9.5 New distributions

9.5.1 Specifying a new sampling distribution

Suppose we wish to use a sampling distribution that is not included in the list of standard distributions, in which an observation y_i contributes a likelihood term L_i. One possibility is the "zeros trick" based on the following.

We invent a set of observations $z_i = 0$, each of which is assumed to be drawn from a Poisson(ϕ_i) distribution. Each then has a likelihood contribution $\exp(-\phi_i)$, and so if ϕ_i is set to $-\log(L_i)$, we will obtain the correct likelihood contribution. (Note that ϕ_i should always be > 0 as it is a Poisson mean, and so we may need to add a suitable constant to ensure that it is positive.) The BUGS code will look like the following:

```
const    <- 10000   # arbitrary, ensures phi[i] > 0
for (i in 1:n) {
  z[i]   <- 0
  z[i]    ~ dpois(phi[i])
  phi[i] <- -log(L[i]) + const
  L[i]   <- ...
}
```

L_i is set to a function of y_i and θ proportional to the likelihood $p(y_i|\theta)$. This trick allows arbitrary sampling distributions to be used and is particularly suitable when, say, dealing with truncated distributions (§9.6.2).

A new observation from the distribution, denoted y_{n+1}, can be predicted by including it as an additional, but missing, observation in the data file and assigning it an improper uniform prior, e.g., `y[n+1]` $\sim$ `dflat()`, defining z_i and ϕ_i in the same way as before for $i = n + 1$. The missing observation is essentially assumed to be an unknown parameter with a uniform prior, but also with a likelihood term corresponding to the sampling distribution.

Note that the DIC (§8.6) for data from distributions specified using the zeros trick, as reported by the WinBUGS or OpenBUGS DIC tool, is calculated with respect to z_i, not y_i. Example 11.6.2 explains how to transform this to the scale of y_i, so it can be compared with the DICs of models for y_i which are specified using built-in sampling distributions.

Example 9.5.1. *A clumsy way of modelling the normal distribution*

We use the "zeros trick" to model a normal distribution with unknown mean μ and unknown standard deviation σ, including predicting a new observation. We

have seven observed values and one missing value as follows: `y = c(-1, -0.3, 0.1, 0.2, 0.7, 1.2, 1.7, NA)`.

```
for (i in 1:8) {
  z[i]   <- 0
  z[i]    ~ dpois(phi[i])
  phi[i] <- log(sigma) + 0.5*pow((y[i] - mu)/sigma, 2)
}
y[8]      ~ dflat()
sigma     ~ dunif(0, 100)
mu        ~ dunif(-100, 100)
```

We must provide an initial value for `y[8]`, via `y = c(NA, NA, NA, NA, NA, NA, NA, 0)`, say, otherwise BUGS will try to generate one from the improper prior and crash.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	0.365	0.4758	0.006864	-0.5948	0.3693	1.316	4001	10000
sigma	1.18	0.481	0.01139	0.6216	1.067	2.415	4001	10000
y[8]	0.3499	1.355	0.03415	-2.345	0.3564	3.095	4001	10000

Whilst the results match those that would be obtained in a standard analysis using `y[i] ~ dnorm(mu, tau); tau <- 1/pow(sigma,2)`, this is an inefficient procedure, particularly for the prediction, and so a long run is necessary. The MC error for the prediction is 0.03 using the zeros trick and 0.01 for the same number of iterations in the equivalent standard analysis.

An alternative to the "zeros trick" is the "ones trick." Here we invent a set of observations equal to 1 instead, and assume each to be Bernoulli distributed with probability p_i. By making each p_i proportional to L_i (i.e., by specifying a scaling constant large enough to ensure $p_i < 1$ for all i), the required likelihood term is provided:

```
const  <- 10000   # arbitrary, ensures p[i] < 1
for (i in 1:n) {
  z[i] <- 1
  z[i]  ~ dbern(p[i])
  p[i] <- L[i]/const
}
```

We will illustrate use of the "ones trick" in §9.6.2, where we consider how to specify truncated sampling distributions.

9.5.2 Specifying a new prior distribution

Suppose we want to use a prior distribution for θ that does not belong to the standard set. Then we can use the "zeros trick" (see above) at the prior level

combined with an improper uniform prior for θ. A single Poisson observation equal to zero, with mean $\phi = -\log(p(\theta))$, contributes a term $\exp(-\phi) = p(\theta)$ to the likelihood for θ; when this is combined with a "flat" prior for θ the correct prior distribution results. This is essentially the same process as predicting from a new distribution covered in the previous section. Summary BUGS code is

```
z     <- 0
z      ~ dpois(phi)
theta ~ dflat()
phi   <- expression for -log(desired prior for theta)
```

For example, if we wished to produce a standard normal prior, we would use

```
phi   <- 0.5*pow(theta, 2)
```

It is important to note that this method produces high auto-correlation, poor convergence, and large MC errors, so it is computationally slow and long runs are necessary. Initial values also need to be specified as the `dflat()` prior cannot be sampled from using `gen.inits`.

New sampling distributions and new prior distributions can also be specified in WinBUGS via the WinBUGS Development Interface (WBDev). This can give big computational savings and clearer BUGS code, at the cost of "lower-level" programming in Component Pascal — see § 12.4.8 for more details. There are similar but less well-documented capabilities in OpenBUGS and JAGS; see Chapter 12.

9.6 Censored, truncated, and grouped observations

9.6.1 Censored observations

A data point is a censored observation when we do not know its exact value, but we do know that it lies above or below a point c, say, or within a specified interval. The most common application is in survival analysis (§11.1), but here we consider general measurement problems. There are two strategies within BUGS:

1. In general, in WinBUGS we can use the `I(,)` construct (§A.2.2), which specifies a restricted range within which the unknown quantity lies. The unknown quantity is then simply treated as a model parameter. Note that in OpenBUGS the `C()` function is preferred (see §12.5.1) and JAGS uses a different syntax altogether (see §12.6.2).

2. Each exact observation y contributes $p(y|\theta)$ to the likelihood of θ, whereas an observation censored at c provides a contribution of $\Pr(Y >$