Introduction to Bayesian Statistics

Andrew Parnell

Learning outcomes

- Know the difference between Frequentist and Bayesian statistics
- Understand the terms posterior, likelihood and prior. Be able to suggest suitable probability distributions for these terms
- Be able to interpret the posterior distribution through plots, summaries, and credible intervals

A bigger aim, either:

1. Stop using SIAR (for dietary proportions) and start writing your own JAGS code
2. Stop using SIAR and start using MixSIAR/simmr instead

Who was Bayes?

An essay towards solving a problem on the doctrine of chances (1763)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

What is Bayesian statistics?

- Bayesian statistics is based on an interpretation of Bayes' theorem
- All quantities are divided up into *data* (i.e. things which have been observed) and *parameters* (i.e. things which haven't been observed)
- We use Bayes' interpretation of the theorem to get the *posterior probability distribution*, the probability of the unobserved given the observed
- Used now in almost all areas of statistical application (finance, medicine, environmetrics, gambling, etc, etc)

Why is this relevant to SIMMs?

- Easy to specify Bayesian models hierarchically in layers so that the data depend on some parameters, which then depend on further parameters, and so on. This allows us to create richer statistical models which will better match reality
- Almost all the modern Stable Isotope Mixing Models (SIMMs) use Bayesian statistics
- MixSIR, SIAR, MixSIAR, simmr, IsotopeR, . . .

What is Bayes' theorem?

Bayes' theorem can be written in words as:

posterior is proportional to likelihood times prior

. . . or . . .

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

Each of the three terms *posterior*, *likelihood*, and *prior* are *probability distributions* (pdfs).

In a Bayesian model, every item of interest is either data (which we will write as x) or parameters (which we will write as θ). Often the parameters are divided up into those of interest, and other *nuisance parameters*

Bayes' theorem in more detail

Bayes' equation is usually written mathematically as:

$$p(\theta|x) \propto p(x|\theta) \times p(\theta)$$

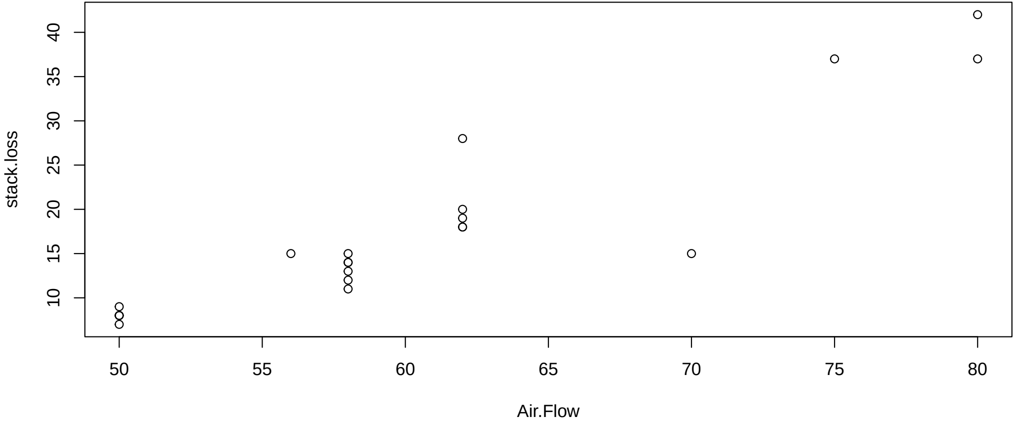
or, more fully:

$$p(\theta|x) = \frac{p(x|\theta) \times p(\theta)}{p(x)}$$

- The *posterior* is the probability of the parameters given the data
- The *likelihood* is the probability of observing the data given the parameters (unknowns)
- The *prior* represents external knowledge about the parameters

A very simple linear regression example

Suppose you had some data that looked like this:



What you are used to doing

```r
model = lm(stack.loss ~ Air.Flow, data = stackloss)
summary(model)
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow, data = stackloss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2896  -1.1272  -0.0459   1.1166   8.8728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -44.13202    6.10586  -7.228 7.31e-07 ***
## Air.Flow      1.02031    0.09995  10.208 3.77e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.098 on 19 degrees of freedom
## Multiple R-squared:  0.8458, Adjusted R-squared:  0.8377
## F-statistic: 104.2 on 1 and 19 DF,  p-value: 3.774e-09
```

What you will now get instead

```
print(model_run)
```

```
## Inference for Bugs model at "4", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved. Running time = 0.04 secs
##             mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## intercept   -43.758   6.535 -56.552 -47.919 -43.881 -39.603 -30.941 1.001  3000
## residual_sd   4.382   0.735   3.220   3.855   4.290   4.792   6.046 1.001  3000
## slope         1.014   0.107   0.806   0.944   1.016   1.083   1.228 1.001  3000
## deviance    120.089   2.673 116.987 118.071 119.387 121.434 127.183 1.001  3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule: pV = var(deviance)/2)
## pV = 3.6 and DIC = 123.7
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Using prior information

- The Bayesian model in the previous slide divided up everything into *parameters* (the intercept, slope and residual standard deviation), and data (the x and y values)
- The software in the background created a posterior probability distribution of the parameters given the data
- The model I fitted used vague *prior information*. However, if we had done a previous experiment that suggested the intercept should be around -30 with standard deviation 5 we can put this in the model

A model with prior information

```r
print(model_run2)
```

```
## Inference for Bugs model at "5", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved. Running time = 0.038 secs
##             mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## intercept   -35.099   4.085 -42.902 -37.872 -35.268 -32.310 -26.839 1.001  3000
## residual_sd   4.541   0.824   3.284   3.948   4.453   4.988   6.395 1.002  1500
## slope         0.874   0.068   0.742   0.827   0.877   0.922   1.005 1.001  3000
## deviance    121.594   2.916 117.433 119.414 121.072 123.287 128.687 1.002  1400
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule: pV = var(deviance)/2)
## pV = 4.2 and DIC = 125.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

An early example of a Bayesian model

- To create the Bayesian version of this model I used the following JAGS code:

```
model_code ='
model {
  # Likelihood
  for(i in 1:N) {
    y[i] ~ dnorm(intercept + slope*x[i], residual_sd^-2)
  }
  # Priors
  intercept ~ dnorm(0,100^-2)
  slope ~ dnorm(0,100^-2)
  residual_sd ~ dunif(0,100)
}
'
```

How do I specify the prior distribution?

There are several choices when it comes to specifying prior distributions:

- *Informative*, when there is information from a previous study, or other good external source, e.g intercept $\sim N(-30, 5^2)$
- *Vague*, when there is only weak information, perhaps as to the likely range of the parameter e.g. intercept $\sim N(0, 100^2)$
- *Flat*, when there is no information at all about a parameter (very rare). In JAGS, write `intercept ~ dflat()`

In fact, choosing the prior and choosing the likelihood are very similar problems

Choosing likelihoods and priors

When creating Bayesian models it's helpful to know a lot of probability distributions. The ones we will use most are:

Distribution	Range of parameter	Useful for
Normal, $N(\mu, \sigma^2)$	$(-\infty, \infty)$	A good default choice
Uniform, $U(a, b)$	(a, b)	Vague priors when we only know the range of the parameter
Binomial, $Bin(k, \theta)$	$[0, k]$	Count or binary data restricted to have an upper value
Poisson, $Po(\lambda)$	$[0, \infty)$	Count data with no upper limit
Gamma, $Ga(\alpha, \beta)$	$(0, \infty)$	Continuous data with a lower bound of zero
Multivariate Normal, $MVN(\mu, \Sigma)$	$(-\infty, \infty)$	Multivariate unbounded data with correlation between parameters/observations

Creating the posterior distribution

- It only takes a few lines of R code (and a few more lines of JAGS code) to calculate the posterior distribution
- However this processes will be slower and harder when we have lots of parameters, and complicated prior distributions
- Almost always in the Bayesian world we have to resort to *simulation* rather than maths to get to the posterior distribution
- This means that we obtain *samples* from the posterior distribution rather than creating the probability distribution directly
- JAGS uses Markov chain Monte Carlo (MCMC) to create these samples. We will talk about this a bit more in later lectures/discussion

Summarising the posterior distribution

- Because we obtain samples from the posterior distribution, we can create any quantity we like from them
- e.g. we can obtain the mean or standard deviation simply from combining the samples together
- We can create quantiles e.g. 50% for the median
- We can create a Bayesian *credible interval* (CI) by calculating lower and upper quantiles
- When the posterior distribution is messy (e.g. multi-modal) we can use a *highest posterior density* (HPD) region

Example:

From the earlier simple example. First 5 posterior samples of the slope

```
post_slope = model_run$BUGSoutput$sims.list$slope
post_slope[1:5]
```

```
## [1] 1.0261186 1.0368179 0.9686494 1.0568446 1.0638896
```

The mean and standard deviation:

```
c(mean(post_slope),sd(post_slope))
```

```
## [1] 1.017123 0.109424
```

A 95% credible interval

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

```
##      2.5%     97.5%
## 0.7992388 1.2294259
```

Why is this better?

The Bayesian approach has numerous advantages:

- It's easier to build complex models and to analyse the parameters you want directly
- We automatically obtain the best parameter estimates and their uncertainty from the posterior samples
- It allows us to get away from (terrible) null hypothesis testing and p-values

Some further reading

- The Bayesian bible: Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis*, Third Edition. CRC Press.
- The MCMC bible: Brooks, S., Gelman, A., Jones, G., & Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press.
- Something simpler: McCarthy, M. A. (2007). *Bayesian Methods for Ecology*. Cambridge University Press.

Summary

- Bayesian statistical models involve a likelihood and a prior. These both need to be carefully chosen. From these we create a posterior distribution
- The likelihood represents the information about the data generating process, the prior represents information about the unknown parameters
- We usually create and analyse samples from the posterior probability distribution of the unknowns (the parameters) given the knowns (the data)
- From the posterior distribution we can create means, medians, standard deviations, credible intervals, etc