Bayesian Cognitive Modeling: A Practical Course

MICHAEL D. LEE AND ERIC-JAN WAGENMAKERS

March 21, 2012

PRELIMINARY DRAFT
SUGGESTIONS FOR IMPROVEMENT WELCOME

Contents

Preface

This book teaches you how to do Bayesian modeling. Using modern computer software—and, in particular, the WinBUGS program—this turns out to be surprisingly straightforward. After working through the examples provided in this book, you should be able to build your own models, apply them to your own data, and draw your own conclusions.

This book is based on three principles. The first is that of *accessibility*: the book's only prerequisite is that you know how to operate a computer; you do not need any advanced knowledge of statistics or mathematics. The second principle is that of *applicability*: the examples in this book are meant to illustrate how Bayesian modeling can be useful for problems that people in cognitive science care about. The third principle is that of *practicality*: this book offers a hands-on, "just do it" approach that we feel keeps students interested and motivated to learn more.

In line with these three principles, this book has little content that is purely theoretical. Hence, you will not learn from this book why the Bayesian philosophy to inference is as compelling as it is; neither will you learn much about the intricate details of modern sampling algorithms such as Markov chain Monte Carlo, even though this book could not exist without them.

The goal of this book is to facilitate and promote the use of Bayesian modeling in cognitive science. As shown by means of examples throughout this book, Bayesian modeling is ideally suited for applications in cognitive science. It is easy to construct a basic model, and then add individual differences, add substantive prior information, add covariates, add a contaminant process, and so on. In other words, Bayesian modeling is flexible and respects the complexities that are inherent in the modeling of cognitive phenomena.

We hope that after completing this course, you will have gained not only a new understanding of statistics (yes, it can make sense), but also the technical skills to implement statistical models that professional but non-Bayesian cognitive scientists dare only dream about.

We like to thank John Miyamoto, Eddy Davelaar, Hedderik van Rijn, and Thomas Palmeri for constructive criticism and suggestions for improvement, and Dora Matzke for her help in programming and plotting.

Michael D. Lee
Irvine, USA
Eric-Jan Wagenmakers
Amsterdam, The Netherlands
August 2011

PART I

GETTING STARTED

(...) the theory of probabilities is basically just common sense reduced to calculus; it makes one appreciate with exactness that which accurate minds feel with a sort of instinct, often without being able to account for it.

Laplace, 1829

1 The Basics of Bayesian Analysis

1.1 General Principles

The general principles of Bayesian analysis are easy to understand. First, uncertainty or "degree of belief" is quantified by probability. Second, the observed data are used to update the *prior* information or beliefs to become *posterior* information or beliefs. That's it!

To see how this works in practice, consider the following example. Assume we give you a test that consists of 10 factual questions of equal difficulty. What we want to estimate is your ability θ—the rate with which you answer questions correctly. Note that we do not directly observe your ability θ; all that we observe is your score on the test.

Before we do anything else (for example, before we start to look at your data) we need to specify our prior uncertainty with respect to your ability θ. This uncertainty needs to be expressed as a probability distribution, called the *prior distribution*. In this case, keep in mind that θ can range from 0 to 1, and that you do not know anything about the topic or about the difficulty level of the questions. Then, a reasonable "prior distribution", denoted by $p(\theta)$, is one that assigns equal probability to every value of θ. This uniform distribution is shown by the dotted horizontal line in Figure 1.1.

Now we consider your performance, and find that you answered 9 out of 10 questions correctly. After having seen these data, the updated knowledge about θ is described by the *posterior distribution*, denoted $p(\theta \mid D)$, where D indicates the observed data. Bayes rule specifies how we can combine the information from the data—that is, the Binomial likelihood $p(D \mid \theta)$—with the information from the prior distribution $p(\theta)$ to arrive at the posterior distribution $p(\theta \mid D)$:

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{p(D)}. \tag{1.1}$$

This equation is often verbalized as

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}. \tag{1.2}$$

Note that the marginal likelihood (i.e., the probability of the observed data) does not involve the parameter θ, and is given by a single number that ensures that the area under the posterior distribution equals 1. Therefore, Equation 1.1 is often

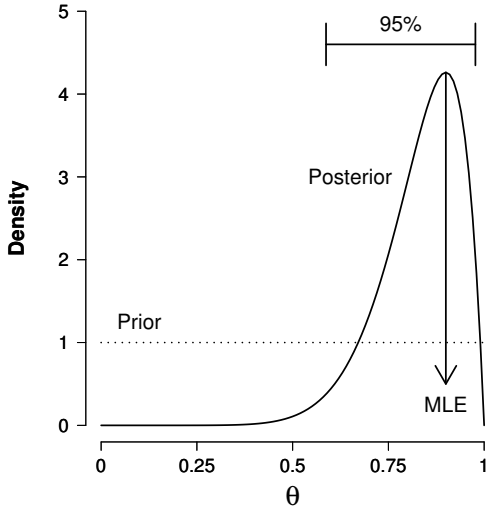


Fig. 1.1 Bayesian parameter estimation for rate parameter θ, after observing 9 correct responses and 1 incorrect response. The mode of the posterior distribution for θ is 0.9, equal to the maximum likelihood estimate, and the 95% confidence interval extends from 0.59 to 0.98.

written as

$$p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta), \tag{1.3}$$

which says that the posterior is proportional to the likelihood times the prior. Note that the posterior distribution is a combination of what we knew before we saw the data (i.e., the information in the prior distribution), and what we have learned from the data.

The solid line in Figure 1.1 shows the posterior distribution for θ, obtained when the uniform prior is updated with the data, that is, $k = 9$ correct answers out of $n = 10$ questions. The central tendency of a posterior distribution is often summarized by its mean, median, or mode. Note that with a uniform prior, the mode of a posterior distribution coincides with the classical maximum likelihood estimate or MLE , $\hat{\theta} = k/n = 0.9$ (Myung, 2003). The spread of a posterior distribution is most easily captured by a Bayesian x% credible interval that extends from the $(x/2)$th to the $(100-x/2)$th percentile of the posterior distribution. For the posterior distribution in Figure 1.1, a 95% Bayesian credible interval for θ extends from 0.59 to 0.98. In contrast to the orthodox confidence interval, this means that one can be 95% confident that the true value of θ lies in between 0.59 and 0.98.

Exercises

Exercise 1.1.1 The famous Bayesian statistician Bruno de Finetti published two big volumes entitled "Theory of Probability" (de Finetti, 1974). Perhaps surprisingly, the first volume starts with the words "probability does not exist".

To understand why de Finetti wrote this, consider the following situation: someone tosses a fair coin, and the outcome will be either heads or tails. What do you think the probability is that the coin lands heads? Now suppose you are a physicist with advanced measurement tools, and you can establish relatively precisely both the position of the coin and the tension in the muscles immediately before the coin is tossed in the air—does this change your probability? Now suppose you can briefly look into the future (Bem, 2011), albeit hazily—is your probability still the same?

Exercise 1.1.2 On his blog, prominent Bayesian Andrew Gelman wrote (March 18, 2010) "Some probabilities are more objective than others. The probability that the die sitting in front of me now will come up '6' if I roll it...that's about 1/6. But not exactly, because it's not a perfectly symmetric die. The probability that I'll be stopped by exactly three traffic lights on the way to school tomorrow morning: that's...well, I don't know exactly, but it is what it is." Was de Finetti wrong, and is there only one clearly defined probability of Andrew Gelman encountering three traffic lights on the way to school tomorrow morning?

Exercise 1.1.3 Figure 1.1 shows that the 95% Bayesian credible interval for θ extends from 0.59 to 0.98. This means that one can be 95% confident that the true value of θ lies in between 0.59 and 0.98. Suppose you did an orthodox analysis and found the same confidence interval. What is the orthodox interpretation of this interval?

Exercise 1.1.4 Suppose you learn that the questions are all true or false questions. Does this knowledge affect your prior distribution? And if so, how would this prior in turn affect your posterior distribution?

1.2 Prediction

The posterior distribution θ contains all that we know about the rate with which you answer questions correctly. One way to use the knowledge is *prediction*.

For instance, suppose we design a new set of 5 questions, all of the same difficulty as before. How can we formalize our expectations about your performance on this new set? In other words, how can we use the posterior distribution $p\left(\theta \mid n = 10, k = 9\right)$—which after all represents everything that we know about θ from the old set—to *predict* the number of correct responses out of the new set of $n^{\text{rep}} = 5$ questions? The mathematical solution is to integrate over the posterior, $\int p\left(k^{\text{rep}} \mid \theta, n^{\text{rep}} = 5\right) p\left(\theta \mid n = 10, k = 9\right)\ \mathrm{d}\theta$, where k^{rep} is the predicted number of correct responses out of the additional set of 5 questions.

Computationally, you can think of this procedure as repeatedly drawing a random value θ_i from the posterior, and using that value to every time determine a single k_i^{rep}. The end result is $p\left(k^{\text{rep}}\right)$, the posterior predictive density of the possible number of correct responses in the additional set of 5 questions. The important

point is that by integrating over the posterior, all predictive uncertainty is taken into account.

Exercises

Exercise 1.2.1 Instead of "integrating over the posterior", orthodox methods often use the "plug-in principle"; in this case, the plug-in principle suggest that we predict $p(k^{\text{rep}})$ solely based on $\hat{\theta}$, the maximum likelihood estimate. Why is this generally a bad idea? Can you think of a specific situation in which this may not be so much of a problem?

1.3 Sequential Updating

Bayesian analysis is particularly appropriate when you want to combine different sources of information. For instance, assume that we present you with a new set of 5 questions. You answer 3 out of 5 correctly. How can we combine this new information with the old? Or, in other words, how do we update our knowledge of θ? Consistent with intuition, Bayes' rule entails that the prior that should be updated based on your performance for the new set is the posterior that was obtained based on your performance for the old set. Or, as Lindley put it, "today's posterior is tomorrow's prior" (Lindley, 1972, p. 2).

When all the data have been collected, however, the precise order in which this was done is irrelevant; the results from the 15 questions could have been analyzed as a single batch, they could have been analyzed sequentially, one-by-one, they could have been analyzed by first considering the set of 10 questions and next the set of 5, or vice versa. For all these cases, the end result, the final posterior distribution for θ, is identical. This again contrasts with orthodox inference, in which inference for sequential designs is radically different from that for non-sequential designs (for a discussion, see, for example, Anscombe, 1963).

Thus, a posterior distribution describes our uncertainty with respect to a parameter of interest, and the posterior is useful—or, as a Bayesian would have it, necessary—for probabilistic prediction and for sequential updating. In general, the posterior distribution or any of its summary measures can only be obtained in closed form for a restricted set of relatively simple models. To illustrate in the case of our binomial example, the uniform prior is a so–called beta distribution with parameters $\alpha = 1$ and $\beta = 1$, and when combined with the binomial likelihood this yields a posterior that is also a beta distribution, with parameters $\alpha + k$ and $\beta + n - k$. In simple *conjugate* cases such as these, where the prior and the posterior belong to the same distributional family, it is possible to obtain closed form solutions for the posterior distribution, but in many interesting cases it is not.

1.4 Markov Chain Monte Carlo

For a long time, researchers could only proceed with Bayesian inference when the posterior was available in closed form. As a result, practitioners interested in models of realistic complexity did not much use Bayesian inference. This situation changed dramatically with the advent of computer-driven sampling methodology generally known as Markov chain Monte Carlo (MCMC: e.g., Gamerman & Lopes, 2006; Gilks, Richardson, & Spiegelhalter, 1996). Using MCMC techniques such as Gibbs sampling or the Metropolis-Hastings algorithm, researchers can directly sample sequences of values from the posterior distribution of interest, foregoing the need for closed form analytic solutions. The current adage is that *Bayesian models are limited only by the user's imagination*.

In order to visualize the increased popularity of Bayesian inference, Figure 1.2 plots the proportion of articles that feature the words "Bayes" or "Bayesian", according to Google Scholar (for a similar analysis for specific journals in statistics and economics see Poirier, 2006). The time line in Figure 1.2 also indicates the publication of three landmark papers, Geman and Geman (1984), Gelfand and Smith (1990), and Casella and George (1992), as well as the introduction of WinBUGS, a general-purpose program that greatly facilitates Bayesian analysis for a wide range of statistical models (Lunn, Thomas, Best, & Spiegelhalter, 2000; Lunn, Spiegelhalter, Thomas, & Best, 2009; Sheu & O'Curry, 1998). Thus, MCMC methods have transformed Bayesian inference to a vibrant and practical area of modern statistics.

For a concrete and simple illustration of Bayesian inference using MCMC, consider again the binomial example of 9 correct responses out of 10 questions, and the associated inference problem for θ, the rate of answering questions correctly. Throughout this book, we use WinBUGS to specify and fit our models, saving us the effort to code the MCMC algorithms ourselves. Although WinBUGS does not work for every research problem application, it will work for many in cognitive science. WinBUGS is easy to learn and is supported by a large community of active researchers.[1]

The WinBUGS program requires you to construct a file that contains the model specification, a file that contains initial values for the model parameters, and a file that contains the data. The model specification file is most important. For our binomial example, we set out to obtain samples from the prior and the posterior of θ. The associated WinBUGS model specification code is three lines long:

```
model {
  theta ~ dbeta(1,1) # the uniform prior for updating by the data
  k ~ dbin(theta,n)  # the data; in our example, k = 9 and n = 10
  thetaprior ~ dbeta(1,1) # a uniform prior not for updating
```

[1] Two alternative software programs are OpenBUGS and JAGS, programs that may be particularly attractive for Mac and Linux users. The model code for OpenBUGS and JAGS is almost identical to WinBUGS, so that the transition from one program to the other is easy.

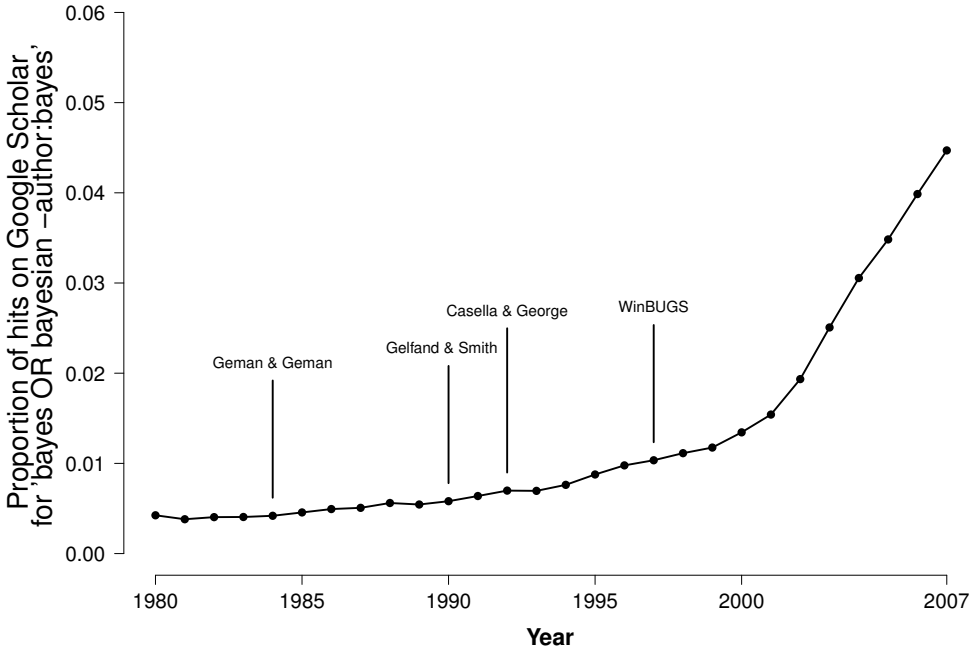


Fig. 1.2 A Google Scholar perspective on the increasing popularity of Bayesian inference.

```
}
```

In this code, the "∼" or twiddle symbol denotes "is distributed as", `dbeta(a,b)` indicates the beta distribution with parameters a and b[2], and `dbin(theta,n)` indicates the binomial distribution with rate theta and n observations. These and many other distributions are build in to the WinBUGS program. The "#" or hash sign is used for commenting out what should not be compiled. As WinBUGS is a declarative language, the order of the three lines is inconsequential.

When this code is executed, you obtain a sequence of samples (i.e., an MCMC chain) from the posterior $p(\theta \mid D)$ and a sequence of samples from the prior $p(\theta)$. This sequence is called a *chain*. In more complex models, it may take some time before a chain converges from its starting value to what is called its stationary distribution. To make sure that we only use those samples that come from the stationary distribution, and hence are unaffected by the starting values, it is good practice to diagnose convergence by running multiple chains. It is often also good practice to discard the first samples from each chain. These discarded samples are called *burn in* samples. Finally, it can also be helpful, especially when the sampling process produce auto-correlates sequences, not to record every sample taken in a

[2] The `dbeta(1,1)` distribution is a uniform distribution from 0 to 1. Therefore, the prior distribution for θ could also have been specified as `theta` ∼ `dunif(0,1)`.

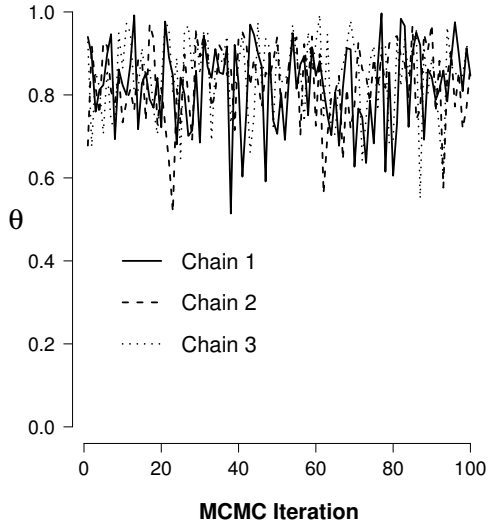


Fig. 1.3 Three MCMC chains for rate parameter θ, after observing 9 correct responses and 1 incorrect response.

chain, but every second, or third, or tenth, or some other subset of samples. This is known as *thinning*.

For example, Figure 1.3 shows the first 100 iterations for three chains that were set up to draw values from the posterior for θ. It is evident that the three chains are "mixing" well, suggesting early convergence. Quantitative measures for diagnosing convergence are also available, such as the Gelman and Rubin (1992) $\hat{R}$ statistic, that compares within–chain to between–chain variability. For more recommendations regarding convergence see Gelman (1996) and Gelman and Hill (2007).

After assuring ourselves that the chains have converged, we can use the sampled values to plot a histogram, construct a density estimate, and compute values of interest. To illustrate, the three chains from Figure 1.3 were run for 3000 iterations each, for a total of 9000 samples for the prior and the posterior of θ. Figure 1.4 plots histograms for the prior (i.e., dotted line) and the posterior (i.e., thick solid line). To visualize how the histograms are constructed from the MCMC chains, the bottom panel of Figure 1.4 plots the MCMC chains sideways; the histograms are created by collapsing the values along the "MCMC iteration" axis and onto the "θ" axis.

In the top panel of Figure 1.4, the thin solid lines represent density estimates. The mode of the density estimate for the posterior of θ is 0.89, whereas the 95% credible interval is $(0.59, 0.98)$, matching the analytical result shown in Figure 1.1.

The key point is that the analytical intractibilities that limited the scope of Bayesian parameter estimation have now been overcome. Using MCMC sampling, posterior distributions can be approximated to any desired degree of accuracy. This book teaches you to use MCMC sampling and Bayesian inference to do research with cognitive science models and data.

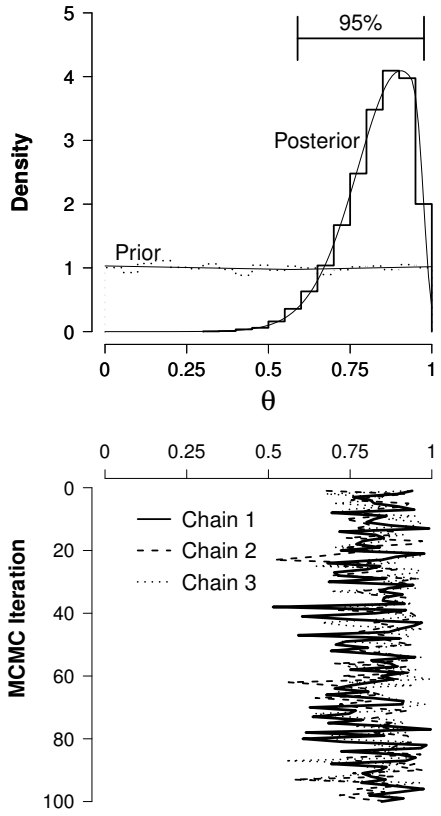


Fig. 1.4 MCMC–based Bayesian parameter estimation for rate parameter θ, after observing 9 correct responses and 1 incorrect response. The thin solid lines indicate the fit of a density estimator. Based on this density estimator, the mode of the posterior distribution for θ is approximately 0.89, and the 95% credible interval extends from 0.59 to 0.98, closely matching the analytical results from Figure 1.1.

Exercise 1.4.1 Use Google and list some other scientific disciplines that use Bayesian inference and MCMC sampling.

Exercise 1.4.2 The text reads: "Using MCMC sampling, posterior distributions can be approximated to any desired degree of accuracy". How is this possible?

1.5 Further Reading

This section provides some references for further reading. We first list Bayesian textbooks and seminal papers, then some texts that specifically deal with WinBUGS. We also note that Smithson (2010) presents a useful comparative review of six introductory textbooks on Bayesian methods.

1.5.1 Bayesian Statistics

This section contains an annotated bibliography on Bayesian articles and books that we believe are particularly useful or inspiring.

Berger, J. O. and Wolpert, R. L. (1988). *The Likelihood Principle* (2nd ed.). Institute of Mathematical Statistics, Hayward (CA). This is a great book if you want to understand the limitations of orthodox statistics. Insightful and fun.

Bolstad, W. M. (2007). *Introduction to Bayesian Statistics* (2nd ed.). Wiley, Hoboken (NJ). Many books claim to introduce Bayesian statistics, but forget to state on the cover that the introduction is "for statisticians" or "for those comfortable with mathematical statistics". The Bolstad book is an exception, as it does not assume much background knowledge.

Gamerman, D., & Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman & Hall/CRC, Boca Raton (FL). This book discusses the details of MCMC sampling; a good book, but too advanced for beginners.

Gelman, A. & Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, This book is an extensive practical guide on how to apply Bayesian regression models to data. WinBUGS code is provided throughout the book. Andrew Gelman also has an active blog that you might find interesting: `http://andrewgelman.com/`

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, Boca Raton (FL). A citation classic in the MCMC literature, this book features many short chapters on all kinds of sampling-related topics: theory, convergence, model selection, mixture models, and so on.

Gill, J. (2002). Bayesian Methods: A Social and Behavioral Sciences Approach. CRC Press, Boca Raton (FL). A well-written book that covers a lot of ground. Readers need some background in mathematical statistics to appreciate the content.

Hoff, P. D. (2009). *A First Course in Bayesian Statistical Methods*. Springer, Dordrecht, The Netherlands. A clear and well-written introduction to Bayesian inference, with accompanying R code. This book requires some familiarity with mathematical statistics.

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge. Jaynes was one of the most ardent supporters of objective Bayesian statistics. The book is full of interesting ideas and compelling arguments, as well as being laced with Jaynes' acerbic wit, but it requires some mathematical background to appreciate all of the content.

Jeffreys, H. (1939/1961). *Theory of Probability*. Oxford University Press, Oxford, UK. Sir Harold Jeffreys is the first statistician who exclusively used Bayesian methods for inference. Jeffreys also invented the Bayesian hypothesis test, and was generally far ahead of his time. The book is not always an easy read, in part because the notation is somewhat outdated. Strongly recommended, but only for those who already have a solid background in mathematical statistics and a firm grasp of Bayesian thinking. See `www.economics.soton.ac.uk/staff/aldrich/jeffreysweb.htm`

Lindley, D. V. (2000). The philosophy of statistics. *The Statistician, 49*, 293-337. Dennis Lindley, one the godfathers of current-day Bayesian statistics, explains why Bayesian inference is right and everything else is wrong. Peter Armitage commented on the paper: "Lindley's concern is with the very nature of statistics, and his argument unfolds clearly, seamlessly and relentlessly. Those of us who cannot accompany him to the end of his journey must consider very carefully where we need to dismount; otherwise we shall find ourselves unwittingly at the bus terminus, without a return ticket."

McGrayne, S. B. (2011). *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant From Two Centuries Of Controversy*. Yale University Press.. A fascinating and accessible overview of the history of Bayesian inference.

Ntzoufras, I. (2009). *Bayesian Modeling using WinBUGS*. Wiley, Hoboken (NJ). A great book for learning how to do regression and ANOVA using WinBUGS. See `www.ruudwetzels.com` for a detailed review.

O'Hagan, A. & Forster, J. (2004). *Kendall's Advanced Theory of Statistics Vol. 2B: Bayesian Inference* (2nd ed.). Arnold, London. If you are willing to read only a single book on Bayesian statistics, this one is it. The book requires a background in mathematical statistics.

Royall, R. M. (1997). *Statistical Evidence: A Likelihood Paradigm*. Chapman & Hall, London. This book describes the different statistical paradigms, and highlights

the deficiencies of the orthodox schools. The content can be appreciated without much background knowledge in statistics. The main disadvantage of this book is that the author is not a Bayesian. We still recommend the book, which is saying something.

1.5.2 WinBUGS Texts

Kruschke, J. K. (2010). *Doing Bayesian Data Analysis: A Tutorial Introduction with R and BUGS*. Academic Press, Burlington (MA). This is one of the first Bayesian books geared explicitly towards experimental psychologists and cognitive scientists. Kruschke explains core Bayesian concepts with concrete examples and OpenBUGS code. The book focuses on statistical models such as regression and ANOVA, and provides a Bayesian approach to data analysis in psychology, cognitive science, and empirical sciences more generally.

Lee, S.–Y. (2007). *Structural Equation Modelling: A Bayesian Approach*. Chichester, UK: Wiley.

Ntzoufras, I. (2009). *Bayesian modeling using WinBUGS*. Hoboken, NJ: Wiley. Provides an easily accessible introduction to the use of WinBUGS. The book also presents a variety of Bayesian modeling examples, with the emphasis on Generalized Linear Models.

Spiegelhalter, D., Best, N. & Lunn, D. (2003). *WinBUGS User Manual 1.4.* MRC Biostatistic Unit, Cambridge, UK. Provides an introduction to the use of WinBUGS, including a useful tutorial and various tips and tricks for new users.

2 Getting Started with WinBUGS

with Dora Matzke

Throughout this course book, you will use the WinBUGS (Lunn et al., 2000) software to work your way through the exercises. Although it is possible to do the exercises using the graphical user interface provided by the WinBUGS package, you can also use the Matlab or R programs to interact with WinBUGS.

In this chapter, we start by working through a concrete example using just WinBUGS. This provides an introduction to the WinBUGS interface, and the basic theoretical and practical components involved in Bayesian graphical model analysis. Completing the example will also quickly convince you that you *do not* want to rely on WinBUGS as your primary means for handling and analyzing data. It is not especially easy to use as a graphical user interface, and does not have all of the data management and visualization features needed for research.

Instead, we encourage you to choose either Matlab or R as your primary research computing environment, and use WinBUGS as an 'add-on' that does the Bayesian inference part of analyses. Some WinBUGS interface capabilities will remain useful, especially in the exploratory stages of research. But either Matlab or R will be primary. Accordingly, this chapter re-works the concrete example, originally done in WinBUGS, using both Matlab and R. You should complete just the one corresponding to your preferred research software. You will then be ready for the following chapters, which assume you are working in either Matlab or R, but understand the basics on the WinBUGS interface.

2.1 Installing WinBUGS, Matbugs, and R2WinBugs

2.1.1 Installing WinBUGS

WinBUGS is a currently free software, and is available at `http://www.mrc-bsu.cam.ac.uk/bugs/`. Download the most recent version, including any patches, and make sure you go to the effort of downloading and applying the registration key. Some of the exercises in this course might work without the registration key, but some of them will not. You can download WinBUGS and the registration key directly from `http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml`.

2.1.2 Installing Matlab and Matbugs

Matlab is a commercial software, and is available at `http://www.mathworks.com/`. As best we know, any reasonably recent version of Matlab should let you do the exercises in this course. Also, as best we know, no toolboxes are required. To give Matlab the ability to interact with WinBUGS, download the freely available `matbugs.m` function and put it in your Matlab working directory. You can download `matbugs.m` directly from `http://www.cs.ubc.ca/~murphyk/Software/MATBUGS/matbugs.html`.

2.1.3 Installing R and R2WinBUGS

R is a free software, and is available at `http://www.r-project.org/`. You can download the Windows version of R directly from `http://cran.nedmirror.nl/bin/windows/base/`. To give R the ability to interact with WinBUGS, you have to install the `R2WinBUGS package`. To install the `R2WinBUGS` package, start R and select the `Install Package(s)` option in the `Packages` menu. Once you chose your preferred CRAN mirror, select `R2WinBUGS` in the `Packages` window and click on `OK`.

2.2 Using the Applications

2.2.1 An Example with the Binomial Distribution

We will illustrate the use of WinBUGS, Matbugs, and R by means of a simple example involving a binary process. A binary process is anything where there are only two possible outcomes. It might be that something either happens or does not happen, or that something either succeeds or fails, or that something takes one value rather than the other. An inference that is often important for these sorts of processes is the underlying *rate* at which the process takes one value rather than the other. Inferences about the rate can be made by observing how many times the process takes each value over a number of trials.

Suppose that one of the values (e.g., the number of successes) happens on k out of n trials. These are known, or observed, data. The unknown variable of interest is the rate θ at which the values are produced. Assuming that what happened on one trial does not influence the other trials the number of successes k follows a Binomial distribution, $k \sim \text{Binomial}(\theta, n)$. This relationship means that by observing the k successes out of n trials, it is possible to update our knowledge about the rate θ. The basic idea of Bayesian analysis is that what we know, and what we do not know, about the variables of interest is always represented by probability distributions. Data like k and n allow us to update prior distributions for the unknown variables into posterior distributions that incorporate the new information.

The graphical model representation of our binomial example is shown in Figure 2.1. The nodes represent all the variables that are relevant to the problem. The graph structure is used to indicate dependencies between the variables, with children depending on their parents. We use the conventions of representing unobserved variables without shading and observed variables with shading, and continuous variables with circular nodes and discrete variables with square nodes.

Thus, the observed discrete counts of the numbers of successes k and the number of trials n are represented by shaded and square nodes, and the unknown continuous rate θ is represented by an unshaded and circular node. Because the number of successes k depends on the number of trials n and on the rate of success θ, the nodes representing n and θ are directed towards the node representing k. We will start with the prior assumption that all possible rates between 0 and 1 are equally likely. We will thus assume a uniform prior $\theta \sim \text{Uniform}\big(0, 1\big)$.

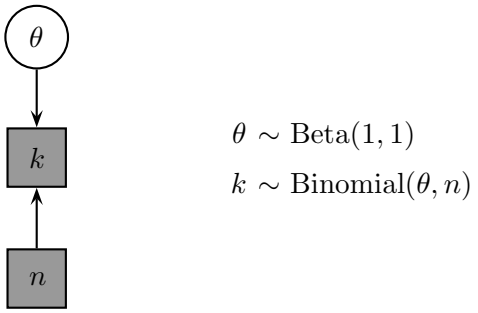


Fig. 2.1 Graphical model for inferring the rate of a binary process.

One advantage of using the language of graphical models is that it gives a complete and interpretable representation of a Bayesian probabilistic model. Another advantage is that WinBUGS can easily implement graphical models, and its various built-in computational algorithms are then able to do all of the inferences automatically.

2.2.2 Using WinBUGS

WinBUGS requires the user to construct a file that contains the data, a file that contains the starting values for the model parameters, and a file that contains the model specification. The WinBUGS model specification code associated with our binomial example is as follows:

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

Note that, even though conceptually the prior on θ is Uniform$(0, 1)$, it has been implemented as Beta$(1, 1)$. These two distributions are the same, but our experience is that WinBUGS seems to have fewer computational problems with the Beta distribution implementation.

Implementing the model shown in Figure 2.1, and obtaining samples from the posterior distribution of θ, can be done by following these steps.

1. Copy the model specification text above and paste it in a text file. Save the file, for instance as "Rate_1.txt".
2. Start WinBUGS. Open your newly created model specification file by selecting the `Open` option in the `File` menu, choosing the appropriate directory, and double-clicking on the model specification file. Do not forget to select files of type "txt", or you might be searching for a long time. Now check the syntax of the model specification code by selecting the `Specification` option in the `Model` menu. Once the `Specification Tool` window is opened, as shown in Figure 2.2, highlight the word "model" at the beginning of the code and click on `check model`. If the model is syntactically correct and all parameters are given priors, the message "model is syntactically correct" will appear in the status bar all the way in the bottom left corner of the WinBUGS window. (Although beware, the letters are very small and difficult to see).
3. Create a text file that contains the data. The content of the file should look like this:

   ```
   list(
   k=5,
   n=10
   )
   ```

   Save the file, for instance as "Data.Rate_1.txt".
4. Open the data file and load the data. To open the data file, select the `Open` option in the `File` menu, select the appropriate directory, and double-click on the data file. To load the data, highlight the word "list" at the beginning of the data file and click on `load data` in the `Specification Tool` window, as shown in Figure 2.2. If the data are successfully loaded, the message "data is loaded" will appear in the status bar.
5. Set the number of chains. Each chain is an independent run of the same model with the same data, although you can vary the set different starting values of parameters for each chain.[1] Chains provide a key test of convergence—something we will discuss in more detail in a later chapter. In our binomial example, we will run two chains. To set the number of chains, type "2" in the field labelled `num of chains` in the `Specification Tool` window, shown in Figure 2.2.
6. Compile the model. To compile the model, click on `compile` in the

[1] Running multiple chains is the best and easiest way to ensure WinBUGS uses different random number sequences in sampling. Doing a single-chain analysis multiple times can use the same random number sequence, and so produce the same results.

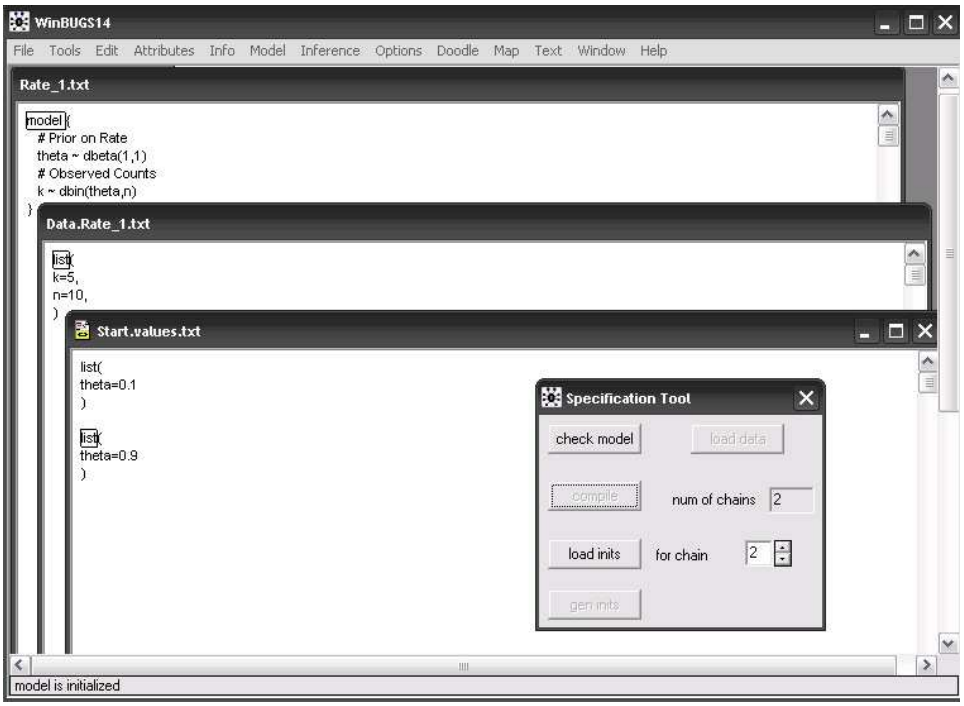


Fig. 2.2 Model Specification Tool.

`Specification Tool` window, shown in Figure 2.2. If the model is successfully compiled, the message "model compiled" will appear in the status bar.

7. Create a text file that contains the starting values of the unobserved variables (i.e., just the parameter θ for this model). If you do not specify the starting values, WinBUGS will try to get them from the prior, which may or may not lead to numerical crashes. It is therefore safer to give a starting value to all unobserved variables, and especially for variables at nodes 'at the top' of the graphical model, which have no parents.
   The content of the file should look like this:

```
list(
theta=0.1
)

list(
theta=0.9
)
```

   Save the file, for instance as "Start.values.txt".
8. Open the file that contains the starting values by selecting the `Open` option in the `File` menu, selecting the appropriate directory, and double-clicking on the file. To load the starting value of θ for the first chain, highlight the word "list" at the beginning of the file and click on `load inits` in the `Specification Tool`

window, shown in Figure 2.2). To load the starting value for the second chain, highlight the second "list" command and click on `load inits` once again. If the starting values are successfully loaded, the message "model is initialized" will appear in the status bar.

9. Set monitors to store the sampled values of the parameters of interest. To set a monitor for θ, select the `Samples` option from the `Inference` menu. Once the `Sample Monitor Tool` window, shown in Figure 2.3, is opened, type "theta" in the field labelled `node` and click on `set`.
10. Specify the number of samples you want to record. To this end, you first have to specifythe total number of samples you want to draw from the posterior of θ, and the number of burn-in samples that you want to discard at the beginning of a sampling run. The number of recorded samples equals the total number of samples minus the number of burn-in samples. In our binomial example, we will not discard any of the samples and will set out to obtain $20,000$ samples from the posterior of θ. To specify the number of recorded samples, type "1" in the field labelled `beg` (i.e., WinBUGS will start recording from the first sample) and type "20000" in the field labelled `end` in the `Sample Monitor Tool` window, shown in Figure 2.3).

Fig. 2.3 Sample Monitor Tool.

11. Set "live" trace plots of the unobserved parameters of interest. WinBUGS allows you to monitor the sampling run in real-time. This can be useful on long sampling runs, for debugging, and for diagnosing whether the chains have converged. To set a "live" trace plot of θ, click on `trace` in the `Sample Monitor Tool` window, shown in Figure 2.3, and wait for an empty plot to appear on the screen. Once WinBUGS starts to sample from the posterior, the trace plot of θ will appear live on the screen.
12. Specify the total number of samples that you want to draw from the posterior. This is done by selecting the `Update` option from the `Model` menu. Once the `Update Tool` window (see 2.4) is opened, type "20000" in the field labelled `updates`. Typically, the number you enter in the `Update Tool` window will correspond to the number you entered in the `end` field of the `Sample Monitor Tool`.
13. Specify how many samples should be drawn between the recorded samples.

You can, for example, specify that only every second drawn sample should be recorded. This ability to "thin" a chain is is important when successive samples are not independent but autocorrelated. In our binomial example, we will record every sample that is drawn from the posterior of θ. To specify this, type "1" in the field labelled `thin` in the `Update Tool` window, shown in Figure 2.4.

14. Specify the number of samples after which WinBUGS should refresh its display. To this end, type "100" in the field labelled `refresh` in the `Update Tool` window, shown in Figure 2.4.
15. Sample from the posterior. To sample from the posterior of θ, click on `update` in the `Update Tool` window, shown in Figure 2.4). During sampling, the message "model updating" will appear in the status bar. Once the sampling is finished, the message "update took x secs" will appear in the status bar.

Fig. 2.4 Update Tool.

16. Specify the output format. WinBUGS can produce two types of output; it can open a new window for each new piece of output or it can paste all output into a single log file. To specify the output format for our binomial example, select `Output options` from the `Options` menu, and click on `log` in the `Output options` window.
17. Obtain summary statistics of the posterior distribution. To request summary statistics based on the sampled values of θ, select the `Samples` option in the `Inference` menu, and click on `stats` in the `Sample Monitor Tool` window, shown in Figure 2.3. WinBUGS will paste a table reporting various summary statistics for θ in the log file.
18. Plot the posterior distribution. To plot the posterior distribution of θ, click on `density` in the `Sample Monitor Tool` window, shown in Figure 2.3. WinBUGS will paste the posterior distribution of θ in the log file.

Figure 2.5 shows the log file that contains the results for our binomial example. The first five lines of the log file document the steps taken to specify and initialize the model. The first output item is the `Dynamic trace` plot that allows the θ variable to be monitored during sampling and is useful for diagnosing whether the chains have reached convergence. In this case, we can be reasonably confident that convergence has been achieved because the two chains, shown in different colors, are overlapping one another. The second output item is the `Node statistics` table that presents the summary statistics for θ. Among others, the table shows the mean,

the standard deviation, and the median of the sampled values of θ. The last output item is the `Kernel density` plot that shows the posterior distribution of θ.

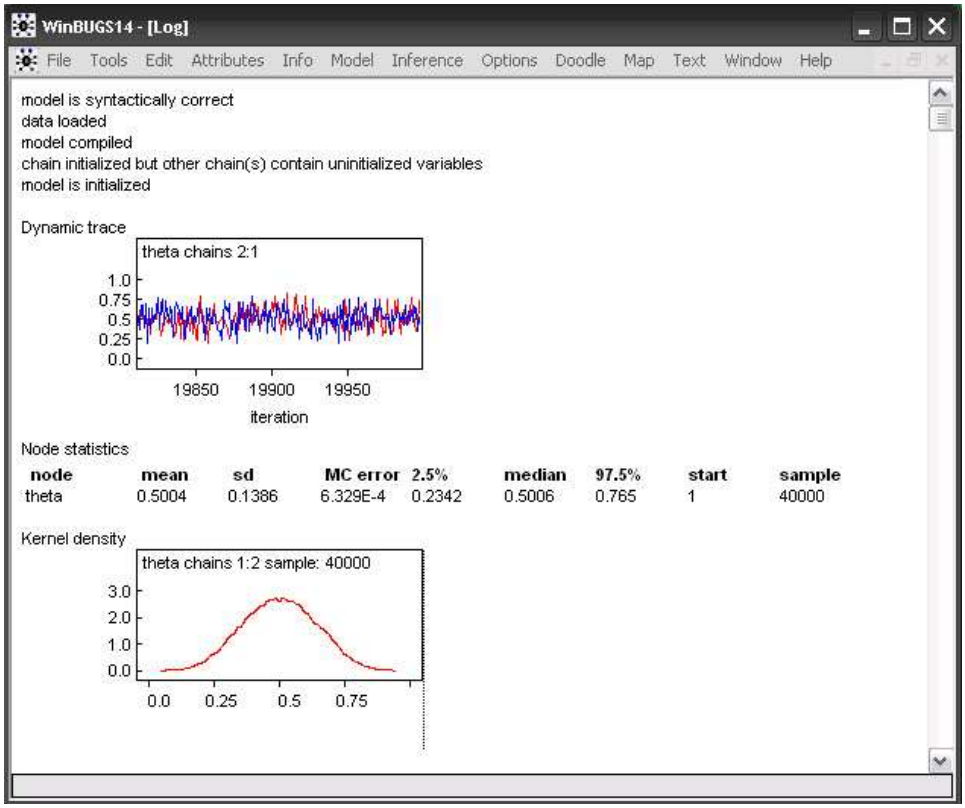


Fig. 2.5 Example of an output log file.

How did WinBUGS produce the results in Figure 2.5? The model specification file implemented the graphical model from Figure 2.1, saying that there is a rate θ with a uniform prior, that generates k successes out of n observations. The data file supplied the observed data, setting $k = 5$ and $n = 10$. WinBUGS then sampled from the posterior of the unobserved variable θ. 'Sampling' means drawing a set of values, so that the relative probability that any particular value will be sampled is proportional to the density of the posterior distribution at that value. For this example, the posterior samples for θ are a sequence of numbers like 0.5006, 0.7678, 0.3283, 0.3775, 0.4126, ... A histogram of these values is an approximation to the posterior distribution of θ.

In one sense, it would be nice to understand exactly how WinBUGS managed to generate the posterior samples. In another sense, if you are interested in building and analyzing models and data, you do not necessarily need to understand the computational basis of posterior sampling (any more than you need to know how SPSS calculates a t-test statistic). If you understand the conceptual basis that

underlies the generation of the posterior samples, you can happily build models and analyze data without worrying about the intricacies of Gibbs Sampling, Adaptive Rejection Sampling, Markov-Chain Monte-Carlo, and all the rest.[2]

Error Messages

If the syntax of your model file is incorrect or the data and starting values are incompatible with your model specification, WinBUGS will balk and produce an error message. Error messages can provide useful information when it comes to debugging your WinBUGS code.[3] The error messages are displayed in the bottom left corner of the status bar, in very small letters.

With respect to errors in the model specification, suppose, for example, that you mistakenly use the "assign" operator (`<-`) to specify the distribution of the prior on the rate parameter θ and the distribution of the observed data `k`:

```
model {
  #Prior on Rate
  theta <- dbeta(1,1)
  #Observed Counts
  k <- dbin(theta,n)
}
```

As WinBUGS requires you to use the tilde symbol "∼" to denote the distributions of the prior and the data, it will produce the following error message: "`unknown type of logical function`", as shown in Figure 2.6. As another example, suppose that you mistype the distribution of the observed counts `k`, and you mistakenly specify the distribution of `k` as follows:

```
k ~ dbon(theta,n)
```

WinBUGS will not recognize `dbon` as an existing probability distribution, and will produce the following error message: "`unknown type of probability density`", as shown in Figure 2.7.

With respect to errors in the data file, suppose that your data file contains the following data: `k` = -5 and `n` = 10. Note, however, that `k` is the number of successes in the 10 trials and it is specified to be binomially distributed. WinBUGS therefore

[2] Some people find the idea that WinBUGS looks after inference, and there is no need to understand the computational sampling routines in detail, to be a relief. Others find it deeply disturbing. For the disturbed, there are many Bayesian texts that give detailed accounts of Bayesian inference using computational sampling. Start with the summary for cognitive scientists presented by Griffiths, Kemp, and Tenenbaum (2008). Continue with the relevant chapters in the excellent book by MacKay (2003), which is freely available on the Web, and follow the more technical references from there.

[3] Although nobody ever accused WinBUGS of being user friendly in this regard. The error trap messaging, in particular, seems to have been written by the same people who did the Dead Sea Scrolls.

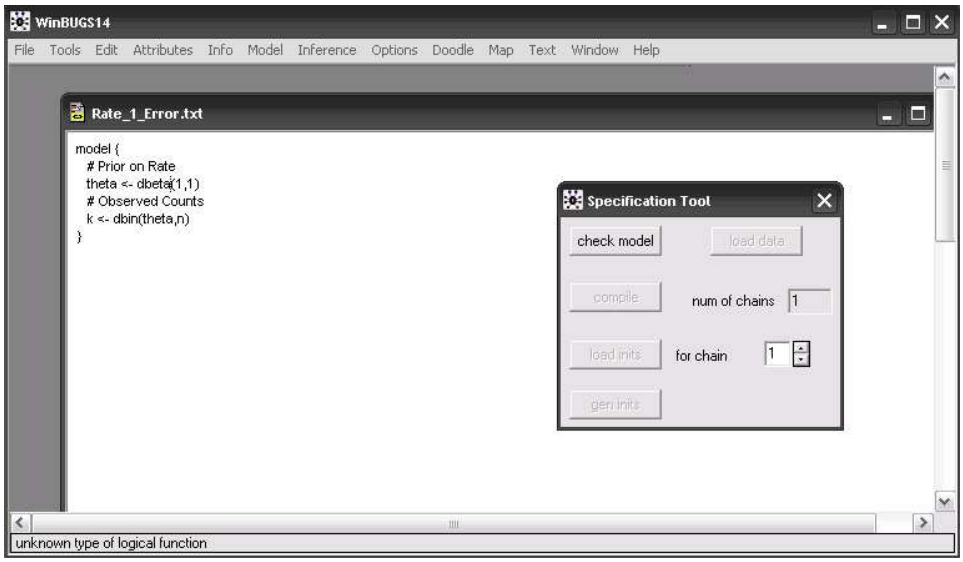


Fig. 2.6 WinBUGS error message as a result of incorrect logical operators.

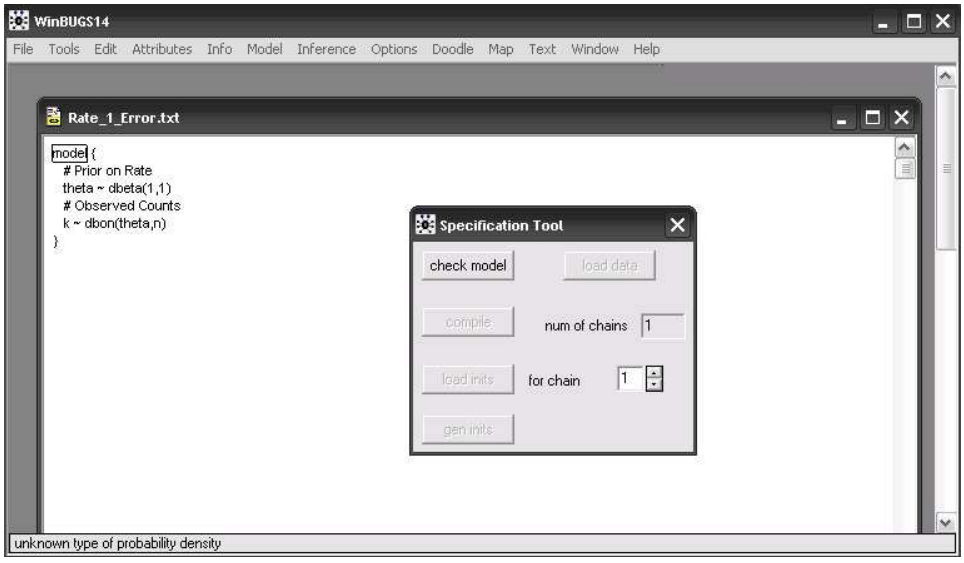


Fig. 2.7 WinBUGS error message as a result of a misspecified probability density.

expects the value of `k` to lie between 0 and `n` and it will produce the following error message: "`value of binomial k must be between zero and order of k`", as shown in Figure 2.8.

Finally, with respect to erroneous starting values, suppose that you chose 1.5 as the starting value of θ for the second chain. Because θ is the *probability* of getting 5 successes in 10 trials, WinBUGS expects the starting value for θ to

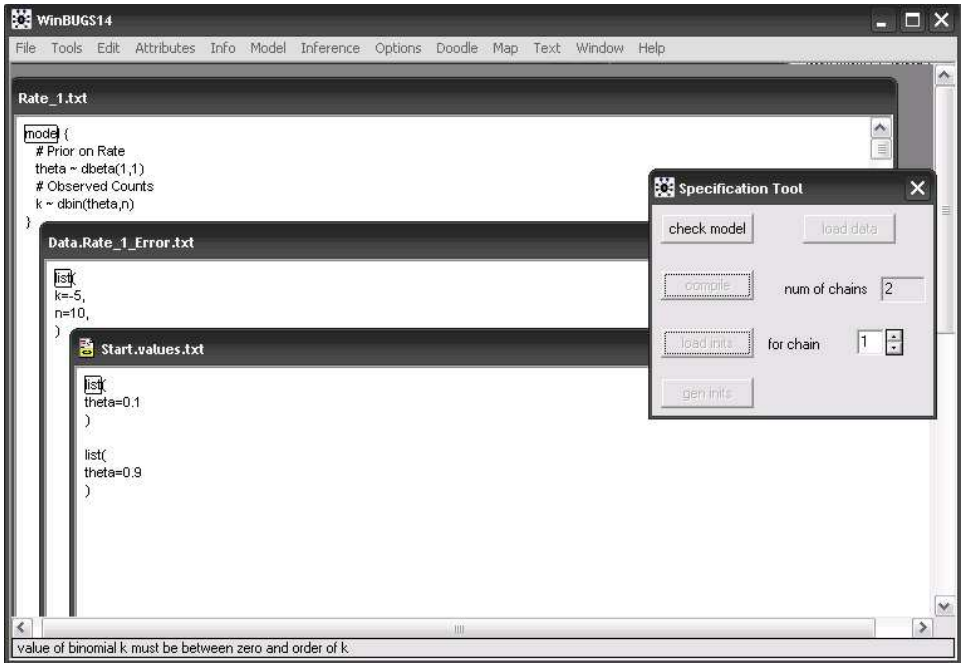


Fig. 2.8 WinBUGS error message as a result of incorrect data.

lie between 0 and 1. Therefore, specifying a value such as 1.5 produces the following error message: "`value of proportion of binomial k must be between zero and one`", as shown in Figure 2.9.

2.2.3 Using Matbugs

We will use the `matbugs` function to call the WinBUGS software from within Matlab, and to return the results of the WinBUGS sampling to a Matlab variable for further analysis. The code we are using to do this follows:

```
% Set the working directory
cd D:\WinBUGS_Book\Matlab_codes

% Data
k=5;n=10;

% WinBUGS Parameters
nchains=2; % How Many Chains?
nburnin=0; % How Many Burn-in Samples?
nsamples=2e4;  %How Many Recorded Samples?

% Assign Matlab Variables to the Observed WinBUGS Nodes
datastruct = struct('k',k,'n',n);

% Initialize Unobserved Variables
start.theta= [0.1 0.9];
```

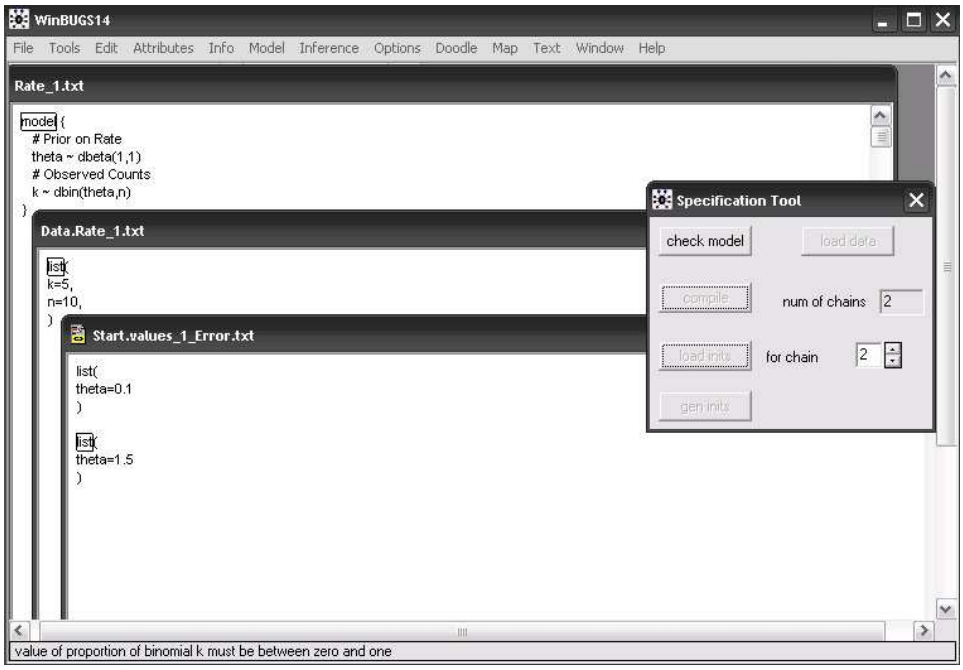


Fig. 2.9 WinBUGS error message as a result of an incorrect starting value.

```
for i=1:nchains
    S.theta = start.theta(i); % An Intial Value for the Success Rate
    init0(i) = S;
end

% Use WinBUGS to Sample
[samples, stats] = matbugs(datastruct, ...
    fullfile(pwd, 'Rate_1.txt'), ...
    'init', init0, ...
    'nChains', nchains, ...
    'view', 1, 'nburnin', nburnin, 'nsamples', nsamples, ...
    'thin', 1, 'DICstatus', 0, 'refreshrate',100, ...
    'monitorParams', {'theta'}, ...
    'Bugdir', 'C:/Program Files/WinBUGS14');
```

Some of the options in the Matbugs function control software input and output.

- `datastruct` contains the data that you want to pass from Matlab to WinBUGS.
- `fullfile` gives the name of the text file that contains the WinBUGS scripting of your graphical model (i.e., the model specification file).
- `view` controls the termination of WinBUGS. If `view` is set to 0, WinBUGS is closed automatically at the end of the sampling. If `view` is set to 1, WinBUGS remains open and it pastes the results of the sampling run in a log output file. To be able to inspect the results in WinBUGS, maximize the log output

file and scroll up to the top of the page. Note that if you subsequently want WinBUGS to return the results to Matlab, you first have to close WinBUGS.
- `refreshrate` gives the number of samples after which WinBUGS should refresh its display.
- `monitorParams` gives the list of variables that will be monitored and returned to Matlab in the `samples` variable.
- `Bugdir` gives the location of the WinBUGS software.

Other options define the values for the computational sampling parameters.

- `init` gives the starting values for the unobserved variables.
- `nChains` gives the number of chains.
- `nburnin` gives the number of 'burn-in' samples.
- `nsamples` gives the number of recorded samples that will be drawn from the posterior.
- `thin` gives the number of drawn samples between those that are recorded.
- `DICstatus` gives an option to calculate the Divergence Information Criterion (DIC) statistic. The DIC statistic is intended to be used for model selection, but is not universally accepted theoretically among Bayesian statisticians. If `DICstatus` is set to 0, the DIC statistic will not be calculated. If it is set to 1, WinBUGS will calculate the DIC statistic.

How did the WinBUGS script and Matlab work together to produce the posterior samples of θ? The WinBUGS model specification script defined the graphical model from Figure 2.1. The Matlab code supplied the observed data and the starting values for θ, and called WinBUGS. WinBUGS then sampled from the posterior of θ and returned the sampled values in the Matlab variable `samples.theta`. You can plot the histogram of these sampled values using Matlab, in the way demonstrated in the script `Rate_1.m`. It should look something like the jagged line in Figure 2.10. Because the probability of any value appearing in the sequence of posterior samples is decided by its relative posterior probability, the histogram is an approximation to the posterior distribution of θ.

Besides the sequence of posterior samples, WinBUGS also returns some useful summary statistics to Matlab. The variable `stats.mean` gives the mean of the posterior samples for each unobserved variable, which approximates its posterior expectation. This can often (but not always, as later exercises explore) be a useful point-estimate summary of all the information in the full posterior distribution. Similarly, `stats.std` gives the standard deviation of the posterior samples for each unobserved variable.

Finally, WinBUGS also returns the so-called $\hat{R}$ statistic in the `stats.Rhat` variable. This is a statistic about the sampling procedure itself, not about the posterior distribution. The $\hat{R}$ statistic is proposed by Brooks and Gelman (1997) and it gives information about convergence. The basic idea is to run two or more chains and measure the ratio of within–to between–chain variance. If this ratio is close to 1, the

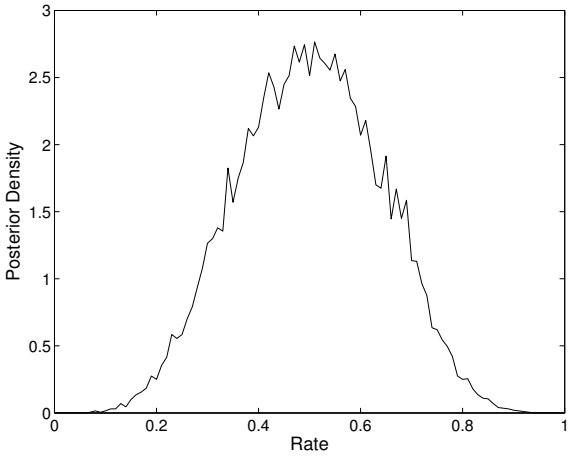


Fig. 2.10 Posterior distribution of rate θ for $k = 5$ successes out of $n = 10$ trials, based on 20,000 posterior samples.

independent sampling sequences are probably giving the same answer, and there is reason to trust the results.

2.2.4 Using R2WinBUGS

We will use the `bugs()` function in the R2WinBUGS package to call the WinBUGS software from within R, and to return the results of the WinBUGS sampling to a R variable for further analysis. The code we are using to do this follows.

```
setwd("D:/WinBUGS_Book/R_codes") #Set the working directory
library(R2WinBUGS) #Load the R2WinBUGS package
bugsdir = "C:/Program Files/WinBUGS14"

k = 5
n = 10

data    = list("k", "n")
myinits = list(
   list(theta = 0.1),
   list(theta = 0.9))

parameters = c("theta")

samples = bugs(data, inits=myinits, parameters,
               model.file ="Rate_1.txt",
               n.chains=2, n.iter=20000, n.burnin=0, n.thin=1,
               DIC=F, bugs.directory=bugsdir,
               codaPkg=F, debug=T)
```

Some of these options control software input and output.

- `data` contains the data that you want to pass from R to WinBUGS.

- `parameters` gives the list of variables that will be monitored and returned to R in the `samples` variable.
- `model.file` gives the name of the text file that contains the WinBUGS scripting of your graphical model (i.e., the model specification file). Avoid using non-alphanumeric characters (e.g., "&" and "*") in the directory and file names. Also, make sure that the name of the directory that contains the model file is not too long, otherwise WinBUGS will generate the following error message : "`incompatible copy`". If WinBUGS fails to locate a correctly specified model file, try to include the entire path in the `model.file` argument.
- `bugs.directory` gives the location of the WinBUGS software.
- `codaPkg` controls the content of the variable that is returned from WinBUGS. If `codaPkg` is set to FALSE, WinBUGS returns a variable that contains the results of the sampling run. If `codaPkg` is set to TRUE, WinBUGS returns a variable that contains the file names of the WinBUGS outputs and the corresponding paths. You can access these output files by means of the R function `read.bugs()`.
- `debug` controls the termination of WinBUGS. If `debug` is set to FALSE, WinBUGS is closed automatically at the end of the sampling. If `debug` is set to TRUE, WinBUGS remains open and it pastes the results of the sampling run in a log output file. To be able to inspect the results in WinBUGS, maximize the log output file and scroll up to the top of the page. Note that if you subsequently want WinBUGS to return the results in the R `samples` variable, you first have to close WinBUGS. In general, you will not be able to use R again until after you terminate WinBUGS.

The other options define the values for the computational sampling parameters.

- `inits` assigns starting values to the unobserved variables. If you want WinBUGS to choose these starting values for you, replace `inits=myinits` in the call to bugs with `inits=NULL`.
- `n.chains` gives the number of chains.
- `n.iter` gives the number of recorded samples that will be drawn from the posterior.
- `n.burnin` gives the number of 'burn-in' samples.
- `n.thin` gives the number of drawn samples between those that are recorded.
- `DIC` gives an option to calculate the DIC statistic. If `DIC` is set to FALSE, the DIC statistic will not be calculated. If it is set to TRUE, WinBUGS will calculate the DIC statistic.

WinBUGS returns the sampled values of θ in the R variable `samples`. You can access these values by typing `samples$sims.array`. You can also plot the histogram of these sampled values using R, in the way demonstrated in the script `Rate_1.R`). Besides the sequence of posterior samples, WinBUGS also returns some

useful statistics to R. You can access the summary statistics of the posterior samples, as well as the $\hat{R}$ statistic mentioned in the previous section by typing `samples`.

2.3 Online Help and Useful URLs

2.3.1 Online Help for WinBUGS

- The BUGS Project webpage `http://www.mrc-bsu.cam.ac.uk/bugs/weblinks/webresource.shtml` provides useful links to various articles, tutorial materials, and lecture notes about Bayesian modeling and the WinBUGS software.
- The BUGS discussion list `https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=bugs` is an online forum where WinBUGS users can exchange tips, ask questions, and share worked examples.

2.3.2 For Mac users

You can run WinBUGS on Macs using emulators, such as Darwine. As best we know, you need a Dual Core Intel based Mac and the latest stable version of Darwine to be able to use R2WinBUGS.

- The Darwine emulator is available at `www.kronenberg.org/darwine/`.
- The R2WinBUGS reference manual on the R-project webpage `cran.r-project.org/web/packages/R2WinBUGS/index.html` provides instructions on how to run R2winBUGS on Macs.
- Further information for running R2WinBUGS on Macs is available at `ggorjan.blogspot.com/2008/10/runnning-r2winbugs-on-mac.html` and `idiom.ucsd.edu/~rlevy/winbugsonmacosx.pdf`.
- Further information for running WinBUGS on Macs using a Matlab or R interface is available at `web.mit.edu/yarden/www/bayes.html` and `www.ruudwetzels.com/macbugs`.

2.3.3 For Linux users

You can run WinBUGS under Linux using emulators, such as Wine and CrossOver.

- The BUGS Project webpage provides useful links to various examples on how to run WinBUGS under Linux `www.mrc-bsu.cam.ac.uk/bugs/faqs/contents.shtml` and how to run WinBUGS using a Matlab interface `www.mrc-bsu.cam.ac.uk/bugs/winbugs/remote14.shtml`.
- The R2WinBUGS reference manual on the R-project webpage `cran.r-project.org/web/packages/R2WinBUGS/index.html` provides instructions on how to run R2winBUGS under Linux.

PART II

PARAMETER ESTIMATION

3 Inferences With Binomial Distributions

3.1 Inferring a Rate

Our first problem completes the introductory example from the "Getting Started with WinBUGS" chapter, 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 $\theta \sim \text{Beta}(1, 1)$.

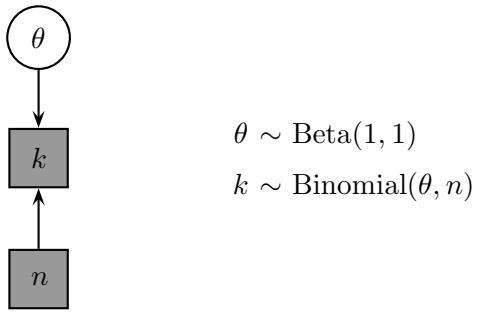


Fig. 3.1 Graphical model for inferring the rate of a binary process.

The script `Rate_1.txt` implements the graphical model in WinBUGS.

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

The code `Rate_1.m` for Matlab or `Rate_1.R` for R 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 from θ_1. The Matlab or R code also plots the

Box 3.1 Beta distributions as conjugate priors

One of the nice properties of using the $\theta \sim \mathrm{Beta}(\alpha, \beta)$ 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 "prior successes" and "prior failures", respectively. This means, using a $\theta \sim \mathrm{Beta}(3, 1)$ 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 $\theta \sim \mathrm{Beta}(3, 1)$ is the same as starting with a $\theta \sim \mathrm{Beta}(1, 1)$, and then seeing data giving two more successes (i.e., the posterior distribution in the second scenario will be 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 posterior to the prior, they are said to have a "conjugate" relationship. This is valued a lot in analytic approaches to Bayesian inference, because it makes for tractable calculations. It is not so important for that reason in computational approaches, as emphasized in this book, because sampling methods can handle easily 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.

posterior distribution of the rate θ. A histogram of the samples looks something like the jagged line in Figure 3.2.[1]

Exercises

Exercise 3.1.1 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 3.1.2 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., ten times as many) by changing the `nsamples` variable in Matlab, or the `niter` variable 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 estimate of the posterior (i.e., the smoothness of the histograms in the figures)?

[1] At least, this is what Matlab produces. The density smoothing used by default in R leads to a different visual appearance.

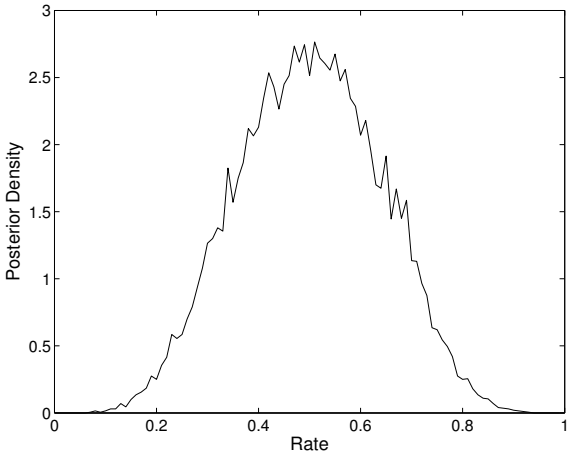


Fig. 3.2 Posterior distribution of rate θ for $k = 5$ successes out of $n = 10$ trials.

Exercise 3.1.3 Alter the data to $k = 99$ and $n = 100$, and comment on the shape of the posterior for the rate θ.

Exercise 3.1.4 Alter the data to $k = 0$ and $n = 1$, and comment on what this demonstrates about the Bayesian approach.

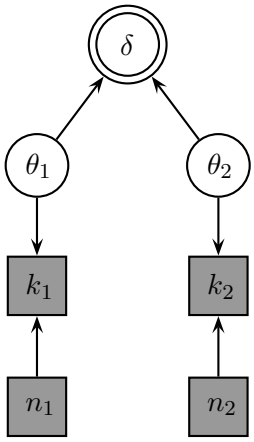
3.2 Difference Between Two Rates

Now suppose that now 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 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 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.

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



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

Fig. 3.3 Graphical model for inferring the difference in the rates of two binary process.

```
   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.

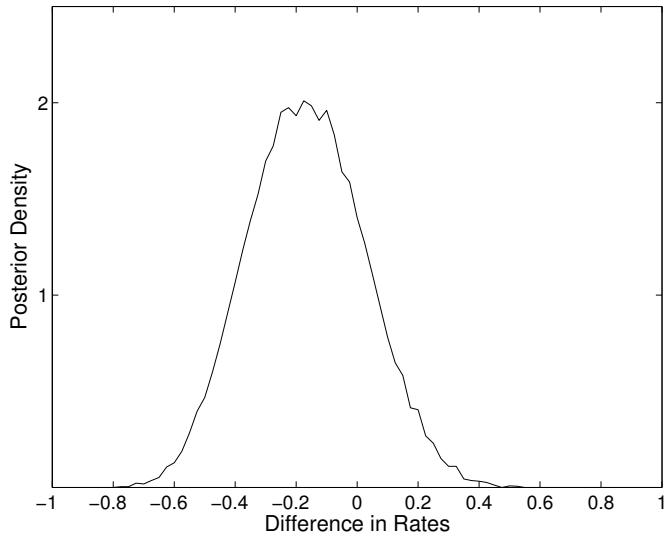


Fig. 3.4 Posterior distribution of the difference between two rates $\delta = \theta_1 - \theta_2$.

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 is the point-estimate corresponding to quadratic loss. That is, it tries to pick a single value close to the truth, with bigger deviations from the truth being punished more heavily.
- The value with maximum density in the posterior samples, approximating the posterior mode. This is known as the maximum a posteriori (MAP) estimate, and is the same as the maximum likelihood estimate (MLE) for 'flat' priors. This point-estimate corresponds to 0-1 loss, which aims to pick the single most likely value. Estimating the mode requires 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 point-estimate corresponding to 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 -0.20, 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 $k_1 = 80$, $n_1 = 100$, $k_2 = 70$, $n_2 = 100$.

Exercise 3.2.2 Try the data $k_1 = 0$, $n_1 = 1$, and $k_2 = 0$, $n_2 = 5$.

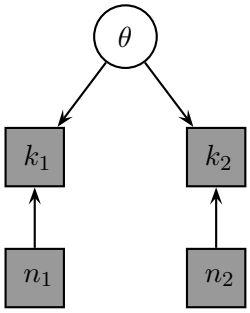
Exercise 3.2.3 In what context might different possible summaries of the posterior distribution of δ (i.e., point estimates, or credible intervals) be reasonable, and when might it be important to show the full posterior distribution?

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.

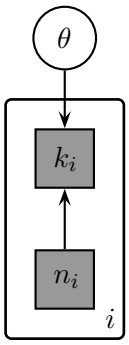
An equivalent graphical model, using plate notation, is shown in Figure 3.6. Plates are bounding rectangles that enclose independent replications of a graphical structure within a whole model. In this case, the plate encloses the two observed



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

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

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 underlying two binary processes, using plate notation.

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

```
# Inferring a Common Rate
model{
   # Observed Counts
   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 call WinBUGS to sample from the graphical model. Note that the R code sets `debug=T`, and so will wait to terminate and return the sampling information. The code also produces a plot of the posterior distribution for the common rate, as shown in Figure 3.7.

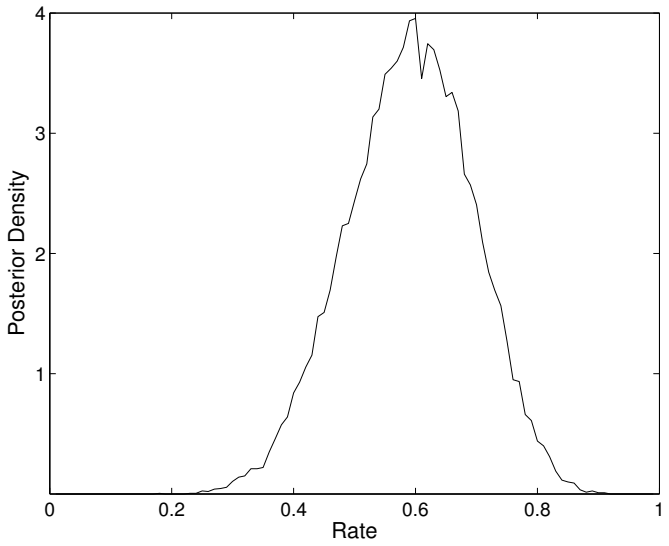


Fig. 3.7 Posterior distribution of common rate θ.

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.

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 measurement of data. The most useful part of this bridge is that data allows 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 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.

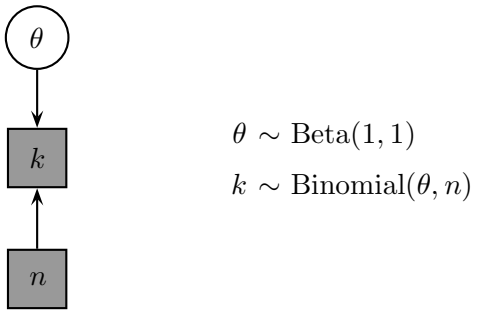


Fig. 3.8 Graphical model for inferring the rate of a binary process.

The script `Rate_4.txt` implements the graphical model in WinBUGS, and provides sampling for not just 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)
}
```

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. Posterior predictive sampling is achieved by the variable `postpredk` that samples predicted data using the same Binomial as the actual observed data.

The code `Rate_4.m` or `Rate_4.R` sets observed data with $k = 1$ successes out of $n = 10$ 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.

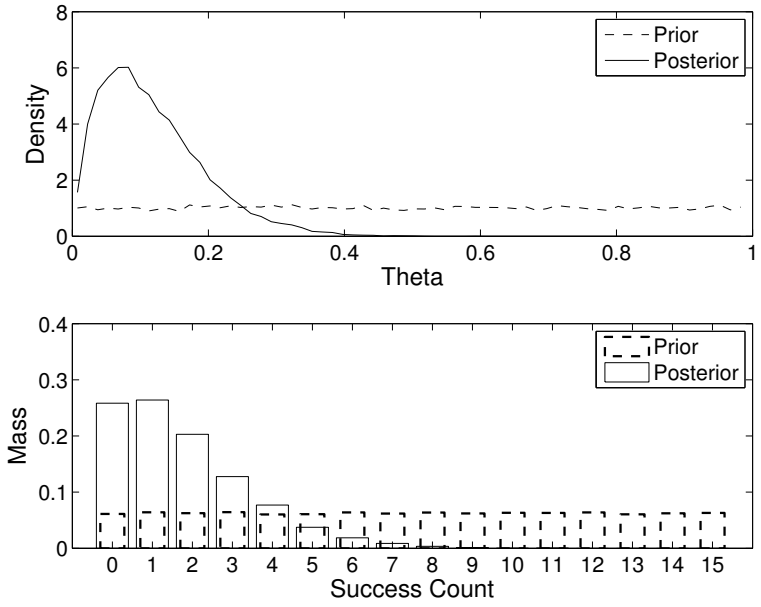


Fig. 3.9 Prior and posterior for the success rate (top panel), and prior and posterior predictive for counts of the number of successes (bottom panel), based on data giving $k = 1$ successes out of $n = 15$ trials.

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, but for any experiment where the inferred

model parameters make predictions. In the current simple Binomial setting, for example, predictive distributions could be found by a new experiment with $n' \neq n$ 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, Hester 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 confidently 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, 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 most 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.

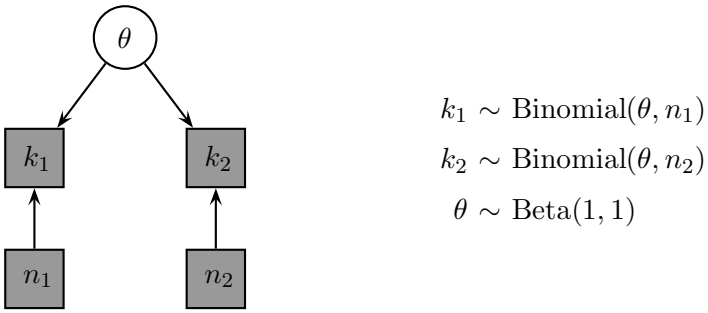


Fig. 3.10 Graphical model for inferring the common rate underlying two binary processes.

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. The code draws the posterior distribution for the rate and the posterior predictive distribution, as shown in Figure 3.11.

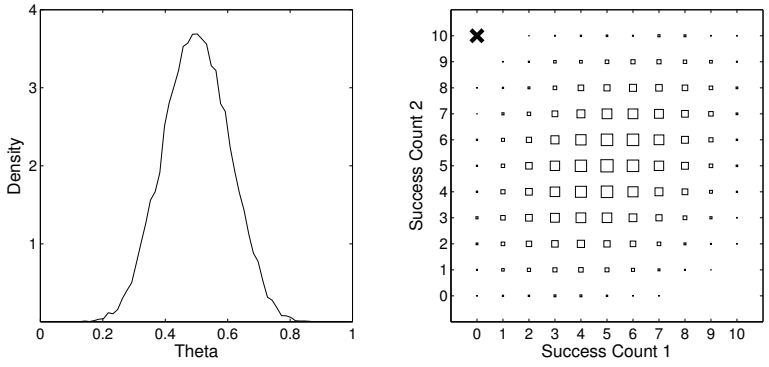


Fig. 3.11 The posterior distribution of the common rate θ for two binary processes (left panel), and the posterior predictive distribution (right panel), based on 0 and 10 successes out of 10 observations.

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 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

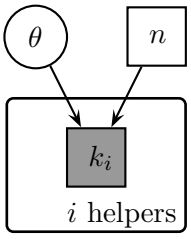
So far, we have almost always assumed that the number of successes k and number of total observations n is known, but that the underlying rate θ is unknown. This has meant that our parameter space has been one-dimensional. Everything learned from data is incorporated into a single probability distribution representing the relative likelihood 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 unknown, and so the problem is to infer both simultaneously from counts of successes k.

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 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 are able to 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.



$$k_i \sim \text{Binomial}(\theta, n)$$
$$\theta \sim \text{Beta}(1, 1)$$
$$n \sim \text{Categorical}(\tfrac{1}{n_{\max}}, \ldots, \tfrac{1}{n_{\max}})$$

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

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 Numbers 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.

It is clear that the joint posterior distributions 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.

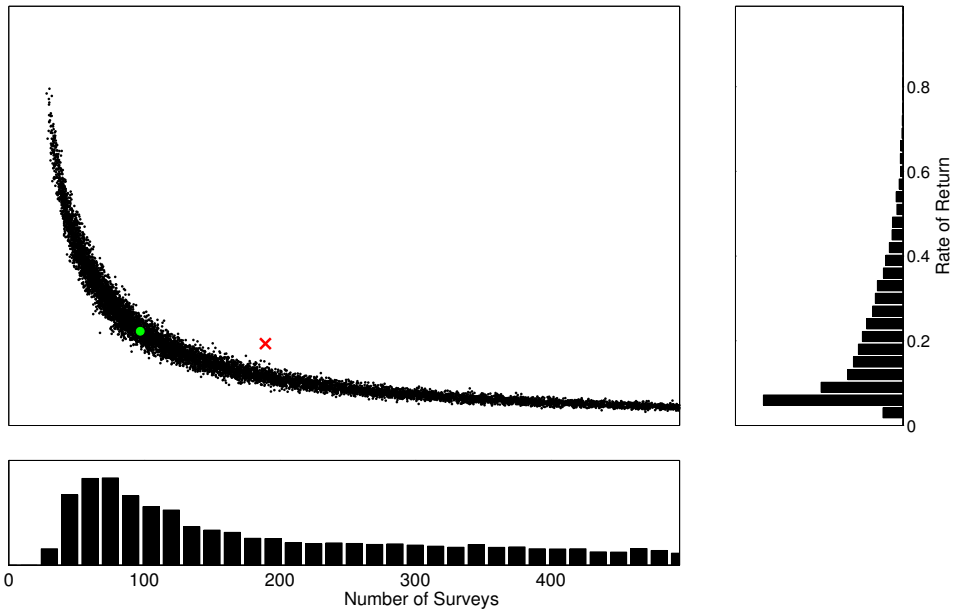


Fig. 3.13 Joint posterior distribution (scatter-plot) of the probability of return θ and the number of surveys m for observed counts $k = \{16, 18, 22, 25, 27\}$. The histograms show the marginal densities. The red cross shows the expected value of the joint posterior, and the green circle shows the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

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 distri-

butions. Imagine sampling a point from the histogram for n, and then sampling one from the histogram for θ, and plotting the two-dimensional point corresponding to these samples. Then imagine repeating this process many times. It should be clear 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 red 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 green 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 expected point, and the maximum likelihood point? If you were comfortable with the mode, are you still comfortable?[2]

Exercise 3.6.4 If you look at the sequence of samples in WinBUGS, 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 to just 500. How is the sequence of samples visually different with thinning?

[2] This example is based heavily on one we read in a book, but we have lost the reference. If you know which one, could you please let us know, so we can acknowledge it?

4 Inferences Involving Gaussian Distributions

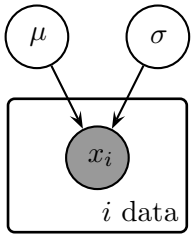
4.1 Inferring Means and Standard Deviations

One of the most common inference problems involves assuming data following a Gaussian (also known as the 'Normal', 'Central', 'Maxwellian') distribution, and inferring the mean and standard deviation of this distribution from a sample of observed independent data.

The graphical model representation for this problem is shown in Figure 4.1. The data are the n observations $x_1, \ldots, x_n$. The mean of the Gaussian is μ and the standard deviation is σ. WinBUGS parameterizes the Gaussian distribution in terms of the mean and precision, not the mean and variance or the mean and standard deviation. These are all simply related, with the variance being σ^2 and the precision being $\lambda = 1/\sigma^2$.

The prior used for μ is intended to be only weakly informative . It is a Gaussian centered on zero, but with very low precision (i.e., very large variance), and gives prior probability to a wide range of possible means for the data. When the goal is to estimate parameters, this sort of approach is relatively non-controversial.

Setting priors for standard deviations (or variances, or precisions) is trickier, and certainly more controversial. If there is any relevant information that helps put the data on scale, so that bounds can be set on reasonable possibilities for the standard deviation, then setting a uniform over that range is advocated by Gelman (2006). In this first example, we assume the data are all small enough that setting an upper bound of 10 on the standard deviation covers all the possibilities.



$$\mu \sim \text{Gaussian}(0, 0.001)$$
$$\sigma \sim \text{Uniform}(0, 10)$$
$$x_i \sim \text{Gaussian}(\mu, \tfrac{1}{\sigma^2})$$

Fig. 4.1 Graphical model for inferring the mean and standard deviation of data generated by a Gaussian distribution.

The script `Gaussian.txt` implements the graphical model in WinBUGS. Note

the conversion of the standard deviation `sigma` into the precision parameter `lambda` used to sample from a Gaussian.

```
# Inferring the Mean and Standard Deviation of a Gaussian
model{
   # Data Come From A Gaussian
   for (i in 1:n){
      x[i] ~ dnorm(mu,lambda)
   }
   # Priors
   mu ~ dnorm(0,.001)
   sigma ~ dunif(0,10)
   lambda <- 1/pow(sigma,2)
}
```

The code `Gaussian.m` or `Gaussian.R` creates some artificial data, and applies the graphical model to make inferences from data. The code does not produce a graph, or any other output. But all of the information you need to analyze the results is in the returned variable `samples`.

Exercises

Exercise 4.1.1 Try a few data sets, varying what you expect the mean and standard deviation to be, and how many data you observe.

Exercise 4.1.2 Plot the *joint* posterior of μ and σ. Interpret the plot.

Exercise 4.1.3 Suppose you knew the standard deviation of the Gaussian was 1.0, but still wanted to infer the mean from data. This is a realistic question: For example, knowing the standard deviation might amount to knowing the noise associated with measuring some psychological trait using a test instrument. The x_i values could then be repeated measures for the same person, and their mean the trait value you are trying to infer. Modify the WinBUGS script and Matlab or R code to do this. What does the revised graphical model look like?

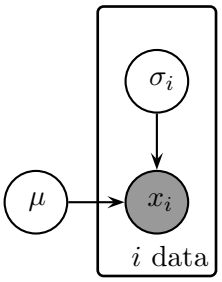
Exercise 4.1.4 Suppose you knew the mean of the Gaussian was zero, but wanted to infer the standard deviation from data. This is also a realistic question: Suppose you know the error associated with a measurement is unbiased, so its average or mean is zero, but you are unsure how much noise there is in the instrument. Inferring the standard deviation is then a sensible way to infer the noisiness of the instrument. Once again, modify the WinBUGS script and Matlab or R code to do this. Once again, what does the revised graphical model look like?

4.2 The Seven Scientists

This problem is from MacKay (2003, p. 309) where it is (among other things) treated to a Bayesian solution, but not quite using a graphical modeling approach, nor relying on computational sampling methods.

Seven scientists with wildly-differing experimental skills all make a measurement of the same quantity. They get the answers $x = \{-27.020, 3.570, 8.191, 9.898, 9.603, 9.945, 10.056\}$. Intuitively, it seems clear that the first two scientists are pretty inept measurers, and that the true value of the quantity is probably just a bit below 10. The main problem is to find the posterior distribution over the measured quantity, telling us what we can infer from the measurement. A secondary problem is to infer something about the measurement skills of the seven scientists.

The graphical model for one (good) way of solving this problem is shown in Figure 4.2. The assumption is that all the scientists have measurements that follow a Gaussian distribution, but with different standard deviations. However, because they are all measuring the same quantity, each Gaussian has the same mean, it is just the standard deviation that differs.



$$\mu \sim \text{Gaussian}(0, 0.001)$$
$$\sigma_i \sim \text{InvSqrtGamma}(0.001, 0.001)$$
$$x_i \sim \text{Gaussian}(\mu, \tfrac{1}{\sigma_i^2})$$

Fig. 4.2 Graphical model for the seven scientists problem.

Notice the different approach to setting priors about the standard deviations used in this example. This approach has a theoretical basis in scale invariance arguments (i.e., choosing to set a prior so that changing the measurement scale of the data does not affect inference). The invariant prior turns is improper (i.e., the area under the curve is unbounded), meaning it is not really a distribution, but the limit of a sequence of distributions (see Jaynes, 2003). WinBUGS requires proper distributions always be used, and so the InvSqrtGamma$(.001, .001)$ is intended as a proper approximation to the theoretically-motivated improper prior. This raises the issue of whether inference is sensitive to the essentially arbitrary value 0.001. Gelman (2006) raises some other challenges to this approach. But, it is still worth knowing about.

The script `SevenScientists.txt` code implements the graphical model in Figure 4.2 in WinBUGS.

```
# The Seven Scientists
model{
   # Data Come From Gaussians With Common Mean But Different Precisions
   for (i in 1:n){
      x[i] ~ dnorm(mu,lambda[i])
   }
   # Priors
   mu ~ dnorm(0,.001)
   for (i in 1:n){
```

```
      lambda[i] ~ dgamma(.001,.001)
      sigma[i] <- 1/sqrt(lambda[i])
   }
}
```

Notice that the Inverse-SquareRoot-Gamma prior distribution is implemented by first setting a prior for the precision, $\lambda \sim \text{Gamma}(.001, .001)$ and then re-parameterization to the standard deviation.

The code `SevenScientists.m` or `SevenScientists.R` applies the seven scientist data to the graphical model.

Exercises

Exercise 4.2.1 Draw posterior samples using the Matlab or R code, and reach conclusions about the value of the measured quantity, and about the accuracies of the seven scientists.

Exercise 4.2.2 Change the graphical model in Figure 4.2 to use a uniform prior over the standard deviation, as was done in Figure 4.1. Experiment with the effect the upper bound of this uniform prior has on inference.

4.3 Repeated Measurement of IQ

In this example, we consider how to estimate the IQ of a set of people, each of whom have done multiple IQ tests. The data are the measures x_{ij} for the $i = 1, \ldots, n$ people and their $j = 1, \ldots, m$ repeated test scores.

We assume that the differences in repeated test scores are Gaussian error with zero mean, but some unknown precision. The mean of the Gaussian of a person's test scores corresponds to their IQ measure. This will be different for each person. The standard deviation of the Gaussians corresponds to the accuracy of the testing instruments in measuring the one underlying IQ value. We assume this is the same for every person, since it is conceived as a property of the tests themselves.

The graphical model for this problem is shown in Figure 4.3. Because we know quite a bit about the IQ scale, it makes sense to set priors for the mean and standard deviation using this knowledge. Our first attempts to set priors (these are re-visited in the exercises) simply assume the actual IQ values are equally likely to be anywhere between 0 and 300, and standard deviations are anywhere between 0 and 100.

The script `IQ.txt` implements the graphical model in WinBUGS.

```
# Repeated Measures of IQ
model{
   # Data Come From Gaussians With Different Means But Common Precision
   for (i in 1:n){
      for (j in 1:m){
         x[i,j] ~ dnorm(mu[i],lambda)
```

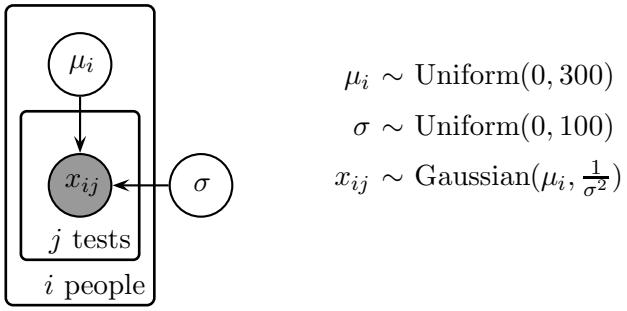


Fig. 4.3 Graphical model for inferring the IQ from repeated measures.

```
      }
   }
   # Priors
   sigma ~ dunif(0,100)
   lambda <- 1/pow(sigma,2)
   for (i in 1:n){
      mu[i] ~ dunif(0,300)
   }
}
```

The code `IQ.m` or `IQ.R` creates a data set corresponding to there being three people, with test scores of $(90, 95, 100)$, $(105, 110, 115)$, and $(150, 155, 160)$, and applies the graphical model.

Exercises

Exercise 4.3.1 Use the posterior distribution for each person's μ_i to estimate their IQ. What can we say about the precision of the IQ test?

Exercise 4.3.2 Now, use a more realistic prior assumption for the μ_i means. Theoretically, IQ distributions should have a mean of 100, and a standard deviation of 15. This corresponds to having a prior of `mu[i] ~ dnorm(100,.0044)`, instead of `mu[i] ~ dunif(0,300)`, because $1/15^2 = 0.0044$. Make this change in the WinBUGS script, and re-run the inference. How do the estimates of IQ given by the means change? Why?

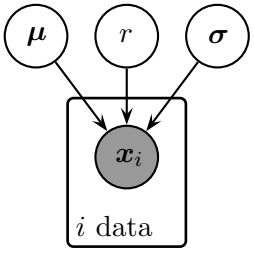
Exercise 4.3.3 Repeat both of the above stages (i.e., using both priors on μ_i) with a new, but closely related, data set that has scores of $(94, 95, 96)$, $(109, 110, 111)$, and $(154, 155, 156)$. How do the different prior assumptions affect IQ estimation for these data. Why does it not follow the same pattern as the previous data?

5 Some Examples Of Data Analysis

5.1 Pearson Correlation

The Pearson-product moment correlation coefficient, usually denoted r, is a very widely-used measure of the relationship between two variables. It ranges between +1, indicating a perfect positive linear relationship, to 0, indicating no linear relationship, to −1 indicating a perfect negative relationship. Usually the correlation r is reported as a single point estimate, perhaps together with a frequentist significance test.

But, rather than just having a single number to measure the correlation, it would be nice to have a posterior distribution for r, saying how likely each possible level of correlation was. There are frequentist confidence interval methods that try to do this, as well as various analytic Bayesian results based on asymptotic approximations (e.g., Donner & Wells, 1986). An advantage of using a computational approach is the flexibility in the assumptions that can be made. It is possible to set up a graphical model that allows inferences about the correlation coefficient for any data generating process and set of prior assumptions about the correlation.



$$
\begin{aligned}
\mu_1, \mu_2 &\sim \text{Gaussian}(0, 0.001) \\
\sigma_1, \sigma_2 &\sim \text{InvSqrtGamma}(0.001, 0.001) \\
r &\sim \text{Uniform}(-1, 1) \\
\boldsymbol{x}_i &\sim \text{MvGaussian}\left((\mu_1, \mu_2), \begin{bmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}^{-1} \right)
\end{aligned}
$$

Fig. 5.1 Graphical model for inferring a correlation coefficient.

One graphical model for doing this is shown in Figure 5.1. The observed data take the form $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ for the ith observation, and, following the theory behind the correlation coefficient, are modeled as draws from a multivariate Gaussian distribution. The parameters of this distribution are the means and standard deviations of the two dimensions, and the correlation coefficient that links them.

In Figure 5.1, the variances are given the approximations to non-informative discussed earlier. The correlation coefficient itself is given a uniform prior over its possible range. All of these choices would be easily modified, with one obvious possible change being to give the prior for the correlation more density around 0.

The script `Correlation_1.txt` implements the graphical model in WinBUGS.

```
# Pearson Correlation
model {
   # Data
   for (i in 1:n){
      x[i,1:2] ~ dmnorm(mu[],TI[,])
   }
   # Priors
   mu[1] ~ dnorm(0,.001)
   mu[2] ~ dnorm(0,.001)
   lambda[1] ~ dgamma(.001,.001)
   lambda[2] ~ dgamma(.001,.001)
   r ~ dunif(-1,1)
   # Reparameterization
   sigma[1] <- 1/sqrt(lambda[1])
   sigma[2] <- 1/sqrt(lambda[2])
   T[1,1] <- 1/lambda[1]
   T[1,2] <- r*sigma[1]*sigma[2]
   T[2,1] <- r*sigma[1]*sigma[2]
   T[2,2] <- 1/lambda[2]
   TI[1:2,1:2] <- inverse(T[1:2,1:2])
}
```

The code `Correlation_1.m` or `Correlation_1.R` includes two data sets. Both involve fabricated data comparing response times (on the x-axis) with IQ measures (on the y-axis), looking for a correlation between simple measures of decision-making and general intelligence.

For the first data set in the Matlab and R code, the results shown in Figure 5.2 are produced. The left panel shows a scatter-plot of the raw data. The right panel shows the posterior distribution of r, together with the standard frequentist point-estimate.

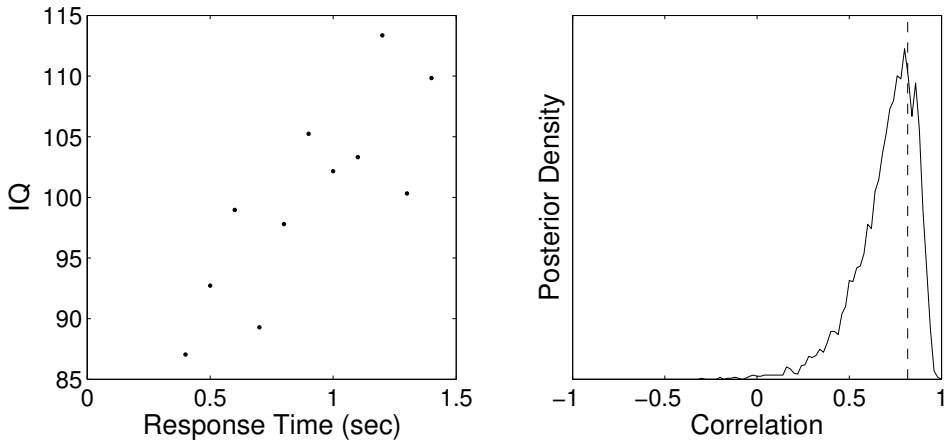


Fig. 5.2 Data (left panel) and posterior distribution for correlation coefficient (right panel). The broken line shows the frequentist point-estimate.

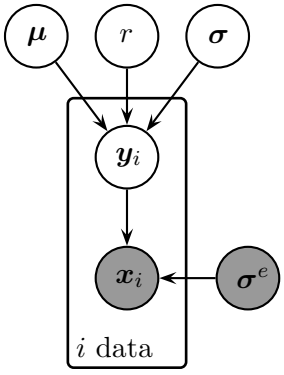
Exercises

Exercise 5.1.1 The second data set in the Matlab and R code is just the first data set from Figure 5.2 repeated twice. Set `dataset=2` to consider these repeated data, and interpret the differences in the posterior distributions for r.

Exercise 5.1.2 The current graphical model assumes that the values from the two variables—the $\boldsymbol{x}_i = (x_{i1}, x_{i2})$—are observed with perfect accuracy. When might this be a problematic assumption? How could the current approach be extended to make more realistic assumptions?

5.2 Pearson Correlation With Uncertainty

We now tackle the problem asked by the last question in the previous section, and consider the correlations when there is uncertainty about the exact values of variables. While it might be plausible that response time could be measured very accurately, the measurement of IQ seems likely to be less precise. This uncertainty should be incorporated in an assessment of the correlation between the variables.



$$
\begin{aligned}
\mu_i &\sim \text{Gaussian}(0, 0.001) \\
\sigma_i &\sim \text{InvSqrtGamma}(0.001, 0.001) \\
r &\sim \text{Uniform}(-1, 1) \\
y_i &\sim \text{MvGaussian}\left((\mu_1, \mu_2), \begin{bmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}^{-1} \right) \\
x_{ij} &\sim \text{Gaussian}(y_{ij}, \sigma_j^e)
\end{aligned}
$$

Fig. 5.3 Graphical model for inferring a correlation coefficient, when there is uncertainty inherent in the measurements.

A simple approach for including this uncertainty is adopted by the graphical model in Figure 5.3. The observed data still take the form $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ for the ith person's response time and IQ measure. But these observations are now draws from a Gaussian distribution, centered on the unobserved 'true' response time and IQ of that person, denoted $\boldsymbol{y}_i = (y_{i1}, y_{i2})$. These true values are then modeled as the $\boldsymbol{x}$ were in the previous model in Figure 5.1, as draws from the Multivariate Gaussian distribution corresponding the correlation.

The precision of the measurements is captured by the standard deviations $\boldsymbol{\sigma}^e = (\sigma_1^e, \sigma_2^e)$ of the Gaussian draws for the observed data, $x_{ij} \sim \text{Gaussian}\big(y_{ij}, \sigma_j^e\big)$. The graphical model in Figure 5.3 assumes that the standard deviations are known.

The script `Correlation_2.txt` implements the graphical model shown in WinBUGS.

```
# Pearson Correlation With Uncertainty in Measurement
model {
   # Data
   for (i in 1:n){
      y[i,1:2] ~ dmnorm(mu[],TI[,])
          for (j in 1:2){
             x[i,j] ~ dnorm(y[i,j],lambdapoints[j])
          }
   }
   # Priors
   mu[1] ~ dnorm(0,.001)
   mu[2] ~ dnorm(0,.001)
   lambda[1] ~ dgamma(.001,.001)
   lambda[2] ~ dgamma(.001,.001)
   r ~ dunif(-1,1)
   # Reparameterization
   sigma[1] <- 1/sqrt(lambda[1])
   sigma[2] <- 1/sqrt(lambda[2])
   T[1,1] <- 1/lambda[1]
   T[1,2] <- r*sigma[1]*sigma[2]
   T[2,1] <- r*sigma[1]*sigma[2]
   T[2,2] <- 1/lambda[2]
   TI[1:2,1:2] <- inverse(T[1:2,1:2])
}
```

The code `Correlation_2.m` uses the same data set as in the previous section, but has different data sets for different assumptions about the uncertainty in measurement. In the first analysis, these are set to the values $\sigma_1^e = .03$ for response times (which seem likely to be measured accurately) and $\sigma_2^e = 1$ for IQ (which seems near the smallest plausible value). The results of this assumption using the model are shown in Figure 5.4. The left panel shows a scatterplot of the raw data, together with error bars representing the uncertainty quantified by the observed standard deviations. The right panel shows the posterior distribution of r, together with the standard frequentist point estimate.

Exercises

Exercise 5.2.1 Compare the results obtained in Figure 5.4 with those obtained earlier using the same data in Figure 5.2, for the model without any account of uncertainty in measurement.

Exercise 5.2.2 Generate results for the second data set, which changes $\sigma_2^e = 10$ for the IQ measurement. Compare these results with those obtained assuming $\sigma_2^e = 1$.

Exercise 5.2.3 The graphical model in Figure 5.3 assumes the uncertainty for each variable is known. How could this assumption be relaxed to the case where the uncertainty is unknown?

Exercise 5.2.4 The graphical model in Figure 5.3 assumes the uncertainty for each variable is the same for all observations. How could this assumption

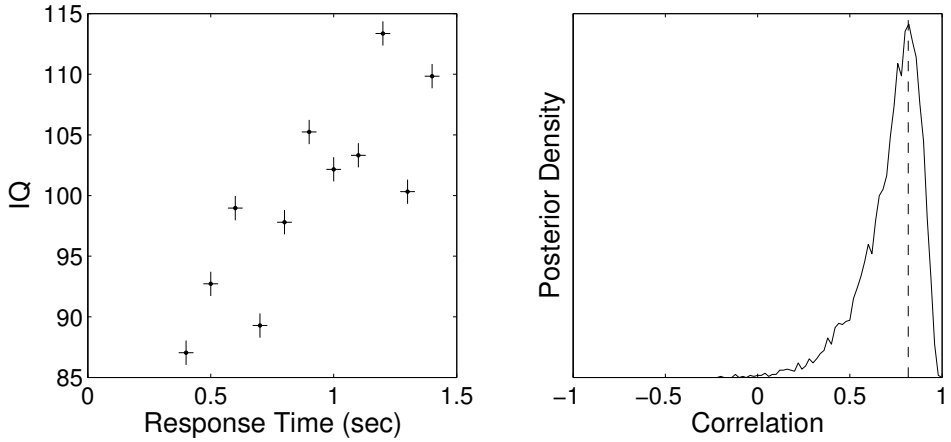


Fig. 5.4 Data (left panel), including error bars showing uncertainty in measurement, and posterior distribution for correlation coefficient (right panel). The broken line shows the frequentist point-estimate.

be relaxed to the case where, for examples, extreme IQs are less accurately measured than IQs in the middle of the standard distribution?

5.3 The Kappa Coefficient of Agreement

An important statistical inference problem in a range of physical, biological, behavioral and social sciences is to decide how well one decision-making method agrees with another. An interesting special case considers only binary decisions, and views one of the decision-making methods as giving objectively true decisions to which the other aspires. This problem occurs often in medicine, when cheap or easily administered methods for diagnosis are evaluated in terms of how well they agree with a more expensive or complicated 'gold standard' method.

For this problem, when both decision-making methods make n independent assessments, the data D take the form of four counts: a observations where both methods decide 'one', b observations where the objective method decides 'one' but the surrogate method decides 'zero', c observations where the objective method decides 'zero' but the surrogate method decides 'one', and d observations where both methods decide 'zero', with $n = a + b + c + d$.

A variety of orthodox statistical measures have been proposed for assessing agreement using these data (but see Basu, Banerjee, & Sen, 2000, for a Bayesian approach). Useful reviews are provided by Agresti (1992), Banerjee, Capozzoli, McSweeney, and Sinha (1999), Fleiss, Levin, and Paik (2003), Kraemer (1992), Kraemer, Periyakoil, and Noda (2004) and Shrout (1998). Of all the measures, however, it is reasonable to argue that the conclusion of Uebersax (1987) that "the kappa

coefficient is generally regarded as the statistic of choice for measuring agreement" (p. 140) remains true.

Cohen's (1960) kappa statistic estimates the level of observed agreement

$$p_o = \frac{a+d}{n}$$

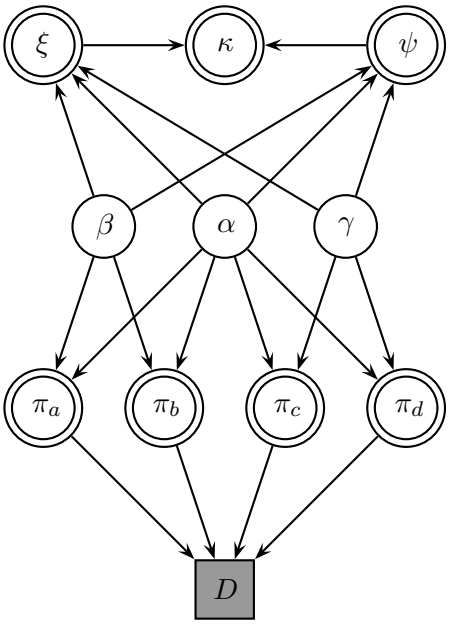
relative to the agreement that would be expected by chance alone (i.e., the overall probability for the first method to decide 'one' times the overall probability for the second method to decide 'one', and added to this the overall probability for the second method to decide 'zero' times the overall probability for the first method to decide 'zero')

$$p_e = \frac{(a+b)(a+c)+(b+d)(c+d)}{n^2},$$

and is given by

$$\kappa = \frac{p_o - p_e}{1 - p_e}.$$

Kappa lies on a scale of -1 to $+1$, with values below 0.4 often interpreted as "poor" agreement beyond chance, values between 0.4 and 0.75 interpreted as "fair to good" agreement beyond chance, and values above 0.75 interpreted as "excellent" agreement beyond chance (Landis & Koch, 1977). The key insight of kappa as a measure of agreement is its correction for chance agreement.



$$\begin{aligned}
\kappa &\leftarrow (\xi - \psi) / (1 - \psi) \\
\xi &\leftarrow \alpha\beta + (1-\alpha)\gamma \\
\psi &\leftarrow (\pi_a + \pi_b)(\pi_a + \pi_c) + (\pi_b + \pi_d)(\pi_c + \pi_d) \\
\alpha, \beta, \gamma &\sim \text{Beta}(1, 1) \\
\pi_a &\leftarrow \alpha\beta \\
\pi_b &\leftarrow \alpha(1-\beta) \\
\pi_c &\leftarrow (1-\alpha)(1-\gamma) \\
\pi_d &\leftarrow (1-\alpha)\gamma \\
D &\sim \text{Multinomial}([\pi_a, \pi_b, \pi_c, \pi_d], n)
\end{aligned}$$

Fig. 5.5 Graphical model for inferring the kappa coefficient of agreement.

The graphical model for a Bayesian version of kappa is shown in Figure 5.5. The

key latent variables are α, β and γ. The rate α is the rate at which the gold standard method decides 'one'. This means $(1-\alpha)$ is the rate at which the gold standard method decides 'zero'. The rate β is the rate at which the surrogate method decides 'one' when the gold standard also decides 'one'. The rate γ is the rate at which the surrogate method decides 'zero' when the gold standard decides 'zero'. The best way to interpret β and γ is that they are the rate of agreement of the surrogate method with the gold standard, for the 'one' and 'zero' decisions respectively.

Using the rates α, β and γ, it is possible to calculate the probabilities that both methods will decide 'one', $\pi_a = \alpha\beta$, that the gold standard will decide 'one' but the surrogate will decide zero, $\pi_b = \alpha\,(1-\beta)$, the gold standard will decide 'zero' but the surrogate will decide 'one', $\pi_c = (1-\alpha)\,(1-\gamma)$, and that both methods will decide 'zero', $\pi_d = (1-\alpha)\,\gamma$.

These probabilities, in turn, describe how the observed data, D, made up of the counts a, b, c, and d, are generated. They come from a Multinomial distribution with n trials, where on each trial there is a π_a probability of generating an a count, π_b probability for a b count, and so on.

So, observing the data D allows inferences to be made about the key rates α, β and γ. The remaining variables in the graphical model in Figure 5.5 just re-express these rates in the way needed to provide an analogue to the kappa measure of chance corrected agreement. The ξ variable measures the observed rate of agreement, which is $\xi = \alpha\beta + (1-\alpha)\,\gamma$. The ψ variable measures the rate of agreement that would occur by chance, which is $\psi = (\pi_a + \pi_b)\,(\pi_a + \pi_c) + (\pi_b + \pi_d)\,(\pi_c + \pi_d)$, and could be expressed in terms of α, β and γ. Finally κ is the chance corrected measure of agreement on the -1 to $+1$ scale, given by $\kappa = (\xi - \psi)\,/\,(1-\psi)$.

The script `Kappa.txt` implements the graphical model in WinBUGS.

```
# Kappa Coefficient of Agreement
model {
   # Underlying Rates
   # Rate objective method decides 'one'
   alpha ~ dbeta(1,1)
   # Rate surrogate method decides 'one' when objective method decides 'one'
   beta ~ dbeta(1,1)
   # Rate surrogate method decides 'zero' when objective method decides 'zero'
   gamma ~ dbeta(1,1)
   # Probabilities For Each Count
   pi[1] <- alpha*beta
   pi[2] <- alpha*(1-beta)
   pi[3] <- (1-alpha)*(1-gamma)
   pi[4] <- (1-alpha)*gamma
   # Count Data
   d[1:4] ~ dmulti(pi[],n)
   # Derived Measures
   # Rate surrogate method agrees with the objective method
   xi <- alpha*beta+(1-alpha)*gamma
   # Rate of chance agreement
   psi <- (pi[1]+pi[2])*(pi[1]+pi[3])+(pi[2]+pi[4])*(pi[3]+pi[4])
   # Chance corrected agreement
   kappa <- (xi-psi)/(1-psi)
}
```

The code `Kappa.m` or `Kappa.R` includes several data sets, described in the Exercises below, to WinBUGS to sample from the graphical model.

Exercises

Exercise 5.3.1 *Influenza Clinical Trial* Poehling, Griffin, and Dittus (2002) reported data evaluating a rapid bedside test for influenza using a sample of 233 children hospitalized with fever or respitory symptoms. Of the 18 children known to have influenza, the surrogate method identified 14 and missed 4. Of the 215 children known not to have influenza, the surrogate method correctly rejected 210 but falsely identified 5. These data correspond to $a = 14$, $b = 4$, $c = 5$, and $d = 210$. Examine the posterior distributions of the interesting variables, and reach a scientific conclusion. That is, pretend you are a consultant for the clinical trial. What would your two- or three-sentence 'take home message' conclusion be to your customers?

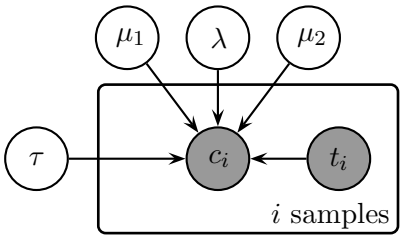
Exercise 5.3.2 *Hearing Loss Assessment Trial* Grant (1974) reported data from a screening of a pre-school population intended to assess the adequacy of a school nurse assessment of hearing loss in relation to expert assessment. Of those children assessed as having hearing loss by the expert, 20 were correctly identified by the nurse and 7 were missed. Of those assessed as not having hearing loss by the expert, 417 were correctly diagnosed by the nurse but 103 were incorrectly diagnosed as having hearing loss. These data correspond to $a = 20$, $b = 7$, $c = 103$, $d = 417$. Once again, examine the posterior distributions of the interesting variables, and reach a scientific conclusion. Once again, what would your two- or three-sentence 'take home message' conclusion be to your customers?

Exercise 5.3.3 *Rare Disease* Suppose you are testing a cheap instrument for detecting a rare medical condition. After 170 patients have been screened, the test results shower 157 did not have the condition, but 13 did. The expensive ground truth assessment subsequently revealed that, in fact, none of the patients had the condition. These data correspond to $a = 0$, $b = 0$, $c = 13$, $d = 157$. Apply the kappa graphical model to these data, and reach a conclusion about the usefulness of the cheap instrument. What is special about this data set, and what does it demonstrate about the Bayesian approach?

5.4 Change Detection in Time Series Data

This case study involves near-infrared spectrographic data, in the form of oxygenated hemoglobin counts of frontal lobe activity during an attention task in Attention Deficit Hyperactivity Disorder (ADHD) adults. This gets up the quinella of sounding neuro and clinical, and so must be impressive and eminently fundable work.

The interesting modeling problem is that a change is expected in the time series of counts because of the attention task. The statistical problem is to identify the change. To do this, we are going to make a number of strong assumptions. In particular, we will assume that the counts come from a Gaussian distribution that always has the same variance, but changes its mean at one specific point in time. The main interest is therefore in making an inference about this change point.



$$\mu_1, \mu_2 \sim \text{Gaussian}(0, 0.001)$$
$$\lambda \sim \text{Gamma}(0.001, 0.001)$$
$$\tau \sim \text{Uniform}(0, t_{\max})$$
$$c_i \sim \begin{cases} \text{Gaussian}(\mu_1, \lambda) & \text{if } t_i < \tau \\ \text{Gaussian}(\mu_2, \lambda) & \text{if } t_i \geq \tau \end{cases}$$

Fig. 5.6 Graphical model for detecting a single change-point in time series.

Figure 5.6 presents a graphical model for detecting the change point. The observed data are the counts c_i at time t_i for the ith sample. The unobserved variable τ is the time at which the change happens, and so controls whether the counts have mean μ_1 or μ_2. A uniform prior over the full range of possible times is assumed for the change point, and generic weakly informative priors are given to the means and the precision.

The script `ChangeDetection.txt` implements this graphical model in WinBUGS.

```
# Change Detection
model {
    # Data Come From A Gaussian
    for (i in 1:n){
       c[i] ~ dnorm(mu[z1[i]],lambda)
    }
    # Group Means
    mu[1] ~ dnorm(0,.001)
    mu[2] ~ dnorm(0,.001)
    # Common Precision
    lambda ~ dgamma(.001,.001)
    sigma  <- 1/sqrt(lambda)
    # Which Side is Time of Change Point?
    for (i in 1:n){
        z[i] <- step(t[i]-tau)
        z1[i] <- z[i]+1
    }
    # Prior On Change Point
    tau ~ dunif(0,tmax)
}
```

Note the use of the `step` function. This function returns 1 if its argument is greater than or equal to zero, and 0 otherwise. The `z1` variable, however, serves as an indicator variable for `mu`, and therefore it needs to take on values 1 and 2. This is

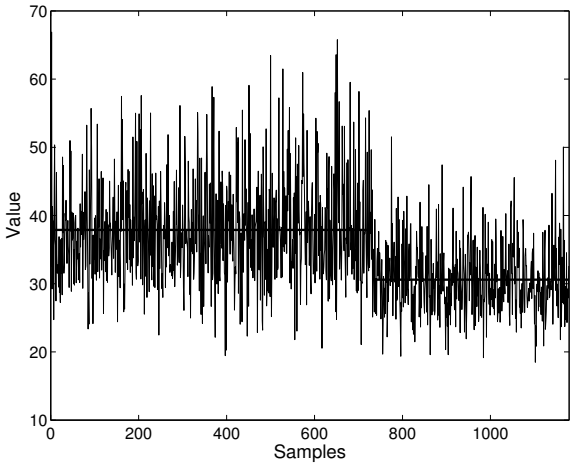


Fig. 5.7 Identification of change-point in time series data.

the reason `z` is transformed to `z1`. Study this code and make sure you understand what the `step` function accomplishes in this example.

The code `ChangeDetection.m` or `ChangeDetection.R` applies the model to the near-infrared spectrographic data. Uniform sampling is assumed, so that $t = 1, \ldots, 1778$.

The code produces a simple analysis, finding the mean of the posteriors for τ, μ_1 and μ_2, and using these summary points to overlay the inferences over the raw data. The result look something like Figure 5.7.

Exercises

Exercise 5.4.1 Draw the posterior distributions for the change-point, the means, and the common standard deviation.

Exercise 5.4.2 Figure 5.7 shows the mean of the posterior distribution for the change-point (this is the point in time where the two horizontal lines meet). Can you think of a situation in which such a plotting procedure can be misleading?

Exercise 5.4.3 Imagine that you apply this model to a data set that has two change-points instead of one. What could happen?

5.5 Censored Data

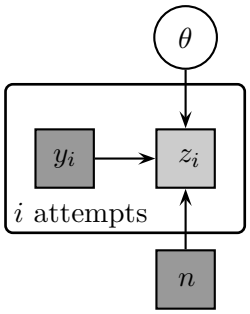
Starting 13 April 2005, Cha Sa-soon, a 68-year old grandmother living in Jeonju, South Korea, repeatedly tried to pass the written exam for a driving license. In South Korea, this exam features 50 four-choice questions. In order to pass, a score of at least 60 points out of a maximum of 100. Accordingly, we assume that each

correct answer is worth 2 points, so that in order to pass one needs to answer at least 30 questions correctly.

What makes Cha Sa-soon special is that she failed to pass the test on 949 consecutive occasions, spending the equivalent of 4,200 US dollars on application fees. In her last, 950th attempt, Cha Sa-soon scored the required minimum of 30 correct questions and finally obtained her written exam. After her 775th failure, in February 2009, Mrs Cha told Reuters news agency "I believe you can achieve your goal if you persistently pursue it. So don't give up your dream, like me. Be strong and do your best."

We know that on her final and 950th attempt, Cha Sa-soon answered 30 questions correctly. In addition, news agencies report that in her 949 unsuccessful attempts, the number of correct answers had ranged from 15 to 25. Armed with this knowledge, what can we say about θ, the latent probability that Cha Sa-soon can answer any one question correctly? Note that we assume each question is equally difficult, and that Cha Sa-soon does not learn from her earlier attempts.

What makes these data special is that for the failed attempts, we do not know the precise scores. We only know that these scores range from 15 to 25. In statistical terms, these data are said to be censored, both from below and above. We follow and approach inspired by Gelman and Hill (2007, p. 405) to apply WinBUGS to the problem of dealing with censored data.



$$\theta \sim \text{Beta}(1,1)$$
$$z_i \sim \text{Binomial}(\theta, n)$$
$$15 \leq z_i \leq 25 \quad \text{if } y_i = 1$$

Fig. 5.8 Graphical model for inferring a rate from observed and censored data.

Figure 5.8 presents a graphical model for dealing with the censored data. The variable z_i represents both the first 949 unobserved, and the final observed attempt. It is called a 'partially observed' variable, and is shaded more lightly to denote this. The variable y_i is a simple binary indicator variable, denoting whether or not the ith attempt is observed. The bounds $z^{\text{lo}} = 15$ and $z^{\text{hi}} = 25$ give the known censored interval for the unobserved attempts. Finally, $n = 50$ is the number of questions in the test. This means that $z_i \sim \text{Binomial}\big(\theta, n\big)_{\mathcal{I}(z^{\text{lo}}, z^{\text{hi}})}$ when y_i indicates a censored attempt, but is not censored for the final known score $z_{950} = 30$. Note that this means z_i is observed once, but not observed the other times. This sort of variable is known as *partially observed*, and is denoted in the graphical model by a lighter shading (between the dark shading of fully observed nodes, and the lack of shading for fully unobserved or latent nodes).

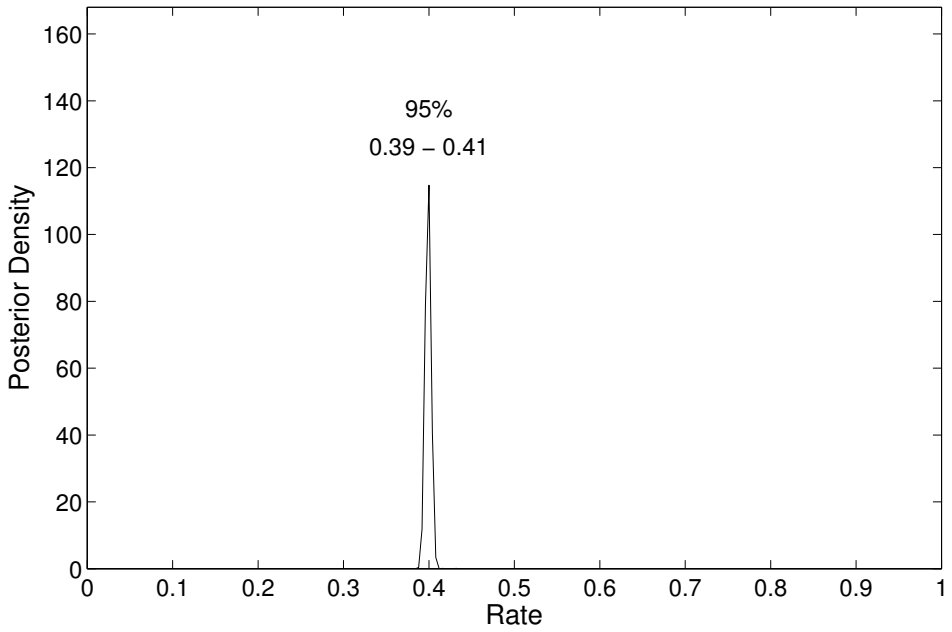


Fig. 5.9 Posterior density for Cha Sa-soon's rate of answering questions correctly.

The script `ChaSaSoon.txt` implements this graphical model in WinBUGS.

```
# ChaSaSoon Censored Data
model
{
  for (i in 1:nattempts){
    # If the Data Were Unobserved y[i]=1, Otherwise y[i]=1
    z.low[i]  <- 15*equals(y[i],1)+0*equals(y[i],0)
    z.high[i] <- 25*equals(y[i],1)+n*equals(y[i],0)
    z[i] ~ dbin(theta,n)I(z.low[i],z.high[i])
  }
  # Uniform Prior on Rate Theta
  theta ~ dbeta(1,1)
}
```

Note the use of the `equals` command, which returns 1 when its arguments match, and 0 when they mismatch. Thus, when `y[i]=1`, for censored data, `z.low[i]` is set to 15 and `z.hi[i]` is set to 25. When `y[i]=0` `z.low[i]` is set to 0 and `z.hi[i]` is set to n. These `z.low[i]` and `z.hi[i]` values are then applied to censor the Binomial distribution that generates the test scores.

The code `ChaSaSoon.m` or `ChaSaSoon.R` applies the model to the data from Cha Sa-soon. The posterior density for θ is shown in Figure 5.9, and can be seen to be relatively peaked. Despite the fact that we do not know the actual scores for 949 of the 950 results, we are still able to infer a lot about θ.

Exercises

Exercise 5.5.1 Do you think Cha Sa-soon could have passed the test by just guessing?

Exercise 5.5.2 What happens when you increase the interval in which you know the data are located, from 15–25 to something else?

Exercise 5.5.3 What happens when you decrease the number of failed attempts?

Exercise 5.5.4 What happens when you increase Cha Sa-soon's final score from 30?

Exercise 5.5.5 Do you think the assumption that all of the scores follow a Binomial distribution with a single rate of success is a good model for these data?

5.6 Population Size

An interesting inference problem that occurs in a number of fields is to estimate the size of a population, when a census is impossible, but repeated surveying is possible. For example, the goal might be to estimate the number of animals in a large woodland area that cannot be search exhaustively. Or, the goal might be to decide how many students are on a campus, but it is not possible to count them all.

A clever sampling approach to this problem is given by capture-and-recapture methods. The basic idea is to capture (i.e., identify, tag, or otherwise remember) a sample at one time point, and then collect another sample. The number of items in the second sample that were also in the first then provides relevant information as to the population size.

Probably the simplest possible version of this approach can be formalized with t as the population total, x as the number in the first sample, n as the number in the second sample, and k as the number in both samples. That is, x animals are tagged or people remembered in the first sample, then k out of n are seen again (i.e., recaptured) in the second sample.

The statistical model to relate the counts, and make inferences about the population size t based on the hypergeometric distribution, so that the probability that the true underlying population is t is given by

$$\frac{\binom{x}{k}\binom{t-x}{n-k}}{\binom{t}{n}}.$$

This makes intuitive sense, since the second sample involves taking n items from a population of t, and has k out of x recaptures, $n-k$ other items out of the other $t-x$ in the population.

The Bayesian approach to this problem involves putting a prior on t, and using the hypergeometric distribution as the appropriate likelihood function. Conceptually,

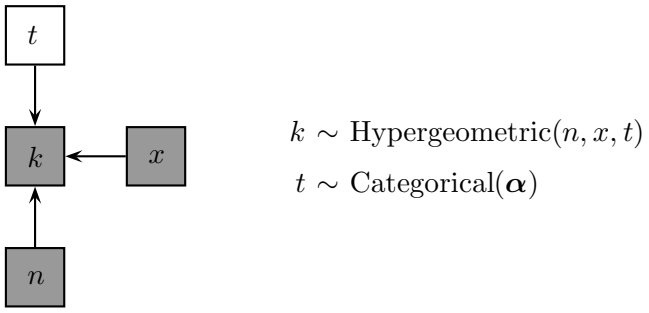


Fig. 5.10 Graphical model for inferring a population from capture and recapture data.

this means $k \sim \text{Hypergeometric}\big(n, x, t\big)$, as in the graphical model in Figure 5.10. The vector $\boldsymbol{\alpha}$ allows for any sort of prior mass to be given to all the possible counts for the population total. Since, $x + (n - k)$ items are known to exist, one reasonable choice of prior might be to make every possibility from $x + (n - k)$ to $t^{\max}$ equally likely, where $t^{\max}$ is a sensible upper bound on the possible population.

While simple conceptually, there is a a difficulty in implementing the graphical model in Figure 5.10. The problem is that WinBUGS does not provide the hypergeometric distribution. It is, however, possible to implement distributions that are not provided, but for which the likelihood function can be expressed in WinBUGS. This can be done using the either the so-called 'ones trick' or 'zeros trick'. These tricks rely on simple properties of the Poisson and Bernoulli distributions. By implementing the likelihood function of the new distribution within the Poisson or Bernoulli distribution, and forcing values of 1 or 0 to be sampled, it can be shown that the samples actually generated will come from the desired distribution.[1]

The script `Population.txt` implements the graphical model in Figure 5.10 in WinBUGS, using the zeros trick. Note how the terms in the log-likelihood expression for the hypergeometric distribution are built up to define `phi`, and a constant `C` is used to insure the Poisson distribution is used with a positive value.

```
# Population
model{
    # Hypergeometric Likelihood Via Ones Trick
    logterm1 <- logfact(x)-logfact(k)-logfact(x-k)
    logterm2 <- logfact(t-x)-logfact(n-k)-logfact((t-x)-(n-k))
    logterm3 <- logfact(t)-logfact(n)-logfact(t-n)
    C <- 1000
    phi <- -(logterm1+logterm2-logterm3)+C
    zeros <- 0
    zeros ~ dpois(phi)
    # Prior on Population Size
    for (i in 1:tmax){
       tptmp[i] <- step(i-(n-k+x))
       tp[i] <- tptmp[i]/sum(tptmp[1:tmax])
```

[1] The negative log-likelihood of a sample of 0 from $\text{Poisson}\big(\phi\big)$ is ϕ. The likelihood of a sample of 1 from $\text{Bernoulli}\big(\theta\big)$ is θ. So, by setting $\log \phi$ or θ appropriately, and forcing 1 or 0 to be observed, sampling effectively proceeds from the distribution defined by ϕ or θ.

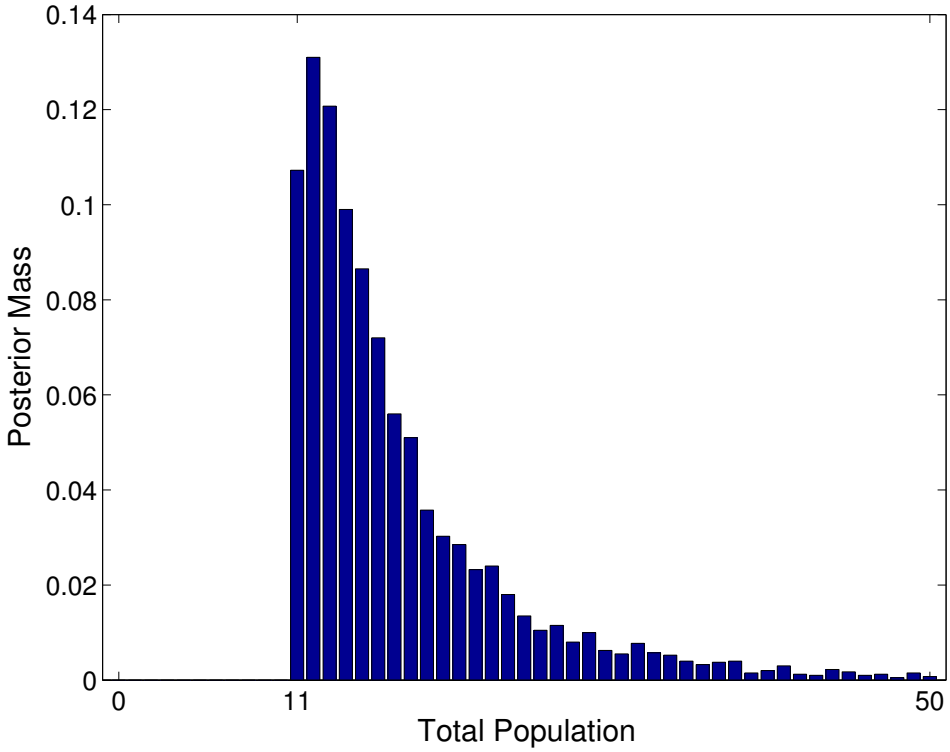


Fig. 5.11 Posterior mass for the population size, known to be 50 or fewer, based on a capture-recapture experiment with $x = 10$ items in the first sample, and $k = 4$ out of $n = 5$ recaptured in the second sample.

```
    }
    t ~ dcat(tp[])
}
```

The code `Population.m` or `Population.R` applies the model to the data $x = 10$, $k = 4$, and $n = 5$, using uniform prior mass for all possible sizes between $x + (n - k) = 11$ and $t^{\max} = 50$. The posterior distribution for t is shown in Figure 5.11. The inference is that it is mostly likely there are not many more than 6 items, which makes intuitive sense, since 4 out of 5 in the second sample were from the original set of 10.

Exercises

Exercise 5.6.1 Try changing the number of items recaptured in the second sample from $k = 4$ to $k = 0$. What inference do you draw about the population size now?

Exercise 5.6.2 How important is it that the upper bound $t^{\max} = 50$ correspond

closely to available information when $k = 4$ and when $k = 0$? Justify your answer by trying both the $k = 4$ and $k = 0$ cases with $t^{\max} = 100$.

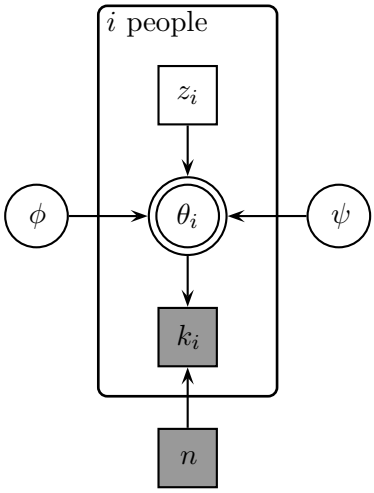
Exercise 5.6.3 Suppose, having obtained the posterior mass in Figure 5.11, the same population was subjected to a new capture-recapture experiment (e.g., with a different means of identifying or tagging). What would be an appropriate prior for t?

6 Latent Mixture Models

6.1 Exam Scores

Suppose a group of 15 people sit an exam made up of 40 true-or-false questions, and they get 21, 17, 21, 18, 22, 31, 31, 34, 34, 35, 35, 36, 39, 36, and 35 right. These scores suggest that the first 5 people were just guessing, but the last 10 had some level of knowledge.

One way to make statistical inferences along these lines is to assume there are two different groups of people. These groups have different probabilities of success, with the guessing group having a probability of 0.5, and the knowledge group having a probability greater than 0.5. Whether each person belongs to the first or the second group is a latent and unobserved variable that can take just two values. Using this approach, the goal is to infer to which group each person belongs, and also the rate of success for the knowledge group.



$$
\begin{aligned}
z_i &\sim \text{Bernoulli}(0.5) \\
\psi &\leftarrow 0.5 \\
\phi &\sim \text{Uniform}(0.5, 1) \\
\theta_i &\sim \begin{cases} \phi \text{ if } z_i = 1 \\ \psi \text{ if } z_i = 0 \end{cases} \\
k_i &\sim \text{Binomial}(\theta_i, n)
\end{aligned}
$$

Fig. 6.1 Graphical modeling for inferring membership of two latent groups, with different rates of success in answering exam questions.

A graphical model for doing this is shown in Figure 6.1. The number of correct answers for the ith person is k_i, and is out of $n = 40$. The probability of success on each question for the ith person is the rate θ_i. This rate is either ϕ_0, if the person is

in the guessing group, or ϕ_1 if the person is in the knowledge group. Which group they are in is determined by their binary indicator variable z_i, with $z_i = 0$ if the ith person is in the guessing group, and $z_i = 1$ is they are in the knowledge group.

We assume each of these indicator variables equally likely to be 0 or 1 a priori, so they have the prior $z_i \sim \text{Bernoulli}\big(1/2\big)$. For the guessing group, we assume that the rate is $\phi_0 = 1/2$. For the knowledge group, we use a prior where all rate possibilities greater than 1/2 are equally likely, so that $\phi_1 \sim \text{Uniform}\big(0.5, 1\big)$.

The script `ExamsQuizzes_1.txt` implements the graphical model in WinBUGS.

```
# Exam Scores
model{
   # Each Person Belongs To One Of Two Latent Groups
   for (i in 1:p){
      z[i] ~ dbern(0.5)
      z1[i] <- z[i]+1
   }
   # First Group Just Guesses
   phi[1] <- 0.5
   # Second Group Has Some Unknown Greater Rate Of Success
   phi[2] ~ dunif(0.5,1)
   # Data Follow Binomial With Rate Given By Each Person's Group Assignment
   for (i in 1:p){
      theta[i] <- phi[z1[i]]
      k[i] ~ dbin(theta[i],n)
   }
}
```

Notice the use of a dummy variable `z1[i] <- z[i]+1`, which—just as in the change-detection example in Section 5.4—allows WinBUGS array structures to be indexed in assigning `theta[i]`.

The code `ExamsQuizzes_1.m` or `ExamsQuizzes_1.R` makes inferences about group membership, and the success rate of the knowledge group, using the model.

Exercises

Exercise 6.1.1 Draw some conclusions about the problem from the posterior distribution. Who belongs to what group, and how confident are you?

Exercise 6.1.2 The initial allocations of people to the two groups in this code is random, and so will be different every time you run it. Check that this does not affect the final results from sampling.

Exercise 6.1.3 Include an extra person in the exam, with a score of 28 out of 40. What does their posterior for z tell you?

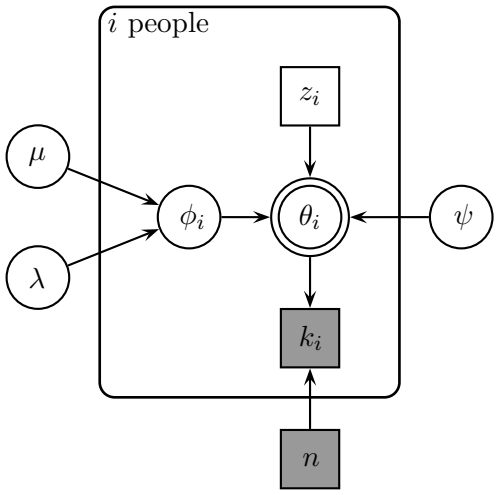
Exercise 6.1.4 What happens if you change the prior on the success rate of the second group to be uniform over the whole range $(0, 1)$, and so allow for worse-than-guessing performance?

Exercise 6.1.5 What happens if you change the initial expectation that everybody is equally likely to belong to either group, and have an expectation that people generally are not guessing, with (say), $z_i \sim \text{Bernoulli}\big(0.9\big)$?

6.2 Exam Scores With Individual Differences

The previous example shows how sampling can naturally and easily find discrete latent groups. But the model itself has at least one big weakness, which is that it assumes all the people in the knowledge group have exactly the same rate of success on the questions.

One straightforward way to allow for individual differences in the knowledge group is to extend the model hierarchically. This involves drawing the success rate for each of the people in the knowledge group from an over-arching distribution. One convenient (but not perfect) choice for this 'individual differences' distribution is a Gaussian. It is a natural statistical model for individual variation, at least in the absence of any rich theory. But it has the problem of allowing for success rates below zero and above one. An inelegant but practical and effective way to deal with this is simply to censor the sampled success rates to the valid range.



$$
\begin{aligned}
z_i &\sim \text{Bernoulli}(0.5) \\
\mu &\sim \text{Uniform}(0.5, 1) \\
\lambda &\sim \text{Gamma}(0.001, 0.001) \\
\phi_i &\sim \text{Gaussian}(\mu, \lambda) \\
\psi &\leftarrow 0.5 \\
\theta_i &\leftarrow \begin{cases} \phi_i \text{ if } z_i = 1 \\ \psi \text{ if } z_i = 0 \end{cases} \\
k_i &\sim \text{Binomial}(\theta_i, n)
\end{aligned}
$$

Fig. 6.2 Graphical model for inferring membership of two latent groups, with different rates of success in answering exam questions, allowing for individual differences in the knowledge group.

A graphical model that implements this idea is shown in Figure 6.2. It extends the original model by having a knowledge group success rate ϕ_{i1} for the ith person. These success rates are drawn from a Gaussian distribution with mean μ and precision λ. The mean μ is given a Uniform prior between 0.5 and 1.0, consistent with the original assumption that people in the knowledge group have a greater than chance success rate.

The script `ExamsQuizzes_2.txt` implements the graphical model in WinBUGS.

```
# Exam Scores With Individual Differences
```

```
model {
   # Data Follow Binomial With Rate Given By Each Person's Group Assignment
   for (i in 1:p){
      k[i] ~ dbin(theta[i,z1[i]],n)
   }
   # Each Person Belongs To One Of Two Latent Groups
   for (i in 1:p){
      z[i] ~ dbern(0.5)
      z1[i] <- z[i]+1
   }
   # The Second Group Now Allows Individual Differences
   # So There Is a Rate Per Person
   for (i in 1:p){
      # First Group Is Still Just Guesses
      theta[i,1] <- 0.5
      # Second Group Drawn From A Censored Gaussian Distribution
      thetatmp[i,2] ~ dnorm(mu,lambda)
          theta[i,2] <- min(1,max(0,thetatmp[i,2])) # Censor The Probability To (0,1)
   }
   # Second Group Mean, Precision (And Standard Deviation)
   mu ~ dunif(0.5,1) # Greater Than 0.5 Average Success Rate
   lambda ~ dgamma(.001,.001)
   sigma <- 1/sqrt(lambda)
   # Posterior Predictive For Second Group
   predphitmp ~ dnorm(mu,lambda)
   predphi <- min(1,max(0,predphitmp))
}
```

Notice that is includes a posterior predictive variable `predphi` for the knowledge group success rates of each person.

The code `ExamsQuizzes_2.m` or `ExamsQuizzes_2.R` makes inferences about group membership, the success rate of each person the knowledge group, and the mean and standard deviation of the over-arching Gaussian for the knowledge group.

Exercises

Exercise 6.2.1 Compare the results of the hierarchical model with the original model that did not allow for individual differences.

Exercise 6.2.2 Interpret the posterior distribution of by the variable `predphi`. How does this distribution relate to the posterior distribution for `mu`?

Exercise 6.2.3 What does the posterior distribution for the variable `theta[1,2]` mean?

Exercise 6.2.4 In what sense could the latent assignment of people to groups in this case study be considered a form of model selection?

6.3 Twenty Questions

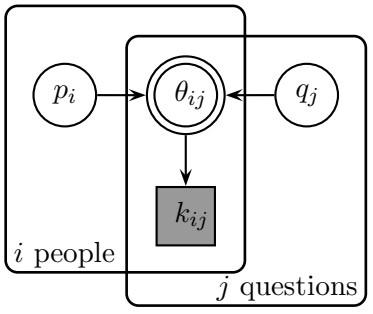
Suppose a group of 10 people attend a lecture, and are asked a set of 20 questions afterwards, with every answer being either correct or incorrect. The pattern of data

is shown in Table 6.1. From this pattern of correct and incorrect answers we want to infer two things. The first is how well each person attended to the lecture. The second is how hard each of the questions was.

Table 6.1 Correct and incorrect answers for 10 people on 20 questions.

	Question																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Person 1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	0	1	0	0
Person 2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Person 3	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
Person 4	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Person 5	1	0	1	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0
Person 6	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0	0
Person 7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Person 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Person 9	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1
Person 10	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0

One way to make these inferences is to specify a model of how a person's attentiveness and a question's difficulty combine to give an overall probability the question will be answered correctly. A very simple model involves assuming each person listens to some proportion of the lecture, and that each question has some probability of being answered correctly if the person was listening at the right point in the lecture.



$$p_i, q_j \sim \text{Beta}(1, 1)$$
$$\theta_{ij} \leftarrow p_i q_j$$
$$k_{ij} \sim \text{Bernoulli}(\theta_{ij})$$

Fig. 6.3 Graphical model for inferring the rate people listened to a lecture, and the difficulty of the questions.

A graphical model that implements this idea is shown in Figure 6.3. Under the model, if the ith person's probability of listening is p_i, and the jth question's probability of being answered correctly if the relevant information is heard is q_j, then the probability the ith person will answer the jth question correctly is just

$\theta_{ij} = p_i q_j$. The observed pattern of correct and incorrect answers, where $k_{ij} = 1$ if the ith person answered the jth question correctly, and $k_{ij} = 0$ if they did not, then is a draw from a Bernoulli distribution with probability θ_{ij}.

The script `TwentyQuestions.txt` implements the graphical model in WinBUGS.

```
# Twenty Questions
model {
  # Correctness Of Each Answer Is Bernoulli Trial
  for (i in 1:np){
     for (j in 1:nq){
        k[i,j] ~ dbern(theta[i,j])
     }
  }
  # Probability Correct Is Product Of Question By Person Rates
  for (i in 1:np){
     for (j in 1:nq){
        theta[i,j] <- p[i]*q[j]
     }
  }
  # Priors For People and Questions
  for (i in 1:np){
     p[i] ~ dbeta(1,1)
  }
  for (j in 1:nq){
     q[j] ~ dbeta(1,1)
  }
}
```

The code `TwentyQuestions.m` or `TwentyQuestions.R` makes inferences about the data in Table 6.1 using the model.

Exercises

Exercise 6.3.1 Draw some conclusions about how well the various people listened, and about the difficulties of the various questions. Do the marginal posterior distributions you are basing your inference on seem intuitively reasonable?

Exercise 6.3.2 Now suppose that three of the answers were not recorded, for whatever reason. Our new data set, with missing data, now take the form shown in Table 6.2.

Bayesian inference will automatically make predictions about these missing values (i.e., "fill in the blanks") by using the same probabilistic model that generated the observed data. Missing data are entered as `nan` ("not a number") in Matlab, and `NA` ("not available") in R or WinBUGS. Including the variable `k` as one to monitor when sampling will then provide posterior values for the missing values. That is, it provides information about the relative likelihood of the missing values being each of the possible alternatives, using the statistical model and the available data.

Look through the Matlab or R code to see how all of this is implemented in the second dataset. Run the code, and interpret the posterior distributions for the three missing values. Are they reasonable inferences?

Table 6.2 Correct, incorrect and missing answers for 10 people on 20 questions.

	Question																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Person 1	1	1	1	1	0	0	1	1	0	1	0	0	?	0	0	1	0	1	0	0
Person 2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Person 3	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
Person 4	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Person 5	1	0	1	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0
Person 6	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0	0
Person 7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Person 8	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Person 9	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1
Person 10	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	?	0	0

6.4 The Two Country Quiz

Suppose a group of people take a historical quiz, and each answer for each person is scored as correct or incorrect. Some of the people are Thai, and some are Moldovan. Some of the questions are about Thai history, and would be very likely to be known by any Thai person, but very unlikely to be known by people from outside the region. The rest of the questions are about Moldovan history, and would be very likely to be known by any Moldovan, but not by others.

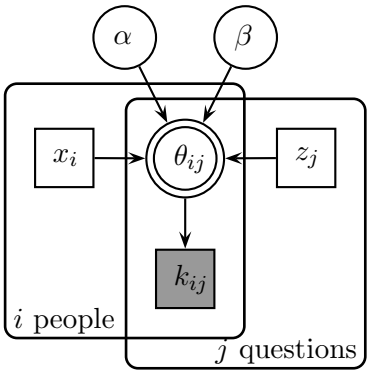
We do not know who is Thai or Moldovan, and we do not know the content of the questions. All we have are the data shown in Table 6.3. Spend some time just looking at the data, and try to infer which people are from the same country, and which questions relate to their country.

A good way to make these inferences formally is to assume there are two types of answers. For those where the nationality of the person matches the origin of the question will be correct with high probability. For those where a person is being asked about the other country will have a very low probability of being correct.

A graphical model that implements this idea is shown in Figure 6.4. The rate α is the (expected to be high) probability of a person from a country correctly answering a question about their country's history. The rate β is the (expected to be low) probability of a person correctly answering a question about the other country's history. To capture the knowledge about the rates, the priors constrain $\alpha \geq \beta$, by defining `alpha` $\sim$ `dunif(0,1)` and `beta` $\sim$ `dunif(0,alpha)`. At first glance, this might seem inappropriate, since it specifies a prior for one parameter in terms of another (unknown, and being inferred) parameter. Conceptually, it is clearer to think of this syntax as a (perhaps clumsy) way to specify a *joint* prior

Table 6.3 Correct, incorrect and missing answers for 8 people on 8 questions.

	Question							
	A	B	C	D	E	F	G	H
Person 1	1	0	0	1	1	0	0	1
Person 2	1	0	0	1	1	0	0	1
Person 3	0	1	1	0	0	1	0	0
Person 4	0	1	1	0	0	1	1	0
Person 5	1	0	0	1	1	0	0	1
Person 6	0	0	0	1	1	0	0	1
Person 7	0	1	0	0	0	1	1	0
Person 8	0	1	1	1	0	1	1	0



$$
\begin{aligned}
\alpha &\sim \text{Uniform}(0,1) \\
\beta &\sim \text{Uniform}(0,\alpha) \\
x_i &\sim \text{Bernoulli}(0.5) \\
\theta_{ij} &\leftarrow \begin{cases} \alpha & \text{if } x_i = z_j \\ \beta & \text{if } x_i \neq z_j \end{cases} \\
k_{ij} &\sim \text{Bernoulli}(\theta_{ij})
\end{aligned}
$$

Fig. 6.4 Graphical model for inferring the country of origin for people and questions.

over α and β in which the $\alpha \geq \beta$. Graphically, the parameter space over (α, β) is a unit square, and the prior being specified is the half of the square on one side of the diagonal line $\alpha = \beta$.

In the remainder of the graphical model, the binary indicator variable x_i assigns the ith person to one or other country, and z_j similarly assigns the jth question to one or other country. The probability the ith person will answer the jth question correctly is θ_{ij}, which is simply α if the country assignments match, and β if they do not. Finally, the actual data k_{ij} indicating whether or not the answer was correct follow a Bernoulli distribution with rate θ_{ij}.

The script `TwoCountryQuiz.txt` implements the graphical model in WinBUGS.

```
# The Two Country Quiz
model {
   # Probability Of Not Forgetting And Guessing
   alpha ~ dunif(0,1) # Match
   beta ~ dunif(0,alpha)  # Mismatch
   # Group Membership For People and Questions
```

```
   for (i in 1:np){
      pz[i] ~ dbern(0.5)
      pz1[i] <- pz[i]+1
   }
   for (j in 1:nq){
      qz[j] ~ dbern(0.5)
      qz1[j] <- qz[j]+1
   }
   # Probability Correct For Each Person-Question Comination By Groups
   # High If Person Group Matches Question Group
   # Low If No Match
   for (i in 1:np){
      for (j in 1:nq){
         theta[i,j,1,1] <- alpha
         theta[i,j,1,2] <- beta
         theta[i,j,2,1] <- beta
         theta[i,j,2,2] <- alpha
      }
   }
   # Data Are Bernoulli By Rate
   for (i in 1:np){
      for (j in 1:nq){
         k[i,j] ~ dbern(theta[i,j,pz1[i],qz1[j]])
      }
   }
}
```

The code `TwoCountryQuiz.m` or `TwoCountryQuiz.R` makes inferences about the data in Table 6.3 using the model.

Exercises

Exercise 6.4.1 Interpret the posterior distributions for `x[i], z[j], alpha` and `beta`. Do the formal inferences agree with your original intuitions?

Exercise 6.4.2 The priors on the probabilities of answering correctly capture knowledge about what it means to match and mismatch, by imposing an order constraint $\alpha \geq \beta$. Change the code so that this information is not included, by using priors `alpha`$\sim$`dbeta(1,1)` and `beta`$\sim$`dbeta(1,1)`. Run a few chains against the same data, until you get an inappropriate, and perhaps counter-intuitive, result. Describe the result, and discuss why it comes about.

Exercise 6.4.3 Now suppose that three extra people enter the room late, and begin to take the quiz. One of them (Late Person 1) has answered the first four questions, the next (Late Person 2) has only answered the first question, and the final new person (Late Person 3) is still sharpening their pencil, and has not started the quiz. This situation can be represented as an updated data set, now with missing data, as in Table 6.4. Interpret the inferences the model makes about the nationality of the late people, and whether or not they will get the unfinished questions correct.

Exercise 6.4.4 Finally, suppose that you are now given the correctness scores for a set of 10 new people, whose data were not previously available, but who

Table 6.4 Correct, incorrect and missing answers for 8 people and 3 late people on 8 questions.

	Question							
	A	B	C	D	E	F	G	H
Person 1	1	0	0	1	1	0	0	1
Person 2	1	0	0	1	1	0	0	1
Person 3	0	1	1	0	0	1	0	0
Person 4	0	1	1	0	0	1	1	0
Person 5	1	0	0	1	1	0	0	1
Person 6	0	0	0	1	1	0	0	1
Person 7	0	1	0	0	0	1	1	0
Person 8	0	1	1	1	0	1	1	0
Late Person 1	1	0	0	1	?	?	?	?
Late Person 2	0	?	?	?	?	?	?	?
Late Person 3	?	?	?	?	?	?	?	?

form part of the same group of people we are studying. The updated data set is shown in Table 6.5. Interpret the inferences the model makes about the nationality of the new people. Revisit the inferences about the late people, and whether or not they will get the unfinished questions correct. Does the inference drawn by the model for the third late person match your intuition? There is a problem here. How could it be fixed?

6.5 Latent Group Assessment of Malingering

Armed with the knowledge from the previous sections we now consider a question of considerable practical interest: how to detect if people are cheating on a test. For instance, people who have been in a car accident may seek financial compensation from insurance companies by feigning cognitive impairment such as pronounced memory loss[1]. When these people are confronted with a memory test that is intended to measure the extent of their impairment, they may deliberately under-perform. This behavior is called malingering, and it may be accompanied by performance that is much worse than that which is displayed by real amnesiacs. For instance, malingerers may perform substantially below chance.

However, malingerers may not always be easy to detect, and this is when Bayesian inference with latent mixture models can be helpful. Not only can we classify people in two categories—those who malinger and those who are truthful or *bona fide*—but we can also quantify our confidence in each classification.

[1] pronounced **mem**-*uh*-ree los.

Table 6.5 Correct, incorrect and missing answers for 8 people, 3 late people, and 10 new people on 8 questions.

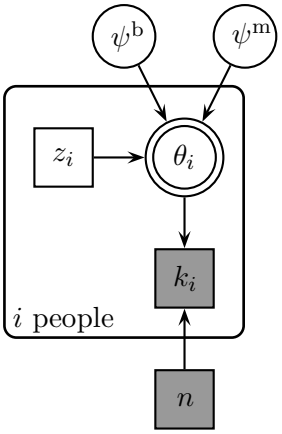
	Question							
	A	B	C	D	E	F	G	H
New Person 1	1	0	0	1	1	0	0	1
New Person 2	1	0	0	1	1	0	0	1
New Person 3	1	0	0	1	1	0	0	1
New Person 4	1	0	0	1	1	0	0	1
New Person 5	1	0	0	1	1	0	0	1
New Person 6	1	0	0	1	1	0	0	1
New Person 7	1	0	0	1	1	0	0	1
New Person 8	1	0	0	1	1	0	0	1
New Person 9	1	0	0	1	1	0	0	1
New Person 10	1	0	0	1	1	0	0	1
Person 1	1	0	0	1	1	0	0	1
Person 2	1	0	0	1	1	0	0	1
Person 3	0	1	1	0	0	1	0	0
Person 4	0	1	1	0	0	1	1	0
Person 5	1	0	0	1	1	0	0	1
Person 6	0	0	0	1	1	0	0	1
Person 7	0	1	0	0	0	1	1	0
Person 8	0	1	1	1	0	1	1	0
Late Person 1	1	0	0	1	?	?	?	?
Late Person 2	0	?	?	?	?	?	?	?
Late Person 3	?	?	?	?	?	?	?	?

In an experimental study on malingering, each of $p = 22$ participants was confronted with a memory test. One group of participants was told to do their best; these are the bona fide participants. The other group of participants was told to under-perform by deliberately simulating amnesia. These are the malingerers. Out of a total of $n = 45$ test items, the participants get 45, 45, 44, 45, 44, 45, 45, 45, 45, 45, 30, 20, 6, 44, 44, 27, 25, 17, 14, 27, 35, and 30 correct. Because this was an experimental study, we know that the first 10 participants were bona fide and the next 12 were instructed to malinger.

The first analysis is straightforward and very similar to the one we did in Section 6.1. We assume that all bona fide participants have the same ability, and so have the same rate ϕ_{b} of answering each question correctly. For the malingerers, the rate of answering questions correctly is given by ϕ_{m}, and $\phi_{\mathrm{b}} > \phi_{\mathrm{m}}$.

The script `Malingering_1.txt` implements the graphical model in WinBUGS.

```
# Malingering
model
{
```



$$
\begin{aligned}
\psi^{\mathrm{b}} &\sim \mathrm{Uniform}(0.5, 1) \\
\psi^{\mathrm{m}} &\sim \mathrm{Uniform}(0, \psi^{\mathrm{b}}) \\
z_i &\sim \mathrm{Bernoulli}(0.5) \\
\theta_i &\leftarrow \begin{cases} \psi^{\mathrm{b}} & \text{if } z_i = 1 \\ \psi^{\mathrm{m}} & \text{if } z_i = 0 \end{cases} \\
k_i &\sim \mathrm{Binomial}(\theta_i, n)
\end{aligned}
$$

Fig. 6.5 Graphical model for the detection of malingering.

```
  # Each person belongs to one of two latent groups
  for (i in 1:p){
    z[i] ~ dbern(0.5)
    z1[i] <- z[i]+1
  }
  # Bonafide group has unknown success rate above chance
  psi[1] ~ dunif(0.5,1)
  # Malingering group has unknown rate of success below bonafide
  psi[2] ~ dunif(0,psi[1])
  # Data are binomial with group rate of each person
  for (i in 1:p){
    theta[i] <- psi[z1[i]]
    k[i] ~ dbin(theta[i],n)
  }
}
```

Notice the restriction in the `dunif` command, which prevents the so-called model indeterminacy or label-switching problem by ensuring that $\phi_{\mathrm{b}} > \phi_{\mathrm{m}}$.

The code `Malingering_1.m` or `Malingering_1.R` allows you to draw conclusions about group membership and the success rate of the two groups.

Exercises

Exercise 6.5.1 What are your conclusions about group membership? Did all participants follow the instructions?

6.6 Individual Differences in Malingering

As before, it may seem needlessly restrictive to assume that all members of a group have the same chance of answering correctly. So now we assume that the ith partic-

ipant in each group has a unique rate parameter, θ_i, which is constrained by group level distributions.

In Section 6.2, we used group level Gaussians. The problem with that approach is that values can lie outside the range 0 to 1. These values can just be excluded from consideration, as we did with the censoring, but this solution is not elegant.

One of several alternatives is to assume that instead of being Gaussian, the group level distribution is Beta(α, β). The α and β values can be thought of as counts of "prior successes" and "prior failures", respectively. Because the Beta distribution is defined on the interval from 0 to 1 it respects the natural boundaries of rates. So we now have a model in which each individual binomial rate parameter is constrained by a group level Beta distribution—this complete model is known as the beta-binomial.

It is useful to transform the α and β parameters from the Beta distribution to a group mean $\mu = \alpha/(\alpha+\beta)$ and a group precision $\lambda = \alpha+\beta$. In a first attempt, one may then assign uniform priors to both μ_b (the group-level mean for the bona fide participants) and μ_m (the group-level mean for the malingerers). Unfortunately, this assignment does not reflect our knowledge that $\mu_\mathrm{b} > \mu_\mathrm{m}$, and so is subject to label-switching. To avoid this problem we could use the `dunif(0,mu_bon)` syntax used in the previous example.

However, here we solve the label-switching problem differently. We first define μ_m as the additive combination of μ_b and a difference parameter, as follows: $\mathrm{logit}(\mu_\mathrm{mal}) = \mathrm{logit}(\mu_\mathrm{b}) - \mu_\mathrm{d}$. Note that this is an additive combination on the *logit* scale, as is customary in beta-binomial models. The logit transformation is defined as $\mathrm{logit}(\theta) \equiv \ln(\theta/(1-\theta))$ and it transforms values on the rate scale (ranging from 0 to 1) to values on the logit scale (ranging from $-\infty$ to ∞). The logit transformation is shown in Figure 6.6.

Next, we assign μ_d a positive-only Gaussian distribution, that is, `mu_diff` $\sim$ `dnorm(0,0.5)I(0,)`. This insures that the group mean of the malingerers is never larger than that of the bona fide participants, and the label-switching problem is solved.

One final note concerns the base rate of malingering ϕ. In the previous example we set the base rate equal to 0.5. Now, we assign the base rate ϕ a prior distribution, and use the data to infer group membership and at the same time learn about the base rate.

A graphical model that implements the above ideas is shown in Figure 6.7. The script `Malingering_2.txt` implements the graphical model in WinBUGS.

```
# Malingering, with individual differences
model {
  # Each person belongs to one of two latent groups
  for (i in 1:p){
     z[i] ~ dbern(phi) # phi is the base rate
     z1[i] <- z[i]+1
  }
  # Relatively uninformative prior on base rate
  phi ~ dbeta(5,5)
```

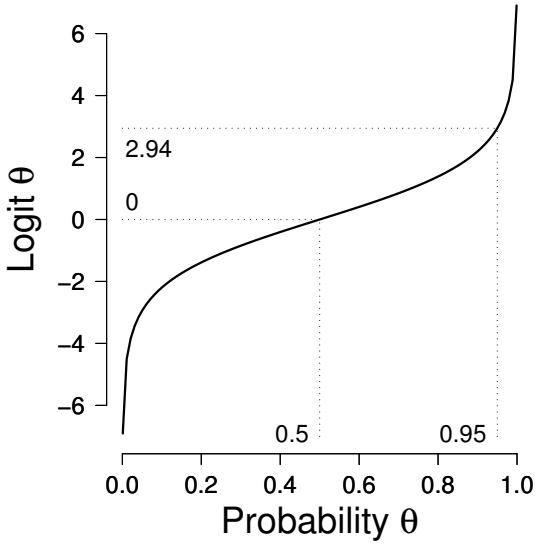
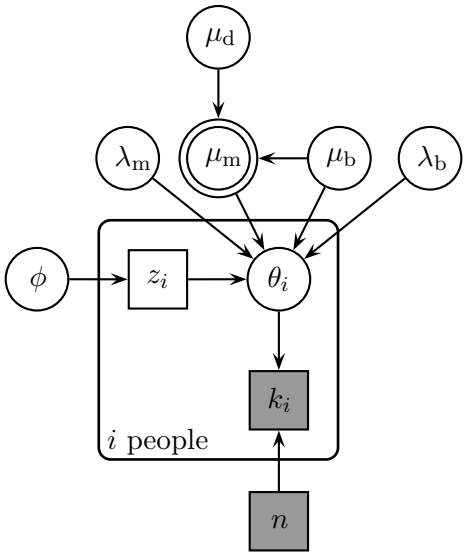


Fig. 6.6 The logit transformation. Probabilities θ range from 0 to 1 and are mapped to the entire real using the logit transform, $\text{logit}(\theta) \equiv \ln(\theta/(1-\theta))$. This transformation is particularly useful for the modeling of additive effects.

```
  # Data are binomial with rate of each person
  for (i in 1:p){
    k[i] ~ dbin(theta[i,z1[i]],n)
    theta[i,1] ~ dbeta(alpha[1],beta[1])
    theta[i,2] ~ dbeta(alpha[2],beta[2])
  }
  # Transformation to group mean and precision
  alpha[1] <- mubon * lambdabon
  beta[1]  <- lambdabon * (1-mubon)
  # Additivity on logit scale
  logit(mumal) <- logit(mubon) - mudiff
  alpha[2] <- mumal * lambdamal
  beta[2]  <- lambdamal * (1-mumal)
  # Priors
  mubon  ~ dbeta(1,1)
  mudiff ~ dnorm(0,0.5)I(0,) # Constrained to be postive
  lambdabon ~ dunif(40,800)
  lambdamal ~ dunif(4,100)
}
```

The code `Malingering_2.m` or `Malingering_2.R` allows you to draw conclusions about group membership and the success rate of the two groups.



$$
\begin{aligned}
\mu_{\mathrm{b}} &\sim \mathrm{Beta}(1,1) \\
\mu_{\mathrm{d}} &\sim \mathrm{Gaussian}(0,0.5)_{\mathcal{I}(0,\infty)} \\
\lambda_{\mathrm{b}} &\sim \mathrm{Uniform}(40,800) \\
\lambda_{\mathrm{m}} &\sim \mathrm{Uniform}(4,100) \\
z_i &\sim \mathrm{Bernoulli}(\phi) \\
\theta_i &\sim \begin{cases} \mathrm{Beta}(\mu_{\mathrm{b}}\lambda_{\mathrm{b}}, (1-\mu_{\mathrm{b}})\,\lambda_{\mathrm{b}}) & \text{if } z_i = 0 \\ \mathrm{Beta}(\mu_{\mathrm{m}}\lambda_{\mathrm{m}}, (1-\mu_{\mathrm{m}})\,\lambda_{\mathrm{m}}) & \text{if } z_i = 1 \end{cases} \\
k_i &\sim \mathrm{Binomial}(\theta_i, n) \\
\mathrm{logit}\mu_{\mathrm{m}} &\leftarrow \mathrm{logit}\mu_{\mathrm{b}} + \mu_{\mathrm{d}} \\
\phi &\sim \mathrm{Beta}(5,5)
\end{aligned}
$$

Fig. 6.7 Graphical model for inferring membership of two latent groups, consisting of malingerers and bona fide participants. Each participant has their own rate of answering memory questions correctly, coming from group level distributions that have their means constrained so that the bona fide group is greater than that for the malingerers.

Exercises

Exercise 6.6.1 Assume you know that the base rate of malingering is 10%. Change the WinBUGS script to reflect this knowledge. Do you expect any differences?

Exercise 6.6.2 Assume you know for certain that participants 1, 2, and 3 are bona fide. Change the code to reflect this knowledge.

Exercise 6.6.3 Suppose you add a new participant. What number of questions answered correctly by this participant would lead to the greatest uncertainty about their group membership?

Exercise 6.6.4 Try to solve the label-switching problem by using the `dunif(0,mu_bon)` trick instead of the logit transform.

Exercise 6.6.5 Why do you think the priors for λ_{b} and λ_{m} are different?

6.7 Alzheimer's Recall Test Cheating

In this section, we apply the same latent mixture model shown in Figure 6.7 to different memory test data. Simple recognition and recall tasks are an important part of screening for Alzheimer's Disease and Related Disorders (ADRD), and are

sometimes administered over the telephone. This practice raises the possibility of people cheating by, for example, writing down the words they are being asked to remember.

The data we use come from an informal experiment, in which 118 people (actually, employees of an insurance company familiar with administering these tests) were asked either to complete the test normally (giving a total of 61 bona fide people), or were instructed to cheat (giving a total of 57 malingerers). The particular test used was a complicated sequence of immediate and delayed free recall tasks, which we simplify to give a simple score correct out of 40 for each person.

We assume that both the bona fide and malingering groups come from different populations, following the model in Figure 6.7. Note that we expect the mean of the malingerers to be higher, since the impact of cheating is to recall more words than would otherwise be the case. The script `Cheating.txt` implements the analysis in WinBUGS

```
# Cheating Latent Mixture Model
model
{
  # Each Person Belongs To One Of Two Latent Groups
  for (i in 1:p){
     z[i] ~ dbern(phi) # phi is the Base Rate
     z1[i] <- z[i]+1
  }
  # Relatively Uninformative Prior on Base Rate
  phi ~ dbeta(5,5)
  # Data Follow Binomial With Rate Given By Each Person's Group Assignment
  for (i in 1:p){
    k[i] ~ dbin(theta[i,z1[i]],n)
    thetatmp[i,1] ~ dbeta(alpha[1],beta[1])
        theta[i,1] <- max(.001,min(.999,thetatmp[i,1]))
    thetatmp[i,2] ~ dbeta(alpha[2],beta[2])
        theta[i,2] <- max(.001,min(.999,thetatmp[i,2]))
  }
  # Transformation to Group Mean and Precision
  alpha[1] <- mubon * lambdabon
  beta[1]  <- lambdabon * (1-mubon)
  # Additivity on Logit Scale
  logit(mumal) <- logit(mubon) - mudiff
  alpha[2] <- mumal * lambdamal
  beta[2]  <- lambdamal * (1-mumal)
  # Priors
  mubon  ~ dbeta(1,1)
  mudifftmp ~ dnorm(0,0.5)
  mudiff <- max(0,mudifftmp) # Constrained to be Postive
  lambdabon ~ dunif(5,50)
  lambdamal ~ dunif(5,50)
  # Correct Count
  for (i in 1:p){
    pct[i] <- equals(z[i],truth[i])
  }
  pc <- sum(pct[1:p])
}
```

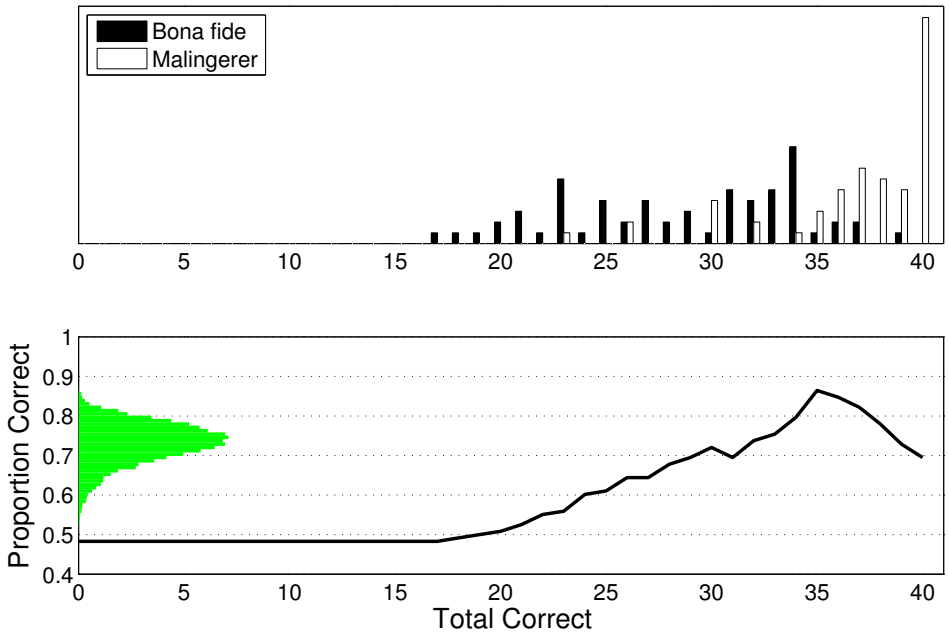


Fig. 6.8 The distribution of total correct recall scores for the Alzheimer's data, and classification performance. The top panel shows the distribution of scores for the bona fide and malingering groups. The bottom panel shows, with the line, the accuracy achieved using various cut offs to separate the groups, and, with the distribution, the accuracy achieved by the latent mixture model.

This script is essentially the same as for the previous example, although it modifies the priors on the precisions of the group distributions. It also includes a variable `pc` that keeps track of the accuracy of each classification sample made in sampling, by comparing each person's latent assignment to the known truth from the experimental design.

The code `Cheating.m` or `Cheating.R` applies the graphical model to the data. We focus our analysis of the results firstly on the classification accuracy of the model. The top panel of Figure 6.8 summarizes the data, showing the distribution of correctly recalled words in both the bona fide and malingering groups. It is clear that malingerers generally recall more words, but that there is overlap between the groups.

One way to provide a benchmark classification accuracy is the consider the best possible 'cut off'. This is a total correct score below which a person is classified as bona fide, and at or above which they are classified as a malingerer. The line in the bottom panel in Figure 6.8 shows the classification accuracy for all possible cut offs, which peaks at 86.4% accuracy using the cut off of 35. The green distribution at the left of the panel is the posterior distribution of the `pc` variable, showing the range of accuracy achieved by the latent mixture model.

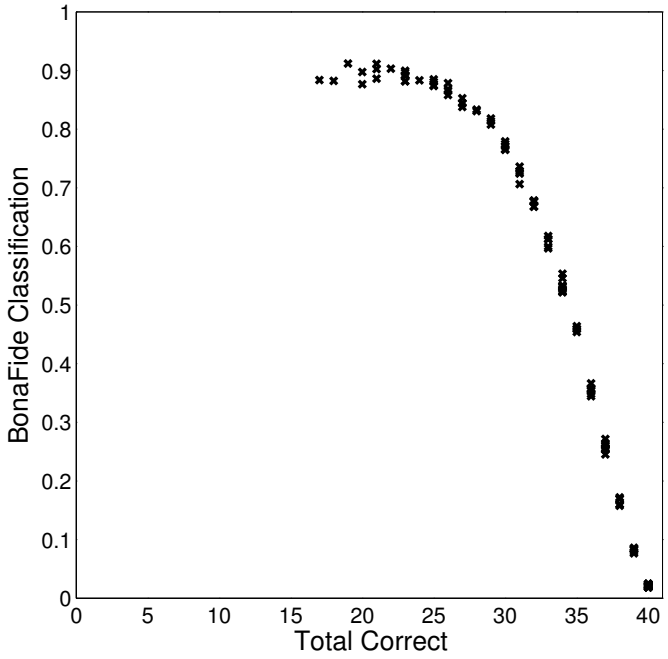


Fig. 6.9 The relationship between each person's total correct recall score, and their posterior classification as belonging to the bona fide or malingering group.

In general, using generative model to solve classification problems is unlikely to work as well as the best discriminative methods from machine learning and statistics. The advantage of the generative model is in providing details about the underlying processes assumed to produce the data, particularly by quantifying uncertainty. A good example of this general feature is shown in Figure 6.9, which shows the relationship between the total correct raw data score, and the posterior uncertainty about classification, for each person. This figure shows that the model infers people with scores below 35 as more likely to be bona fide. But it also shows how certain the model is about each classification, which provides more information (and more probabilistically coherent information) than many machine learning methods.

This information about uncertainty is useful, for example, if there are costs or utilities associated with different classification decisions. Suppose that raising a false-alarm and suspecting someone of cheating on the screening test costs \$25, perhaps through a wasted follow-up in person test, but that missing someone cheated on the screening test costs \$100, perhaps through providing insurance that should not have been provided. With these utilities, the decision should be to classify people as bona fide only if it is four times more likely than them being a malingerer. In other words, we need 80% certainty they are bona fide. The posterior distribution of the latent assignment variable `z` provides exactly this information. It is clear

from Figure 6.9 only people with a total correct score below 30 (not 35) should be treated as bona fide, under this set of utilities.

Exercises

Exercise 6.7.1 Suppose the utilities are very different, so that a false-alarm costs \$100, because of the risk of litigation in a false accusation, but misses are relatively harmless, costing \$10 in wasted administrative costs. What decisions should be made about bona fide and malingering now?

Exercise 6.7.2 What other potential information, besides the uncertainty about classification, does the model provide? Give at least one concrete example.

PART III

ANSWERS

Preface

This document contains answers to the exercises from the book "A Course in Bayesian Graphical Modeling for Cognitive Science". Contrary to popular belief, statistical modeling is rarely a matter of right or wrong; instead, the overriding concerns are reasonableness and plausibility. Therefore, you may find yourself disagreeing with some of our intended solutions. Please let us know if you believe your answer is clearly superior to ours, and—if we agree with you—we will adjust the text accordingly.

Michael D. Lee
Irvine, USA
Eric-Jan Wagenmakers
Amsterdam, The Netherlands
August 2011

2 The Basics of Bayesian Analysis

2.1 Answers: General Principles

Exercise 2.1.1 The famous Bayesian statistician Bruno de Finetti published two big volumes entitled "Theory of Probability" (de Finetti, 1974). Perhaps surprisingly, the first volume starts with the words "probability does not exist". To understand why de Finetti wrote this, consider the following situation: someone tosses a fair coin, and the outcome will be either heads or tails. What do you think the probability is that the coin lands heads? Now suppose you are a physicist with advanced measurement tools, and you can establish relatively precisely both the position of the coin and the tension in the muscles immediately before the coin is tossed in the air—does this change your probability? Now suppose you can briefly look into the future (Bem, 2011), albeit hazily—is your probability still the same?

These different situations illustrate how the concept of probability is always conditional on background knowledge, and does not exist in a vacuum. This idea is central to the subjective Bayesian school, a school that stresses how inference is, in the end, dependent on personal beliefs.

Exercise 2.1.2 On his blog, prominent Bayesian Andrew Gelman wrote (March 18, 2010) "Some probabilities are more objective than others. The probability that the die sitting in front of me now will come up '6' if I roll it...that's about 1/6. But not exactly, because it's not a perfectly symmetric die. The probability that I'll be stopped by exactly three traffic lights on the way to school tomorrow morning: that's...well, I don't know exactly, but it is what it is." Was de Finetti wrong, and is there only one clearly defined probability of Andrew Gelman encountering three traffic lights on the way to school tomorrow morning?

A detailed knowledge of the layout of the traffic signs along the route will influence your assessment of this probability, as well as your knowledge of how busy traffic will be tomorrow morning, how often the traffic signs malfunction, whether traffic will be rerouted because of construction, and so on When you can look one day into the future, the probability of Andrew Gelman encountering

three traffic lights on the way to school is either zero or one. As before, probability statements are conditional on your background knowledge.

Exercise 2.1.3 Figure 1.1 shows that the 95% Bayesian credible interval for θ extends from 0.59 to 0.98. This means that one can be 95% confident that the true value of θ lies in between 0.59 and 0.98. Suppose you would do an orthodox analysis and find the same confidence interval. What is the orthodox interpretation of this interval?

The orthodox interpretation is that if you repeat the experiment very many times, and every time determine the confidence interval in the same way as you did for the observed data, then the true value of θ falls inside the computed intervals for 95% of the replicate experiments. Note that this says nothing about the confidence for the current θ, but instead refers to the long-run performance of the confidence interval method across many hypothetical experiments.

Exercise 2.1.4 Suppose you learn that the questions are all true/false questions. Does this knowledge affect your prior distribution? And if so, how would this prior in turn affect your posterior distribution?

With true or false questions, zero ability corresponds to guessing, that is, $\theta = .5$. Because negative ability is deeply implausible (unless the questions are deliberately misleading), it makes sense to have a uniform prior that ranges from .5 to 1, and hence has zero mass below .5. Because there is no prior mass below .5, there will also be no posterior mass below 0.5.

2.2 Answers: Prediction

Exercise 2.2.1 Instead of "integrating over the posterior", orthodox methods often use the "plug-in principle"; in this case, the plug-in principle suggest that we predict $p(k^{\text{rep}})$ solely based on $\hat{\theta}$, the maximum likelihood estimate. Why is this a bad idea in general? And can you think of a specific situation in which this may not be so much of a problem?

The plug-in principle ignores uncertainty in θ, and therefore lead to predictions that are overconfident, that is, predictions that are less variable than they should be (?, ?). The overconfidence increases with the width of the posterior distribution. This also means that when the posterior is very peaked, that is, when we are very certain about θ (for instance because we have observed many data), the plug-in principle will only result in very little overconfidence.

2.3 Answers: Sequential Updating

No exercises.

2.4 Answers: Markov Chain Monte Carlo

Exercise 2.4.1 Use Google and list some other scientific disciplines that use Bayesian inference and MCMC sampling.

Bayesian inference is used in almost all scientific disciplines (but we wanted you to discover this yourself).

Exercise 2.4.2 The text reads: "Using MCMC sampling, you can approximate posterior distributions to any desired degree of accuracy". How is this possible?

By drawing more and more MCM samples, the discrepancy between the true distribution and the histogram can be made arbitrarily small. Or, in other words, longer chains yield better approximations.

3 Inferences Involving Binomial Distributions

3.1 Answers: Inferring a Rate

Exercise 3.1.1 Alter the data to $k = 50$ and $n = 100$, and compare the posterior for the rate θ to the original with $k = 5$ and $n = 10$.

When you have more information (i.e., high n) the posterior becomes more peaked. This means that you are more certain about what values for the difference are plausible, and what values are not.

Exercise 3.1.2 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., ten times as many) by changing the `nsamples` variable in Matlab or the `niter` variable 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 estimate of the posterior (i.e., the smoothness of the histograms in the figures)?

The width of the posterior distribution, expressing the uncertainty in the single true underlying rate, is controlled by the available information in the data. Thus, higher n leads to narrower posterior distributions. The quality of the estimate, visually evident by the smoothness of the posterior histogram, is controlled by how many samples are collected to form the approximation. Note that these two aspects of the analysis are completely independent. It is possible to have many data but just collect a few samples in a quick data analysis, to get a crude approximation to a narrow posterior. Similarly, it is possible to have only a few data, but collect many samples, to get a very close approximation to a very broad posterior.

Exercise 3.1.3 Alter the data to $k = 99$ and $n = 100$, and comment on the shape of the posterior for the rate θ.

The posterior distribution is not symmetric, because of the 'edge effect' given by the theoretical upper bound of one for the rate. This goes some way

to demonstrating how a Bayesian posterior distribution can take any form, and certainly does not have to be symmetric, or Gaussian, or in any other simple form.

Exercise 3.1.4 Alter the data to $k = 0$ and $n = 1$, and comment on what this demonstrates about the Bayesian approach.

The fact that a posterior distribution exists at all shows that Bayesian analysis can be done even when there are very few data. The posterior distribution is very broad, reflecting the large uncertainty following from the lack of information, but nonetheless represents (as always) everything that is known and unknown about the parameter of interest.

3.2 Answers: Difference Between Two Rates

Exercise 3.2.1 Compare the data sets $k_1 = 8$, $n_1 = 10$, $k_2 = 7$, $n_2 = 10$ and $k_1 = 80$, $n_1 = 100$, $k_2 = 70$, $n_2 = 100$.

When you have more information (i.e., high n) the posteriors–for the individual rates, as well as for the difference between them that is of interest—become more peaked. This means that you are more certain about what values for the difference are plausible, and what values are not.

Exercise 3.2.2 Try the data $k_1 = 0$, $n_1 = 1$, $k_2 = 0$, $n_2 = 5$.

The key to understanding the posterior is that you can be relatively sure that θ_2 is small, but you cannot be so sure about the value of θ_1.

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?

In general, point estimates (usually mean, median, or mode) and credible intervals are appropriate when they convey much the same information as would be gained from examining the whole posterior distribution. For example, if the posterior distribution is symmetric and with a small variance, its mean is a good summary of the entire distribution.

3.3 Answers: Inferring a Common Rate

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?

One reasonable reporting strategy here might be to use a measure for central tendency, such as a mean, median, or mode, together with a credible interval, for instance a 95% credible interval.

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?

The analysis wants you to believe that the most plausible value for the common rate is around 0.5. This example highlights that the posterior distributions generated by a Bayesian analysis are conditional on the truth of the observed data, and of the model. If the model is wrong in an important way, the posteriors will be correct for that model, but probably not useful for the real problem. If a single rate really did underly $k_1 = 0$ and $k_2 = 10$ then the rate must be near a half, since it is the most likely way to generate those data. But the basic assumption of a single rate seems problematic. The data suggest that a rate of 0.5 is one of the least plausible values. Perhaps the data are generated by two different rates, instead of one common rate.

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.

The results for these data sets will be exactly the same. Because the model assumes a common rate, both data sets can in fact be re-described as having $k = k_1 + k_2 = 10$, $n = n_1 + n_2 = 20$.

3.4 Answers: Prior and Posterior Prediction

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.

Line plots are for continuous quantities (e.g., rate parameter θ) and bar plots are

for discrete quantities (e.g., success counts of data).

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.

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 "prior successes" and "prior failures", respectively. This means, 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 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 posterior to the prior, they are said to have a "conjugate" relationship. This is valued a lot in analytic approaches to Bayesian inference, because it makes for tractable calculations. It is not so important for that reason in computational approaches, because sampling methods can handle easily 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.

Exercise 3.4.3 Predictive distributions are not restricted to exactly the same experiment as the observed data, but for any experiment where the inferred model parameters make predictions. In the current simple Binomial setting, for example, predictive distributions could be found by a new experiment with $n' \neq n$ observations. Change the graphical model, and Matlab or R code, to implement this more general case.

The script `Rate_4_answer.txt` implements the modified graphical model.

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

```
    # Prior Predictive
    thetaprior ~ dbeta(1,1)
    priorpredk ~ dbin(thetaprior,npred)
}
```

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 confidently 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, 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?

The 95% credible interval on the predicted number of bullied elderly (out of a total of 121) ranges from approximately (depending on sampling) 13 to 38. This means that the percentage varies from $13/121 \approx 10.7\%$ to $38/121 \approx 31.4\%$. This is about a 20% spread, considerably more than Trompetter estimated.

3.5 Answers: Posterior Prediction

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?

There is only one parameter, the rate θ, but there are two data, the success counts k_1 and k_2.

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?

The posterior predictive mass, shown by the squares, is very small for the actual outcome of the experiment, shown by the cross. The posterior prediction is concentrated on outcomes (around $k_1 = k_2 = 5$) that are very different from the data, and so the model does not seem descriptively adequate.

Exercise 3.5.3 What can you conclude about the parameter θ?

If the model is a good one, the posterior distribution for θ indicates that it is somewhere between about 0.2 and 0.8, and most likely around 0.5. *But*, it seems

unlikely the model is a good one, and so it is not clear anything useful can be concluded about θ.

3.6 Answers: Joint Distributions

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 red 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?

In general, it seems unhelpful to have a point summary that is not a plausible estimate of the true underlying value. One way to think about this result is in terms of the goal of the point estimate. The mean in this example is trying to minimize squared loss to the true value, and the possible values follow a curved surface, causing it to lie in the interior. Another way to think about the location of the mean is physically. It is the center of mass of the joint posterior (i.e., the place where you would put your finger to make the curved scatterplot balance). More mundanely, the expectation of the joint posterior is (by mathematical fact) the combination of the expectations for each parameter taken independently. Looking at the marginal posteriors, it is clear why the cross lies where it does.

Exercise 3.6.2 The green 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?

This estimate seems to be more useful, at least in the sense that it falls on values that are plausible. In fact, it falls on the values with the highest density in the (estimated) posterior. Think of it as sitting on top of the hill surface traced out by the scatterplot. Nonetheless, it still seems unwise to try and summarize the complicated and informative curved pattern shown by the joint posterior scatterplot by a single set of values.

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 expected point, and the maximum likelihood point? If you were comfortable with the mode, are you still comfortable?

The minor change to the data hardly affects the mean, but greatly shifts

the mode. This shows that the mode can be very sensitive to the exact information available, and is a non-robust summary in that sense. Metaphorically, the hill traced out by the joint density scatterplot has a 'ridge' running along the top that is very flat, and the single highest point can move a long way if the data are altered slightly.

Exercise 3.6.4 If you look at the sequence of samples in WinBUGS, 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 to just 500. How is the sequence of samples visually different with thinning?

With thinning, the sequence of samples no longer shows the visual pattern of autocorrelation, as resembles more of a 'block' than a 'curve'. One colorful description of the ideal visual appearance of samples is as a 'fat hairy caterpillar'. Thinning is needed in this example to achieve that type of visual appearance.

4 Inferences Involving Gaussian Distributions

4.1 Answers: Inferring Means and Standard Deviations

Exercise 4.1.1 Try a few data sets, varying what you expect the mean and standard deviation to be, and how many data you observe.

As usual, posterior distributions become more peaked the more data you observe. The posterior distribution for μ should be located around the sample average. Highly variable numbers lead to a low precision λ, that is, a high standard deviation σ. Note that with many data points, you may estimate the standard deviation σ quite accurately (i.e., the posterior for σ can be very peaked). In fact, with an infinite number of data, the posterior distribution converges to a single point. This happens independently of whether the standard deviation σ is large or small; for instance, after observing a large sequence of highly variable data you can be relatively certain that the standard deviation is very high.

Exercise 4.1.2 Plot the *joint* posterior of μ and σ. Interpret the plot.

There is a tendency for the joint posterior to be U-shaped. This is because extreme values of μ are only plausible when σ is high.

Exercise 4.1.3 Suppose you knew the standard deviation of the Gaussian was 1.0, but still wanted to infer the mean from data. This is a realistic question: For example, knowing the standard deviation might amount to knowing the noise associated with measuring some psychological trait using a test instrument. The x_i values could then be repeated measures for the same person, and their mean the trait value you are trying to infer. Modify the WinBUGS script and Matlab or R code to do this. What does the revised graphical model look like?

The script can be adjusted in several ways. The easiest is probably just to replace the statement `x[i]` $\sim$ `dnorm(mu,lambda)` with `x[i]` $\sim$ `dnorm(mu,1)`. In the graphical model. This change means that the node for σ is now shaded, because σ is no longer an unknown quantity that needs to be inferred.

Exercise 4.1.4 Suppose you knew the mean of the Gaussian was zero, but wanted

to infer the standard deviation from data. This is also a realistic question: Suppose you know the error associated with a measurement is unbiased, so its average or mean is zero, but you are unsure how much noise there is in the instrument. Inferring the standard deviation is then a sensible way to infer the noisiness of the instrument. Once again, modify the WinBUGS script and Matlab or R code to do this. Once again, what does the revised graphical model look like?

Again, the script can be adjusted in several ways. Again, the easiest is probably just to replace the statement `x[i]` $\sim$ `dnorm(mu,lambda)` with `x[i]` $\sim$ `dnorm(0,lambda)`. In the graphical model, this change means that the node for μ is now shaded, because μ is no longer an unknown quantity that needs to be estimated. Follow-up question: if you set μ to zero as suggested above, WinBUGS still provides a trace plot for μ. Why?

4.2 Answers: The Seven Scientists

Exercise 4.2.1 Draw posterior samples using the Matlab or R code, and reach conclusions about the value of the measured quantity, and about the accuracies of the seven scientists.

The posterior distributions for most standard deviations are very skewed. As a result, the posterior mean will be dominated by relatively low proportion of extreme values. For this reason, it is more informative to look at the posterior median. As expected, the first two scientists are pretty inept measurers and have high estimates of sigma. The third scientist does better than the first two, but also appears more inept than the remaining four.

Exercise 4.2.2 Change the graphical model in Figure 4.2 to use a uniform prior over the standard deviation, as was done in Figure 4.1. Experiment with the effect the upper bound of this uniform prior has on inference.

This exercise requires you to put a uniform distribution on sigma, so that the code needs to read (for an upper bound of 100): `sigma[i]`$\sim$ `dunif(0,100)`. Then `lambda[i] <-1/pow(sigma[i],2)`. Note that this change also requires that you change the Matlab or R code to assign initial values to sigma instead of lambda, because now sigma is assigned a prior and `lambda` is calculated deterministically from `sigma`, instead of the other way around.

When you make these changes you can see that the difference between the scientists is reduced. To get a more accurate idea of what is going on you may want to set the number of MCMC samples to 100,000 (and, optionally,

set a thinning factor to 10, so that only every tenth sample is recorded – this reduces the autocorrelation in the MCMC chains). As before, posterior median for the first scientist is largest, followed by that of numbers two and three.

4.3 Answers: Repeated Measurement of IQ

Exercise 4.3.1 Use the posterior distribution for each person's μ_i to estimate their IQ. What can we say about the precision of the IQ test?

The posterior means for the `mu` parameters are very close to the sample means. The precision is $1/\sigma^2$, and because the posterior for `sigma` is concentrated around 6 the posterior precision is concentrated around $1/36 \approx 0.03$.

Exercise 4.3.2 Now, use a more realistic prior assumption for the μ_i means. Theoretically, IQ distributions should have a mean of 100, and a standard deviation of 15. This corresponds to having a prior of `mu[i] ~dnorm(100,.0044)`, instead of `mu[i] ~dunif(0,300)`, because $1/15^2 = 0.0044$. Make this change in the WinBUGS script, and re-run the inference. How do the estimates of IQ given by the means change? Why?

Parameter `mu[3]` is now estimated to be around 150, which is 5 points lower than the sample mean. Extremely high scores are tempered by the prior expectation. That is, an IQ of 150 is much more likely, according to the prior, than an IQ of 160. The same strong effect of the prior on inference is not evident for the other people, because their IQ scores have values over a range for which the prior is (relatively) flat.

Exercise 4.3.3 Repeat both of the above stages (i.e., using both priors on μ_i) with a new, but closely related, data set that has scores of $(94, 95, 96)$, $(109, 110, 111)$, and $(154, 155, 156)$. How do the different prior assumptions affect IQ estimation for these data. Why does it not follow the same pattern as the previous data?

The tempering effect of prior expectation has now disappeared, and even under realistic prior assumptions the posterior means for `mu` are close to the sample means. This happens because the data suggest that the test is very accurate, and accurate data are more robust against the prior. One helpful way to think about this is that the IQ test is now more informative (because it measures more accurately), and that extra information now overwhelms the prior. Notice how this example shows that is is not necessarily *more data* that is needed to

remove the influence of priors, but rather *more information*. Often, of course, the best way to get more information is to collect more data. But, another way is to develop data that are more precisely measured, or in some other way more informative.

5 Some Examples Of Basic Data Analysis

5.1 Answers: Pearson Correlation

Exercise 5.1.1 The second data set in the Matlab and R code is just the first data set from Figure 5.2 repeated twice. Interpret the differences in the posterior distributions for r for these two data sets.

With more data the posterior distribution for r becomes more peaked, showing that there is less uncertainty about the true correlation coefficient when more information is available.

Exercise 5.1.2 The current graphical model assumes that the values from the two variables—the $\boldsymbol{x}_i = (x_{i1}, x_{i2})$—are observed with perfect accuracy. When might this be a problematic assumption? How could the current approach be extended to make more realistic assumptions?

Very often in psychology, as with all empirical sciences, data are not measured with arbitrary precision. Other than nominal or ordinal variables (gender, color, occupation, and so on), most variables are measured imperfectly. Some, like response time, might be quite precise, consistent with measurement in the physical sciences. Others, like IQ, or personality traits, are often very imprecise. The current model makes the assumption that these sorts of measurements are perfectly precise. Since they are the basis for the correlation coefficient, the inference understates the uncertainty, and could lead to conclusions that are too confident, or otherwise inappropriate. The next section shows one approach to extending the model to address this problem.

5.2 Answers: Pearson Correlation With Uncertainty

Exercise 5.2.1 Compare the results obtained in Figure 5.4 with those obtained earlier using the same data in Figure 5.2, for the model without any account of uncertainty in measurement.

The posterior distributions are (suprisingly, perhaps) quite similar.

Exercise 5.2.2 Generate results for the second data set, which changes $\sigma_2^e = 10$ for the IQ measurement. Compare these results with those obtained assuming $\sigma_2^e = 1$.

These results are very different. Allowing the large (but perhaps plausibly large, depending on the measurement instrument) uncertainty in the IQ data introduces large uncertainty into inference about the correlation coefficient. Larger values are more likely, but all possible values, including negative correlations, remain plausible. Note also that the expectation of this posterior is the same as in the case where the uncertainty of measurement is low or non-existent. This is a good example of the need to base inference on posterior distributions, rather than point estimates.

Exercise 5.2.3 The graphical model in Figure 5.3 assumes the uncertainty for each variable is known. How could this assumption be relaxed to the case where the uncertainty is unknown?

Statistically, it is straightforward to extend the graphical model, making the $\boldsymbol{\sigma}^e$ variables into parameters with prior distributions, and allowing them to be inferred from data. Whether the current data would be informative enough about the uncertainty of measurement to allow helpful inference is less clear. It might be that different sorts of data, like repeated measurements of the same people's IQs, are needed for this model to be effective. But is it straightforward to implement.

Exercise 5.2.4 The graphical model in Figure 5.3 assumes the uncertainty for each variable is the same for all observations. How could this assumption be relaxed to the case where, for examples, extreme IQs are less accurately measured than IQs in the middle of the standard distribution?

The basic statistical idea would be to model the σ_{i2}^e variables, representing the ith persons error of measurement in their IQ score as a functions of μ_{i2}, representing their IQ itself. This would express a relationship between where people lie on the IQ scale, and how precisely their IQ can be measured. Whatever relationship is chosen is itself a statistical model, formalizing assumptions about this relationship, and so can have parameters that are given priors and inferred from data.

5.3 Answers: The Kappa Coefficient of Agreement

Exercise 5.3.1 *Influenza Clinical Trial* Poehling et al. (2002) reported data evaluating a rapid bedside test for influenza using a sample of 233 children hospitalized with fever or respitory symptoms. Of the 18 children known to have influenza, the surrogate method identified 14 and missed 4. Of the 215 children known not to have influenza, the surrogate method correctly rejected 210 but falsely identified 5. These data correspond to $a = 14$, $b = 4$, $c = 5$, and $d = 210$. Plot posterior distributions of the interesting variables, and reach a scientific conclusion. That is, pretend you are a consultant for the clinical trial. What would your two- or three-sentence 'take home message' conclusion be to your customers?

The surrogate method does a better job detecting the absence of influenza than it does detecting the presence of influenza. The 95% Bayesian confidence interval for kappa is $(.51, .84)$, suggesting that the test is useful.

Exercise 5.3.2 *Hearing Loss Assessment Trial* Grant (1974) reported data from a screening of a pre-school population intended to assess the adequacy of a school nurse assessment of hearing loss in relation to expert assessment. Of those children assessed as having hearing loss by the expert, 20 were correctly identified by the nurse and 7 were missed. Of those assessed as not having hearing loss by the expert, 417 were correctly diagnosed by the nurse but 103 were incorrectly diagnosed as having hearing loss. These data correspond to $a = 20$, $b = 7$, $c = 103$, $d = 417$. Once again, plot posterior distributions of the interesting variables, and reach a scientific conclusion. Once again, what would your two- or three-sentence 'take home message' conclusion be to your customers?

Compared to the expert, the nurse displays a bias to classify children as having hearing loss. In addition, the nurse misses 7 out of 27 children with hearing loss. The nurse is doing a poor job, and this is reflected in the 95% credible interval for kappa of (approximately, up to sampling) $(.12, .29)$.

Exercise 5.3.3 *Rare Disease* Suppose you are testing a cheap instrument for detecting a rare medical condition. After 170 patients have been screened, the test results shower 157 did not have the condition, but 13 did. The expensive ground truth assessment subsequently revealed that, in fact, none of the patients had the condition. These data correspond to $a = 0$, $b = 0$, $c = 13$, $d = 157$. Apply the kappa graphical model to these data, and reach a conclusion about the usefulness of the cheap instrument. What is special about this data set, and what does it demonstrate about the Bayesian approach?

The posterior mean for kappa is approximately .05, with a 95% credible interval of approximately $(0, .24)$. The data are noteworthy because the disease has never been observed, so there are two zero cells, and a zero column sum. This poses a challenge for frequentist estimators. In order to deal with the problem of zero counts a frequentist may add a "1" to each cell in the design, but this amounts to fabricating data. An attractive property of the Bayesian approach is that it is always possible to do the analysis.

5.4 Answers: Change Detection in Time Series Data

Exercise 5.4.1 Draw the posterior distributions for the change-point, the means, and the common standard deviation.

When you look at the trace plots, you may see that it takes a few samples for the chains to lose their dependence on the initial value that was used as a starting point. These initial values are non-representative outliers, and they also stretch out the y-axis of the trace plots. In the call to bugs, set burn-in to 10 and observe the change. We discuss this issue in detail in the chapter on convergence.

With respect to the posterior distributions, it is worthwhile to note that the key parameter tau is estimated relatively precisely around 732. One of the reasons for this is that μ_1 and μ_2 are relatively easy to tell apart.

Exercise 5.4.2 Figure 5.7 shows the mean of the posterior distribution for the change-point (this is the point in time where the two horizontal lines meet). Can you think of a situation in which such a plotting procedure can be misleading?

One case in which this procedure may be misleading is when the posterior distribution is relatively wide (i.e., not peaked around its mean); in such a situation, there is a lot of uncertainty about the location of the change-point, and the plotting procedure, based on a point estimate, falsely suggests that the location is determined precisely.

Exercise 5.4.3 Imagine that you apply this model to a data set that has two change-points instead of one. What could happen?

In this case the model is seriously misspecified. The model assumes that there are two regimes, but in reality there are three. One thing that could happen is that the model groups together the two adjacent regimes that are most similar

to each other and treats them as one. The problems that result should be visible from mismatches between the posterior predictive distribution and the data.

5.5 Answers: Censored Data

Exercise 5.5.1 Do you think Cha Sa-soon could have passed the test by just guessing?

It is unlikely that Cha Sa-soon was just guessing. First, the posterior distribution for `theta` is relatively peaked around .40, whereas chance performance in a four-choice task is only 0.25. Second, the probability of scoring 30 or more correct answers when guessing equals .00000016 (in R: `1-pbinom(29,50,.25)`). With this success probability, the number of attempts to pass the exam follows a geometric distribution. Therefore we know that when guessing, the average number of attempts equals $1/.00000016 \approx 6,097,561$, considerably more than Cha Sa-soon required. The probability of guessing and "only" needing 950 attempts is a relatively low .00016 (in R: `pgeom(950, prob=.00000016)`).
In contrast, with a `theta` of .4 the the probability of scoring 30 or more correct answers equals .0034 (in R: `1-pbinom(29,50,.40)`). With this probability, the associated expected number of attempts until success is 294, and the probability of passing the exam and "only" needing 950 attempts is a relatively high 0.96.

Exercise 5.5.2 What happens when you increase the interval in which you know the data are located, from 15–25 to something else?

Increasing the interval increases the posterior uncertainty for `theta`.

Exercise 5.5.3 What happens when you decrease the number of failed attempts?

When the number of failed attempts becomes low (say 20), the posterior for `theta` becomes wider and shifts to values that are somewhat higher.

Exercise 5.5.4 What happens when you increase Cha Sa-soon's final score from 30?

Not that much! Apparently, the extra information about Cha Sa-soon's final score is much less informative than the knowledge that she had failed 949 times (and with scores ranging from 15 to 25).

Exercise 5.5.5 Do you think the assumption that all of the scores follow a Binomial distribution with a single rate of success is a good model for these data?

It seems to be a poor model. The chance of the same underlying rate generating all of the censored scores below 25, and then producing the 30, can be calculated according to the model, and is tiny. Alternative models would assume some sort of change in the underlying rate. This could psychologically correspond to learning, for at least some of the problems in the test, at some point in the sequence of 950 attempts.

6 Exams, Quizzes, Latent Groups, and Missing Data

6.1 Answers: Exam Scores

Exercise 6.1.1 Draw some conclusions about the problem from the posterior distribution. Who belongs to what group, and how confident are you?

Inspection of the `z[i]` nodes confirms that the first five people are confidently assigned to the guessing group, and the remaining ten people are confidently assigned to the knowledge group. The high confidence is clear from the fact that the posteriors for `z[i]` are approximately located either at 0 or 1.

Exercise 6.1.2 The initial allocations of people to the two groups in this code is random, and so will be different every time you run it. Check that this does not affect the final results from sampling.

This is easily done by running the code again a few times.

Exercise 6.1.3 Include an extra person in the exam, with a score of 28 out of 40. What does their posterior for z tell you?

Performance of the new participant is completely ambiguous. The posterior for z is approximately 0.5, indicating that this participant is as likely to belong to the guessing group as they are to belong to the knowledge group.

Exercise 6.1.4 What happens if you change the prior on the success rate of the second group to be uniform over the whole range $(0, 1)$, and so allow for worse-than-guessing performance?

Nothing much. While the original prior assumption makes more sense, since it captures information we have about the problem, the data are sufficiently informative that inference is not significantly affected by this change.

Exercise 6.1.5 What happens if you change the initial expectation that everybody is equally likely to belong to either group, and have an expectation that people generally are not guessing, with (say), $z_i \sim \text{Bernoulli}\big(0.9\big)$?

This expectation does not change the classification of the first 15 participants much, because these participants are unambiguous in terms of their performance. However, the new participant with a score of 28 is inferred to be in the knowledge group with probability 0.9, whereas this was 0.5 before. Because the data for this participant are ambiguous it is the prior expectation that largely determines how this participant is classified.

6.2 Answers: Exam Scores With Individual Differences

Exercise 6.2.1 Compare the results of the hierarchical model with the original model that did not allow for individual differences.

For the first 15 participants the results are essentially unchanged. The new participant with a score of 28 is now inferred to be in the knowledge group with probability 0.8, compared to the original 0.5. This happens because the new participant is more likely to be a low-knowledge member of the knowledge group than a member of the guessing group. The fact that the current model allows for individual differences helps it account for the relatively low score of 28.

Exercise 6.2.2 Interpret the posterior distribution of the variable `predphi`. How does this distribution relate to the posterior distribution for `mu`?

The variable `predphi` is based on a draw from a Gaussian distribution with mean `mu` and standard deviation `sigma`. This predictive distribution indicates what we can expect about the success rate of a new, as yet unobserved participant from the knowledge group. If there were no individual differences, `sigma` would be zero and draws for `predphi` are effectively draws from the posterior of `mu`. But because there are individual differences, this adds uncertainty to what we can expect for a new participant and hence the posterior distribution for `predphi` is wider than that of `mu`.

Exercise 6.2.3 What does the posterior distribution for the variable `theta[1,2]` mean?

Participant 1 clearly belongs to the guessing group. In the samples, this participant is almost never assigned to the knowledge group. The value for `theta[1,2]` is therefore not directly informative. It is like saying "`theta[1,2]` is what we expect the success rate to be *if* participant 1 was in the knowledge group." Perhaps the only sense in which this is useful information is if it is conceived as hypothetically how the participant would have performed if they had listened in class and put themselves in the knowledge group.

Exercise 6.2.4 In what sense could the latent assignment of people to groups in this case study be considered a form of model selection?

There are two rival explanations, specified as statistical models, for the data. These are the guessing or knowledge-based responding accounts. When a participant is assigned to the guessing group this means that, for that particular participant, we believe the guessing model is a better explanation for the observed data than is the knowledge-based model.

6.3 Answers: Twenty Questions

Exercise 6.3.1 Draw some conclusions about how well the various people listened, and about the difficulties of the various questions. Do the marginal posterior distributions you are basing your inference on seem intuitively reasonable?

This question is best answered by tallying the total number of correct responses separately for each participant and for each item. The result show that, first, participants who answer most items correctly have the highest estimated values for p, and, second, items answered correctly most often have the highest estimated values for q. These results are consistent with intuition.

Exercise 6.3.2 Now suppose that three of the answers were not recorded. Think of a Scantron[1] with coffee spilled on it being eaten by a dog. This gives the new data set shown in Table 6.2. Bayesian inference will automatically make predictions about these missing values (i.e., "fill in the blanks") by using the same probabilistic model that generated the observed data. Missing data are entered as `nan` ("not a number") in Matlab, and `NA` ("not available") in R or WinBUGS. Including the variable `k` as one to monitor when sampling will then provide posterior values for the missing values. That is, it provides information about the relative likelihood of the missing values being each of the possible alternatives, using the statistical model and the available data. Look through the Matlab or R code to see how all of this is implemented in the second dataset. Run the code, and interpret the posterior distributions for the three missing values. Are they reasonable inferences? Finally, think of a more realistic application for inferring missing values in cognitive modeling than dogs eating coffee flavored Scantrons.

The estimates are reasonable. One of the nice things about Bayesian inference is that, given that the model is appropriate, the estimates are *always* reasonable. Sometimes the reasonableness may be hidden from your intuition,

[1] A machine-readable form on which students mark answers to academic test questions.

but this just means that your intuition was faulty. Consider person 1 on item M. We know that person 1 is relatively attentive, because they answer relatively many questions, and we also know that item M is relatively easy, because many other people answer item M correctly. This item-person combination looks like it could have resulted in a correct answer. The inferred probability for M-1 being correct is approximately 0.74. For item-person combination E-8 the reverse holds. The item is difficult and the participant is inattentive. Consequently, the inferred probability for E-8 being correct is approximately 0.01. Finally, combination R-10 is middle-of-the-road on both dimensions, and the inferred probability for it being correct is 0.41.

These inferred probabilities are directly related to the knowledge about each participant and item. In fact, if you multiply the estimated p's and q's you can recover the inferred probabilities. For instance, `p[1]` is approximately 0.88, and `q[13]` (the M) is approximately 0.84. The multiplication of these probabilities yields .74. The same calculations may be performed for the other missing item-participant calculations.

The last part of the question deals with plausible scenarios for missing data in cognitive modeling. Participants in speeded response time experiments may not answer fast enough, in designs where the trial is terminated without a response and the next trial is presented; people answering a questionnaire may fail to answer a specific question because they overlooked it; participants in neuroscientific experiments may blink their eyes, distorting the signal and leading to the removal of the trial; participants in longitudinal experiments may quit the study prematurely; participants who are asked to monitor some aspect of their lives at regular intervals over several days may sometimes forget to register their response, and so on and so forth.

In general, data can be missing in several ways. When data are missing "completely at random" it is easiest handled. It is more difficult when there is a relationship between the missing-ness and the parameters of interest. For example, say person 10 does not complete item Q because he or she realizes the answer may well be wrong and time is better spend answering easier items. To handle this situation we need more complicated models of the missing-ness. Bayesian models, of course.

6.4 Answers: The Two Country Quiz

Exercise 6.4.1 Interpret the posterior distributions for `x[i]`, `z[j]`, `alpha` and `beta`. Do the formal inferences agree with your original intuitions?

Yes, people 1, 2, 5, and 6 form one group, and people 3, 4, 7, and 8 form the other group. The model also groups together questions A, D, E, and H versus questions B, C, F, and G. And the rates of correct decisions for matched and mismatched groups make sense, too.

Exercise 6.4.2 The priors on the probabilities of answering correctly capture knowledge about what it means to match and mismatch, by imposing an order constrain $\alpha \geq \beta$. Change the code so that this information is not included, by using priors `alpha`$\sim$`dbeta(1,1)` and `beta`$\sim$`dbeta(1,1)`. Run a few chains against the same data, until you get an inappropriate, and perhaps counter-intuitive, result. Describe the result, and discuss why it comes about. The result you get from the analysis with uniform prior can change from chain to chain, switching the inferences about `alpha` and `beta`. This is a basic and common problem for mixture models, known as *model indeterminacy*. The probability α is used whenever $x_i = z_j$. If this corresponds to a Thai person answering a Thai question, then α should be high, as we expect. But there is nothing stopping the model, without the order constraint, from coding Thai people as $x_i = 1$ and Moldovan questions as $z_j = 1$, in which case α will be low. Effectively, with this coding, α and β will swap roles. Overall, there are four possibilities (two ways people can be encoded, by two ways questions can be encoded). Our semantics of α being knowledge-based and β being ignorance-based will apply for 2 of these 4 possible encodings, but will be reversed for the other two. The core problem is that α and β are statistically defined the same way in the revised model. This is the indeterminacy. A practical but inelegant way to solve this problem is by being flexible in interpretation. A better way is, as per the original model and code, by defining the statistical model itself more carefully, introducing the order constraint, and removing the indeterminacy.

Exercise 6.4.3 Now suppose that three extra people enter the room late, and begin to take the quiz. One of them (Late Person 1) has answered the first four questions, the next (Late Person 2) has only answered the first question, and the final new person (Late Person 3) is still sharpening their pencil, and has not started the quiz. This situation can be represented as an updated data set, now with missing data, as in Table 6.4. Interpret the inferences the model makes about the nationality of the late people, and whether or not they will get the unfinished questions correct.

Late person 1 is confidently placed in the same category as people 1, 2, 5, and 6. This is also reflected in the probabilities of answering the remaining four questions correctly: 0.88, 0.07, 0.05, 0.90, predicting a "1 0 0 1" pattern that was also observed for people 1, 2, 5, and 6.

Late person 2 only answered a single question, but this information suf-

fices to assign this person with probability .89 to the same category as persons 3, 4, 7, and 8. This is reflected in the probabilities of answering the remaining seven questions correctly: .80, .80, .15, .15, .80, .80, .13 predicting a "1 1 0 0 1 1 0" pattern that was relatively typical for people 3, 4, 7, and 8.

Late person 3 did not answer a single question and is equally likely to belong to either group. Because each group has an opposite pattern of answering any particular question correctly, the model predicts that the performance of late person 3 will be around chance (slightly worse than chance because not all questions are answered correctly even if the question matches the nationality).

Exercise 6.4.4 Finally, suppose that you are now given the correctness scores for a set of 10 new people, whose data were not previously available, but who form part of the same group of people we are studying. The updated data set shown in Table 6.5. Interpret the inferences the model makes about the nationality of the new people. Revisit the inferences about the late people, and whether or not they will get the unfinished questions correct. Does the inference drawn by the model for the third late person match your intuition? There is a problem here. How could it be fixed?

The new people are all classified in the same group as people 1, 2, 5, and 6. However, late person 3 is still equally likely to be classified in either group. This is a problem in the sense that the model is insensitive to changes in baseline proportions: if we know that there are many more people in one category than another this knowledge should affect our prediction for late person 3. In this case, late person 3 is likely to belong to the same category as the new persons. The model can be extended to deal with baseline proportions by changing the line `pz[i]` $\sim$ `dbern(0.5)`, which assumes equal baselines, to `pz[i]` $\sim$ `dbern(phi)`, and `phi` $\sim$ `dbeta(1,1)`, which now estimates the baseline.

6.5 Answers: Latent Group Assessment of Malingering

Exercise 6.5.1 What are your conclusions about group membership? Did all participants follow the instructions?

The expectation of the posterior for the indicator variable z shows everybody to have (essentially) the value 0 or 1. This means there is certainly of the classification into the bona fide and malingering classifications, with 0 corresponding to bona fide and 1 to malingering. The first 10 people, as expected, are bona fide. The rest, with two exceptions, are inferred to be malingerers. The

two exceptions are the 14th and 15th people, who scored 44. It seems likely these people did not follow the instruction to malinger.

6.6 Answers: Individual Differences in Malingering

Exercise 6.6.1 Assume you know that the base rate of malingering is 10%. Change the WinBUGS script to reflect this knowledge. Do you expect any differences?

The change can be made by changing the definition of the indicator variables to `z[i]~dbern(0.1)`, This base rate is different from the one that is inferred for ϕ, which has posterior expectation of about 0.5, so it is reasonable to expect inference to be affected. Specifically, when the base rate of malingering is very low we should be less confident about the classification of participants who performed poorly.

Exercise 6.6.2 Assume you know for certain that participants 1, 2, and 3 are bona fide. Change the code to reflect this knowledge.

This change can be made by defining `z[1] <- 0`, `z[2] <- 0` and `z[3] <- 0`. It is important, once this change is made, to be sure that initial values are not set for these three parameters, since they are no longer stochastic variables to be inferred. This is not always straightforward, in terms of the data structures being passed to and from WinBUGS from Matlab or R. An effective solution is to define `z[4] <- ztmp[1]` to `z[22] <- ztmp[19]`, and set the initial values directly on the complete `ztmp` vector.

Exercise 6.6.3 Suppose you add a new participant. What number of questions answered correctly by this participant would lead to the greatest uncertainty about their group membership?

A little trial-and-error experimentation finds that an extra score of 41 questions correct leads to a posterior expectation of about 0.45. This is more uncertain than the neighboring possibilities of 40 questions correct, with posterior expectation about 0.8, and 42 questions correct, with posterior expectation about 0.15.

Exercise 6.6.4 Try to solve the label-switching problem by using the `dunif(0,mu_bon)` approach to specifying a joint prior instead of the logit transform.

To be written

Exercise 6.6.5 Why do you think the priors for λ_{b} and λ_{m} are different?

Conceptually, it might be reasonable to expect less variability for the bona fide group. Doing the task correctly seems more constraining than the freedom to simulate amnesia and malinger. Computationally, the lack of variability in the bona fide scores can cause under- and over-flow issues in sampling, and limiting the priors to reasonable and computational ranges is a practical approach to address this issue.

6.7 Answers: Alzheimer's Recall Test Cheating

Exercise 6.7.1 Suppose the utilities are very different, so that a false-alarm costs \$100, because of the risk of litigation in a false accusation, but misses are relatively harmless, costing \$10 in wasted administrative costs. What decisions should be made about bona fide and malingering now?

If it is 10 times more important not to make false-alarms (i.e., \$100 vs \$10), then you should only treat someone as a cheater if that is 10 times more likely. This means the expectation of z_i should be less than 0.1 before the decision is made to classify them as having cheated. It is clear from Figure 6.9 that his applies only to the few people who scored 39 or 40 on the test.

Exercise 6.7.2 What other potential information, besides the uncertainty about classification, does the model provide? Give at least one concrete example.

The model provides information about the base rate of cheating, as well as information about the levels of performance, and variability in that performance, for the different groups of people, By providing a complete model of the data-generating process, Bayesian inference is able to provide a much more complete analysis of the data than a simple set of classifications.

References

Agresti, A. (1992). Modelling patterns of agreement and disagreement. *Statistical Methods in Medical Research*, *1*, 201–218.

Anscombe, F. J. (1963). Sequential medical trials. *Journal of the American Statistical Association*, *58*, 365–383.

Banerjee, M., Capozzoli, M., McSweeney, L., & Sinha, D. (1999). Beyond kappa: A review of interrater agreement measures. *The Canadian Journal of Statistics*, *27*, 3–23.

Basu, S., Banerjee, M., & Sen, A. (2000). Bayesian inference for kappa from single and multiple studies. *Biometrics*, *56*, 577–582.

Bem, D. J. (2011). Feeling the future: Experimental evidence for anomalous retroactive influences on cognition and affect. *Journal of Personality and Social Psychology*, *100*, 407–425.

Brooks, S. P., & Gelman, A. (1997). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–455.

Casella, G., & George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, *46*, 167–174.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*, 37–46.

de Finetti, B. (1974). *Theory of Probability, Vol. 1 and 2*. New York: John Wiley & Sons.

Donner, A., & Wells, G. (1986). A comparison of confidence interval methods for the intraclass correlation coefficient. *Biometrics*, *42*(2), 401–412.

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). *Statistical methods for rates and proportions* (Third ed.). New York: Wiley.

Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. Boca Raton, FL: Chapman & Hall/CRC.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*, 398–409.

Gelman, A. (1996). Inference and monitoring convergence. In W. R. Gilks, S. Richardson, & D. J. Spiegelhalter (Eds.), *Markov chain Monte Carlo in practice* (pp. 131–143). Boca Raton (FL): Chapman & Hall/CRC.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, *1*, 515–534.

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge: Cambridge University Press.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, *7*, 457–472.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice*. Boca Raton (FL): Chapman & Hall/CRC.

Grant, J. A. (1974). Evaluation of a screening program. *American Journal of Public Health*, *64*, 66–71.

Griffiths, T. L., Kemp, C., & Tenenbaum, J. B. (2008). Bayesian models of cognition. In R. Sun (Ed.), *Cambridge Handbook of Computational Cognitive Modeling* (pp. 59–100). Cambridge, MA: Cambridge University Press.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge, UK: Cambridge University Press.

Kraemer, H. C. (1992). Measurement of reliability for categorical data in medical research. *Statistical Methods in Medical Research*, *1*, 183–199.

Kraemer, H. C., Periyakoil, V. S., & Noda, A. (2004). Kappa coefficients in medical research. In R. B. D'Agostino (Ed.), *Tutorials in biostatistics volume 1: Statistical methods in clinical studies*. New York: Wiley.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*, 159–174.

Lindley, D. V. (1972). *Bayesian Statistics, A Review*. Philadelphia (PA): SIAM.

Lunn, D. J., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, *28*, 3049–3067.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*, 90–100.

Poehling, K. A., Griffin, M. R., & Dittus, R. S. (2002). Bedside diagnosis of influenzavirus infections in hospitalized children. *Pediatrics*, *110*, 83–88.

Poirier, D. J. (2006). The growth of bayesian methods in statistics and economics since 1970. *Bayesian Analysis*, *1*, 969–980.

Sheu, C.-F., & O'Curry, S. L. (1998). Simulation–based Bayesian inference using BUGS. *Behavioral Research Methods, Instruments, & Computers*, *30*, 232–237.

Shrout, P. E. (1998). Measurement reliability and agreement in psychiatry. *Statistical Methods in Medical Research*, *7*, 301–317.

Smithson, M. (2010). A review of six introductory texts on Bayesian methods. *Journal of Educational and Behavioral Statistics*, *35*, 371–374.

Uebersax, J. S. (1987). Diversity of decision-making models and the measurement of interrater agreement. *Psychological Bulletin*, *101*, 140–146.

Index