3 Inferences with binomials

3.1 Inferring a rate

Our first problem completes the introductory example in Chapter 2, and involves inferring the underlying success rate for a binary process. The graphical model is shown again in Figure 3.1. Recall that shaded nodes indicate known values, while unshaded nodes represent unknown values, and that circular nodes correspond to continuous values, while square nodes correspond to discrete values.

The goal of inference in the graphical model is to determine the posterior distribution of the rate θ, having observed k successes from n trials. The analysis starts with the prior assumption that all possible rates between 0 and 1 are equally likely. This corresponds to the uniform prior distribution $\theta \sim \text{Uniform}(0, 1)$, which can equivalently be written in terms of a beta distribution as $\theta \sim \text{Beta}(1, 1)$.

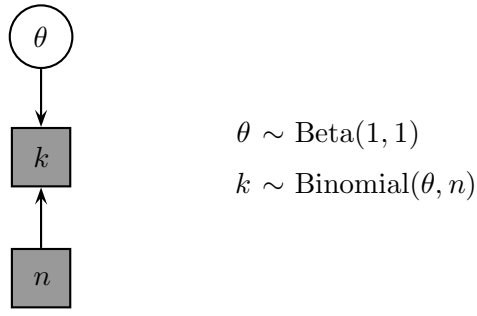


Fig. 3.1 Graphical model for inferring the rate θ of a binary process.

The script `Rate_1.txt` implements the graphical model in WinBUGS. The script is available at `www.bayesmodels.com` and is shown below:

```
# Inferring a Rate
model{
   # Prior Distribution for Rate Theta
   theta ~ dbeta(1,1)
   # Observed Counts
   k ~ dbin(theta,n)
}
```

The code `Rate_1.m` for Matlab or `Rate_1.R` for R, both available at `www.bayesmodels.com`, sets $k = 5$ and $n = 10$ and calls WinBUGS to sample from the graphical model. WinBUGS then returns to Matlab or R the posterior samples

Box 3.1 Beta distributions as conjugate priors

One of the nice properties of using the $\theta \sim \text{Beta}(\alpha, \beta)$ prior distribution for a rate θ is that it has a natural interpretation. The α and β values can be thought of as counts of, respectively, "prior successes" and "prior failures." This means that using a $\theta \sim \text{Beta}(3, 1)$ prior corresponds to having the prior information that 4 previous observations have been made, and 3 of them were successes. Or, more elaborately, starting with a $\theta \sim \text{Beta}(3, 1)$ is the same as starting with a $\theta \sim \text{Beta}(1, 1)$, and then seeing data giving two more successes (i.e., the posterior distribution in the second scenario will be the same as the prior distribution in the first). As always in Bayesian analysis, inference starts with prior information, and updates that information—by changing the probability distribution representing the uncertain information—as more information becomes available. When a type of likelihood function (in this case, the binomial) does not change the type of distribution (in this case, the beta) going from the prior to the posterior, they are said to have a "conjugate" relationship. This property is valued a lot in analytic approaches to Bayesian inference, because it makes for tractable calculations. It is not so important in the computational approaches emphasized in this book, because sampling methods can handle much more general relationships between parameter distributions and likelihood functions. But conjugacy is still useful in computational approaches because of the natural semantics it gives in setting prior distributions.

from θ. The Matlab or R code also plots the posterior distribution of the rate θ. A histogram of the samples looks something like the jagged line in Figure 3.2.

Exercises

Exercise 3.1.1 Carefully consider the posterior distribution for θ given $k = 5$ successes out of $n = 10$ trials. Based on a visual impression, what is your estimate of the probability that the rate θ is higher than 0.4 but smaller than 0.6? How did you arrive at your estimate?

Exercise 3.1.2 Consider again the posterior distribution for θ given $k = 5$ successes out of $n = 10$ trials. Based on a visual impression, what is your estimate of how much more likely it is that the rate θ is equal to 0.5 rather than 0.7? How did you arrive at your estimate?

Exercise 3.1.3 Alter the data to $k = 50$ and $n = 100$, and compare the posterior for the rate θ to the original with $k = 5$ and $n = 10$.

Exercise 3.1.4 For both the $k = 50$, $n = 100$ and $k = 5$, $n = 10$ cases just considered, re-run the analyses with many more samples (e.g., 10 times as many) by changing the `nsamples` variable in Matlab, or the `n.iter` variable

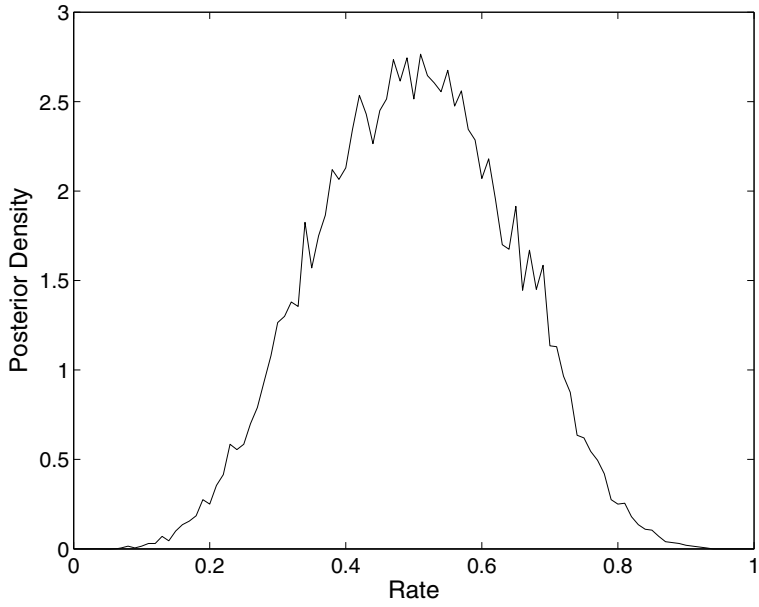


Fig. 3.2 Posterior distribution of rate θ for $k = 5$ successes out of $n = 10$ trials.

in R. This will take some time, but there is an important point to understand. What controls the width of the posterior distribution (i.e., the expression of uncertainty in the rate parameter θ)? What controls the quality of the approximation of the posterior (i.e., the smoothness of the histograms in the figures)?

Exercise 3.1.5 Alter the data to $k = 99$ and $n = 100$, and comment on the shape of the posterior for the rate θ.

Exercise 3.1.6 Alter the data to $k = 0$ and $n = 1$, and comment on what this demonstrates about the Bayesian approach.

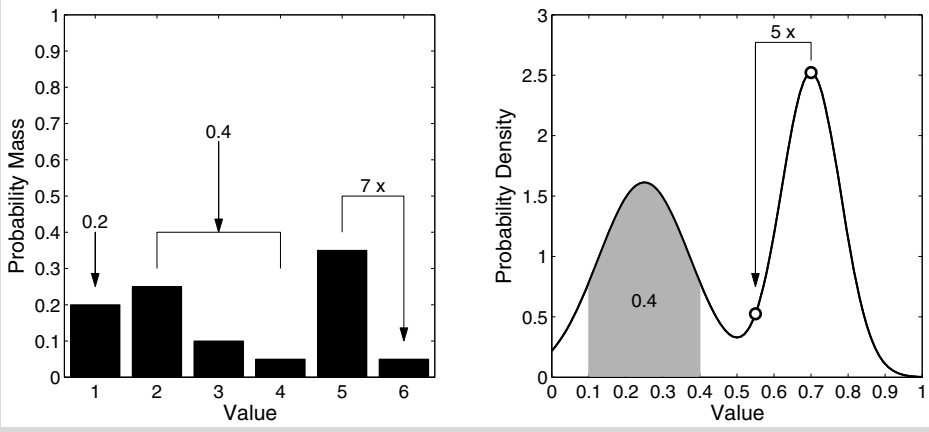
3.2 Difference between two rates

Now suppose that we have two different processes, producing k_1 and k_2 successes out of n_1 and n_2 trials, respectively. First, we will make the assumption that the underlying rates are different, so they correspond to different latent variables θ_1 and θ_2. Our interest is in the values of these rates, as estimated from the data, and in the difference $\delta = \theta_1 - \theta_2$ between the rates.

The graphical model representation for this problem is shown in Figure 3.3. The new notation is that the deterministic variable δ is shown by a double-bordered node. A deterministic variable is one that is defined in terms of other variables, and inherits its distribution from them. Computationally, deterministic nodes are

Box 3.2 Interpreting distributions

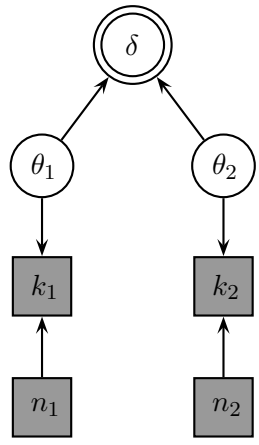
Since the essence of Bayesian inference is using probability distributions to represent uncertainty, it is important to be able to interpret probability mass functions and probability density functions. Probability mass functions are for discrete variables, which take a finite number of values, while probability density functions are for continuous variables, which take infinitely many values.



The panel on the left shows a probability mass function for a variable with 6 values. Each bar represents the probability of that value, so that, for example, the probability of the value 1 is 0.2. The probability of a range of values is the sum of their probabilities, so that the probability that the value is between 2 and 4 inclusive is 0.4. The ratio between the probabilities determines how much more likely one value is than another, so that the value 5 is 7 times more likely than the value 6. And, the sum of all of the probabilities (i.e., the height of the bars stacked on each other) is always 1. The panel on the right shows a probability density function for a variable that is between 0 and 1. The total area under the curve is always 1, which means the densities of individual points can (and often do) exceed 1. They cannot be interpreted as probabilities. But the probability of a range of values can be determined by the relevant area under the curve. In the right panel, the probability that the value is between 0.1 and 0.4 is 0.4. And ratios can still be interpreted in a relative way, so it is 5 times more likely the value is 0.7 than 0.55.

unnecessary—all inference could be done with the variables that define them—but they are often conceptually very useful to include, to communicate the meaning of a model.

The script `Rate_2.txt` implements the graphical model in WinBUGS:



$$
\begin{aligned}
k_1 &\sim \text{Binomial}(\theta_1, n_1) \\
k_2 &\sim \text{Binomial}(\theta_2, n_2) \\
\theta_1 &\sim \text{Beta}(1, 1) \\
\theta_2 &\sim \text{Beta}(1, 1) \\
\delta &\leftarrow \theta_1 - \theta_2
\end{aligned}
$$

Fig. 3.3 Graphical model for inferring the difference, $\delta = \theta_1 - \theta_2$, in the rates of two binary processes.

```
# Difference Between Two Rates
model{
   # Observed Counts
   k1 ~ dbin(theta1,n1)
   k2 ~ dbin(theta2,n2)
   # Prior on Rates
   theta1 ~ dbeta(1,1)
   theta2 ~ dbeta(1,1)
   # Difference Between Rates
   delta <- theta1-theta2
}
```

The code `Rate_2.m` or `Rate_2.R` sets $k_1 = 5$, $k_2 = 7$, $n_1 = n_2 = 10$, and then calls WinBUGS to sample from the graphical model. WinBUGS returns to Matlab or R the posterior samples from θ_1, θ_2, and δ. If the main research question is how different the rates are, then δ is the most relevant variable, and its posterior distribution is shown in Figure 3.4.

There are many ways the full information in the posterior distribution of δ might usefully be summarized. The Matlab or R code produces a set of these from the posterior samples, including:

- The mean value, which approximates the expectation of the posterior. This summary tries to pick a single value close to the truth, with bigger deviations from the truth being punished more heavily. Statistically, it corresponds to the point estimate under quadratic loss.
- The value with maximum density in the posterior samples, approximating the posterior mode. This summary aims to pick the single most likely value. This is known as the maximum a posteriori (MAP) estimate, and is the same as the maximum likelihood estimate (MLE) for "flat" priors. Statistically, it corresponds to the point estimate under zero–one loss. Estimating the mode requires

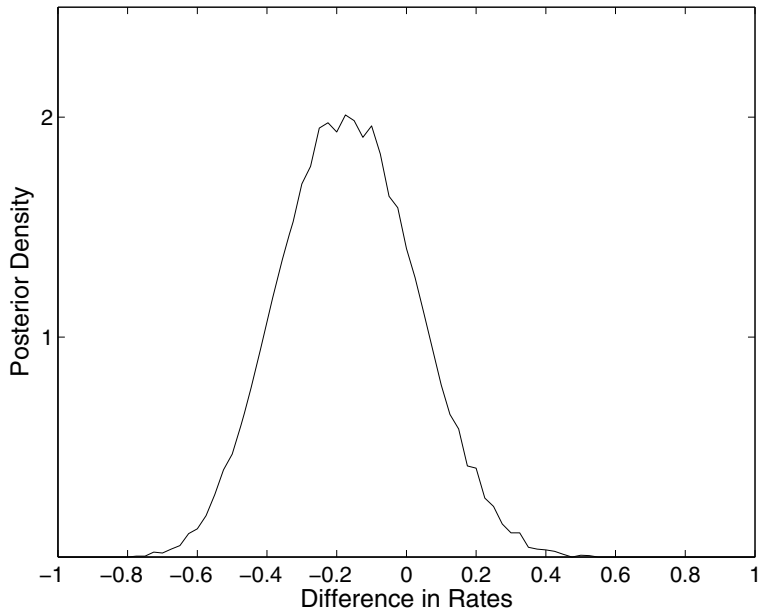


Fig. 3.4 Posterior distribution of the difference between two rates $\delta = \theta_1 - \theta_2$.

evaluating the likelihood function at each posterior sample, and so requires a bit more post-processing work in Matlab or R.

- The median value, which is the value that separates the highest 50% of the posterior distribution from the lowest 50%, and so finds the middle-most value. Statistically, it corresponds to the point estimate under linear loss.
- The 95% credible interval. This gives the upper and lower values between which 95% of samples fall. Thus, it approximates the bounds on the posterior distribution that contain 95% of the posterior density. The Matlab or R code can be modified to produce credible intervals for criteria other than 95%.

For the current problem, the mean of δ estimated from the returned samples is approximately -0.17, the mode is approximately -0.17, the median is approximately -0.17, and the 95% credible interval is approximately $[-0.52, 0.21]$.

Exercises

Exercise 3.2.1 Compare the data sets $k_1 = 8$, $n_1 = 10$, $k_2 = 7$, $n_2 = 10$ and $k_1 = 80$, $n_1 = 100$, $k_2 = 70$, $n_2 = 100$. Before you run the code, try to predict the effect that adding more trials has on the posterior distribution for δ.

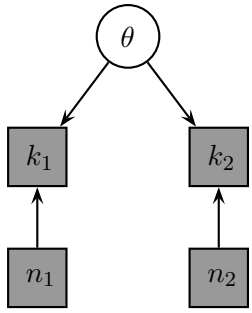
Exercise 3.2.2 Try the data $k_1 = 0$, $n_1 = 1$ and $k_2 = 0$, $n_2 = 5$. Can you explain the shape of the posterior for δ?

Exercise 3.2.3 In what context might different possible summaries of the posterior distribution of δ (i.e., point estimates, or credible intervals) be reasonable, and when might it be important to show the full posterior distribution?

3.3 Inferring a common rate

We continue to consider two binary processes, producing k_1 and k_2 successes out of n_1 and n_2 trials, respectively, but now assume the underlying rate for both is the same. This means there is just one rate, θ.

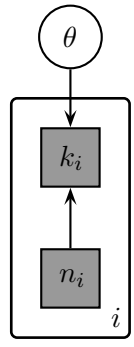
The graphical model representation for this problem is shown in Figure 3.5.



$$k_1 \sim \text{Binomial}(\theta, n_1)$$
$$k_2 \sim \text{Binomial}(\theta, n_2)$$
$$\theta \sim \text{Beta}(1, 1)$$

Fig. 3.5 Graphical model for inferring the common rate θ of two binary processes.

An equivalent graphical model, using plate notation, is shown in Figure 3.6. Plates are bounding rectangles that enclose independent replications of a graphical structure within a whole model. In this case, the plate encloses the two observed counts and numbers of trials. Because there is only one latent rate θ (i.e., the same probability drives both binary processes) it is not iterated inside the plate. One way to think of plates, which some people find helpful, is as "for loops" from programming languages (including WinBUGS itself).



$$k_i \sim \text{Binomial}(\theta, n_i)$$
$$\theta \sim \text{Beta}(1, 1)$$

Fig. 3.6 Graphical model for inferring the common rate θ underlying a number of binary processes, using plate notation.

The script `Rate_3.txt` implements the graphical model in WinBUGS:

```
# Inferring a Common Rate
model{
   # Observed Counts
```

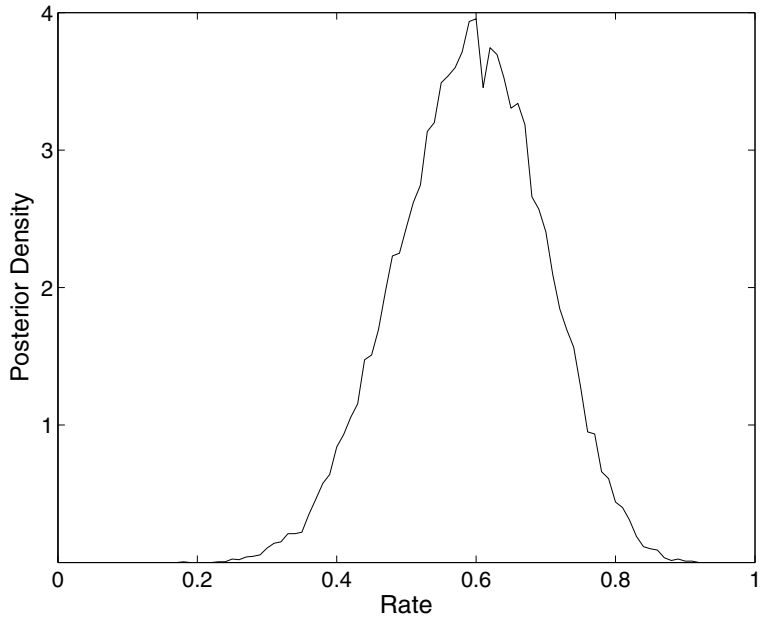


Fig. 3.7 Posterior distribution of the common rate θ of two binary processes.

```
   k1 ~ dbin(theta,n1)
   k2 ~ dbin(theta,n2)
   # Prior on Single Rate Theta
   theta ~ dbeta(1,1)
}
```

The code `Rate_3.m` or `Rate_3.R` sets k_1, k_2, n_1, and n_2, and then calls WinBUGS to sample from the graphical model.[1] The code also produces a plot of the posterior distribution for the common rate, as shown in Figure 3.7.

Exercises

Exercise 3.3.1 Try the data $k_1 = 14$, $n_1 = 20$, $k_2 = 16$, $n_2 = 20$. How could you report the inference about the common rate θ?

Exercise 3.3.2 Try the data $k_1 = 0$, $n_1 = 10$, $k_2 = 10$, $n_2 = 10$. What does this analysis infer the common rate θ to be? Do you believe the inference?

Exercise 3.3.3 Compare the data sets $k_1 = 7$, $n_1 = 10$, $k_2 = 3$, $n_2 = 10$ and $k_1 = 5$, $n_1 = 10$, $k_2 = 5$, $n_2 = 10$. Make sure, following on from the previous question, that you understand why the comparison works the way it does.

[1] Note that the R code specifies `debug=T`, and this means that WinBUGS needs to be closed (not minimized) before the sampling information can be returned to R. WinBUGS is ready as soon as the message "updates took x s" appears in the status bar.

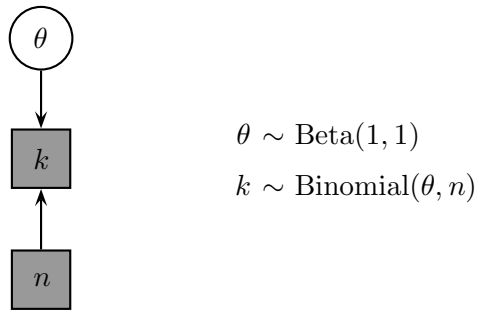


Fig. 3.8 Graphical model for inferring the rate θ of a binary process.

3.4 Prior and posterior prediction

One conceptual way to think about Bayesian analysis is that Bayes' rule provides a bridge between the unobserved parameters of models and the observed data. The most useful part of this bridge is that data allow us to update the uncertainty, represented by probability distributions, about parameters. But the bridge can handle two-way traffic, and so there is a richer set of possibilities for relating parameters to data. There are really four distributions available, and they are all important and useful.

- First, the *prior distribution* over parameters captures our initial assumptions or state of knowledge about the psychological variables they represent.
- Secondly, the *prior predictive distribution* tells us what data to expect, given our model and our current state of knowledge. The prior predictive is a distribution over data, and gives the relative probability of different observable outcomes before any data have been seen.
- Thirdly, the *posterior distribution* over parameters captures what we know about the psychological variables having updated the prior information with the evidence provided by data.
- Finally, the *posterior predictive distribution* tells us what data to expect, given the same model we started with, but with a current state of knowledge that has been updated by the observed data. Again, the posterior predictive is a distribution over data, and gives the relative probability of different observable outcomes after data have been seen.

As an example to illustrate these distributions, we return to the simple problem of inferring a single underlying rate. Figure 3.8 presents the graphical model, and is the same as Figure 3.1.

The script `Rate_4.txt` implements the graphical model in WinBUGS, and provides sampling not just for the posterior, but also for the prior, prior predictive, and posterior predictive:

```
# Prior and Posterior Prediction
model{
   # Observed Data
   k ~ dbin(theta,n)
   # Prior on Rate Theta
   theta ~ dbeta(1,1)
   # Posterior Predictive
   postpredk ~ dbin(theta,n)
   # Prior Predictive
   thetaprior ~ dbeta(1,1)
   priorpredk ~ dbin(thetaprior,n)
}
```

Posterior predictive sampling is achieved by the variable `postpredk` that samples predicted data using the same binomial as the actual observed data. To allow sampling from the prior, we use a dummy variable `thetaprior` that is identical to the one we actually do inference on, but is itself independent of the data, and so is never updated. Prior predictive sampling is achieved by the variable `priorpredk` that samples data using the same binomial, but relying on the prior rate.

The code `Rate_4.m` or `Rate_4.R` sets observed data with $k = 1$ successes out of $n = 15$ observations, and then calls WinBUGS to sample from the graphical model. The code also draws the four distributions, two in the parameter space (the prior and posterior for θ), and two in the data space (the prior predictive and posterior predictive for k). It should look something like Figure 3.9.

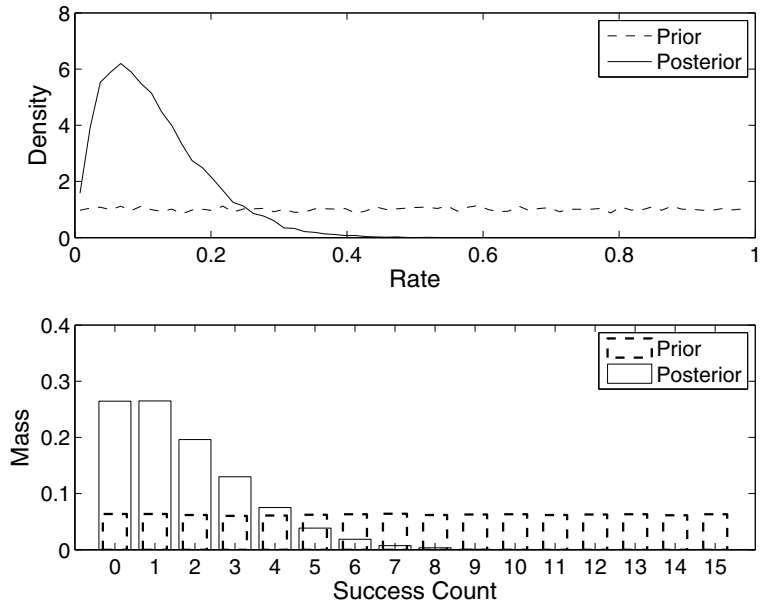


Fig. 3.9 Prior and posterior for the success rate θ (top panel), and prior and posterior predictive for counts of the number of successes (bottom panel), based on data giving $k = 1$ successes out of $n = 15$ trials.

Exercises

Exercise 3.4.1 Make sure you understand the prior, posterior, prior predictive, and posterior predictive distributions, and how they relate to each other (e.g., why is the top panel of Figure 3.9 a line plot, while the bottom panel is a bar graph?). Understanding these ideas is a key to understanding Bayesian analysis. Check your understanding by trying other data sets, varying both k and n.

Exercise 3.4.2 Try different priors on θ, by changing $\theta \sim \text{Beta}(1,1)$ to $\theta \sim \text{Beta}(10,10)$, $\theta \sim \text{Beta}(1,5)$, and $\theta \sim \text{Beta}(0.1,0.1)$. Use the figures produced to understand the assumptions these priors capture, and how they interact with the same data to produce posterior inferences and predictions.

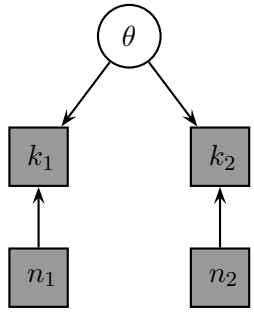
Exercise 3.4.3 Predictive distributions are not restricted to exactly the same experiment as the observed data, and can be used in the context of any experiment where the inferred model parameters make predictions. In the current simple binomial setting, for example, predictive distributions could be found by an experiment that is different because it has $n' \neq n$ observations. Change the graphical model, and Matlab or R code, to implement this more general case.

Exercise 3.4.4 In October 2009, the Dutch newspaper *Trouw* reported on research conducted by H. Trompetter, a student from the Radboud University in the city of Nijmegen. For her undergraduate thesis, Trompetter had interviewed 121 older adults living in nursing homes. Out of these 121 older adults, 24 (about 20%) indicated that they had at some point been bullied by their fellow residents. Trompetter rejected the suggestion that her study may have been too small to draw reliable conclusions: "If I had talked to more people, the result would have changed by one or two percent at the most." Is Trompetter correct? Use the code `Rate_4.m` or `Rate_4.R`, by changing the `dataset` variable (Matlab) or changing the values for `k` and `n` (R), to find the prior and posterior predictive for the relevant rate parameter and bullying counts. Based on these distributions, do you agree with Trompetter's claims?

3.5 Posterior prediction

One important use of posterior predictive distributions is to examine the descriptive adequacy of a model. It can be viewed as a set of predictions about what data the model expects to see, based on the posterior distribution over parameters. If these predictions do not match the data already seen, the model is descriptively inadequate.

As an example to illustrate this idea of checking model adequacy, we return to the problem of inferring a common rate underlying two binary processes. Figure 3.10 presents the graphical model, and is the same as Figure 3.5.



$$k_1 \sim \text{Binomial}(\theta, n_1)$$
$$k_2 \sim \text{Binomial}(\theta, n_2)$$
$$\theta \sim \text{Beta}(1, 1)$$

Fig. 3.10 Graphical model for inferring the common rate θ underlying two binary processes.

The script `Rate_5.txt` implements the graphical model in WinBUGS, and provides sampling for the posterior predictive distribution:

```
# Inferring a Common Rate, With Posterior Predictive
model{
   # Observed Counts
   k1 ~ dbin(theta,n1)
   k2 ~ dbin(theta,n2)
   # Prior on Single Rate Theta
   theta ~ dbeta(1,1)
   # Posterior Predictive
   postpredk1 ~ dbin(theta,n1)
   postpredk2 ~ dbin(theta,n2)
}
```

The code `Rate_5.m` or `Rate_5.R` sets observed data with $k_1 = 0$ successes out of $n_1 = 10$ observations, and $k_2 = 10$ successes out of $n_2 = 10$ observations, as considered in Exercise 3.3.2. The code draws the posterior distribution for the rate and the posterior predictive distribution, as shown in Figure 3.11.

The left panel shows the posterior distribution over the common rate θ for two binary processes, which gives density to values near 0.5. The right panel shows the posterior predictive distribution of the model, with respect to the two success counts. The size of each square is proportional to the predictive mass given to each

Box 3.3 The fundamental problem of inference

"The fundamental problem of inference and induction is to use past data to predict future data. Extensive observations on the motions of heavenly bodies enables their future positions to be calculated. Clinical studies on a drug allow a doctor to give a prognosis for a patient for whom the drug is prescribed. Sometimes the uncertain data are in the past, not the future. A historian will use what evidence he has to assess what might have happened where records are missing. A court of criminal law enquires about what had happened on the basis of later evidence." (Lindley, 2000, p. 304).

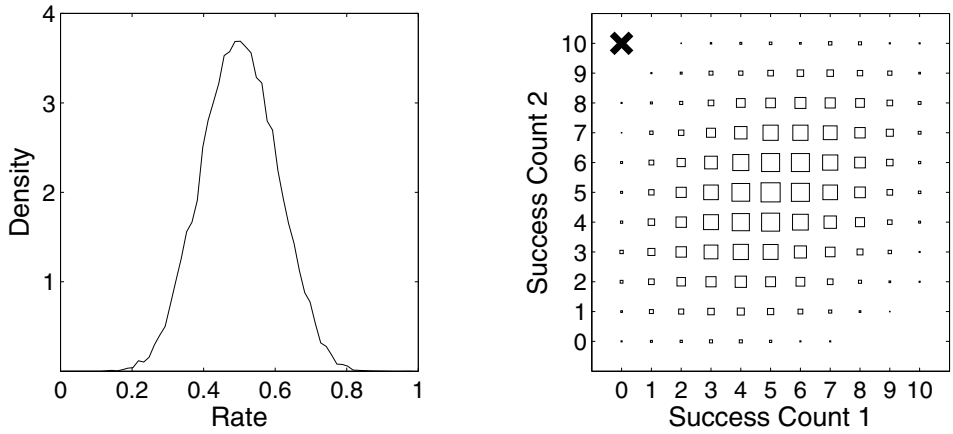


Fig. 3.11 The posterior distribution of the common rate θ for two binary processes (left panel), and the posterior predictive distribution (right panel), based on 0 and 10 successes out of 10 observations.

possible combination of success count observations. The actual data observed in this example, with 0 and 10 successes for the two counts, are shown by the cross.

Exercises

Exercise 3.5.1 Why is the posterior distribution in the left panel inherently one-dimensional, but the posterior predictive distribution in the right panel inherently two-dimensional?

Exercise 3.5.2 What do you conclude about the descriptive adequacy of the model, based on the relationship between the observed data and the posterior predictive distribution?

Exercise 3.5.3 What can you conclude about the parameter θ?

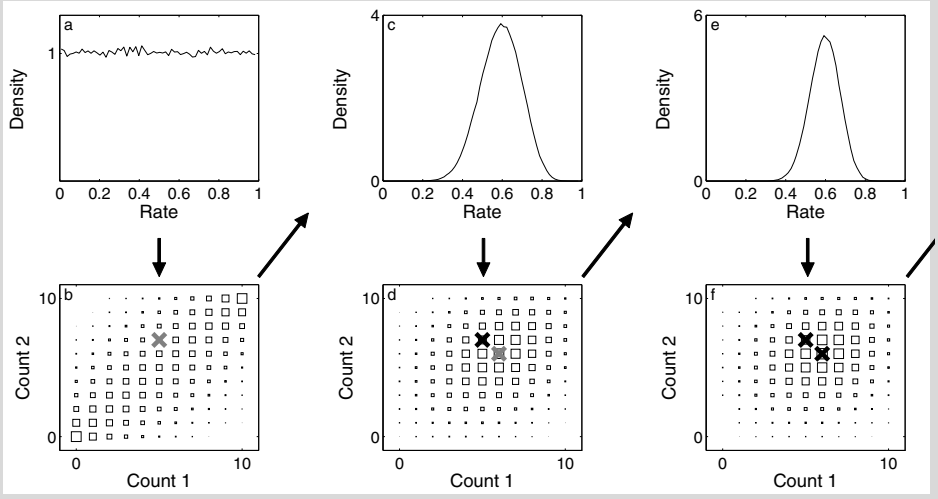
3.6 Joint distributions

So far, we have assumed that the number of successes k and number of total observations n is known, but that the underlying rate θ is unknown. This means that our parameter space has been one-dimensional. Everything learned from data is incorporated into a single probability distribution representing the relative probabilities of different values for the rate θ.

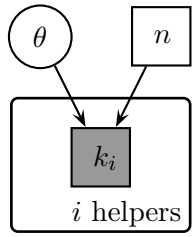
For many problems in cognitive science (and more generally), however, there will be more than one unknown variable of interest, and they will interact. A simple case of this general property is a binomial process in which both the rate θ *and* the total number n are unknown, and so the problem is to infer both simultaneously from counts of successes k.

Box 3.4 Today's posterior is tomorrow's prior

The idea that prior information about parameters can be transformed into posterior information, and hence prior predictive information about data can be transformed into posterior predictive information, can be continued indefinitely. As more information becomes available, usually as more data are collected, uncertainty about parameters and predictive distributions are naturally updated in the Bayesian approach.



The figure shows the incorporation of a sequence of data for the common model in Figure 3.10. Panel "a" shows the uniform prior over the common rate. Panel "b" shows the prior predictive, for the two counts of successes out of 10 trials. The gray cross corresponds to the observed data, which has yet to be incorporated, but can be compared to the prior predictive distribution. Panel "c" shows the posterior on the rate that now incorporates the data, and panel "d" shows the resulting posterior predictive. The first data are now shown by the black cross in this posterior predictive, since they are incorporated, but a new second data set, in the form of the different gray cross, is about to arrive. These new data are incorporated into the posterior distribution over the rate in panel "e," which leads to the posterior prediction in panel "f." And so the process can continue. Notice how the distribution over the rate parameter in panel "c" is the posterior distribution with respect to the first data set, but acts as the prior for the second data set. This leads to Lindley's Bayesian motto "Today's posterior is tomorrow's prior."



$$k_i \sim \text{Binomial}(\theta, n)$$
$$\theta \sim \text{Beta}(1,1)$$
$$n \sim \text{Categorical}(\underbrace{\frac{1}{n_{\max}}, \ldots, \frac{1}{n_{\max}}}_{m})$$

Fig. 3.12 Graphical model for the joint inference of n and θ from a set of m observed counts of successes $k_1, \ldots, k_m$.

To make the problem concrete, suppose there are five helpers distributing a bundle of surveys to houses. It is known that each bundle contained the same number of surveys, n, but the number itself is not known. The only available relevant information is that the maximum bundle is $n_{\max} = 500$, and so n must be between 1 and $n_{\max}$.

In this problem, it is also not known what the rate of return for the surveys is. But, it is assumed that each helper distributed to houses selected in a random enough way that it is reasonable to believe the return rates are the same. It is also assumed to be reasonable to set a uniform prior on this common rate $\theta \sim \text{Beta}(1,1)$.

Inferences can simultaneously be made about n and θ from the observed number of surveys returned for each of the helpers. Assuming the surveys themselves can be identified with their distributing helper when returned, the data will take the form of $m = 5$ counts, one for each helper, giving the number of returned surveys for each.

The graphical model for this problem is shown in Figure 3.12, and the script `Survey.txt` implements the graphical model in WinBUGS. Note the use of the categorical distribution, which gives probabilities to a finite set of nominal outcomes:

```
# Inferring Return Rate and Number of Surveys from Observed Returns
model{
  # Observed Returns
  for (i in 1:m){
     k[i] ~ dbin(theta,n)
  }
  # Priors on Rate Theta and Number n
  theta ~ dbeta(1,1)
  n ~ dcat(p[])
  for (i in 1:nmax){
    p[i] <- 1/nmax
  }
}
```

The code `Survey.m` or `Survey.R` uses the data $k = \{16, 18, 22, 25, 27\}$, and then calls WinBUGS to sample from the graphical model. Figure 3.13 shows the joint posterior distribution over n and θ as a scatter-plot, and the marginal distributions of each as histograms.

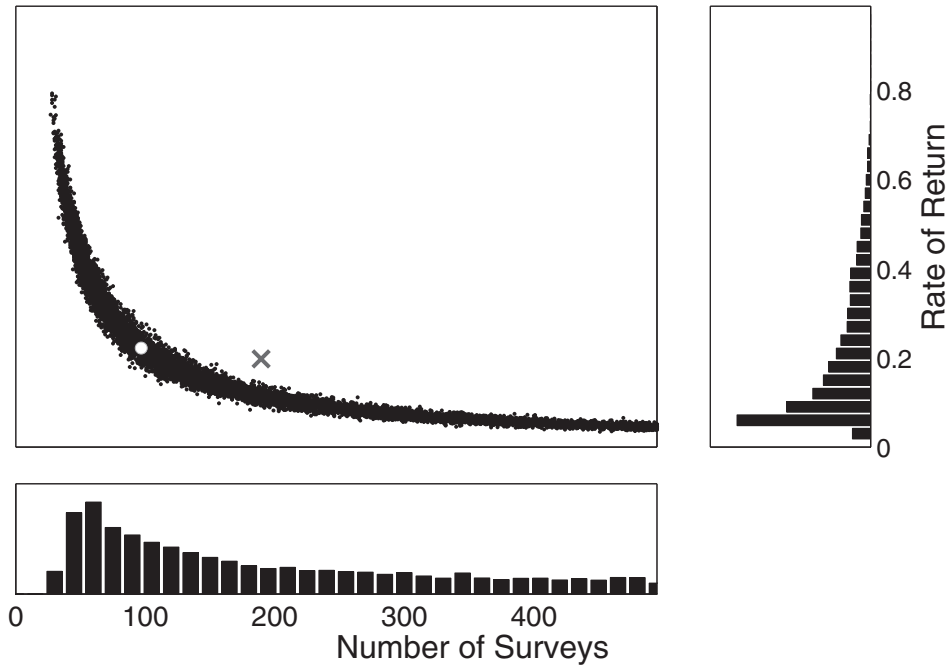


Fig. 3.13 Joint posterior distribution of the probability of return θ and the number of surveys n for $m = 5$ observed counts $k = \{16, 18, 22, 25, 27\}$. The histograms show the marginal densities. The cross shows the expected value of the joint posterior, and the circle shows the mode (i.e., maximum likelihood), both estimated from the posterior samples.

It is clear that the joint posterior distribution carries more information than the marginal posterior distributions. This is very important. It means that just looking at the marginal distributions will not give a complete account of the inferences made, and may provide a misleading account.

An intuitive graphical way to see that there is extra information in the joint posterior is to see if it is well approximated by the product of the marginal distributions. Imagine sampling a point from the histogram for n where there is non-negligible marginal density, such as at $n = 300$. Imagine also sampling points from the histogram for θ, where there is non-negligible marginal density, such as at $\theta = 0.4$. These choices correspond to a single point in the joint posterior density space. Now imagine repeating this process many times. It should be clear that the resulting scatter-plot would be different from the joint posterior scatter-plot in Figure 3.13. So, the joint distribution carries information not available from the marginal distributions.

For this example, it is intuitively obvious why the joint posterior distribution has the clear non-linear structure it does. One possible way in which 20 surveys might be returned is if there were only about 50 surveys, but 40% were returned. Another

possibility is that there were 500 surveys, but only a 4% return rate. In general, the number and return rate can trade-off against each other, sweeping out the joint posterior distribution seen in Figure 3.13.

Exercises

Exercise 3.6.1 The basic moral of this example is that it is often worth thinking about joint posterior distributions over model parameters. In this case the marginal posterior distributions are probably misleading. Potentially even more misleading are common (and often perfectly appropriate) point estimates of the joint distribution. The cross in Figure 3.13 shows the expected value of the joint posterior, as estimated from the samples. Notice that it does not even lie in a region of the parameter space with any posterior mass. Does this make sense?

Exercise 3.6.2 The circle in Figure 3.13 shows an approximation to the mode (i.e., the sample with maximum likelihood) from the joint posterior samples. Does this make sense?

Exercise 3.6.3 Try the very slightly changed data $k = \{16, 18, 22, 25, 28\}$. How does this change the joint posterior, the marginal posteriors, the expectation, and the mode? If you were comfortable with the mode, are you still comfortable?

Exercise 3.6.4 If you look at the sequence of samples in the trace plot, some autocorrelation is evident. The samples "sweep" through high and low values in a systematic way, showing the dependency of a sample on those immediately preceding. This is a deviation from the ideal situation in which posterior samples are independent draws from the joint posterior. Try thinning the sampling, taking only every 100th sample, by setting `nthin=100` in Matlab or `n.thin=100` in R. To make the computational time reasonable, reduce the number of samples collected after thinning to just 500 (i.e., run 50,000 total samples, so that 500 are retained after thinning). How is the sequence of samples visually different with thinning?[2]

[2] A note for R2jags users: at the time of writing, R2jags mistakenly randomizes the values in the `sims.array` object whenever you run a single chain. Until this error is fixed it is safest to run multiple chains, at least when you are interested in examining autocorrelation. See also the last few posts here: `http://sourceforge.net/p/mcmc-jags/discussion/610037/thread/cc61b820/?limit=50#83b4`.