QUEEN MARY, UNIVERSITY OF LONDON

MTH6102: Bayesian Statistical Methods

Practical 5

2023-2024

1 Posterior inference for exponential likelihood/gamma prior example

We use again Exponential likelihood/gamma prior example to find the posterior median and a 95% credible interval. We use the dataset available in R called `faithful`, containing data on eruptions of the Old Faithful geyser. The waiting time between successive eruptions can be accessed as

```
faithful$waiting
```

You could store this column of the dataset in a new vector to simplify the later code

```
t = faithful$waiting
```

We assume that these values follow an exponential distribution, with parameter λ.

We saw that a gamma distribution is conjugate to the exponential likelihood. The help files for the R commands for the gamma distribution may be found using

```
?dgamma
```

There are two common ways of specifying the parameters for the gamma distribution. The form used in this module is

$$f(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, \ x > 0$$

The mean of the gamma distribution is α/β. In R, the pdf for this distribution is given by the following

```
dgamma(x, shape=alpha, rate=beta)
```

1. Calculate the parameters for the posterior gamma distribution when combining a gamma prior distribution with the likelihood based on the Old Faithful data. Do this for $\alpha = 1, \beta = 1$ and then $\alpha = 200, \beta = 200$.

2. Calculate the posterior mean for λ using each prior distribution.

3. Calculate the posterior median using each prior distribution, using the `qgamma` command. For a Gamma(a, b) distribution, the median is given by:

```
qgamma(0.5, shape=a, rate=b)
```

This is because `qgamma` is the quantile function, i.e. the inverse of the cdf.

4. Also use the `qgamma` command to calculate an equal tail 95% credible interval for λ with each choice of prior distribution.

```
qgamma(c(0.025,0.975),shape=alpha_posterior,rate=beta_posterior)
```

5. Now generate a random sample of size 10,000 from the posterior distribution using either $\alpha = 1, \beta = 1$ or $\alpha = 200, \beta = 200$ for prior parameters. Store the sample in a vector called `post_lambda`.

```
post_lambda = rgamma(NS, shape=alpha, rate=beta)
```

In the command above use the posterior parameters!

6. Use the random sample to calculate a 95% credible interval. One way of doing this (for an equal tail interval) is to sort the sample in ascending order and then take the values 2.5% and 97.5% of the of the way from first to last. This can be done using:

```
post_lambda_sorted = sort(post_lambda)
post_lambda_sorted[NS*0.025]
post_lambda_sorted[NS*0.975]
```

Check that the credible interval found from the random sample approximately agrees with the exact values.

There is also a function called `quantile` which could be used instead to calculate a credible interval from a posterior sample. The syntax would be:

```
quantile(post_lambda, probs=c(0.025, 0.975))
```

2 Posterior inference for other distributions

Each of the standard probability distributions has a similar set of functions. A help page with a list of them can be found as follows:

```
?distributions
```

In each case (if we can write down the exact posterior distribution), it is the quantle function that is needed to find the median and equal tail credible interval.

Consider the `iris` dataset in R that we saw question 3 of exercise sheet 3 . Let $y_1, \ldots, y_n$ be the column `iris$Sepal.Length`, and assume that they are independent. Suppose that each $y_i \sim N(\mu, \sigma^2)$, where we assume σ is known and equal to 0.9. For μ, we assign a normal prior distribution $N(\mu_0, \sigma_0^2)$, where $\mu_0 = 5$, $\sigma_0 = 2$.

1. Find the posterior distribution of μ, after seeing the `iris` dataset.
2. Plot the posterior and the prior density functions
3. Find the posterior median, and a 95% credible interval for μ, given the same prior distribution assumed before. This would use the functions associated with the normal distribution, such as `qnorm` for the quantile function.

Simple graph options

We have used named colours such as red and blue in some examples, but there are many more available. The R command `colors()` lists them, while there are documents elsewhere on the web that show them visually, which is more helpful. One is here.

For example, to draw a histogram and choose the colour of the bars:

```
hist(iris$Petal.Length, col="papayawhip")
```

Some other useful options are the following, which are illustrated in examples below. They can found among the many options in `?par`.

- `lwd` - line width, default is 1;
- `lty` - line type, e.g. `"dashed"`, `"dotted"`;
- `pch` - plotting character;
- `main` - graph title;
- `xlab, ylab` - axis titles;
- `xlim, ylim` - limits of axis ranges;
- `legend` - add a legend.

```
x = 1:8
y = 3 + 2*x + 0.4*rnorm(length(x))
z = y + 1
plot(x=x, y=y, xaxs="r", yaxs="r", pch=15, col="royalblue")
points(x=x, y=z, pch=17, col="seagreen")
```

The following two lines show the symbols available for plotting points:

```
xp=0:25
plot(x=xp, y=rep(1, length(xp)), pch=xp)
```

A legend can be added, but it does not automatically pick up elements such as colour and line type from the graph, so these need to be specified both in the plot and the legend:

```
plot(dnorm, xlim=c(-3, 3), ylim=c(0, 0.5), lwd=2, col="slateblue",
    main="Normal pdf", xlab="y", ylab="pdf")
x = seq(from=-3, to=3, by=0.1)
y = dnorm(x, mean=0, sd=0.8)
lines(x=x, y=y,  lty="dashed", lwd=2, col="red3")
legend("topright", legend=c("sd = 1", "sd = 0.8"),
    col=c("slateblue", "red3"), lty=c("solid", "dashed"), lwd=c(2, 2))
```

Saving graphs

If you are using Microsoft Word, then just copying graphs from R into your document may be all you need. But for other purposes, it can be useful to save graphs to disk. For use in a Latex document, saving the graph as a pdf is a good option. The second line can be replaced with multiple lines for drawing the graph.

```
pdf("norm.pdf", width=5, height=4)
plot(dnorm, xlim=c(-3, 3), col="seagreen3")
dev.off()
```

This will save the file to our current working directory, which we can check using `getwd()` and change using `setwd(...)`.

`?Devices` lists ways of saving in different graph formats.