Chapter 1

Introduction to Bayesian Statistics and R

Bayesian inference is the process of fitting probability model to a set of data and summarizing the result by a probability distribution on the parameters of the model and on unobserved quantities such as prediction for new observation.

By Bayesian data analysis, we mean practical methods for making inferences from data using probability models for quantities we observed and for quantities about which we wish to learn. The essential characteristic of Bayesian methods is their explicit use of probability for quantifying uncertainty in inferences based on statistical data analysis.

The process of Bayesian data analysis can be idealized by dividing it into the following three steps:

1. Setting up a full probability model — a joint probability distribution for all observable and unobservable quantities in a problem. The model should be consistent with knowledge about the underlying scientific problem and the data collection process.

2. Conditioning on observed data — calculating and interpreting the appropriate posterior distribution — the conditional probability distribution of the unobserved quantities of ultimate interest, given the observed data.

3. Evaluating the fit of the model and the implication of the resulting posterior distribution.
1.1 General notation for statistical inference

Statistical inference is concerned with drawing conclusions, from numerical data, about quantities that are not observed.

For example, a clinical trial of a new cancer drug might be designed to compare the five year survival probability in a population given the new drug with that in a population under standard treatment. These survival probabilities refer to a large population of patients and it is neither feasible nor ethically acceptable to experiment with an entire population. Therefore inferences about the true probabilities and, in particular, their differences must based on a sample of patients.

In this example, even if it were possible to expose the entire population to one or the other treatment, it is obviously never possible to expose anyone to both treatments, and therefore statistical inference would still be needed to assess the causal inference — the comparison between the observed outcome in each patient and that patient's unobserved outcome if exposed to the other treatment.

We distinguish between two kinds of estimands — unobserved quantities for which statistical inferences are made — first, potentially observable quantities, such as future observations of a process, or the outcome under the treatment not received in the clinical trial example; and second, quantities that are not directly observable, that is, parameters that govern the hypothetical process leading to the observed data. The distinction between two kinds of estimands is not always precise, but is generally useful as a way of understanding how a statistical model for a particular problem fit into the real world.

Parameters, data, and prediction

As general notation, we let \( \theta \) denote unobservable vector quantities or population parameters of interest, \( y \) denote the observed data, and \( \tilde{y} \) denote unknown, but potentially observable, quantities. In general these symbols represent multivariate quantities.

When using matrix notation, we consider vectors as column vector throughout; for example, if \( u \) is a vector with \( n \) components, then \( u^T u \) is a scalar and \( uu^T \) an \( n \times n \) matrix.

Observational units and variables

In many statistical studies, data are gathered on each of a set of \( n \) objects or units, and we can write the data as a vector, \( y=(y_1, \ldots, y_n) \). In the clinical
trial example, we might label \( y_i \) as 1 if patient \( i \) is alive after 5 years or 0 if patient dies. If several variables are measured on each unit, then each \( y_i \) is actually a vector, and the entire data set \( y \) is a matrix. The \( y \) variables are called the outcomes and are considered random in the sense that, when making inferences, we wish to allow for the possibility that the observed value of the variables could have turned out otherwise, due to the sampling process and the natural variation of the population.

**Exchangeability**

The usual starting point of a statistical analysis is the assumption that the \( n \) values \( y_i \) may be regarded as exchangeable, meaning that the joint probability density \( p(y_1, \ldots, y_n) \) should be invariant to permutations of the indexes. A nonexchangeable model would be appropriate if information relevant to the outcome were conveyed in the unit indexes rather than by explanatory variables.

Generally, it is useful and appropriate to model data from an exchangeable distribution as independently and identically distributed \( iid \) given some unknown parameter vector \( \theta \) with distribution \( p(\theta) \). In the clinical trial example, we might model the outcome \( y_i \) as \( iid \), given \( \theta \), the unknown probability of survival.

**Explanatory variables**

It is common to have observations on each unit that we do not bother to model as random. In the clinical trial example, such variables might include the age and previous health status of each patient in the study. We call this second class of variables explanatory variables, or covariates, and label them \( x \). We use \( X \) to denote the entire set of explanatory variables for all \( n \) units; if there are \( k \) explanatory variables, then \( X \) is a matrix with \( n \) rows and \( k \) columns. Treating \( X \) as random, the notation of exchangeability can be extended to require the distribution of the \( n \) values of \( (x, y)_i \) to be unchanged by arbitrary permutations of the indexes. It is always appropriate to assume an exchangeable model after incorporating sufficient relevant information in \( X \) that the indexes can be thought of as randomly assigned. It follows from the assumption of exchangeability that the distribution of \( y \), given \( x \), is the same for all units in the study in the sense that if two units have the same value of \( x \), then their distribution of \( y \) are the same. Any of the explanatory variables \( x \) can of course be moved into the \( y \) category if we wish to model them.
Hierarchical modeling

Hierarchical models, which are used when information is available on several different levels of observation units. In a hierarchical model, it is possible to speak of exchangeability at each level of units. For example, suppose two medical treatments are applied, in separate randomized experiments, to patients in several different cities. Then, if no other information is available, it would be reasonable to treat the patients within each city as exchangeable and also treat the results from different cities as themselves exchangeable. In practice it would make sense to include, as explanatory variables at the city levels, whatever relevant information we have on each city, as well as the explanatory variables mentioned before at the individual level, and then the conditional distribution given these explanatory variables would be exchangeable.

1.2 Bayesian inference

Bayesian statistical conclusion about a parameter \( \theta \), or unobserved data \( \tilde{y} \), are made in terms of probability statements. These probability statements are conditional on the observed value of \( y \), and in our notation are written simply as \( p(\theta | y) \) or \( p(\tilde{y} | y) \). We also implicitly condition on the known values of any covariates, \( x \). It is at the fundamental level of conditioning on observed data that Bayesian inference depart from the approach to statistical inference, which is based on a retrospective evaluation of the procedure used to estimate \( \theta \) (or \( y \)) over the distribution of possible \( y \) values conditional on the true unknown value of \( \theta \). Despite this difference, it will be seen that in many simple analysis, superficially similar conclusions result from the two approaches to statistical inference. However, analysis obtained using Bayesian methods can be easily extended to more complex problems.

Bayes’ rule

In order to make probability statements about \( \theta \) given \( y \), we must begin with a model providing a joint probability distribution for \( \theta \) and \( y \). The joint probability mass of density function can be written as a product of two densities that are often referred to as the prior distribution \( p(\theta) \) and the sampling distribution (or data distribution) \( p(y|\theta) \) respectively:

\[
p(\theta, y) = p(\theta)p(y|\theta)
\]

simply conditioning on the known value of the data \( y \), using the basic property on conditional probability known as Bayes’ rule, yield the posterior
1.2 Bayesian inference

Density:

\[ p(\theta | y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y | \theta)}{p(y)} \tag{1.2} \]

where \( p(y) = \sum_\theta p(\theta)p(y | \theta) \), and the sum is over all possible value of \( \theta \) (or \( p(y) = \int p(\theta)p(y | \theta) d\theta \) in the case of continuous \( \theta \)). An equivalent form of (1.1) omits the factor \( p(y) \), which does not depend on \( \theta \) and, with fixed \( y \), can thus be considered as constant, yielding the unnormalized posterior density, which is the right side of (1.2):

\[ p(\theta | y) \propto p(\theta)p(y | \theta) \tag{1.3} \]

These simple expression encapsulate the technical core of Bayesian inferences: the primary task of any specific application is to develop the model \( p(\theta, y) \) and perform the necessary computation to summarize \( p(\theta | y) \) in appropriate ways.

**Prediction**

To make inference about an unknown observable, often called predictive inference, we follow a similar logic. Before the data \( y \) are considered, the distribution of the unknown but observable \( y \) is,

\[ p(y) = \int p(y, \theta) d\theta = \int p(\theta)p(y | \theta) d\theta \tag{1.4} \]

This is often called the marginal distribution of \( y \), but a more informative name is the prior predictive distribution: prior because it is not conditional on a previous observation of the process, and predictive because it is the distribution for a quantity that is observable.

After the data \( y \) have been observed, we can predict an unknown observable, \( \hat{y} \), from the same process. For example, \( y = (y_1, \ldots, y_n) \) may be the vector of recorded weights of an object weighed \( n \) times on a scale, \( \theta = (\mu, \sigma^2) \) may be the unknown true weight of the object and the measurement variance of the scale, and \( \hat{y} \) may be the yet to be recorded weight of the object in a planned new weighing. The distribution of \( \hat{y} \) is called the posterior predictive distribution, posterior because it is conditional on the observed \( y \) and predictive
because it is a prediction for an observable $\tilde{y}$:

$$p(\tilde{y}|y) = \int p(\tilde{y}, \theta|y)d\theta$$

$$= \int p(\tilde{y}|\theta, y)p(\theta|y)d\theta$$

$$= \int p(\tilde{y}|\theta)p(\theta|y)d\theta$$  \hspace{1cm} (1.5)

The second and third lines display the posterior predictive distribution as an average of conditional predictions over the posterior distribution of $\theta$. The last equation follows because $y$ and $\tilde{y}$ are conditionally independent given $\theta$ in this model.

**Likelihood**

Using Bayes rule with a chosen probability model means that the data $y$ affect the posterior inference Equation (1.3) only through the function $p(y | \theta)$, which, when regarded as a function of $\theta$, for fixed $y$, is called the likelihood function. In this way Bayesian inference obeys what is sometimes called the likelihood principle, which states that for a given sample of data, any two probability models $p(y | \theta)$ that have the same likelihood function yield the same inference for $\theta$.

The likelihood function is reasonable, but only within the framework of the model or family of models adopted for a particular analysis. In practice, one can rarely be confident that the chosen model is correct model. Sampling distribution plays an important role in checking model assumptions. In fact, our view of an applied Bayesian statistician is one who is willing to apply Bayes’ rule under a variety of possible models.

**Highest posterior density**

A region $R$ in parameter space $\Theta$ is called a highest posterior density region of content $(1-\alpha)$ if

$$p(\theta \in R | y) = 1 - \alpha$$  \hspace{1cm} (1.6)

for,

$$\theta_1 \in R,$$  \hspace{1cm} (1.7)

and

$$p(\theta_1 | y) \geq p(\theta_2 | y)$$  \hspace{1cm} (1.8)

This means that the content of the posterior is $1 - \alpha$ and it is also evident from above statement that the value of the posterior within the HPD region is always greater than outside of it. (Box and Tiao, 1973, p. 123)
1.3 Example: inference about a genetic probability

The following example is not typical of statistical applications of the Bayesian methods, because it deals with a very small amount of data and concerns a single individual’s state (gene carrier or not) rather than with the estimation of a parameter that describe an entire population. Nevertheless it is a real example of the very simplest type of Bayesian calculation, where estimand and the individual item of data each have only two possible values.

Human males have \(X\) chromosomes and \(Y\) chromosomes, whereas females have two \(X\) chromosomes, each chromosome being inherited from one parent. Hemophilia is a disease that exhibits \(X\) chromosomes linked recessive inheritance, meaning that a male who inherits the gene that causes the disease on the \(X\) chromosomes is affected, whereas a female carrying the gene on only one of her two \(X\) chromosomes is not affected. The disease is generally fatal for women who inherit two such gene, and this is very rare, since the frequency occurrence of the gene is low in human populations.

The prior distribution

Consider a woman who has an affected brother, which implies that her mother must be a carrier of the hemophilia gene with one good and one bad hemophilia gene. We are also told that her father is not affected; thus the woman herself has a fifty-fifty chance of having the gene. The unknown quantity of interest, the state of the woman, has just two values: the woman is either a carrier of the gene \(\theta = 1\) or not \(\theta = 0\). Based on the information provided thus far, the prior distribution for the unknown \(\theta\) can be expressed simply as \(p(\theta = 1) = p(\theta = 0) = \frac{1}{2}\).

The model and likelihood

The data used to update this prior information consist of the affection status of the woman’s son. Suppose she has two sons, neither of whom is affected. Let \(y_1 = 1\) or \(0\) denote an affected or unaffected son, respectively. The outcome of the two sons are exchangeable and, conditional of the unknown \(\theta\), are independent; we assume the sons are not identical twins. The two items of independent data generates the following likelihood function:

\[
p(y_1 = 0, y_2 = 0 | \theta = 1) = (0.5)(0.5) = 0.25 \tag{1.9}
\]

\[
p(y_1 = 0, y_2 = 0 | \theta = 0) = (1)(1) = 1 \tag{1.10}
\]
These expression follows from the fact that if the woman is a carrier, then each of her sons will have a 50 percent chance of inheriting the gene and so being affected, whereas if she is not a carrier then there is a probability very close to 1 that a son of hers will be unaffected.

The posterior distribution

Bayes' rule can now be used to combine the information in the data with the prior probability, in particular, interest is likely to focus on the posterior probability that the woman is a carrier. Using \( y \) to denote the joint data \((y_1, y_2)\), this is simply,

\[
p(\theta = 1|y) = \frac{p(y|\theta = 1)p(\theta = 1)}{p(y|\theta = 1)p(\theta = 1) + p(y|\theta = 0)p(\theta = 0)}
\]

\[
= \frac{.25 \times .25}{.24 \times .5 + 1 \times .5}
\]

\[
= \frac{.25}{.625}
\]

\[
= .20
\]

Intuitively it is clear that if a woman has unaffected children, it is less probable that she is a carrier, and Bayes' rule provides a formal mechanism for determining the extend of the correction. The result can also be described in terms of prior and posterior odds. The prior odds of the woman being a carrier are \( 0.5 \). The likelihood ratio based on the information about her two unaffected sons is \( 0.25 \), so the posterior odds are obtained very simply as \( 0.25 \).

1.4 Some useful results from probability theory

For two quantities \( u \) and \( v \), we write the joint density as \( p(u,v) \); if specific values need to be referenced, this notation will be further abused as with, for example, \( p(u,v=1) \).

In Bayesian calculation relating to a joint density \( p(u,v) \), we will often refer to a conditional distribution or density function such as \( p(u|v) \) and a marginal density such as. In this notation either or both \( u \) and \( v \) can be vectors. Typically it will be clear from the context that the range integration in the latter expression refers to the entire range of the variable being integrated out. It is also often useful to factor a joint density as a product of marginal and conditional densities; for example, \( p(u,v,w) = p(u|v,w)p(v|w)p(w) \). We use the standard notation, \( E(.) \) and \( V(.) \), for mean and variance, respectively.
for a vector parameter $u$, the expression for the mean is the same, and the
covariance matrix is defined. where $u$ is considered a column vector(we use
the terms 'variance matrix' and 'covariance matrix' interchangeably). This
notation is slightly imprecise, because $E(u)$ and $\text{var}(u)$ are really functions
of the distribution functions $p(u)$, not of the variable $u$. In an expression
involving an expectation, any variable that does not appear explicitly as a
conditioning variable is assumed to be integrated out in the expectation;
for example, $E(u|v)$ refers to the conditional expectation of $u$ with $v$ held
fixed — that is, the conditional expectation as a function $v$—whereas $E(u)$ is
the expectation of $u$, averaging over $v$ (as well as $u$).

Means and variances of the conditional distributions

It is often useful to express the means and variance of a random variable $u$
in terms of the conditional mean and variance given some related quantity
$v$. The mean of $u$ can be obtained by averaging the conditional mean over
the marginal distribution of $v$,

$$E(u) = E(E(u|v)),$$  \hspace{1cm} (1.11)

where the inner expectation averages over $u$, conditional on $v$, and the outer
expectation averages over $v$. Identity (1.5) is easy to derive by writing the
expectation in terms of the joint distribution of $u$ and $v$ and then factoring
the joint distribution:

$$E(u) = \int \int up(u, v)dudv$$
$$= \int \int up(u|v)dup(v)dv$$
$$= \int E(u|v)p(v)dv$$ \hspace{1cm} (1.12)

The corresponding result for the variance include two terms, the mean of the
conditional variance and the variance of the conditional mean:

$$\text{var}(u) = E(\text{var}(u|v)) + \text{var}(E(u|v))$$ \hspace{1cm} (1.13)

This result can be derived by expanding the terms on the right side of the
above expression:

$$E(\text{var}(u|v)) + \text{var}(E(u|v)) = E[E(u^2|v) - (E(u|v))^2] + \text{var}[(E(u|v))^2] - (E[E(u|v)])^2$$
$$= E(u^2) - E[(E(u|v))^2] + \text{var}[(E(u|v))^2] - (E(u))^2$$
$$= E(u^2) - (E(u))^2$$
$$= \text{var}(u)$$
These identities also hold if \( u \) is a vector, in which case \( E(u) \) is a vector and \( \text{var}(u) \) a matrix.

### 1.5 Summarizing inferences by simulation

Simulation forms a central part of much applied Bayesian analysis, because of the relative ease with which samples can often be generated from a probability distribution, even when the density function cannot be explicitly integrated. In performing simulations, it is helpful to consider the duality between a probability density function and a histogram of a set of random draws from the distribution: given a large enough sample, the histogram can provide practically complete information about the density, and in particular, various sample moments, percentiles, and other summary statistics provide estimates of any aspects of the distribution, to a level of precision that can be estimated. Here we discuss some simulation tools that are:

1. Monte Carlo sampling
2. Gibbs sampler
3. Metropolis Hasting algorithm

#### 1.5.1 Monte Carlo simulation

Analytically summarizing posterior distributions is typically impossible. Over the last twenty years, Bayesian Statisticians have harnessed the Monte Carlo methods (Metropolis and Ulam, 1949 and Metropolis et al. (1953)) to perform this summarization numerically. The name Monte Carlo was coined by Metropolis (inspired by Ulam’s interest in poker) during the Manhattan project of World War II, because of similarity of Statistical simulation to games of chance, and Monte Carlo, the capital of Monaco was a center for gambling. Monte Carlo now refers to any method that utilizes sequences of random numbers to perform Statistical simulation. The main requirement to use Monte Carlo method for simulation of a physical system is that it must be possible to describe the system in terms of probability density function. Once the density function is known, then the simulation begins by random sampling from it. While these methods can be employed to study any distribution, the discussion here will solely on Monte Carlo methods commonly used in Bayesian Statistics. We are interested in learning about the posterior distribution \( p(\theta | y) \), which we will call the target distribution because it is the distribution from which we intend to simulate.
1.5 Summarizing inferences by simulation

Monte Carlo approximation, is based on random sampling and its implementation does not require a deep knowledge of calculus or numerical analysis. Let \( \theta \) be a parameter of interest and let \( y_1, \ldots, y_n \) be the numerical value of a sample from a distribution \( p(y_1, \ldots, y_n|\theta) \). Suppose we could sample some numbers \( S \) of independent, random \( \theta \) values from the posterior distribution \( p(\theta|y_1, \ldots, y_n) \):

\[
\theta^{(1)}, \ldots, \theta^{(S)} \sim \text{i.i.d } p(\theta | y_1, \ldots, y_n)
\]

Then the empirical distribution of the samples \( (\theta^{(1)}, \ldots, \theta^{(S)}) \) would approximated \( p(\theta|y_1, \ldots, y_n) \), with the approximation improving with increasing \( S \). The empirical distribution of \( (\theta^{(1)}, \ldots, \theta^{(S)}) \) is known as a Monte Carlo approximation to \( p(\theta|y_1, \ldots, y_n) \).

The law of large numbers says that if \( \theta^{(1)}, \ldots, \theta^{(S)} \) are i.i.d samples from \( p(\theta|y_1, \ldots, y_n) \), then

\[
\frac{1}{S} \sum_{s=1}^{S} g(\theta^{(s)}) \rightarrow E(g(\theta)|y_1, \ldots, y_n) \rightarrow \int g(\theta)p(\theta|y_1, \ldots, y_n)d\theta
\]

this implies that as \( S \to \infty \)

- \( \bar{\theta} = \frac{1}{S} \sum_{s=1}^{S} \theta^{(s)} \to E[\theta|y_1, \ldots, y_n] \)
- \( \sum_{s=1}^{S} (\theta^{(s)} - \bar{\theta})^2/(S-1) \to \text{var}[\theta|y_1, \ldots, y_n] \)
- \( (\theta^{(s)} \leq c) \to Pr(\theta \leq c|y_1, \ldots, y_n) \)
- the empirical distribution of \( \theta^{(1)}, \ldots, \theta^{(S)} \to p(\theta|y_1, \ldots, y_n) \)
- the median of \( \theta^{(1)} \ldots \theta^{(S)} \to \theta_{1/2} \)
- the \( \alpha \) percentile of \( \theta^{(1)} \ldots, \theta^{(S)} \to \theta_{\alpha} \)

To use Monte Carlo methods to summarize posterior distributions, it is necessary to introduce you to algorithms that are well suited to producing draws from commonly found target distributions. Two algorithms,

- Gibbs sampling
- Metropolis Hastings algorithm

have proven to be very useful for applied Bayesian work.
1.5 Summarizing inferences by simulation

1.5.2 Gibbs sampler

For many multiparameter models the joint posterior distribution is non standard and difficult to sample from directly. However, it is often the case that it is easy to sample from the full conditional distribution of each parameter. In such cases, posterior approximation can be made with Gibbs sampler, an iterative algorithm that constructs a dependent sequence of parameter values whose distribution converges to the target joint posterior distribution.

The distribution \( p(\theta | \sigma^2, y_1 \ldots y_n) \) and \( p(\sigma^2 | \theta, y_1 \ldots, y_n) \) are called the full conditional distributions of \( \theta \) and \( \sigma^2 \), respectively, as they are each a conditional distribution of a parameter given everything else.

General properties of Gibbs sampler

Suppose a vector of parameters \( \phi = (\phi_1, \ldots, \phi_p) \) and your information about \( \phi \) is measured with \( p(\phi) = p(\phi_1, \ldots, \phi_p) \). For example, in the normal model \( \phi = (\theta, \sigma^2) \), and the probability measure of interest is \( p(\theta, \sigma^2 | y_1, \ldots, y_n) \). Given a starting point,

\[
\phi^0 = (\phi_1^{(0)}, \ldots, \phi_p^{(0)}),
\]

Figure 1.1: Histograms and kernel density estimates of Monte Carlo approximations to the gamma(68,45) distribution, with true density in gray.
the Gibbs sampler generates $\phi^{(s)}$ from $\phi^{(s-1)}$ as follow:

$$
\begin{align*}
\phi_1^{(s)} &\sim p(\phi_1|\phi_2^{(s-1)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)}) \\
\phi_2^{(s)} &\sim p(\phi_2|\phi_1^{(s)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)}) \\
&\vdots \\
\phi_p^{(s)} &\sim p(\phi_p|\phi_1^{(s)}, \phi_2^{(s)}, \ldots, \phi_{p-1}^{(s)})
\end{align*}
$$

This algorithm generates a dependent sequence of vectors:

$$
\begin{align*}
\phi^{(1)} &= \phi_1^{(1)}, \ldots, \phi_p^{(1)} \\
\phi^{(2)} &= \phi_1^{(2)}, \ldots, \phi_p^{(2)} \\
&\vdots \\
\phi^{(s)} &= \phi_1^{(s)}, \ldots, \phi_p^{(s)}
\end{align*}
$$

In this sequence, $\phi^{(s)}$ depends on $\phi^{(0)}, \ldots, \phi^{(s-1)}$ only through $\phi^{(s-1)}$, i.e., $\phi^{(s)}$ is conditionally independent of $\phi^{(0)}, \ldots, \phi^{(s-2)}$ given $\phi^{(s-1)}$; this is called the Markov property, and so the sequence is called a Markov chain. Under some conditions that will be met for all of the model discussed in this text,

$$
p(\phi^{(s)} \in A) \to \int_A P(\phi) d\phi \quad \text{as } S \to \infty
$$

In other words, the sampling distribution of $\phi^{(S)}$ approaches the target distribution as $s \to \infty$, no matter what the starting value $\phi^{(0)}$ is (although some starting values will get you to the target sooner than others.) More importantly, for most function $g$ of interest,

$$
\frac{1}{S} \sum_{s=1}^{S} g(\phi^{(s)}) \to E[g(\phi)] = \int g(\phi)p(\phi) d\phi \quad \text{as } S \to \infty \quad (1.15)
$$

This means we can approximate $E[g(\phi)]$ with the sample average of $[g(\phi^{(1)}), \ldots, g(\phi^{(S)})]$, just as in Monte Carlo approximation. For this reason, we call such approximations Markov chain Monte Carlo approximations, and the procedure an MCMC algorithm.

1.5.3 Metropolis algorithm

When conjugate or semi-conjugate prior distributions are used, the posterior distribution can be approximated with the Monte Carlo methods or the
Gibbs sampler. In situations where a conjugate prior distribution is unavailable or undesirable, the full conditional distribution of the parameters do not have a standard form and the Gibbs sampler can not be easily used. The Metropolis algorithm as a generic method of approximating posterior distribution corresponding to any combination of prior distribution and sampling model.

Suppose we have a working collection \( \theta^{(1)}, \ldots, \theta^{(s)} \) to which we would like to add a new value \( \theta^{(s+1)} \). Let’s consider adding a value \( \theta^* \) which is nearby \( \theta^{(s)} \). The Metropolis algorithm proceeds by sampling a proposal value \( \theta^* \) nearby the current value \( \theta^{(s)} \) using a symmetric proposal distribution \( J(\theta^*|\theta^{(s)}) \). Symmetric here means that \( J(\theta^*|\theta^{(s)}) = J(\theta^{(s)}|\theta^*) \), i.e the probability of proposing \( \theta^* = \theta^* \) given that \( \theta^{(s)} = \theta^{(s)} \).

Usually the proposal distribution which are used:

\[
J(\theta^*|\theta^{(s)}) = \text{uniform}(\theta^{(s)} - \delta, \theta^{(s)} + \delta) \quad (1.16)
\]

\[
J(\theta^*|\theta^{(s)}) = \text{normal}(\theta^{(s)}, \delta^2) \quad (1.17)
\]

The value of the parameter \( \delta \) is generally chosen to make the approximation algorithm run efficiently.

Having obtained a proposal value \( \theta^* \), we add either it or a copy of \( \theta^{(s)} \) to our set, depending on the ratio

\[
r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} \quad (1.18)
\]

Specifically, given \( \theta^{(s)} \), the Metropolis algorithm generates a value \( \theta^{(s+1)} \) as follows:

1. Sample \( \theta^* \sim J(\theta | \theta^{(s)}) \);

2. Compute the acceptance ratio

\[
r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})} \quad (1.19)
\]

3. Let

\[
\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability min}(r,1) \\ \theta^{(s)} & \text{with probability } 1-\text{min}(r,1). \end{cases}
\]

Step 3 can be accomplished by sampling \( u \sim \text{uniform}(0, 1) \) and setting \( \theta^{(s+1)} = \theta^* \) if \( u < r \) and setting \( \theta^{(s+1)} = \theta^{(s)} \) otherwise.
Example: normal distribution with known variance

Let's try out the Metropolis algorithm for the conjugate normal model with a known variance, a situation where we know the correct posterior distribution. Let, we have sampling model,

\[(y_1, \ldots, y_n | \theta) \sim N(\theta, \sigma^2)\]  \hspace{1cm} (1.20)

prior,

\[\theta \sim N(\mu, \tau^2)\]  \hspace{1cm} (1.21)

the posterior distribution of \(\theta\) is \(N(\mu_n, \tau_n^2)\) where,

\[\mu_n = \frac{n \bar{y} / \sigma^2}{n / \sigma^2 + 1 / \tau^2} + \frac{\mu / \tau^2}{n / \sigma^2 + 1 / \tau^2}\]

\[\tau_n^2 = \frac{1}{n / \sigma^2 + 1 / \tau^2}\]

Suppose, \(\sigma^2 = 1, \tau^2 = 10, n = 5, y = (9.37, 10.18, 9.16, 11.60, 10.33)\)

For these data, \(\mu_n = 10.03\) and \(\tau_n^2 = .20\) and so \(p(\theta | y) = \text{dnorm}(10.03, 44)\)

Now suppose that for some reason we were unable to obtain the formula for this posterior distribution and needed to use the Metropolis algorithm to approximate it. Based on this model and prior distribution, the acceptance ratio comparing a proposed value \(\theta^*\) to a current value \(\theta^{(s)}\) is

\[r = \frac{p(\theta^* | y)}{p(\theta^{(s)} | y)}\]

\[= \frac{\prod_{i=1}^n \text{dnorm}(y_i, \theta^*, \sigma)}{\prod_{i=1}^n \text{dnorm}(y_i, \theta^{(s)}, \sigma)} \frac{\text{dnorm}(\theta^*, \mu, \tau)}{\text{dnorm}(\theta^{(s)}, \mu, \tau)}\]  \hspace{1cm} (1.22)

The R-code below generates 10,000 iterations of the Metropolis algorithm, starting at \(\theta^{(0)} = 0\) and using a normal proposal distribution, \(\theta^{(s+1)} \sim N(\theta^{(s)}, \delta^2)\) with \(\delta^2 = 2\).

```r
> s2<-1
> t2<-10 ; mu<-5
> set.seed(1)
> n<-5
> y<-round(rnorm(n,10,1),2)
> mu.n<-( mean(y)*n/s2 + mu/t2 )/( n/s2+1/t2)
> t2.n<-1/(n/s2+1/t2)
> ######
> s2<-1 ; t2<-10 ; mu<-5
```
1.5 Summarizing inferences by simulation

Figure 1.2: Results from the Metropolis algorithm for the normal model.

```r
> y<-c(9.37, 10.18, 9.16, 11.60, 10.33)
> theta<-0 ; delta<-2 ; S<-10000 ; THETA<-NULL ; set.seed(1)
> for(s in 1:S).
+ {  
+ theta.star<-rnorm(1,theta,sqrt(delta))
+ log.r<-( sum(dnorm(y,theta.star,sqrt(s^2),log=TRUE)) +
+ dnorm(theta.star,mu,sqrt(t^2),log=TRUE)) -
+ ( sum(dnorm(y,theta,sqrt(s^2),log=TRUE)) +
+ dnorm(theta,mu,sqrt(t^2),log=TRUE) )
+ if(log(runif(1))<log.r) { theta<-theta.star }
+ THETA<-c(THETA,theta)
+ }
> ######### Commands for plotting graph ##########
> par(mar=c(3,3,1,1),mgp=c(1.75,.75,0))
> par(mfrow=c(1,2))
> skeep<-seq(10,S,by=10)
> plot(skeep,THETA[skeep],type="l",xlab="iteration",ylab=expression(theta))
> hist(THETA[-(1:50)],prob=TRUE,main="",xlab=expression(theta),ylab="density")
> th<-seq(min(THETA),max(THETA),length=100)
> lines(th,dnorm(th,mu.n,sqrt(t2.n)) )
```
1.5 Summarizing inferences by simulation

The first panel of Figure 1.2 plots these 10,000 simulated values as a function of iteration number. Although the value of \( \theta \) starts nowhere near the posterior mean of 10.03, it quickly arrives there after a few iterations. The second panel gives a histogram of the 10,000 \( \theta \)-values, and includes a plot of the \( \text{normal}(10.03, 0.20) \) density for comparison. Clearly the empirical distribution of the simulated values is very close to the true posterior distribution.

Example: song sparrow reproductive success

A sample from population of 52 female song sparrows was studied over the course of a summer and their reproductive activities were recorded. In particular, the age and number of new offspring were recorded for each sparrow (Arcese et al., 1992). This is not surprising from a biological point of view: One-year-old birds are in their first mating season and are relatively inexperienced compared to two-year-old birds. As birds age beyond two years they experience a general decline in health and activity.

Suppose we wish to fit a probability model to these data, perhaps to understand the relationship age and reproductive success, or to make population forecasts for this group of birds. Since the number of offspring for each bird is a non-negative integer \( \{0, 1, 2, \ldots\} \), a simple probability model for \( Y = \text{number of offspring} \) conditional on \( x = \text{age} \) would be a Poisson model,

\[
Y \mid x \sim \text{Poisson}(\theta_x)
\]

1.5.4 The Metropolis algorithm for Poisson regression

Let's implement the Metropolis algorithm for the Poisson regression model. The model is that \( Y_i \) is a sample from a Poisson distribution with a log mean given by \( \log[\text{E}(Y_i \mid x_i)] = \beta_1 + \beta_2 x_i + \beta_3 x_i^2 \), where \( x_i \) is the age of the sparrow \( i \). We will abuse notation slightly by writing \( x_i = (1, x_i, x_i^2) \) so that \( \log[\text{E}(Y_i \mid x_i)] = \beta^T x_i \). The prior distribution was the regression coefficients were i.i.d \( \text{normal}(0, 100) \), the acceptance ratio for the Metropolis algorithm is

\[
r = \frac{p(\beta^* \mid X, y)}{p(\beta \mid X, y)}
\]

\[
= \frac{\prod_{i=1}^n \text{dpois}(y_i, x_i^T \beta^*) \prod_{j=1}^3 \text{dnorm}(\beta_j^*, 0, 10)}{\prod_{i=1}^n \text{dpois}(y_i, x_i^T \beta) \prod_{j=1}^3 \text{dnorm}(\beta_j, 0, 10)}
\]

(1.23) (1.24)
1.5 Summarizing inferences by simulation

1.5.5 The Metropolis Hastings algorithm

Consider a simple example where our target probability distribution is \( p_0(u,v) \), a bivariate distribution for two random variables \( U \) and \( V \). In the one sample normal problem, for example, we would have \( U = \theta, V = \sigma^2 \) and \( p_0(u,v) = p(\theta,\sigma^2 \mid y) \). Recall that the Gibbs sampler proceeds by iteratively samplings values \( U \) and \( V \) from their conditional distribution: Given \( x^{(s)} = (u^{(s)},v^{(s)}) \), a new value of \( x^{(s+1)} \) is generated as follows:

1. update \( U \): sample \( u^{(s+1)} \sim p_0(u \mid v^{(s)}) \)
2. update \( V \): sample \( v^{(s+1)} \sim p_0(v \mid u^{(s)}) \)

Alternatively, we could have first sampled \( v^{(s+1)} \sim p_0(v \mid u^{(s)}) \) and then \( u^{(s+1)} \sim p_0(u \mid v^{(s)}) \). In contrast, the Metropolis algorithm proposes changes to \( X = (U,V) \) and then accepts or rejects those changes based on \( p_0 \). In the Poisson regression example the proposed vector differed from its current value at each element of the vector, but this is not necessary. An alternative way to implement the Metropolis algorithm is to propose and then reject changes to one element at a time:

1. update \( U \):
   - sample \( u^* \sim J_u(u \mid u^{(s)}) \)
   - compute \( r = p_0(u^*,v^{(s)}) / p_0(u^{(s)},v^{(s)}) \)
   - set \( u^{(s+1)} \) to \( u^* \) or \( u^{(s)} \) with probability \( \min(1,r) \) and \( \max(0,1-r) \)
2. update \( V \):
   - sample \( v^* \sim J_v(v \mid v^{(s)}) \)
1.5 Summarizing inferences by simulation

- compute \( r = \frac{p_0(u^{(s+1)}, v^*)}{p_0(u^{(s+1)}, v^{(s)})} \)
- set \( v^{(s+1)} \) to \( v^* \) or \( v^{(s)} \) with probability \( \min(1, r) \) and \( \max(0, 1-r) \)

Here, \( J_u \) and \( J_v \) are separate symmetric proposal distributions for \( U \) and \( V \). This Metropolis algorithm generates proposals from \( J_u \) and \( J_v \) and accepts them with some probability \( \min(1, r) \). Similarly, each step of the Gibbs sampler can be seen as generating a proposal from a full conditional distribution and then accepting it with probability \( 1 \). The Metropolis-Hastings algorithm generalizes both of these approaches by allowing arbitrary proposal distributions. The proposal distributions can be symmetric around the current values, full conditional distributions, or something else entirely.

A Metropolis-Hastings algorithm for approximating \( p_0(u,v) \) runs as follows:

1. update \( U \):
   - sample \( u^* \sim J_u(u \mid u^{(s)}, v^{(s)}) \)
   - compute the acceptance ratio
     \[
     r = \frac{p_0(u^*, v^{(s)}) J_u(u^{(s)} \mid u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)}) J_u(u^{(s)} \mid u^{(s)}, v^{(s)})}
     \]
   - set \( u^{(s+1)} \) to \( u^* \) or \( u^{(s)} \) with probability \( \min(1, r) \) and \( \max(0, 1-r) \)

2. update \( V \):
   - sample \( v^* \sim J_v(v \mid u^{(s+1)}, v^{(s)}) \)
   - compute the acceptance ratio
     \[
     r = \frac{p_0(u^{(s+1)}, v^*) J_v(v^{(s)} \mid u^{(s+1)}, v^*)}{p_0(u^{(s+1)}, v^{(s)}) J_v(v^{(s)} \mid u^{(s+1)}, v^{(s)})}
     \]
   - set \( v^{(s+1)} \) to \( v^* \) or \( v^{(s)} \) with probability \( \min(1, r) \) and \( \max(0, 1-r) \)

In this algorithm the proposal distribution \( J_u \) and \( J_v \) are not required to be symmetric. In fact, the only requirement is that they do not depend on \( U \) or \( V \) values in our sequence previous to the most current values. This requirement ensures that the sequence is a Markov chain.

The Metropolis-Hastings algorithm looks a lot like the Metropolis algorithm, except that the acceptance ratio contains an extra factor, the ratio of the probability of generating the current value from the proposed to the probability of generating the proposed from the current.
1.6 Computation and software

The software used here is R. It has great features of computation and graphics. R is both a software and a language considered as a dialect of the S language created by the AT and T Bell laboratories.

- R is available in several forms: the sources (written mainly in C and some routines in Fortran) essentially for unix and linux machines, or some pre-compile binaries for windows, linux, and machintosh.

- R has many function for statistical analysis and graphics; the later we are visualized immediately in their own window and can be saved in various format (jpg, png, bmp, ps, pdf, emf, pictex, xfig)

Since, heavier use of graphics is an essential part of most analysis; we make heavier use of R software than other softwares. With the help of R software, data can be easily simulated from the posterior distribution and thereby results can be drawn instantly. In any case, we typically work within R to plot and transform the data before model fitting, and to display inference and model checks afterwards:

Specific computational tasks that arises in Bayesian data analysis include:

1. Vector and matrix manipulations
2. Computing probability density functions
3. Drawing simulation from probability distribution
4. Structured programming (including looping and customized functions.)
5. Calculating the linear regression estimate and variance matrix.
6. Graphics, including scatter plots with overlain lines and multiple graphs per page.

R software

Applied researchers interested in Bayesian Statistics are increasingly attracted to R because of the ease of which one can code algorithm to sample from posterior distributions as well as the significant number of packages contributed to the Comprehensive R Archive Network (CRAN) that provide tools for Bayesian inference.

Now we discuss packages that address Bayesian models or specialized methods in Bayesian Statistics. Now we briefly introduced a few important packages ment for Bayesian analysis.
Bayesian packages for general model fitting

- **MCMCpack()**
  MCMCpack provides model-specific Markov chain Monte Carlo (MCMC) algorithms for a wide range of models commonly used in the social and behavioral sciences. It contains R functions to fit a number of regression models (linear regression, logit, ordinal probit, probit, Poisson regression, etc.), measurement models (item response theory and factor models), changepoint models (binary and Poisson) and models for ecological inference. It also contains a generic Metropolis sampler that can be used to fit arbitrary models.

- **arm()**
  The arm package contains R functions for Bayesian inference using lm, glm, mer, and polr objects.

Post-estimation tools

- **coda()**
  The coda (Convergence Diagnosis and Output Analysis) package is a suite of functions that can be used to summarize, plot and diagnose convergence from MCMC samples. The package coda also defines an MCMC object and related methods which are used by other packages. It can easily import MCMC output from WinBUGS, OpenBUGS, and JAGS, or from plain matrices. This package contains the Gelman and Rubin Geweke, Heidelberger and Welch, and Raftery and Lewis diagnostics.

Packages for learning Bayesian statistics

- **LearnBayes()**
  The LearnBayes package contains a collection of functions helpful in learning the basic tenets of Bayesian statistical inference. It contains functions for summarizing basic one and two parameter posterior distributions and predictive distributions and MCMC algorithm for summarizing posterior distributions defined by the user. It also contains functions for regression models, hierarchical models, Bayesian tests, and illustration of Gibbs sampling.

R is the GNU licensed implementation of S language. It is an open software version available from its official website http://cran.r-project.org. The developers of R Development Core Team, an international group that includes John Chambers and other Bell Lab. researchers.
1.7 Discussion and further reference


Gelman et al. (2003), Hoff PD (2007, 2010) and Tanner (1996) provides a user's guide to Bayesian calculations. Gelman et al. (2003) covers several statistical models most most familiar to ecologist from a Bayesian perspective. Most of the methods presented by Gelman (2003) can be computed using R software packages. Ghosh et al. (2006) contains modern tools of computation like Laplace approximation and simulation techniques. Several review articles in the statistical literature, such as Breslow (1990) Racine et al.(1986), have appeared that discuss, in general terms, areas of application in which Bayesian methods have been useful. The volumes edited by Gatsonis et al.(1993-2002) are collection of Bayesian analysis, including extensive discussions about choices in the modeling process and the relations between the statistical methods and the applications.

the foundations of probability and Bayesian statistics are an important topic that we treat only very briefly. Bernardo and Smith (1994) give a thorough review of the foundations of Bayesian models and inference with a comprehensive list of references. Jefferys (1961) is a self contained book about Bayesian statistics that comprehensively presents an inductive view of inference; Good (1950) is another important early work. Jaynes (1983) is a collection of reprinted articles that present a deductive view of Bayesian inference, which we believe is quite similar to ours. Jaynes (1996) focuses on connections between statistical inference and the philosophy of science and includes several examples of physical probability.