. the standard deviation of $p(\tilde{y} > 1.5|\theta^i)$ across the θ^i samples.

Laplace Approxmation

An alternative to MCMC which is sometimes used in Bayesian statistics is the Laplace Approximation. This is often not very good in terms of performance, but it is quick and simple to implement.

The Laplace approximation is based on the fact that as the number of observations increases, the posterior distribution will asymptotically converge to a Normal distribution (Why? Because we know the likelihood function will asymptotically be Normal, and as the number of data poitns increase, the prior becomes less and less relevant so the posterior converges to the likelihood). As such, we can sometimes approximate the posterior distribution by a Normal distribution.

Specifically, we approximate the posterior by a $N(\hat{k}, \sigma_k^2)$ distribution, where $\hat{k}$ is the value of k which maximises the posterior, and σ_k^2 is 1 divided by the negative of the second derivative of the log-posterior evaluated at $\hat{k}$, i.e.

$$\sigma_k^2 = -\frac{1}{\frac{\partial^2}{\partial k^2} \log p(k = \hat{k}|y)}$$

7. Sample 10,000 values from the Laplace approximation to the Weibull distribution above, with the Gamma prior (i.e. compute the mean and standard deviation of the above Normal distribution). Compute the mean, median and standard deviation of the posterior. Hint: when you are using optim() to find the posterior maximum, you can add the argument hessian=TRUE and R will return the second derivative of the log-posterior evaluated at its maximum value, which is the quantity you need in the formula for σ_k^2 above – you do not need to calculate this manually!

8. On the same plot, plot a kernel density estimate (or histogram) of the 10,000 samples you got above using MCMC, and the 10,000 samples you got using the Laplace approximation (use a differrent colour for each). Include this plot in your submission file.