Basics of Bayesian inference

Calculations (numerical and algebraic) are usually required only up to a proportionally constant. We, therefore, write the posterior as:

$$p(\theta | y, \lambda) \propto p(\theta | \lambda) \times f(y | \theta).$$

If $\lambda$ are known/fixed, then the above represents the desired posterior. If, however, $\lambda$ are unknown, we assign a prior, $p(\lambda)$, and seek:

$$p(\theta, \lambda | y) \propto p(\lambda)p(\theta | \lambda)f(y | \theta).$$

The proportionality constant does not depend upon $\theta$ or $\lambda$:

$$\frac{1}{p(y)} = \int p(\lambda)p(\theta | \lambda)f(y | \theta)|\lambda|d\theta$$

The above represents a joint posterior from a hierarchical model. The marginal posterior distribution for $\theta$ is:

$$p(\theta | y) = \int p(\lambda)p(\theta | \lambda)f(y | \theta)|\lambda|d\lambda.$$
A simple example: Normal data and normal priors

Interpret: Posterior mean is a weighted mean of prior mean and data.

\[
\frac{\theta}{\sigma^2 + n\tau^2} \mu + \frac{n\tau^2}{\sigma^2 + n\tau^2} \bar{y}
\]

\[
N\left(\mu, \tau^2 \bar{y}, \sigma^2 + n\tau^2\right)
\]

Example: Consider a single data point \( y \) from a Normal distribution: \( y \sim N(\theta, \sigma^2) \); assume \( \sigma \) is known.

\[
f(y|\theta) = N(y|\theta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(y - \theta)^2\right)
\]

\[
\theta \sim N(\mu, \tau^2), \text{i.e. } p(\theta) = N(\theta|\mu, \tau^2); \mu, \tau^2 \text{ are known.}
\]

Posterior distribution of \( \theta \)

\[
p(\theta|y) \propto N(\theta|\mu, \tau^2) \times N(y|\theta, \sigma^2)
\]

\[
= N\left(\theta|\frac{1}{\tau^2 + \sigma^2} \mu + \frac{\sigma^2}{\tau^2 + \sigma^2} \bar{y}, \frac{1}{\tau^2 + \sigma^2}\right)
\]

\[
= N\left(\theta|\frac{\sigma^2}{\sigma^2 + n\tau^2} \mu + \frac{n\tau^2}{\sigma^2 + n\tau^2} \bar{y}, \frac{\sigma^2 + n\tau^2}{\sigma^2 + n\tau^2}\right)
\]

Another simple example: The Beta-Binomial model

{\text{drawing samples from it. This replaces numerical integration (quadrature) by "Monte Carlo integration".}}

One important advantage: we only need to know \( p(\theta|\bar{y}) \) up to the proportionality constant.

Suppose \( \Theta = (\theta_1, \theta_2) \) and we know how to sample from the marginal posterior distribution \( p(\theta_2|\bar{y}) \) and the conditional distribution \( P(\theta_1|\theta_2, \bar{y}) \).

How do we draw samples from the joint distribution: \( p(\theta_1, \theta_2|\bar{y}) \)?
Bayesian predictions

Suppose we want to predict new observations, say \( \tilde{y} \), based upon the observed data \( y \). We will specify a joint probability model \( p(\tilde{y}, y | \theta) \), which defines the conditional predictive distribution:

\[
p(\tilde{y} | y, \theta) = \frac{p(\tilde{y}, y | \theta)}{p(y | \theta)}
\]

Bayesian predictions follow from the posterior predictive distribution that averages out the \( \theta \) from the conditional predictive distribution with respect to the posterior:

\[
p(\tilde{y} | y) = \int p(\tilde{y} | y, \theta)p(\theta | y)d\theta.
\]

This can be evaluated using composition sampling:
- First obtain: \( \theta^{(j)} \sim p(\theta | y), \quad j = 1, \ldots, M \)
- For \( j = 1, \ldots, M \) sample \( \tilde{y}^{(j)} \sim p(\tilde{y} | y, \theta^{(j)}) \)
- The \( \{\tilde{y}^{(j)}\}_{j=1}^{M} \) are samples from the posterior predictive distribution \( p(\tilde{y} | y) \).

Some remarks on sampling-based inference

Direct Monte Carlo: Some algorithms (e.g., composition sampling) can generate independent samples exactly from the posterior distribution. In these situations there are NO convergence problems or issues. Sampling is called exact.

Markov Chain Monte Carlo (MCMC): In general, exact sampling may not be possible/feasible. MCMC is a far more versatile set of algorithms that can be invoked to fit more general models. Note: anywhere where direct Monte Carlo applies, MCMC will provide excellent results too.

Convergence issues: There is no free lunch! The power of MCMC comes at a cost. The initial samples do not necessarily come from the desired posterior distribution. Rather, they need to converge to the true posterior distribution. Therefore, one needs to assess convergence, discard output before the convergence and retain only post-convergence samples. The time of convergence is called burn-in.

Diagnosing convergence: Usually a few parallel chains are run from rather different starting points. The sample values are plotted (called trace-plots) for each of the chains. The time for the chains to "mix" together is taken as the time for convergence.

Good news! All this is automated in WinBUGS. So, as users, we need to only configure how to specify good Bayesian models and implement them in WinBUGS.