

Mattias Villani

Bayesian Learning

- draft version 0.5.2



Copyright © 2025 Mattias Villani

PUBLISHED BY

TYPESET BY L^AT_EX USING TEMPLATE FROM TUFTE-LATEX.GITHUB.IO

I will have to figure out how to license this work. For the moment the license is restrictive.

First edition, September 2025

Contents

1	<i>The Bayesics</i>	13
1.1	<i>Learning probability models</i>	14
1.2	<i>The likelihood function</i>	15
1.3	<i>Maximum likelihood estimation and frequentist properties</i>	17
1.4	<i>Subjective Probability</i>	19
1.5	<i>Bayesian Learning</i>	21
2	<i>Single-parameter models</i>	27
2.1	<i>Bernoulli data</i>	27
2.2	<i>Gaussian data - known variance</i>	32
2.3	<i>Online learning</i>	36
2.4	<i>Poisson data</i>	38
2.5	<i>Summarizing a posterior distribution</i>	41
2.6	<i>Coverage probabilities of Bayesian credible intervals</i>	43
2.7	<i>Bayesian learning and the likelihood principle</i>	47
2.8	<i>Exponential Family and Sufficiency*</i>	49
3	<i>Multi-parameter models</i>	57
3.1	<i>Joint posterior distributions</i>	57
3.2	<i>Marginalization</i>	58
3.3	<i>Gaussian data with unknown variance</i>	58
3.4	<i>A first look at Monte Carlo simulation</i>	62
3.5	<i>Multinomial data</i>	65

3.6	<i>Multivariate normal data with known covariance</i>	71
3.7	<i>Multivariate normal data with unknown covariance*</i>	72
3.8	<i>Likelihood and Information</i>	74
4	<i>Priors</i>	79
4.1	<i>Time series</i>	79
4.2	<i>Past or other data</i>	81
4.3	<i>Expert opinion</i>	81
4.4	<i>Structured regularization priors</i>	82
4.5	<i>Hierarchical priors</i>	83
4.6	<i>Elicitation through prior predictive distributions</i>	83
4.7	<i>Noninformative priors</i>	86
4.8	<i>Invariant priors</i>	87
5	<i>Linear Regression</i>	91
5.1	<i>The linear Gaussian regression model</i>	91
5.2	<i>Maximum likelihood</i>	92
5.3	<i>Non-informative prior</i>	94
5.4	<i>Conjugate prior</i>	96
6	<i>Prediction and Decision making</i>	109
6.1	<i>Bayesian prediction</i>	109
6.2	<i>Bayesian decisions</i>	117
7	<i>Normal posterior approximation</i>	125
7.1	<i>Intractable posterior and approximation</i>	125
7.2	<i>Taylor approximation of the posterior</i>	127
7.3	<i>Normal posterior approximation and large sample asymptotics</i>	128
7.4	<i>Computing the normal posterior approximation numerically</i>	135
7.5	<i>Reparametrization</i>	136

8	<i>Classification and Generalized regression</i>	141
8.1	<i>Classification problems</i>	141
8.2	<i>Logistic regression</i>	143
8.3	<i>Multi-class logistic regression</i>	150
8.4	<i>Poisson regression</i>	152
8.5	<i>Generalized linear models and beyond</i>	157
8.6	<i>Bayesian discriminant analysis and Naive Bayes</i>	161
9	<i>Gibbs sampling</i>	165
9.1	<i>The Gibbs sampling algorithm</i>	165
9.2	<i>Gibbs sampling for probit regression</i>	173
9.3	<i>Gibbs sampling for logistic regression</i>	176
9.4	<i>Autoregressive processes</i>	181
10	<i>Markov Chain Monte Carlo simulation</i>	187
10.1	<i>Monte Carlo</i>	188
10.2	<i>Importance sampling</i>	191
10.3	<i>Rejection sampling</i>	197
10.4	<i>Markov Chain Monte Carlo</i>	200
10.5	<i>Hamiltonian Monte Carlo</i>	200
10.6	<i>Probabilistic programming frameworks</i>	200
11	<i>Variational inference</i>	209
12	<i>Regularization</i>	211
12.1	<i>Model complexity and overfitting</i>	211
12.2	<i>L₂-regularization and ridge regression</i>	213
12.3	<i>Bayesian learning of the L₂ regularization parameter</i>	218
12.4	<i>L₁-regularization and the Lasso estimator</i>	220
12.5	<i>Global-local regularization and the horseshoe prior</i>	226
12.6	<i>Regularized nonlinear regression</i>	231

13	<i>Mixture models and Bayesian nonparametrics</i>	239
13.1	<i>Mixture of normals as flexible data models</i>	239
13.2	<i>Simulating data from the mixture of normals model</i>	242
13.3	<i>Inference for the mixture of normals model</i>	244
13.4	<i>Mixture of Poissons for count data</i>	247
13.5	<i>Exponential family mixtures and multivariate mixtures</i>	252
13.6	<i>Mixture of regressions and mixture of experts</i>	252
13.7	<i>Bayesian histograms</i>	255
13.8	<i>Dirichlet process priors</i>	258
13.9	<i>Dirichlet process mixtures</i>	262
14	<i>Model comparison and variable selection</i>	265
14.1	<i>Posterior model probabilities and the marginal likelihood</i>	265
14.2	<i>Normal model</i>	267
14.3	<i>The Laplace approximation of the marginal likelihood</i>	272
14.4	<i>Log predictive score</i>	274
14.5	<i>Bayesian estimators of generalization performance</i>	277
14.6	<i>Bayesian variable selection</i>	278
15	<i>Gaussian processes</i>	283
15.1	<i>Gaussian processes priors</i>	283
15.2	<i>Gaussian process regression</i>	290
15.3	<i>Learning the kernel hyperparameters</i>	298
15.4	<i>Heteroscedastic Gaussian processes regression</i>	301
15.5	<i>Gaussian processes for classification</i>	302
15.6	<i>Gaussian processes for Poisson regression</i>	304
15.7	<i>Bayesian optimization</i>	306
16	<i>Interaction models</i>	307
16.1	<i>Surface splines</i>	307
16.2	<i>Bayesian regression trees</i>	307

<i>17 Dynamic models and sequential inference</i>	309
<i>17.1 Some examples of state-space models</i>	309
<i>17.2 The linear Gaussian state-space model</i>	313
<i>17.3 Bayesian filtering</i>	315
<i>17.4 Bayesian filtering in linear Gaussian models</i>	316
<i>17.5 Bayesian smoothing in linear Gaussian models</i>	325
<i>17.6 Parameter inference in linear Gaussian state-space models</i>	326
<i>17.7 Non-linear non-Gaussian models</i>	329
<i>17.8 Sequential inference in non-linear non-Gaussian models</i>	332
<i>Bibliography</i>	337
<i>Appendix: Some Mathematical results</i>	341
<i>A.1 Some calculus</i>	341
<i>A.2 Some linear algebra</i>	342
<i>A.3 Taylor approximation</i>	351
<i>Index</i>	355

*To all Bayesians who persevered
through the difficult decades.*

Preface

Who's this book for?

This book can be used as a first book in Bayesian statistics at the advanced undergraduate or master level. It is written to also accommodate students in engineering and computer science with an interest in Bayesian learning for applications in Data Science and Machine Learning, but may not be as heavily trained in probability and statistics.

In fact, the book grew out of a Bayesian course that I taught for groups of heterogeneous students, with roughly half of the students from statistics and the other half from engineering and computer science. To my surprise, I found that it was indeed possible to teach the same material to all students, even if half the class had a much more extensive background in statistics. Students from both camps thought that the course was on the right level for them. There are two main explanations for this. First, since most bachelor level Statistics are non-Bayesian in methods and thinking, taking a first course in Bayesian inference is in some way like starting from scratch. There are of course several overlapping concepts and probability is the underlying technical language (although with highly different interpretations), but there is nevertheless a lot of effort spent in basic statistics courses that are not needed prerequisites for a Bayesian course. Second, my courses are very computational, as is most of the Bayesian field, with a lot of computer labs and also a partly computerized exam. Engineering and particularly computer science students tend to have a comparative advantage in computing and programming. So the additional time that students from statistics had to spend on programming, computer science students could spend on catching up on statistical concepts. In order to accommodate both groups of students, my lectures covers also some rather elementary concepts, especially in the early part of the course, before moving over to territory unknown to all students. This book is written in the same style using Tufte style margin notes and figures to fill in potential missing gaps in probability and statistics, without breaking the flow of the main

text.

Some programming experience is useful for the exercises, or at least basic familiarity with R, Python or Julia or a similar datacentric language. I use pseudo code for certain smaller algorithms and Julia for real code; Julia is used to present algorithms in the book since the ability to use mathematical symbols in Julia (via unicode) makes the code easy to read, almost like pseudo code. All graphs were made in Julia using the Plots package with GR as backend.

Why the term Bayesian learning?

I have used the term Bayesian *learning* in the book's title instead of the more traditional Bayesian *inference* or Bayesian *statistics*. There are several reasons for this.

First, I want the book to be welcoming to students in fields neighboring statistics, such as machine learning, computer science, and parts of engineering. This reflects a belief that a modern statistician or machine learner should be a little of a renaissance person that understands both probability, statistical modelling, and computing. An ideal class is therefore a mix of students from nearby disciplines that learn from each other's competences as much as they learn from my classes or this book.

Second, the term learning instead of inference was chosen since Bayesian statistics is about learning from data, often in a very sequential way where incrementally collected information updates our knowledge about the world.

Finally, the title is meant to convey the message that this is not a traditional book in statistics. The approach taken here, especially in later chapters, is very computationally driven with many algorithms for real-world data analysis. It is also inspired by machine learning in that much of the focus is on prediction and decision making, and almost none on hypothesis testing.

Acknowledgment

This section will be much more complete when the book is finished, but I want to note already now that this book has been influenced by many other excellent textbooks on Bayesian methods. This is particularly true for two books that I have used as course literature over the years. I taught my first Bayes course in the year of 2000 using the book *Statistical Inference - An Integrated Approach* by Migon and Gamerman. Second, I have used the book *Bayesian Data Analysis* by Gelman et al. for a number of years while teaching. I imagine that I have been more influenced by these two books than I know, and I thank the authors for taking the time to write them. I now

appreciate them even more: it takes a lot of time to write a book! The presentation of some parts of the chapter on sequential inference was inspired by the book 'Probabilistic Robotics' by [Thrun et al. \(2006\)](#).

1 The Bayesics

Uncertainty is an ever-present aspect of life. Much of the environment remains unknown to us, as does the future. Yet we constantly face situations where we need to make decisions whose consequences depend on factors that are unknown at the time of the decision. The outcome of a seemingly simple decision, such as whether to leave the house without an umbrella, depends on how the weather develops. In other cases the stakes are far greater, for instance when choosing between two demanding cancer treatments with uncertain prognoses, or when determining how much reinforcement to use when building a bridge that could collapse under heavy load.

So what can we do? One option is to gather more information to reduce the uncertainty. Information may take the form of numerical data, for instance body measurements in a medical setting or sensors on a self-driving car, or it may be qualitative, such as verbal advice from a physician or the perceived safety of a passenger in an autonomous vehicle. Using both quantitative and qualitative information sources, we can learn about unknowns and reduce the uncertainty to reach a more informed decision.

In this book, uncertainty will be quantified by probabilities, and a Bayesian framework will be used to learn probabilistic models from data. A probabilistic model, with parameters learned from data, can be used to generate probabilistic predictions and to support formal decision making under uncertainty. The Bayesian approach offers several important advantages compared to alternative frameworks, including the ability to combine numerical data with qualitative sources of information such as expert judgement. Later in the book, we learn that such qualitative information may be simple smoothness assumptions about some unknown function, which is one of several reasons why Bayesian uncertainty quantification is popular in machine learning. Throughout the book, the Bayesian approach is shown to give a structured, coherent, way to learn from data about unknowns, to compute predictive probabilities, and ultimately to make optimal decisions under uncertainty.

In this chapter we will take the first stumbling steps by introduc-

ing probability models for modeling data, and briefly reviewing some parts of non-Bayesian statistics. The Bayesian approach with its basis in subjective probability is then shown to combine data information and other more qualitative sources of information in a natural way.

1.1 Learning probability models

Throughout this book, we will work almost exclusively with probability models. Such models provide a precise quantification of uncertainty that can be directly applied to decision making in real-world problems.

A central task in statistics and machine learning is to infer an unknown parameter $\theta \in \Theta$ in a probability model $p(X_1, \dots, X_n | \theta)$ given a dataset of n observations x_1, \dots, x_n . The **parameter space** Θ is the set of admissible parameter values. Examples of single-parameter problems is estimating the voting share of a political party from exit polls, predicting the number of bugs in a software release and inferring a one-dimensional measure of a person's intelligence from IQ tests.

While the initial chapters focus on learning parameters in models, it is important to recognize that parameter inference is typically an intermediate step toward the final aim of prediction or decision making under uncertainty. For instance, the predictions and actions of a robot rely on a probability model with network weights learned from training data, and public health authorities estimate the basic reproduction number R_0 within probabilistic models in order to forecast disease spread and guide intervention strategies. The Bayesian approach to predictions and decisions will be presented in Chapter 6, and used in many places throughout the book.

Many problems require models with multiple parameters. A prominent case is the deep neural networks widely used in artificial intelligence (AI), which often involve millions, or even billions, of parameters to be inferred from training data. To focus on core ideas and to keep mathematical derivation short and transparent to build intuition, we will restrict attention in the first two chapters to single-parameter models. Later chapters tackle more complex models and present methods specifically designed for high-dimensional parameter spaces.

The observations X_1, \dots, X_n are initially assumed to be **independent and identically distributed (iid)** conditional on θ so that we can write the joint distribution as a product

$$p(X_1, \dots, X_n | \theta) = \prod_{i=1}^n p(X_i | \theta).$$

parameter space



Figure 1.1: Artificial intelligence and infectious disease models are examples where Bayesian learning is often used for quantifying uncertainty.

independent and identically distributed

iid

We denote this by $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} p(X|\theta)$. In this setting we can refer to ‘the probability model’ as the probability distribution $p(X|\theta)$ for a single observation, coupled with the iid assumption. For example, the phrase ‘the Bernoulli model’, or the more complete ‘the iid Bernoulli model’, refers to the model in the following example.

EXAMPLE: A binary random variable $X \in \{0, 1\}$ follows a **Bernoulli distribution** if its **probability mass function (pmf)** is

$$\Pr(X = x|\theta) = \begin{cases} \theta & \text{for } x = 1 \\ 1 - \theta & \text{for } x = 0 \end{cases}$$

which can be written more compactly as

$$p(x|\theta) = \theta^x(1 - \theta)^{1-x}. \quad (1.1)$$

A typical example of iid Bernoulli data occurs when a coin is flipped n times (also called **Bernoulli trials**) and the sequence of heads ($x = 1$) and tails ($x = 0$) are recorded. It is common to refer to the outcome $X = 1$ as a success, and $X = 0$ as a failure. The properties of the Bernoulli distribution is given in Box 1.1 and the distribution is illustrated in Figure 1.2. All distributions in this book have interactive versions which can be explored in the PDF version of the book by either clicking on the distribution’s name in properties box or on the example graph of the distribution. A reader of the paper version of the book can visit the whole collection of distributions at observablehq.com/collection/@mattiasvillani/distributions.

We make the usual distinction between *random variables* denoted by capital letters and their *realizations (data)*, so $X = x$ means a random variable X with outcome x . As we will see later on, this distinction will often be less relevant in a Bayesian world where all inferences are conditioned on the observed data; we will therefore be less careful with this distinction in later chapters, but no harm will come from this.

1.2 The likelihood function

The likelihood function is a key component of Bayesian learning, and indeed in all of Statistics. Given a probability model $p(X_1, \dots, X_n | \theta)$ for some discrete random variables X_1, \dots, X_n , the **likelihood function** $p(x_1, \dots, x_n | \theta)$ is the *joint* probability of observing the dataset x_1, \dots, x_n considered as a function of the parameter θ . If the data are iid we can express the likelihood in terms of the univariate distributions $p(X|\theta)$ as

$$p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n p(x_i | \theta). \quad (1.2)$$

Bernoulli distribution

probability mass function (pmf)

Bernoulli trials

Bernoulli distribution

$$X \sim \text{Bern}(\theta)$$

Support: $X \in \{0, 1\}$

$$p(x) = \theta^x(1 - \theta)^{1-x}$$

$$\mathbb{E}(X) = \theta$$

$$\mathbb{V}(X) = \theta(1 - \theta)$$

Box 1.1: The Bernoulli distribution.

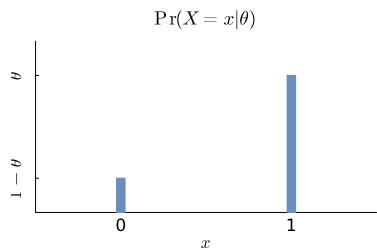


Figure 1.2: Bernoulli distribution with success probability $\theta = 0.8$.

likelihood function

EXAMPLE: In the case of iid Bernoulli data the likelihood function is obtained by multiplying together the probability of success θ for the observations where $x_i = 1$ and the probability of failure $1 - \theta$ when $x_i = 0$, giving the likelihood

$$p(x_1, \dots, x_n | \theta) = \theta^s (1 - \theta)^f, \quad (1.3)$$

where $s = \sum_{i=1}^n x_i$ is the number of successes in the sample, and $f = n - s$ is the number of failures. For example, imagine that you have observed the following sample of size $n = 5$ with $s = 2$ successes and $f = 3$ failures: $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0$. The likelihood function is then

$$p(x_1, \dots, x_5 | \theta) = \theta(1 - \theta)\theta(1 - \theta)(1 - \theta) = \theta^2(1 - \theta)^3.$$

Note that (1.3) is the probability if the data is provided in a form that records the *order* that the successes and failures occurred in the n trials. If we instead only get the data in terms of the *number* of successes and failures, without knowing the precise *order* of successes and failures, then the likelihood function will be slightly different because a given number of successes and failures can be obtained in many different ways (it will be based on the Binomial distribution for s instead of Bernoulli). It turns out however that the difference in the likelihood between the cases when the order is known or unknown will only be a constant multiplicative factor, which we later show is unimportant for Bayesian inference. Hence, the likelihood can be taken to be (proportional to) the expression in (1.3) even if we only know the number of successes and failures, and not the order in which they were observed.

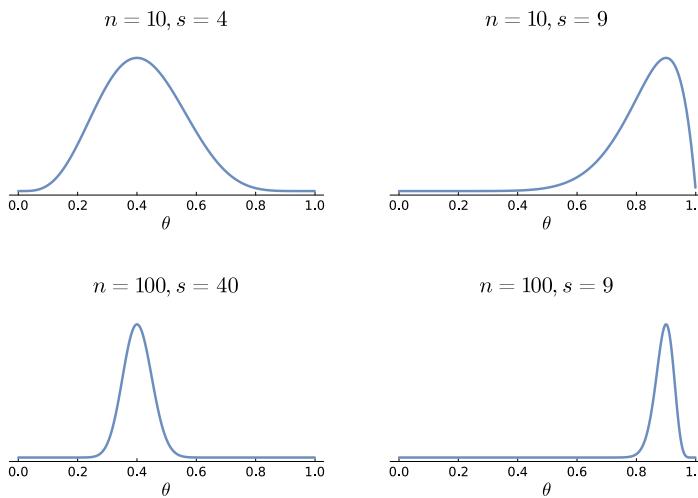


Figure 1.3: Bernoulli likelihood functions from four different datasets.

It is essential to have a mental image of the likelihood function when thinking about statistical modeling. Figure 1.3 illustrates the

likelihood function for the Bernoulli model based on datasets with $s = 4$ successes in $n = 10$ trials (top left) and $s = 9$ successes in $n = 10$ trials (top right). The two graphs in the lower part of Figure 1.3 show results for $n = 100$ trials with the same success ratio s/n as in corresponding graphs in the upper part of the figure; larger datasets make the likelihood more concentrated, i.e. large dataset are more informative about the plausibility of different θ values.

Figure 1.3 nicely illustrates how the likelihood function can inform us about the plausibility of any given θ for any given dataset. We can for example see that the dataset $n = 100$ and $s = 40$ is much more likely to have been generated with a Bernoulli model with $\theta = 0.4$ than with a Bernoulli model with $\theta = 0.8$.

When the data are recorded as **continuous random variables** the probability of any dataset is zero, and we instead define the likelihood function by letting $p(x_1, \dots, x_n | \theta)$ be the joint probability density function (**pdf**) of the observed data. See Figure 1.4 for an illustration of a **probability density function**. When data are iid we can similarly define the likelihood as the product of the individual pdf's for each data point $p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n p(x_i | \theta)$.

1.3 Maximum likelihood estimation and frequentist properties

If we want to select a single value, an **estimate**, of θ , a natural candidate is the **maximum likelihood estimator**

$$\hat{\theta}_{MLE} = \arg \max_{\theta \in \Theta} p(x_1, \dots, x_n | \theta). \quad (1.4)$$

It makes some intuitive sense to estimate θ by the value that maximizes the probability of the observed data; the estimator $\hat{\theta}_{MLE}$ also enjoys several other attractive properties, particularly in large samples, i.e. when n is large. Using a single value such as $\hat{\theta}_{MLE}$ as an estimate of θ is called **point estimation**.

It is quite easy to derive $\hat{\theta}_{MLE}$ for iid Bernoulli data. Rather than maximizing $p(x_1, \dots, x_n | \theta)$ directly with respect to θ it is often easier to maximize the *log-likelihood function*

$$\log p(x_1, \dots, x_n | \theta) = s \log \theta + f \log(1 - \theta).$$

Since the logarithm is a monotonically increasing function we obtain the same estimator if we maximize the likelihood or the log-likelihood function. We can now easily find $\hat{\theta}_{MLE}$ by taking the first derivative of the log-likelihood function with respect to θ , setting that derivative to zero and solving for θ . Solving

$$\frac{d \log p(x_1, \dots, x_n | \theta)}{d\theta} = \frac{s}{\theta} - \frac{f}{1 - \theta} = 0,$$

continuous random variables

pdf

probability density function

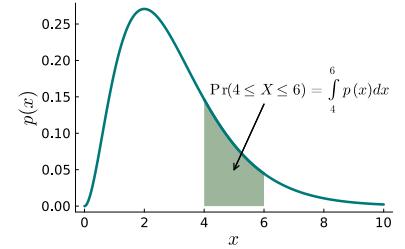


Figure 1.4: The probability density function (pdf) $p(x)$ for a continuous random variable X is a non-negative function that can be integrated to compute the probabilities of the form $\Pr(a \leq X \leq b) = \int_a^b p(x)dx$, and the total area under the pdf is one.

estimate

maximum likelihood estimator

point estimation

gives the unique solution $\hat{\theta}_{MLE} = s/n$, the fraction of successes in the data. It is straightforward to show that this is indeed a maximum by checking that the second derivative is negative at $\theta = \hat{\theta}_{MLE}$.

The maximum likelihood estimator is **unbiased** in this example, i.e. it is correct *on average over all possible samples* from the model:

$$\mathbb{E} [\hat{\theta}_{MLE}(X_1, \dots, X_n)] = \mathbb{E} \left(\frac{S}{n} \right) = \frac{n\theta}{n} = \theta,$$

where we have written out explicitly that an estimator is function of the sample. Note that the number of successes is random in this calculation as we are considering the variability over all possible samples, hence the use of capital letter S to denote that it is a random variable. We have also used that if $X_1, \dots, X_n | \theta \stackrel{iid}{\sim}$ Bernoulli then $S|\theta \sim \text{Binomial}(n, \theta)$ with mean $E(S) = n\theta$; see Box 1.2 for some properties of the **Binomial distribution** and Figure 1.5 for an graph of the probability mass function. The concept of unbiasedness is abstractly illustrated in Figure 1.6.

The **sampling variance** of an estimator can be used to assess the quality of an estimator. It is easily calculated for $\hat{\theta}_{MLE}$ in the Bernoulli example as

$$\mathbb{V} [\hat{\theta}_{MLE}(X_1, \dots, X_n)] = \mathbb{V} \left(\frac{S}{n} \right) = \frac{1}{n^2} \mathbb{V} (S) = \frac{\theta(1-\theta)}{n},$$

since $\mathbb{V}(S) = n\theta(1-\theta)$ when $S|\theta \sim \text{Binomial}(n, \theta)$.

It is important to understand that the above mean and variance of $\hat{\theta}_{MLE}$ are computed with respect to the **sampling distribution**, i.e. the distribution of the estimator as we repeatedly sample new datasets of size n from the assumed data generating process, see Figure 1.6. They are **long-run properties** of the estimation method, telling us how the estimator would perform on average over many repeatedly sampled datasets. Such long-run properties play a very limited role in the Bayesian approach where one can directly condition the inferences on the single dataset that we have observed. While sampling properties such as $\mathbb{E}(\hat{\theta}_{MLE})$ and $\mathbb{V}(\hat{\theta}_{MLE})$ are not used in the Bayesian approach, the likelihood *function* is at the core of Bayesian learning.

The likelihood functions in Figure 1.3 look like a probability distribution for θ , and it is tempting to compute probabilities for θ , for example $\Pr(\theta \leq c | x_1, \dots, x_n)$ for some c . Of course, such probabilities only make sense if θ is a random variable, and we have so far considered θ to be a fixed unknown constant. So while $p(X_1, \dots, X_n | \theta)$ is a probability distribution for a random sample X_1, \dots, X_n for a fixed θ , the likelihood function is only the probability of a *fixed* sample x_1, \dots, x_n considered as a function of θ ; the likelihood is therefore *not* a probability distribution for θ . Figure 1.7 reminds us of this error.

Binomial distribution

$S \sim \text{Binom}(n, \theta)$ for $S \in \{0, 1, \dots, n\}$

$$p(s) = \binom{n}{s} \theta^s (1-\theta)^{n-s}$$

$$\mathbb{E}(X) = n\theta$$

$$\mathbb{V}(X) = n\theta(1-\theta)$$

where

$$\binom{n}{s} = \frac{n!}{s!(n-s)!}$$

is the **binomial coefficient** that counts the number of ways that s successes can be obtained in n trials.

Box 1.2: The binomial distribution.

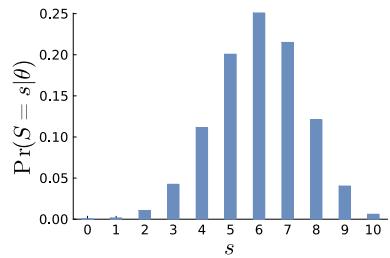
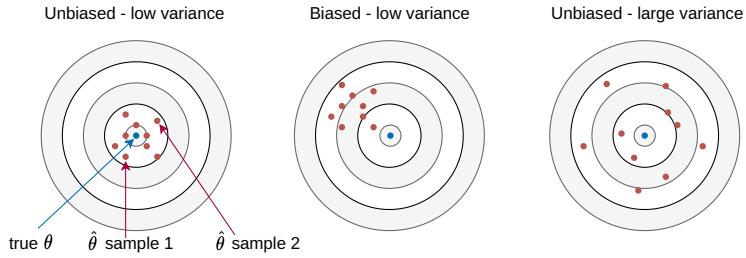


Figure 1.5: Binomial distribution with $n = 10$ and $\theta = 0.7$.

unbiased
Binomial distribution
sampling variance
sampling distribution
long-run properties



This is somewhat disappointing since having a probability distribution for θ would be very useful, for example when making a decision whose consequences depend on the unknown θ ; see Chapter 6. But again, it only makes sense to speak about probabilities for θ when θ is random, in some sense. And this is where our Bayesian story begins.

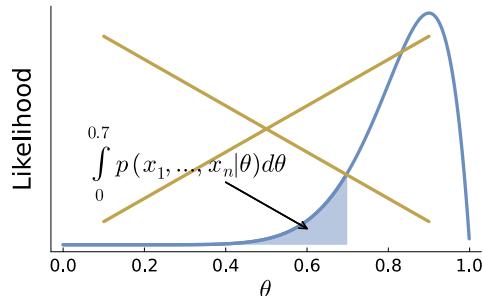


Figure 1.6: Illustration of the bias and variance of an estimator by darts on a dartboard. Each dart throw represents an estimate from a sample, and the sampling variance of the estimator is given by the spread of the dart throws. An unbiased estimator aims correctly at the center of the dart board (left). A biased estimator has an aim that is systematically off-center (middle). The graph to the right shows an unbiased estimator with a large sampling variance.

Figure 1.7: Areas under the likelihood function are **not** probabilities for the parameter.

1.4 Subjective Probability

What is the probability that the 10th decimal of π is 7? This may seem like a silly question since there is nothing intrinsically random about the 10th decimal of π ; it is a fixed quantity that does not vary. A Bayesian will however argue that if *you do not know its value* then you should express that uncertainty by a probability distribution. The Italian mathematician Bruno de Finetti, one of the founders of Bayesian learning, has expressed this well:

The only relevant thing is uncertainty - the extent of our knowledge and ignorance. The actual fact of whether or not the events considered are in some sense determined, or known by other people, and so on, is of no consequence.

Bruno de Finetti in his 1974 book 'A Theory of Probability' Vol 1.

Probability is the language of uncertainty and Bayesian learning is based on a subjective probability. A **subjective probability** measures

π



Figure 1.8: Bruno de Finetti, 1906-1985, a founder of subjective probability.

subjective probability

the **personal degree of belief** of a person. Since different persons have different knowledge and experience, such beliefs will vary between persons. A person that has no idea about the 10th decimal of π may use a uniform distribution on the integers 0-9, while someone that knows that this decimal is 5 assigns a probability of 1 to that outcome and zero to all other integers between 0 and 9. Again, whether or not the event is in some sense intrinsically random or not is of no consequence; the only relevant thing is *your* uncertainty. Einstein's famous statement "God does not play dice with the universe" in connection to quantum mechanics is interesting to ponder about, but has no bearing on subjective probability and Bayesian learning.

The notion of probability in Bayesian learning is therefore radically different from the frequentist interpretation of probability taught in most basic statistics courses. The **frequentist probability** of an event A is defined as the proportion of times that event A occurs in an (imagined) infinite number of repetitions of an experiment; for example the tossing of a coin with the event of interest $A = \{\text{coin lands with Head up}\}$.

A subjective probability measure is instead defined as the personal degree of belief in the event A for a person. Note that subjective probabilities can be used to quantify uncertainties also for events that are unrepeatable, for example the probability of a nuclear disaster at a particular location under certain conditions; the frequentist definition instead requires that the event must be infinitely repeatable, at least in principle. A subjective probability distribution can also contain useful information that may not directly come from observed data, for example expert knowledge. As we will see, the Bayesian approach combines such subjective information with objective data in a natural way.

Luckily, the computational rules for probabilities $0 \leq \Pr(A) \leq 1$ are the same for both frequentist and subjective interpretations of probability:

$$\Pr(A^c) = 1 - \Pr(A) \text{ for the complementary event } A^c$$

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

$$\Pr(A \cap B) = \Pr(A)\Pr(B|A) \text{ for independent events,}$$

where $A \cup B$ denotes the *union* and $A \cap B$ the *intersection* of the two events A and B , respectively (see Box 1.3). The rules can be motivated by considering subjective probabilities as the result of pricing of bets. Imagine that you are given the chance to enter a bet where you win \$1 if event A occurs. How much would you be willing to pay for that bet? Surely not more than \$1 as then you would lose money with certainty. If you strongly believe that A will occur you would probably be willing to pay closer to \$1, but if you believe

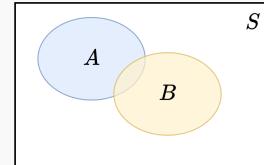
personal degree of belief

frequentist probability

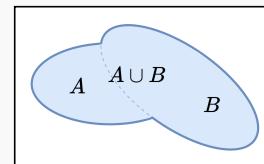
Venn diagram, Union and Intersection of events

Let A and B be two events in some sample space S .

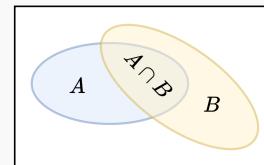
A **Venn diagram** is a visual representation of events in a sample space S . The sample space is represented by a rectangle, and events are represented by ellipses.



The **union** event $A \cup B$ occurs when A **or** B (or both) occurs.



The **intersection** event $A \cap B$ occurs when **both** A and B occur.



Box 1.3: Venn diagram, union and intersection of events.

that A is nearly impossible your price for the bet would be close to \$0. The highest price that you would be willing to pay for the bet is your subjective probability in the event A. One can show that your subjective probabilities must satisfy the usual axioms/rules for probabilities, otherwise you would be willing to enter a sequence of bets where you would lose an infinite amount with certainty; this is the **dutch book argument** for subjective probabilities. Objections have been raised against this argument, for example that the utility from the bet may not increase linearly with the monetary gain, and some people may even get utility just by the excitement in gambling; subsequent refinements of this argument have therefore completely disposed with the notion of money in favor of a more general notion of utility; see Chapter 6.

dutch book argument

1.5 Bayesian Learning

The general recipe for Bayesian learning about an event A is:

- Formulate your subjective *prior beliefs* $\Pr(A)$ about A.
- *Collect data* that inform you about A.
- *Update* your prior beliefs with the observed data.

The big question is *how* to update prior beliefs with data. Bayesian learning gets its name from using Bayes' theorem for this updating. The most basic version of **Bayes' theorem** computes the **conditional probability** of an event A given that some other event B has occurred:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}.$$

One way to think about this result is that it 'reverses the conditioning', i.e. it computes $\Pr(A|B)$ from knowledge of $\Pr(B|A)$.

Before moving on to Bayesian learning for model parameters, let us first use Bayes' theorem to solve a problem with meaningful real-life events. During the Covid pandemic it was common to take a quick home test to detect Covid, most commonly a cotton swab for nostrils and throat. Assume that you had just taken such a home test during the pandemic, with a positive result. Let us define the events of having Covid A = {covid} and getting a positive test B = {pos}. The test that you are using contains a leaflet with the following information:

- The **sensitivity** of the test is 96.77%. This is the probability of a positive test given that one has Covid, hence $\Pr(B|A) = 0.9677$.
- The **specificity** of the test is 99.20%. This is the probability of a



Figure 1.9: Reverend Thomas Bayes, ca 1701-1761, whose famous theorem was published after his death. Interestingly and somewhat ironically, we are not quite sure that the man in the photo actually is Thomas Bayes. *Probably not.*

Bayes' theorem

conditional probability

sensitivity

specificity

negative test when one does not have Covid, hence $\Pr(B^c|A^c) = 0.9920$, where A^c is the complement to the event A.

So a positive test is very unlikely if you do not have the disease, and you naturally start to worry.

But what you really want to know is the probability of having Covid given a positive home test, i.e. $\Pr(A|B)$. To compute this you need to know the so called *prior* probability of A before you took the test. Let us first assume we know nothing more than that the current prevalence of Covid in the population is around 5%, i.e. we use $\Pr(A) = 0.05$. Bayes' theorem reverses the conditioning and gives us the sought conditional probability:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|A^c)\Pr(A^c)} \approx 0.864,$$

where we have expressed $\Pr(B)$ in the numerator using a version of the **law of total probability** in Box 1.4 with $A_1 = A$ and $A_2 = A^c$. The probability $\Pr(B|A^c)$ is not given directly in the problem, but can be computed with the complement rule $\Pr(B|A^c) = 1 - \Pr(B^c|A^c)$. Hence, even though the test has increased the probability of having the disease by a factor of $0.864/0.05 = 17.28$, the probability of actually having Covid is far from conclusive: there is a good chance $1 - 0.864 = 0.136$ of not having Covid after a positive test.

A crucial assumption in this calculation is that your prior probability of having Covid before you took test is the prevalence of Covid in the population. This may be sensible if you were randomly selected to take the test *without any other symptoms*, but the reason why you took the test in the first place is probably because you had some symptoms of Covid (fever, coughing etc). Given such symptoms, you may assess your prior probability to be $\Pr(A) = 0.7$ and the posterior probability after the positive test then rises to $\Pr(A|B) = 0.9965$. It is now almost certain that you have Covid. The lesson here is that prior probabilities matter. This [observable widget](#) lets you experiment with different sensitivity, specificity and prior probability.

To see how Bayes' theorem can be used for Bayesian learning from data, let us consider the event $B = \{\text{Data } x_1, \dots, x_n \text{ was observed}\}$ which we write simply as $B = \{x_1, \dots, x_n\}$. We can now use Bayes' theorem to update the initial beliefs $\Pr(A)$ about some event A with data $B = \{x_1, \dots, x_n\}$ by the formula

$$\Pr(A|x_1, \dots, x_n) = \frac{\Pr(x_1, \dots, x_n|A)\Pr(A)}{\Pr(x_1, \dots, x_n)}.$$

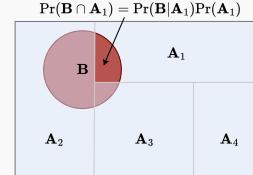
The initial belief $\Pr(A)$ is called a **prior** probability since it refers to a belief about the event A *before* the data x_1, \dots, x_n was observed. In the same way, $\Pr(A|x_1, \dots, x_n)$ is referred to as the **posterior** probability

law of total probability

Law of total probability for events

Let A_1, \dots, A_K be events that partitions the outcome space S , i.e. non-overlapping events $A_k \subset S$ covering all of S . Let $B \subset S$ another event. Then

$$\Pr(B) = \sum_{k=1}^K \Pr(B|A_k)\Pr(A_k)$$



Box 1.4: Law of total probability for events.

prior

posterior

since it is the probability of A *after* the dataset is observed.

The final step is to show how Bayes' theorem can be used to infer a parameter in a probability model $p(X_1, \dots, X_n | \theta)$. One way to see the connection between a continuous parameter θ and the events A mentioned so far is by defining A to be the event that the model parameter θ belongs to an interval $\theta \in [a, b]$, for some constants $a < b$. We first take a simplified approach where the only possible parameter values are on a grid of values $\theta_1, \theta_2, \dots, \theta_K$; for example 0.1, 0.2, ..., 0.9 for the success probability $\theta \in [0, 1]$ in the iid Bernoulli model. Let $B = \{x_1, \dots, x_n\}$ be the event of observing a specific dataset and $A_k = \{\theta_k\}$ be the event that $\theta = \theta_k$. The posterior probability for each $A_k = \{\theta_k\}$ is then given by Bayes' theorem as

$$\Pr(\theta_k | x_1, \dots, x_n) = \frac{\Pr(x_1, \dots, x_n | \theta_k) \Pr(\theta_k)}{\sum_{j=1}^K \Pr(x_1, \dots, x_n | \theta_j) \Pr(\theta_j)}. \quad (1.5)$$

Note how we again used the law of total probability in the denominator to express $\Pr(B) = \Pr(x_1, \dots, x_n)$. This denominator is only there to guarantee that the posterior is a probability distribution, i.e. that $\sum_{j=1}^K \Pr(\theta_j | x_1, \dots, x_n) = 1$.

The really interesting thing is however in the numerator of (1.5) and we will therefore often write Bayes' theorem in proportional form

$$\Pr(\theta_k | x_1, \dots, x_n) \propto \Pr(x_1, \dots, x_n | \theta_k) \Pr(\theta_k), \quad (1.6)$$

where the symbol \propto is read as 'is proportional to', meaning that a multiplicative normalizing constant is missing in the expression. Now here is the really crucial thing: the factor $\Pr(x_1, \dots, x_n | \theta_k)$ in Equation (1.6) is the *likelihood function* evaluated in the point θ_k . Equation (1.6) therefore expresses the fundamental updating rule of Bayesian learning:

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}.$$

The top row of Figure 1.11 illustrates the updating from prior to posterior for the Bernoulli model with data $n = 10$ and $s = 9$ over a grid of θ values. Note how the posterior is a compromise between the prior information and the data information (likelihood).

Finally, taking a finer and finer grid in Equation 1.5 we get the following Bayes' theorem for a continuous parameter θ in the limit

$$p(\theta | x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n | \theta) p(\theta)}{\int p(x_1, \dots, x_n | \theta) p(\theta) d\theta}, \quad (1.7)$$

where $p(\theta)$ is now a continuous **prior density** that gets updated with new data via the likelihood function $p(x_1, \dots, x_n | \theta)$ to a **posterior density** $p(\theta | x_1, \dots, x_n)$. The normalizing constant is now given by an

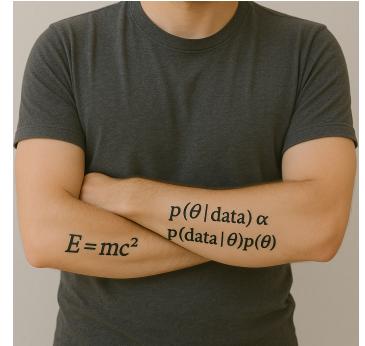


Figure 1.10: Great theorems make great tattoos.

prior density

posterior density

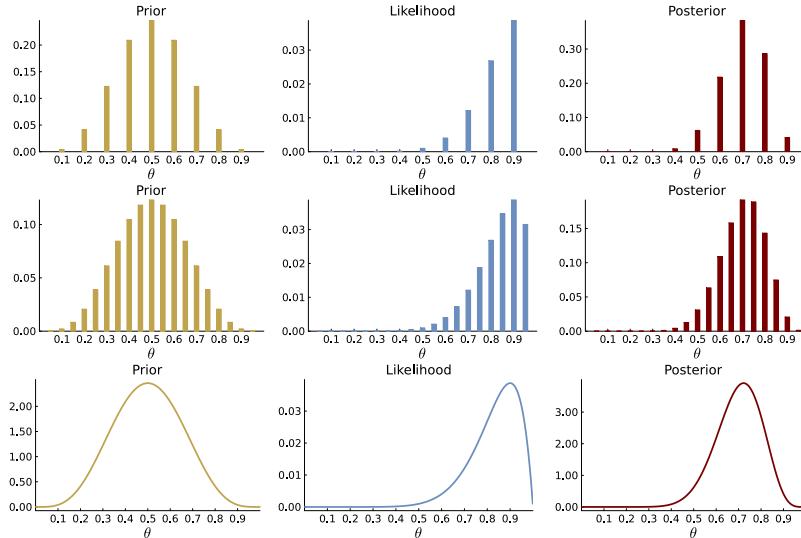


Figure 1.11: Prior, likelihood and posterior for Bernoulli model with $n = 10$ and $s = 9$. The top row plots over a coarse grid of θ values. The middle row uses a finer grid. Finally, the bottom row plot uses the continuous version of Bayes' theorem and plots the prior and posterior as density functions.

integral over θ and is a continuous version of the law of total probability, see Box 1.5. We can again hide the unimportant normalizing constant to get the core form of the **Bayesian updating** rule:

$$p(\theta|x_1, \dots, x_n) \propto p(x_1, \dots, x_n|\theta)p(\theta). \quad (1.8)$$

The middle row of Figure 1.11 illustrates the updating from prior to posterior for the Bernoulli model with data $n = 10$ and $s = 9$ over a finer grid of θ values. The bottom row of the same figure uses the continuous version of Bayes' theorem and plots the prior and posterior as density functions; the next chapter presents a detailed Bayesian analysis of the Bernoulli model with a continuous prior density.

It is important to note that the posterior distribution $p(\theta|x_1, \dots, x_n)$ is a probability distribution for the parameter θ ; it completely describes the knowledge about θ for a person with the prior $p(\theta)$ after having observed the data x_1, \dots, x_n . In contrast to the likelihood, with the posterior distribution we actually *can* compute probabilities for θ , for example the posterior probability that θ is below some constant c :

$$\Pr(\theta \leq c|x_1, \dots, x_n) = \int_{-\infty}^c p(\theta|x_1, \dots, x_n)d\theta$$

or any other posterior probability of interest. It is the prior $p(\theta)$ that makes it possible to use Bayes' theorem to revert the conditioning in the likelihood $p(x_1, \dots, x_n|\theta)$ into the conditional probability that we really care about, the posterior $p(\theta|x_1, \dots, x_n)$; you need the prior to get the posterior. As Leonard Jimmie Savage, a founder of Bayesian analysis, has famously said:

Law of total probability - continuous form

Let X and Y be continuous random variables with marginal densities $p_X(x)$ and $p_Y(y)$ and conditional density $p(x|y)$. Then

$$p_X(x) = \int p(x|y)p_Y(y)dy$$

Box 1.5: Law of total probability for continuous random variables.

You can't cook the Bayesian omelette without breaking the Bayesian eggs.

Leonard Jimmy Savage

The ability to use prior information is a strength, especially when one has to make a decision based on very little or weak data. Later in the book we will see how priors can be used to convey the idea that a functional relationship between two variables is in some sense smooth, and how this can prevent models from overfitting the data. Nevertheless, the subjective elements of a Bayesian analysis can complicate the reporting of scientific evidence, where objectivity is the ideal. One can argue that objectivity is simply unattainable, and that the supposedly objective alternatives to Bayesian learning just sweeps the subjective elements under the carpet. A more pragmatic Bayesian approach for scientific communication is presented in Section 4.7 where priors are intentionally chosen to be neutral or minimally informative. Section 4.8 gives an alternative approach to so called objective priors using invariance arguments.

There are two aspects of the Bayesian approach that gives it a clear scientific character. The prior distribution is subjective, and therefore varies from person to person, but the rule that updates the beliefs with new data is objective: we *should* use Bayes' theorem and the data *should* enter the updating *only through the likelihood function*. The word 'should' is emphasized here since one can mathematically derive this result from some simple axioms, and it can be proved to be the optimal way to process information; see [Bernardo and Smith \(2009\)](#) and Section 2.7. Second, one can prove that the effect of the prior vanishes asymptotically as the sample size n grows large; objectivity is attained by a **subjective consensus**: persons with wildly different priors will eventually reach the same posterior distribution as we collect more data. This result is given in Chapter 8 and we will see an empirical demonstration of this effect already in the next chapter.

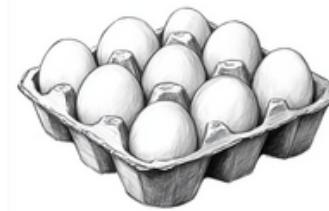


Figure 1.12: Making a Bayesian omelette.

subjective consensus

EXERCISES

Exercise 1.1

A university uses an automatic tool to detect plagiarism in student essays. The tool has a sensitivity of 0.95 (probability of flagging plagiarism when the essay is plagiarized) and a specificity of 0.90 (probability of not flagging plagiarism when the essay is not plagiarized). Assume that 1% of all students actually plagiarize. If a student is flagged by the tool, what is the probability that the student actually has plagiarized?

Exercise 1.2

Think about an event that you are uncertain about, for example the event that your favorite sports team wins their next game. Try to elicit your subjective probability for the event by considering a betting situation where you win \$100 if the event occurs. Start with a price of \$1 and ask yourself if you would be willing to take the bet. Then gradually increase the price of the bet until you are indifferent between taking the bet or not.

Exercise 1.3

Think about a political party that you care about. Elicit a histogram to represent your prior distribution for the party's support in percent, $0 \leq \theta \leq 100$, in the next national election, by asking yourself questions about the probability of certain intervals. For example, what is the probability that the party's support is below 10%? Between 10% and 20%? And so on. Make sure that the final histogram integrates to one over the full support, and plot it.

Exercise 1.4

Reproduce the first row of Figure 1.11 by writing your own code in your favorite programming language.

2 Single-parameter models

Now that we have covered the basics of Bayesian updating — how prior beliefs are revised in light of new data — we turn to models with a single parameter. Working with these simple models gives us the opportunity to practice deriving posterior distributions in clear, manageable settings. The drawback of simple models is that they do not show anywhere near the full potential of Bayesian methods. But as with any skill, one must crawl before walking, and a bit of patience is needed before we reach the more interesting models in later chapters.

2.1 Bernoulli data

Let us return to the iid Bernoulli model for binary data where

$$X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} \text{Bern}(\theta), \quad (2.1)$$

where $0 \leq \theta \leq 1$ is the success probability. The starting point for any Bayesian analysis is to formulate a prior distribution, $p(\theta)$, that summarizes the beliefs about θ , before observing the data. The act of extracting a prior distribution from a person, for example an expert, is called **prior elicitation**. Prior elicitation can be done in many ways, and the techniques involve ideas from psychology. The most common procedures consist of asking a series of questions, followed by checks for internal consistency of the elicited prior beliefs, and potential adjustments of the initial prior. One can in principle elicit a distribution nonparametrically, e.g. in the form of a histogram, but the most common approach is to use a suitable distributional family and then elicit the hyperparameters within the family.

Since θ is a probability in the Bernoulli model, and is therefore restricted to the unit interval $\theta \in [0, 1]$, the **Beta distribution** is a commonly used two-parameter family with quite a lot of flexibility; see Box 2.1 for some properties of the Beta distribution and Figure 2.1 for plots of a few members of the Beta family. Note that we are using the word *family* to describe that that the Beta distribution is a

Beta distribution

$X \sim \text{Beta}(\alpha, \beta)$ for $X \in [0, 1]$.

$$p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

$$\mathbb{E}(X) = \frac{\alpha}{\alpha + \beta}$$

$$\mathbb{V}(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, and $\Gamma(\alpha)$ is the Gamma function.

Box 2.1: The Beta distribution.

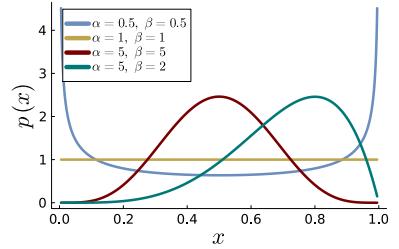


Figure 2.1: Some Beta distributions.

prior elicitation

Beta distribution

whole family of different shapes with each choice of α and β determining one member of the family. Setting $\alpha = \beta = 1$ makes the pdf constant over $[0, 1]$, so the family member Beta(1, 1) is the **uniform distribution** (Box 2.2).

It is common to refer to α and β as **prior hyperparameters**, since they are parameters that determine the shape of the prior distribution. The parameter θ that appears in the likelihood function is then called a *model parameter*, or simply a *parameter*, to distinguish it from the hyperparameters α and β in the prior. This distinction will be become more blurred in later chapters when we work with hierarchical models and priors.

We will now show that the Beta family is particularly convenient as a prior for the iid Bernoulli model since the posterior turns out to also be a Beta distribution. To derive the posterior distribution of a parameter θ we always start with Bayes' theorem (1.8):

$$p(\theta|x_1, \dots, x_n) \propto p(x_1, \dots, x_n|\theta)p(\theta),$$

where $p(x_1, \dots, x_n|\theta) = \theta^s(1-\theta)^f$ is the likelihood for iid Bernoulli data from (1.3), and $p(\theta)$ is the Beta(α, β) prior for θ . So, using the pdf of the Beta distribution from Box 2.1 (with θ as the random variable), we get

$$\begin{aligned} p(\theta|x_1, \dots, x_n) &\propto \theta^s(1-\theta)^f \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} \\ &\propto \theta^{\alpha+s-1}(1-\theta)^{\beta+f-1}, \end{aligned} \quad (2.2)$$

where in the second line we drop the normalizing constant $1/B(\alpha, \beta)$ in the Beta prior by absorbing that factor in the missing proportionality constant. Note that $1/B(\alpha, \beta)$ is a multiplicative constant that does *not* depend on θ and will therefore not affect the shape of the posterior distribution, only its vertical scale. In the final step we will reintroduce the normalizing constant so that $p(\theta|x_1, \dots, x_n)$ integrates to one over its support, as required. This is a common pattern in Bayesian derivations: constants that do not depends on θ , are quickly removed during the derivation of the posterior to simplify the calculations.

From the functional form of the pdf of the Beta distribution in Box 2.1, we can recognize that the expression in (2.2) is proportional to a Beta distribution. With some practice this becomes easy to spot: the expression in (2.2) is of the form

$$\theta^{a-1}(1-\theta)^{b-1},$$

where $a = \alpha + s$ and $b = \beta + f$, and any function of that form is proportional to a particular member of the Beta family. It follows that the function in (2.2) is proportional to the density of a

uniform distribution
prior hyperparameters

Uniform distribution

$X \sim \text{Uniform}(a, b)$, $X \in [a, b]$.

$$\begin{aligned} p(x) &= \frac{1}{b-a} \\ \mathbb{E}(X) &= \frac{a+b}{2} \\ \mathbb{V}(X) &= \frac{(b-a)^2}{12} \end{aligned}$$

Box 2.2: The uniform distribution.

$\theta \sim \text{Beta}(\alpha + s, \beta + f)$ distribution. To turn this proportionality into an normalized probability density, we simply restore the missing normalizing constant, which must be $1/B(\alpha + s, \beta + f)$ so that the pdf integrates to one over the support. The Bayesian updating for the Bernoulli model is summarized in Box 2.3.

Conjugate analysis - Bernoulli model

Model: $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{Bern}(\theta)$

Prior: $\theta \sim \text{Beta}(\alpha, \beta)$

Posterior: $\theta|x_1, \dots, x_n \sim \text{Beta}(\alpha + s, \beta + f)$

where $s = \sum_{i=1}^n x_i$ and $f = n - s$.

Box 2.3: Bayesian updating for the iid Bernoulli model with a Beta prior.

Using a Beta prior for the Bernoulli parameter is convenient because the posterior distribution then belongs to the *same distributional family* as the prior distribution; a Beta prior for the Bernoulli model gives a Beta posterior. The Beta family is said to be *conjugate* to the Bernoulli model, or equivalently that the Beta distribution is the **conjugate prior** for the Bernoulli model. Conjugate priors are easy to use, since updating is simple: to update a Beta prior with Bernoulli data, we just add the number of successes s to α and the number of failures f to β .

The way that α and β enter the posterior makes it clear that we can interpret the information in a $\text{Beta}(\alpha, \beta)$ prior as if it contained the information from a prior dataset with α successes and β failures. We usually do not have an explicit prior sample at hand, and α and β need not even be integers. Nevertheless, it is helpful to view the prior hyperparameters α and β as an **imaginary prior sample**, summarizing our prior beliefs before seeing the actual data.

Similar conjugate results for several other models will be presented in this book, but there are many models for which a known conjugate prior do not exist. For such models, the posterior is often not available in closed form, but several easy-to-use approximation or simulation methods are presented in later chapters.

EXAMPLE: SPAM EMAILS. The **SpamBase dataset** from the UCI repository¹ consists of 4601 emails sent to a single person that have been manually classified as *spam* (junk email) or *ham* (non-junk email). The dataset also contains a vector of covariates/features for each email, such as the number of capital letters or \$-signs; this information can be used to build a spam filter that automatically separates

conjugate prior

imaginary prior sample

SpamBase dataset

¹ Dua, D. and Graff, C. (2017). UCI machine learning repository

spam from ham. We will in this chapter only analyze the proportion of spam emails without using the covariates; we return to the more interesting case with features in Chapter 8 on classification.

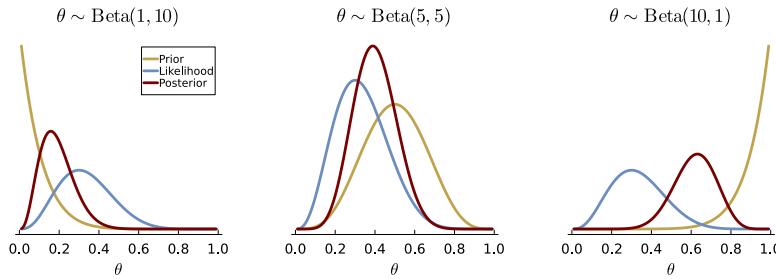


Figure 2.2: Bayesian analysis of $n = 10$ randomly chosen emails from the SpamBase data using three different priors. The likelihood is normalized.

So, let $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} \text{Bern}(\theta)$ for the $n = 4601$ emails, where $x_i = 1$ if the email is spam and $x_i = 0$ for ham. The unknown parameter θ is the probability of spam, and we use a $\theta \sim \text{Beta}(\alpha, \beta)$ prior. To illustrate the incremental learning process in Bayesian learning we start off by analyzing only $n = 10$ randomly sampled emails, out of which $s = 4$ are spam. Figure 2.2 shows the posterior distribution (red curves) of θ for three persons with very different priors (yellow curves). The likelihood is also shown in blue and has been normalized to integrate to one to have the same vertical scale as the prior and posterior density for visualization purposes. With only $n = 10$ data points, the posteriors for the three persons with highly diverse prior beliefs are naturally quite different. Figure 2.3 adds another 90 data points, so the posterior is now based on $n = 100$ emails. The posteriors are now in rather close, but not perfect agreement. Finally, Figure 2.4 shows the posterior for the full dataset with $n = 4601$ data points; there is now a complete subjective consensus between the three persons that initially had very different beliefs about the spam probability.

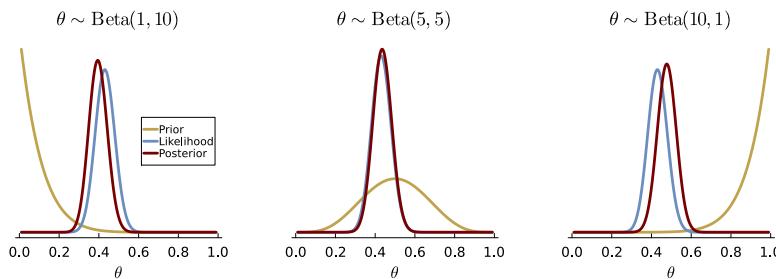


Figure 2.3: Bayesian analysis of $n = 100$ randomly chosen emails from the SpamBase data using three different priors. The likelihood is normalized.

From this dataset we have learned that around we expected around 40% of all emails to this particular person to be spam, and

we are also quite certain about this as the posterior distribution is tightly concentrated around 0.4. This information is however not useful for building a spam filter where one instead needs the spam probability for each email to be a function of the text in that specific email (e.g. the number of \$-signs). We will achieve this in Chapter 8 when we derive the posterior for a binary regression model and use the methods in Chapter 6 to construct Bayesian spam predictions.

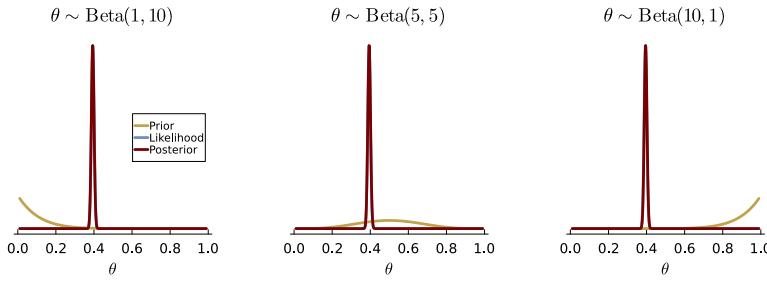


Figure 2.4: Bayesian analysis of all $n = 4601$ emails from the SpamBase data using three different priors. The likelihood is normalized.

Note that we have implicitly assumed that we have access to all the n binary data observations x_1, \dots, x_n when constructing the Bernoulli likelihood in (1.3). Sometimes we only get the data in summarized form as s recorded successes in n trials, without knowing exactly which trials were successful and which were failures. For example, a dataset summarized as $s = 2$ successes in $n = 3$ trials can be obtained in three different ways:

1. $x_1 = 0, x_2 = 1$ and $x_3 = 1$
2. $x_1 = 1, x_2 = 0$ and $x_3 = 1$
3. $x_1 = 1, x_2 = 1$ and $x_3 = 0$

In general, a dataset with s successes in n trials can be obtained in $\binom{n}{s} = \frac{n!}{s!(n-s)!}$ ways. When the data are available only in summarized form — as the number of successes and failures — the likelihood is given by the binomial distribution in Box 1.2

$$p(s) = \binom{n}{s} \theta^s (1-\theta)^{n-s}, \quad (2.3)$$

instead of the Bernoulli distribution; note the extra binomial factor $\binom{n}{s}$. As we explain more fully in Section 2.7, since the extra factor $\binom{n}{s}$ does not depend on θ , it has no effect on the posterior distribution. So whether or not we are given the full dataset x_1, \dots, x_n or just the **summary statistics** s and f does not matter for Bayesian learning. This is an example of the *likelihood principle* discussed in Section 2.7. Of course, we cannot use *any* summary statistic and expect it to contain all the relevant data information about θ ; the summary statistics must be *sufficient* for θ , as is the case for s and f in the Bernoulli

summary statistics

model. The concept of sufficient statistics is discussed in more detail in Section 2.8.

It is interesting to compare a Bayesian analysis of Bernoulli data with the maximum likelihood estimator $\hat{\theta}_{MLE} = s/n$. A common **Bayes estimator**, or Bayesian point estimator, is the posterior mean $E(\theta|x_1, \dots, x_n) = \frac{\alpha+s}{\alpha+\beta+n}$, which follows directly from the formula for the mean of a Beta distribution. Let us also assume a uniform prior for θ as some sort of non-informative prior, i.e. our prior is the Beta(1,1) distribution. Consider the case when we have observed no successes ($s = 0$) in a small number of trials n . We then have the quite unreasonable MLE of $\hat{\theta}_{MLE} = 0$, whereas the Bayes estimator is $E(\theta|x_1, \dots, x_n) = 1/(n+2) > 0$. We will return to this example and the idea of a non-informative prior in Sections 4.7 and 4.8.

Bayes estimator

2.2 Gaussian data - known variance

In this section we derive the posterior distribution for the mean in the iid Gaussian model

$$X_1, \dots, X_n | \theta, \sigma^2 \stackrel{iid}{\sim} N(\theta, \sigma^2), \quad (2.4)$$

see Box 2.4 for some properties of the Normal distribution and Figure 2.5 for plots of a few members of the Normal family of distributions. Since this chapter is about models with a single parameter, we will assume the variance σ^2 to be known; this is rarely the case in practice and we return to the Gaussian model with both parameters unknown in Chapter 3.

Uniform prior

We will first derive the posterior for a so called non-informative prior, i.e. a prior that is supposed to contain no, or at least very little, prior information. The most common non-informative prior for θ is a uniform distribution $p(\theta) = c$ for $\theta \in \mathbb{R}$ where $c > 0$ is a constant; the idea is that this flat distribution does not favor any particular value for θ . A uniform distribution over an unbounded space is not a proper distribution since $\int_{-\infty}^{\infty} p(\theta)d\theta = \infty$. It is nevertheless possible to use this somewhat strange prior since the resulting posterior is proper after observing a single data point. We can also think about the uniform prior as a limiting normal distribution with a variance that tends to infinity; in practice there is no difference between a uniform prior over \mathbb{R} and a $N(0, 1000000^2)$ distribution, but the latter is a proper distribution.

By Bayes' theorem, the posterior distribution for θ from a uniform

Normal distribution

$$X \sim N(\mu, \sigma^2)$$

Support: $X \in (-\infty, \infty)$

$$p(x) = \frac{\exp(-\frac{1}{2\sigma^2}(x-\mu)^2)}{\sqrt{2\pi\sigma^2}}$$

$$E(X) = \mu$$

$$V(X) = \sigma^2$$

Box 2.4: The Normal (Gaussian) distribution.

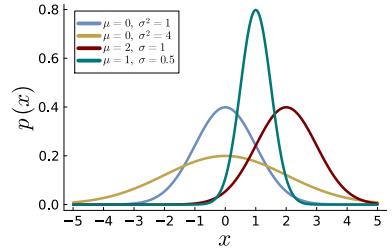


Figure 2.5: Some Normal distributions.

prior is

$$p(\theta|x_1, \dots, x_n) \propto p(x_1, \dots, x_n|\theta)p(\theta) \quad (2.5)$$

$$= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \theta)^2\right) \cdot c \quad (2.6)$$

$$\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \theta)^2\right), \quad (2.7)$$

where the multiplicative constants $\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}}$ and c have been removed since they do not depend on θ ; remember that σ^2 is assumed known. Let $\bar{x}_n = n^{-1} \sum_{i=1}^n x_i$ be the sample mean, we can add and subtract \bar{x} inside the parenthesis in the sum to get

$$\sum_{i=1}^n (x_i - \theta)^2 = \sum_{i=1}^n (x_i - \bar{x} - (\theta - \bar{x}))^2 = \sum_{i=1}^n (x_i - \bar{x})^2 + n(\theta - \bar{x})^2, \quad (2.8)$$

since the cross term $2(\theta - \bar{x}) \sum_{i=1}^n (x_i - \bar{x})$ is zero because $\sum_{i=1}^n (x_i - \bar{x}) = \sum_{i=1}^n x_i - n\bar{x} = 0$. Note that the term $\sum_{i=1}^n (x_i - \bar{x})^2$ does not depend on θ and by inserting (2.8) in (2.5) we therefore get

$$p(\theta|x_1, \dots, x_n) \propto \exp\left(-\frac{n}{2\sigma^2}(\theta - \bar{x})^2\right). \quad (2.9)$$

A function proportional to the exponential of a quadratic function of θ is proportional to a normal density, and hence that the posterior for θ in (2.9) can be recognized as

$$\theta|x_1, \dots, x_n \sim N\left(\bar{x}, \frac{\sigma^2}{n}\right). \quad (2.10)$$

The maximum likelihood estimator of θ in the iid normal model is the sample mean $\hat{\theta}_{MLE} = \bar{X}$, with sampling variance $V(\bar{X}) = \sigma^2/n$. The Bayesian posterior in (2.10) there agrees with the classical frequentist result in this case since we used a non-informative, uniform, prior. Even so, the Bayesian posterior has a direct *probabilistic interpretation* as the updated beliefs about θ *conditional* on the observed data, which is not the case for the frequentist sampling distribution of the MLE.

Normal prior

Consider now a normal prior, $\theta \sim N(\mu_0, \tau_0^2)$; following Gelman et al. (2013) the subscript 0 is used to denote that these are **hyperparameters** in the prior, i.e. based on 0 observations. The user must determine the prior mean μ_0 as the most probable value for θ a priori, and also how sure she is by setting the prior standard deviation, τ_0 . One way to elicit these prior hyperparameters is to ask the user

hyperparameters

for a 95% prior probability interval (l, u) for θ and then solve for μ_0 and τ_0 from the equations

$$\begin{aligned} l &= \mu_0 - 1.96\tau_0 \\ u &= \mu_0 + 1.96\tau_0, \end{aligned}$$

with solution $\mu_0 = (l + u)/2$ and $\tau_0 = (u - l)/(2 \cdot 1.96)$.

The posterior distribution for θ can be derived by using Bayes' theorem

$$p(\theta|x_1, \dots, x_n) \propto \exp\left(-\frac{n}{2\sigma^2}(\theta - \bar{x})^2\right) \times \exp\left(-\frac{1}{2\tau_0^2}(\theta - \mu_0)^2\right),$$

where the first term is proportional to the likelihood in (2.9) and the second term is the normal prior, and the normalizing constants have again been removed. In Exercise 2.6 you are asked to complete the squares in this expression to prove that this expression is proportional to a normal density of the form given in Box 2.5.

Conjugate analysis - Gaussian model with known variance

Model: $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$, σ^2 known

Prior: $\theta \sim N(\mu_0, \tau_0^2)$

Posterior: $\theta|x_1, \dots, x_n \sim N(\mu_n, \tau_n^2)$

Posterior precision: $\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$

Posterior mean: $\mu_n = w\bar{x} + (1-w)\mu_0$, where $\bar{x} = n^{-1} \sum_{i=1}^n x_i$

Posterior weight: $w = \frac{n/\sigma^2}{n/\sigma^2 + 1/\tau_0^2}$

Box 2.5: Bayesian updating for normal data with known variance and a normal prior for the mean.

The normal prior is therefore conjugate to the normal model with known variance: a normal prior gives a normal posterior. The interpretations of the posterior mean μ_n and the posterior variance τ_n^2 in Box 2.5 are quite intuitive. Note first that the expression for the posterior variance τ_n^2 is written in terms of the **precision**

precision

$$\text{Precision} = \frac{1}{\text{Variance}},$$

so that a large variance is the same as a low precision, and vice versa. The first term n/σ^2 is the precision in the data. This can be seen in several ways, for example since the frequentist sampling variance is $V(\bar{X}) = \sigma^2/n$ we can write $n/\sigma^2 = 1/V(\bar{X}) = \text{Precision}(\bar{X})$. Hence the formula for the posterior precision

$$\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$$

can be read

$$\text{Posterior precision} = \text{Data precision} + \text{Prior precision}.$$

The posterior mean $\mu_n = w\bar{x} + (1 - w)\mu_0$ is a weighted average of the data mean \bar{x} and the prior mean. The weight w on \bar{x} in Box 2.5 is the data precision relative to the prior precision. The posterior therefore puts more emphasis on the data when n is large, σ small or τ_0 is large. It will not always be possible to get this clear a view of the Bayesian updating in other models, but the same logic will apply also there: the posterior combines the data and prior information, and puts more weight on the source of information that is most precise.

EXAMPLE: INTERNET CONNECTION SPEED. The maximum internet connection speed downstream in my home is 50 Mbit/sec. This maximum will typically never be reached, but my internet service provider (ISP) claims that the average speed is *at least* 20Mbit/sec. To test this, I collect a total of five measurements, $\mathbf{x} = (15.77, 20.5, 8.26, 14.37, 21.09)$, over the course of five consecutive days using a speed testing internet service; I will call this the **Internet speed dataset**. The measurements are assumed to be $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$, where θ is the average speed; we ignore for simplicity that the normal model is not appropriate here in a strict sense as it allows for negative measurements, but it is a decent approximation since the measurements are far enough from zero in relation to their spread. The measurements are reported to have a standard deviation of $\sigma = 5$ by the speed testing service. I will use a prior centered on the average speed claimed by the ISP, $\mu_0 = 20$, with a prior standard deviation of $\tau_0 = 5$. My prior beliefs are therefore that $\theta \in (10, 30)$ with approximately 95% probability.

Internet speed dataset

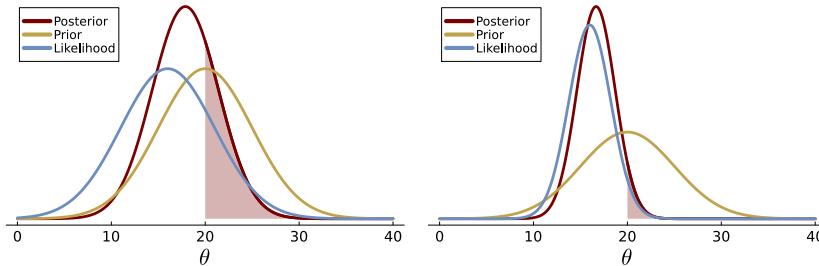


Figure 2.6: Internet speed data. Posterior updating based on $n = 1$ observation (left) and $n = 5$ observations (right). The shaded region marks out $\Pr(\theta > 20 | x_1, \dots, x_n)$.

Figure 2.6 (left) displays the prior, normalized likelihood and posterior of θ based on only the first measurement $x_1 = 15.77$ Mbit/sec; the probability of interest $\Pr(\theta \geq 20 | x_1, \dots, x_n) \approx 0.275$ is marked out by the shaded red region. Since the prior precision

happened to be equal to the data precision of a single observation, the weight on the data in the posterior mean μ_n is exactly $w = 0.5$. Figure 2.6 (right) shows the updated posterior using all $n = 5$ data points with $\bar{x} = 16.001$; we are beginning to be rather confident that the ISP's claim that $\theta \geq 20$ is false since we now have $\Pr(\theta \geq 20 | x_1, \dots, x_n) \approx 0.051$, which is rather small. The weight w is now 0.833 so the data is starting to dominate the prior. An [interactive Observable notebook](#) is available where you can change the prior and data to see how the posterior changes.

2.3 Online learning

Figure 2.6 illustrates a situation where the posterior is computed by combining the prior at day 0, $N(\mu_0, \tau_0^2)$, with the likelihood for all x_1, \dots, x_n data points; hence the posterior on day n is computed as

$$p(\theta | x_1, \dots, x_n) \propto p(x_1, \dots, x_n | \theta) p(\theta). \quad (2.11)$$

We can however equally well compute this posterior by updating yesterday's posterior $p(\theta | x_1, \dots, x_{n-1})$ with today's measurement x_n by

$$p(\theta | x_1, \dots, x_n) \propto p(x_n | \theta) p(\theta | x_1, \dots, x_{n-1}). \quad (2.12)$$

Note how yesterday's posterior $p(\theta | x_1, \dots, x_{n-1})$ plays the role of today's prior in (2.12). The word 'prior' is here used in the sense of *before today's data point x_n* . The updating in (2.11) and (2.12) give the same result, but (2.12) can be used sequentially in what is often called **online learning** or **sequential learning**, where "yesterday's posterior becomes today's prior". Note that the online result in (2.12) is not specific for normal data with a normal prior, but is a general property of Bayesian updating. Figure 2.7 illustrates Bayesian online learning for the internet speed data.

online learning
sequential learning

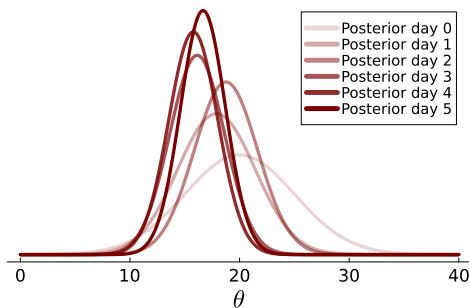


Figure 2.7: Illustration of Bayesian online learning for the internet speed data. The figure shows how the posterior changes when each new data point arrives. The posterior at Day 0 is just the original prior.

The same online learning holds also for dependent data, e.g. time

series, as is easily proved as follows

$$\begin{aligned} p(\theta|x_1, \dots, x_n) &\propto p(x_1, \dots, x_n|\theta)p(\theta) \\ &= p(x_n|\theta, x_1, \dots, x_{n-1})p(x_1, \dots, x_{n-1}|\theta)p(\theta) \\ &\propto p(x_n|\theta, x_1, \dots, x_{n-1})p(\theta|x_1, \dots, x_{n-1}), \end{aligned} \quad (2.13)$$

where the second line follows from the decomposition results in Box 2.6. For iid data we have the additional simplification

$$p(x_n|\theta, x_1, \dots, x_{n-1}) = p(x_n|\theta),$$

hence showing the equivalence of (2.11) and (2.12) in the iid case.

By the same proof we also see that Bayesian methods are directly applicable in **batch learning**, where the posterior can be incrementally updated using batches of *several* observations, since for any $1 \leq m \leq n - 1$

$$p(\theta|x_1, \dots, x_n) \propto p(x_{m+1}, \dots, x_n|\theta)p(\theta|x_1, \dots, x_m). \quad (2.14)$$

Implementing online or batch learning is straightforward for conjugate models since:

- any intermediate posterior $p(\theta|x_1, \dots, x_m)$ belongs to the same distribution family as the original prior $p(\theta)$ and
- the prior is conjugate to the likelihood for any data, and the intermediate posterior $p(\theta|x_1, \dots, x_m)$ is therefore also conjugate to the likelihood of the new batch $p(x_{m+1}, \dots, x_n|\theta)$.

In the case of the iid normal model with known variance we have the recursions for observation $i = 1, 2, \dots$

$$\begin{aligned} \frac{1}{\tau_i^2} &= \frac{1}{\sigma^2} + \frac{1}{\tau_{i-1}^2} \\ w_i &= \frac{\sigma^{-2}}{\sigma^{-2} + \tau_{i-1}^{-2}} \\ \mu_i &= w_i x_i + (1 - w_i) \mu_{i-1}. \end{aligned}$$

Note that the posterior mean after observation i is a weighted average of the single new observation x_i and the *posterior mean after observation $i - 1$* , denoted by μ_{i-1} . The weight w_i is the data precision σ^{-2} relative to the posterior precision τ_{i-1}^{-2} after observation $i - 1$, so the weight on the new observation x_i decreases as more data is observed. When the prior is not conjugate one has to resort to numerical methods that can be more or less computationally attractive in online mode; see Chapter 9 on Gibbs sampling, Chapter 10 on Markov Chain Monte Carlo methods and Chapter 11 on approximate inference using variational methods.

Decomposing distributions

For two random variables X, Y the joint distribution can be decomposed as a conditional distribution times a marginal distribution

$$p(x, y) = p(y|x)p(x)$$

For n random variables

$$\begin{aligned} p(x_1, \dots, x_n) &= p(x_1)p(x_2|x_1) \times \\ &\dots \times p(x_n|x_1, \dots, x_{n-1}) \\ \text{and conditional on } \theta & \\ p(x_1, \dots, x_n|\theta) &= p(x_1|\theta) \times \\ &\dots \times p(x_n|x_1, \dots, x_{n-1}, \theta) \end{aligned}$$

Box 2.6: Marginal-Conditional decomposition of a joint distribution.

batch learning

2.4 Poisson data

Data on a count variable $X \in \{0, 1, 2, \dots\}$ is a common data type in many applications: for example, the number of software bugs, the number of lethal car accidents in a region, the number of people in intensive care during a pandemic, or the number of electric scooters available at a given pick-up station.

One of the simplest — and probably most commonly used — model for count data, is the **Poisson distribution**; see Box 2.7 for basic properties of the Poisson distribution and Figure 2.8 for plots of several members of the Poisson family. The mean and variance of a Poisson variable are equal, which can be restrictive in some applications, yet the model often fits many real datasets surprisingly well, or can be extended to do so.

The iid Poisson model is written as

$$X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} \text{Pois}(\theta), \quad (2.15)$$

where the parameter $\theta > 0$ is both the mean and the variance of the distribution. The likelihood function from iid Poisson observation is obtained by multiplying the individual Poisson densities for each observation to get

$$p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n \frac{\theta^{x_i} e^{-\theta}}{x_i!} \propto \theta^{\sum_{i=1}^n x_i} e^{-n\theta}. \quad (2.16)$$

Comparing the functional form of the Poisson likelihood in (2.16) with a list of common probability distributions we can see that the likelihood from iid Poisson data looks very much like a **Gamma distribution** in θ ; see the functional form of the density $p(x)$ in Box 2.8. Note that we are here looking for a distribution where the random variable is θ (in the subjective probability sense), and the data $\sum_{i=1}^n x_i$ is just a fixed constant at this stage. The form of the Gamma distribution suggests that a Gamma prior may indeed combine nicely with this likelihood. So let us try if $\theta \sim \text{Gamma}(\alpha, \beta)$ is conjugate to the iid Poisson model:

$$\begin{aligned} p(\theta | x_1, \dots, x_n) &\propto p(x_1, \dots, x_n | \theta) p(\theta) \\ &\propto \theta^{\sum_{i=1}^n x_i} e^{-n\theta} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta} \\ &= \theta^{\alpha+\sum_{i=1}^n x_i-1} e^{-(\beta+n)\theta}, \end{aligned}$$

where the normalizing constant of the Gamma prior $\frac{\beta^\alpha}{\Gamma(\alpha)}$ is absorbed in the proportionality sign. This expression is indeed proportional to a Gamma distribution with updated posterior hyperparameters $\alpha + \sum_{i=1}^n x_i$ and $\beta + n$. We summarize this result in Box 2.9.

Poisson distribution

$X \sim \text{Pois}(\theta)$ for $X = 0, 1, 2, \dots$

$$p(x) = \frac{\theta^x e^{-\theta}}{x!}$$

$$\mathbb{E}(X) = \theta$$

$$\mathbb{V}(X) = \theta$$

Box 2.7: The Poisson distribution.

Poisson distribution

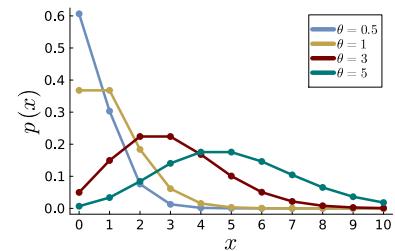


Figure 2.8: Some Poisson distributions.

Gamma distribution

$X \sim \text{Gamma}(\alpha, \beta)$ for $X > 0$.

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

$$\mathbb{E}(X) = \frac{\alpha}{\beta}$$

$$\mathbb{V}(X) = \frac{\alpha}{\beta^2}$$

Box 2.8: Gamma distribution.

Gamma distribution

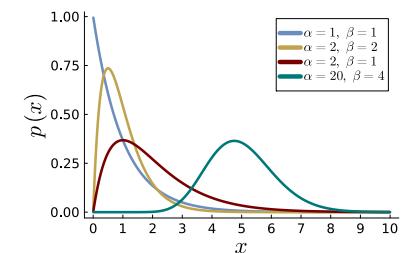


Figure 2.9: Some Gamma distributions.

Conjugate analysis - Poisson model

Model: $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{Pois}(\theta)$

Prior: $\theta \sim \text{Gamma}(\alpha, \beta)$

Posterior: $\theta|x_1, \dots, x_n \sim \text{Gamma}(\alpha + \sum_{i=1}^n x_i, \beta + n)$

Box 2.9: Bayesian updating for iid Poisson data with a Gamma prior.

EXAMPLE: INTERNET AUCTION DATA. The **eBayCoin dataset** collected by [Wegmann and Villani \(2011\)](#) and made available in the UCI repository² consist of data from 1000 eBay auctions of collectors coins. For each auction, the dataset records the final price of the auctioned coin, the number of bidders in the auction and a number of covariates such as the quality of the sold coin, the lowest price that the seller would agree to sell for etc. We will here analyze the number of bidders using an iid Poisson model without covariates. We return to this dataset in Chapter 8 where we make use of the covariates in a Poisson regression model for predicting the number of bidders.

Our aim here is to compute the posterior distribution for θ , the average number of bidders in an auction. From Box 2.9 we need the summary statistic $\sum_{i=1}^n x_i = 3635$. The sample mean in the $n = 1000$ auctions is therefore $\bar{x} = 3.635$ bidders per auction. We use a Gamma prior with $\alpha = 2$ and $\beta = 1/2$ for illustration; this prior has a mean of $E(\theta) = 4$ and a standard deviation of $S(\theta) = 2.283$, which seems like reasonable prior beliefs. The Gamma prior and the posterior updated with data from $n = 1000$ auctions are shown in Figure 2.10; note the different ranges on the horizontal axis. We are now more or less certain that the average number of bidders is in the interval $\theta \in (3.4, 3.9)$.

² <http://archive.ics.uci.edu/ml/datasets/eBayCoin/>

eBayCoin dataset

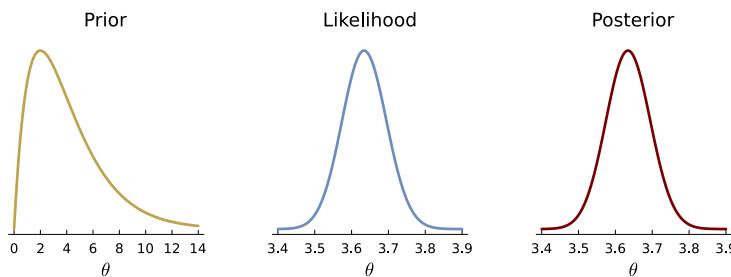


Figure 2.10: Bayesian analysis of the numbers of bidders in $n = 1000$ eBay coin auctions.

Figure 2.11 a) shows a histogram of the data and overlays the fitted Poisson distribution with θ set to the posterior mean. The fit is poor and it is clear that the Poisson distribution is too restrictive for this dataset. The poor fit likely reflects the heterogeneity of the

auctions; the auctioned coins differ in book value, quality and other features, and the auctions also vary along other dimensions. One important factor is the posted reservation price — the lowest bid the seller will accept — which varies substantially across auctions; an auction with a low reservation price attracts many bidders, while an auction with a high reservation price may discourage bidders from entering.

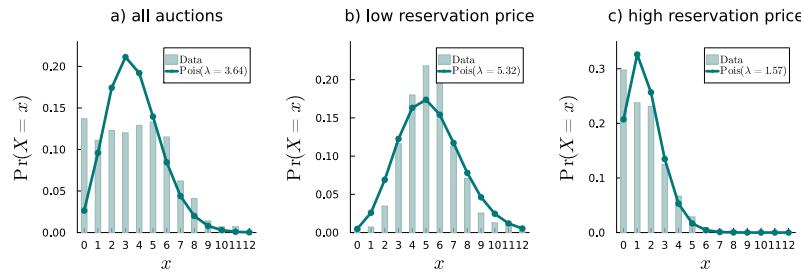


Figure 2.11: Assessing the fit of the Poisson model with the posterior mean estimate of θ .

To explore the effect of the reservation price we split the data into auctions with low and high reservation prices, and analyze the two auction types separately. The prior for the auction with low reservation prices is set to $\text{Gamma}(4, 1/2)$ to reflect a belief that such auctions are likely to attract more bidders (prior mean is 8 bidders). The prior for the auctions with high reservation prices is set to $\text{Gamma}(1, 1/2)$ (prior mean is 2 bidders). The Bayesian updating is shown in Figure 2.12. The posteriors in the right hand graph are clearly different in the two subpopulations. The Poisson model fits better on the two subpopulations, as shown in Figure 2.11 b) and c), but there is room for improvement. We will return to this dataset in Chapter 8 using a Poisson regression with the reservation price and other auction specific covariates.

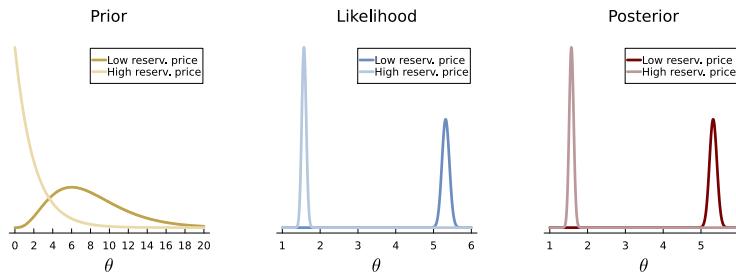


Figure 2.12: eBay auctions. Bayesian analysis of the numbers of bidders in $n = 550$ auctions with a low reservation price and $n = 450$ auctions with a high reservation price.

We can summarize the conjugate priors encountered so far by

- the Beta prior is conjugate to the Bernoulli likelihood
- the Normal prior is conjugate to the Normal likelihood
- the Gamma prior is conjugate to the Poisson likelihood,

and we will see some more examples of models with conjugate priors in the next chapters. It is important to remember that a prior is conjugate to a *specific* model, or equivalently, to a family of likelihood functions. For example, the iid Bernoulli model gives rise to a family of likelihood functions of the form $\theta^s(1-\theta)^f$, where each specific θ value in the parameter space Θ corresponds to one member of the family. The Beta prior is conjugate to this family of likelihoods. Here is a formal definition of a conjugate prior that highlights this point.

Definition (Conjugate prior). A family of prior distributions \mathcal{P} is *conjugate* to a family of likelihoods $\mathcal{L} = \{p(\mathbf{x}|\theta), \theta \in \Theta\}$ if

$$p(\theta) \in \mathcal{P} \Rightarrow p(\theta|\mathbf{x}) \in \mathcal{P} \quad \text{for all } p(\mathbf{x}|\theta) \in \mathcal{L}.$$

2.5 Summarizing a posterior distribution

The posterior distribution for models with a single parameter is easy to plot and provides a clear visual summary of the uncertainty. Starting in the next chapter, our models will typically involve more than one parameter, often quite a few. In such cases, plotting the entire posterior distribution becomes impractical, so we will instead explore commonly used numerical summaries of the posterior, such as point estimates and posterior probability intervals.

A point estimate of θ summarizes the posterior with a single point. The three most commonly used Bayesian point estimates are:

- The posterior mean $\hat{\theta}_{\text{mean}} \equiv \mathbb{E}(\theta|x_1, \dots, x_n)$.
- The posterior median $\hat{\theta}_{\text{med}}$, the 50th quantile of $p(\theta|x_1, \dots, x_n)$.
- The posterior mode $\hat{\theta}_{\text{mode}} \equiv \arg \max_{\theta \in \Theta} p(\theta|x_1, \dots, x_n)$.

These point estimates have different properties, and we will see in Chapter 6 that the choice of point estimate can be formalized as a decision problem.

A point estimate says nothing about the variability in the posterior. One way to quantify the uncertainty is the posterior standard deviation $S(\theta|x_1, \dots, x_n) = \sqrt{\text{V}(\theta|x_1, \dots, x_n)}$.

EXAMPLE: INTERNET AUCTION DATA. As we saw earlier, the posterior for the parameter θ in a Poisson distribution with a $\theta \sim \text{Gamma}(\alpha, \beta)$ prior is $\theta|x_1, \dots, x_n \sim \text{Gamma}(\alpha + \sum_{i=1}^n x_i, \beta + n)$. From properties of the Gamma distribution in Box 2.8, the posterior mean estimate is therefore $(\alpha + \sum_{i=1}^n x_i)/(\beta + n)$ and the posterior variance is $(\alpha + \sum_{i=1}^n x_i)/(\beta + n)^2$. For the eBay data presented in

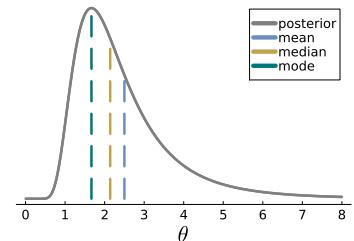


Figure 2.13: Three common point estimates for summarizing a posterior.

Section 2.4 we have $\mathbb{E}(\theta|x_1, \dots, x_n) = \frac{2+3635}{0.5+1000} \approx 3.635$ bidders and

$$S(\theta|x_1, \dots, x_n) = \sqrt{\frac{2+3635}{(0.5+1000)^2}} \approx 0.060.$$

A useful way to summarize a posterior distribution is by an *interval* with a specific coverage probability. Recall first the definition of a frequentist confidence interval: a 95% **confidence interval** for a parameter θ is a random interval $[l(X_1, \dots, X_n), u(X_1, \dots, X_n)]$ that contains the true θ in 95% of *all possible datasets* X_1, \dots, X_n from the data generating process. As with frequentist methods in general, we are guaranteed a long-run performance over all possible datasets. However, the presented confidence interval $[l(x_1, \dots, x_n), u(x_1, \dots, x_n)]$ for a given dataset either covers the true θ or it does not, and this is all we can say about the realized interval.

confidence interval

A Bayesian interval is defined in a much more direct way, and is conditional on the actually observed dataset. This simpler definition is possible since the posterior is a probability distribution; we have broken the Bayesian eggs and can enjoy the Bayesian omelette. A 95% posterior **credibility interval** for $\theta \in \Theta \subset \mathbb{R}$ is an interval $[l, u] \subset \Theta$ such that $\Pr(\theta \in [l, u] | x_1, \dots, x_n) = 0.95$, i.e. an interval that contains 95% of the posterior probability mass. We can generalize this to a more general region than an interval, for example a union of disjoint intervals, to multi-dimensional parameters, and of course to other probability coverages than 95%.

credibility interval

There are many ways to construct a credibility interval with a certain coverage probability. An **equal tail credibility interval** is an interval that cuts off equal probability in the left and right tail; for example, a 95% equal tail interval sets l and u to the 2.5% and 97.5% posterior quantile, respectively. Another popular interval construction is the highest posterior density (HPD) region which, as the name suggests, is made up of the θ values with the highest posterior density. We use the word *region* instead of interval here since HPD regions need not be intervals. A HPD interval is the shortest interval for a given coverage probability. Here is the definition.

equal tail credibility interval

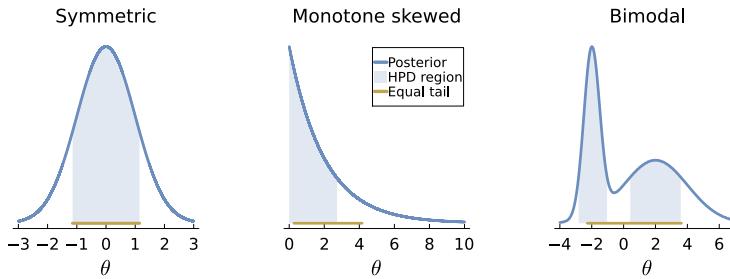
Definition (HPD region). A **Highest Posterior Density (HPD) region** for $\theta \in \Theta$ with coverage probability γ is a region $R \subset \Theta$ such that:

Highest Posterior Density (HPD) region

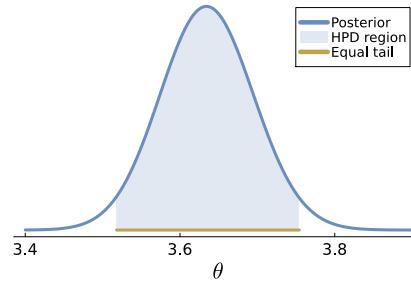
- $\Pr(\theta \in R | x_1, \dots, x_n) = \gamma$ and
- $p(\theta_{\text{in}} | x_1, \dots, x_n) \geq p(\theta_{\text{out}} | x_1, \dots, x_n)$ for all $\theta_{\text{in}} \in R$ and $\theta_{\text{out}} \notin R$.

Figure 2.14 illustrates the difference between equal tail intervals (yellow horizontal lines) and HPD regions (shaded area) for some example densities. Note how the equal tail interval construction can exclude θ values that actually have the highest posterior density

(middle graph) and how HPD regions can be disconnected (right hand graph). This [observable widget](#) lets you further explore credible intervals for some different posterior distributions.



A disadvantage of HPD regions is that they are not invariant to reparametrization: if $[a, b]$ is an HPD region for θ , and $\phi = f(\theta)$ is a non-linear transformation, then $[f(a), f(b)]$ is typically *not* an HPD region for the transformed parameter ϕ ; that is, you cannot just transform the interval endpoints and expect that to be an HPD region for the transformed parameter.



EXAMPLE: INTERNET AUCTION DATA. The 95% equal tail interval for the mean number of bidders in the iid Poisson model is $[3.518, 3.754]$ which is virtually indistinguishable from the HPD interval $[3.517, 3.754]$ since the posterior is essentially symmetric, see Figure 2.15.

2.6 Coverage probabilities of Bayesian credible intervals

A 90% frequentist confidence interval is constructed so that it is guaranteed to cover the true parameter value in 90% of all possible datasets; see Figure 2.16 for an illustration, and this [observable widget](#). This ideal coverage cannot be attained in all situations, and the actual coverage probability can vary depending on the true population parameter. Bayesian credible intervals are conditioned on

Figure 2.14: Illustration of HPD regions (shaded areas) and equal tail intervals (yellow horizontal line).

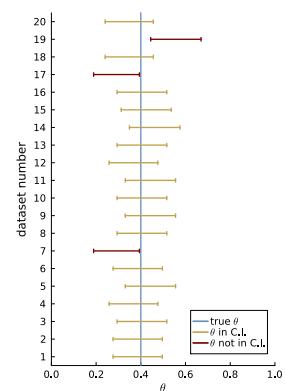


Figure 2.16: Illustration of how a 90% frequentist confidence interval either covers (yellow interval) or does not cover (red interval) the true population proportion θ (light blue line) across 20 different dataset from the population.

the observed dataset and do not come with the same repeated sampling guarantees. We will here first explore the frequentist coverage of Bayesian intervals and later in the section explain how Bayesian intervals have an alternative coverage guarantee.

The most widely taught confidence interval for a population proportion θ in the iid Bernoulli model is the *Wald* interval. The Wald interval with coverage probability q (i.e. a $100q\%$ interval) is

$$\hat{\theta} \pm z_q \sqrt{\frac{\hat{\theta}(1 - \hat{\theta})}{n}}, \quad (2.17)$$

where $\hat{\theta} = s/n$ is the maximum likelihood estimate, $s = \sum_{i=1}^n x_i$ is the number of successes in n trials and z_q is the value such that $\Pr(-z_q \leq Z \leq z_q) = q$ where $Z \sim N(0, 1)$; for example $z_q = 1.96$ for a 95% interval. The Wald interval is based on a normal approximation to the binomial distribution, an approximation which is known to be accurate if both $n\theta$ and $n(1 - \theta)$ are not too small; see this [observable widget](#).

An interval with better coverage properties is the *Wilson* confidence interval

$$\tilde{p} \pm z_q \frac{\sqrt{\frac{\hat{p}(1 - \hat{p})}{n} + \frac{z_q^2}{4n^2}}}{1 + z_q^2/n} \quad (2.18)$$

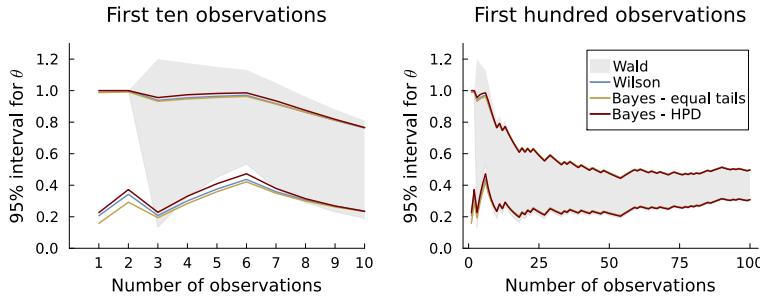
where the midpoint of the interval is

$$\tilde{p} = \hat{p} + \frac{z_q^2}{2n} \frac{1 - 2\hat{p}}{1 + z_q^2/n}.$$

Since the posterior for θ in the Bernoulli model with a $\theta \sim \text{Beta}(\alpha, \beta)$ prior is $\theta \sim \text{Beta}(\alpha + s, \beta + n - s)$, an equal tail 95% credible interval is easily obtained by the 0.025 and 0.975 quantiles of this posterior distribution. Similarly, a HPD interval can be obtained as explained in the previous section.

EXAMPLE: SPAM DATA. Figure 2.17 examines Wald, Wilson and Bayesian credible intervals for the spam probability θ using the uniform $\theta \sim \text{Beta}(1, 1)$ prior for increasing sample sizes. The Wald interval behaves badly when n is small with intervals collapsing at the single point $\theta = 1$ or even extending outside the $[0, 1]$ interval. The other three intervals are nearly identical already for $n = 10$.

The top row of Figure 2.18 shows that the Wald interval can have much lower actual coverage than the nominal target coverage of $q = 0.95$, particularly when the true θ is close to zero or one, a well known fact in the statistical literature (Andersson, 2023). The second row of Figure 2.18 shows that the coverage of the Wilson interval is



much closer to the nominal $q = 0.95$. The last three rows of Figure 2.18 show the frequentist coverage of equal tail Bayesian credible intervals from the posterior $\theta|x_1, \dots, x_n \sim \text{Beta}(\alpha + s, \beta + n - s)$ for three different priors: i) a uniform $\alpha = \beta = 1$, ii) $\alpha = \beta = 1/2$ and iii) a prior with zero prior sample size, i.e. $\alpha = \beta = \epsilon$ with $\epsilon \rightarrow 0$. Note how the coverage of the Bayesian interval from the uniform prior is almost identical to the Wilson interval; in fact the two intervals are nearly identical for any dataset (Jin et al., 2017). The interval from the $\text{Beta}(\epsilon, \epsilon)$ prior has more or less the same coverage profile as the Wald interval. The coverage of Bayesian intervals based on the uniform prior and the $\text{Beta}(1/2, 1/2)$ prior are both interestingly close to the target coverage of 95%, considering that Bayesian intervals are not constructed to have a prespecified frequentist coverage. This [observable widget](#) lets you explore the coverage probabilities of Bayesian intervals for different α and β in the prior $\theta \sim \text{Beta}(\alpha, \beta)$.

A $100q\%$ Bayesian interval constructed using an *informative* prior is not expected to have a repeated sampling coverage of q for all θ . An informative prior will move the posterior toward this prior, and if the true θ is far from where the prior assigns most of its mass, then Bayesian intervals will tend to miss the true θ in more than $1 - q$ fraction of repeated samples and we get undercoverage. If instead the true θ lies in a region where the prior has a lot of probability mass, then we get overcoverage of the credible interval. The reason why the $\text{Beta}(1, 1)$ and $\text{Beta}(1/2, 1/2)$ prior worked for most θ is that they are rather non-informative with little effect on the posterior and the credible intervals.

Since we already know that the effect of the prior will vanish as we collect more data, it is not surprising that a Bayesian interval can be shown to obtain the correct frequentist coverage in large samples, for any θ . Specifically, the actual coverage of a $100q\%$ Bayesian interval can be shown to be $q + c \cdot n^{-1/2}$, for some constant $c > 0$, so a Bayesian credible interval will approach the correct frequentist coverage in large samples, and this holds for essentially any prior. The asymptotic convergence is illustrated in Figure 2.19 where the cover-

Figure 2.17: 95 % confidence and credible intervals for the spam probability θ in the spam dataset. The intervals are computed by incrementally adding more data points up to a sample size of $n = 10$ (left graph) and $n = 100$ (right graph). The Wald interval collapses to the point $\theta = 1$ for the first two observations.

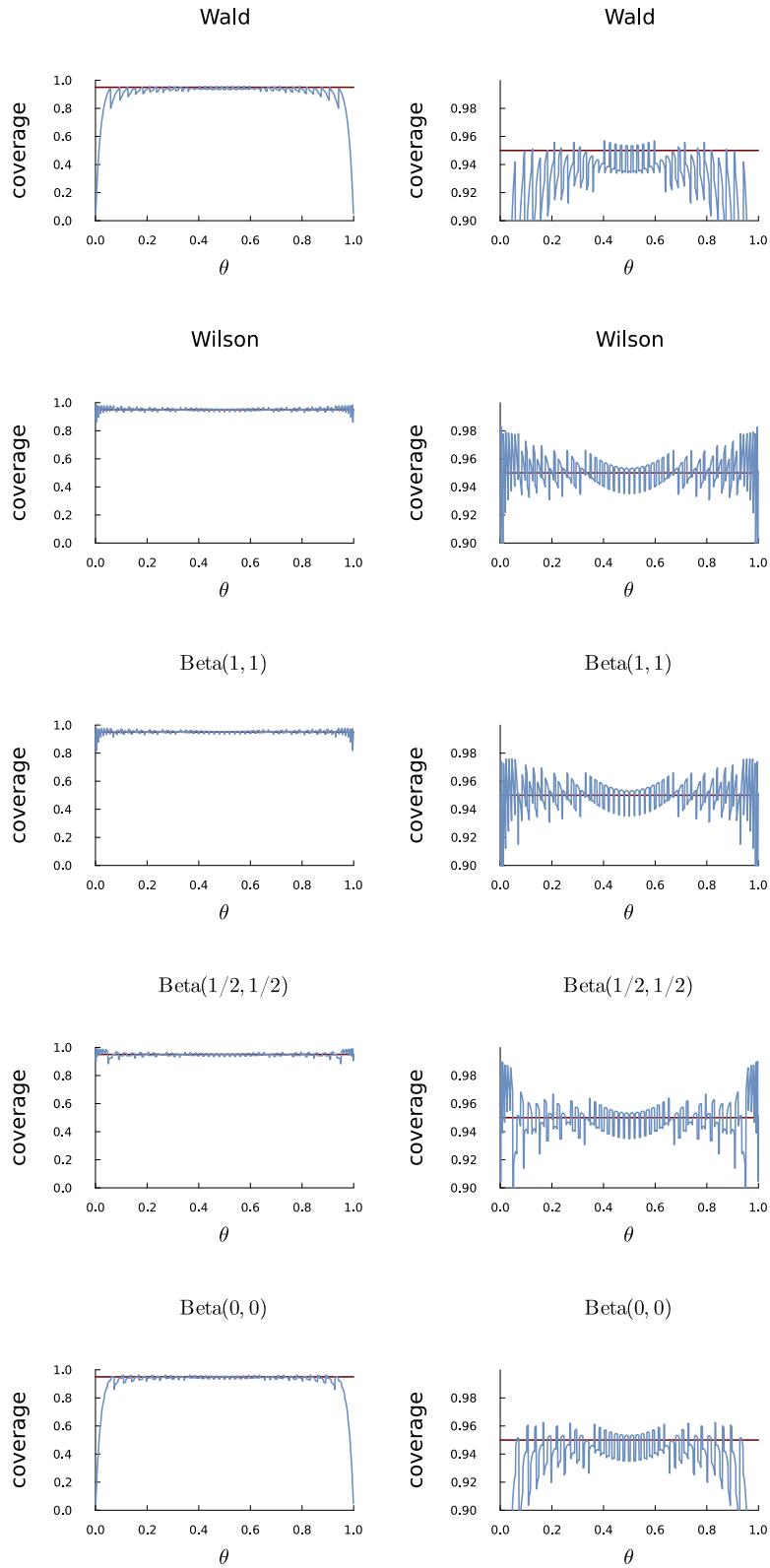
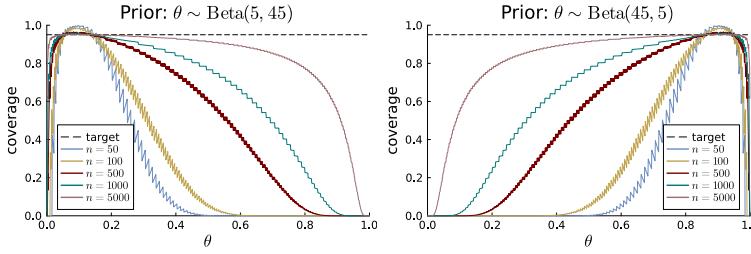


Figure 2.18: Coverage probabilities for frequentist (Wald and Wilson) and three Bayesian equal tail intervals for a Bernoulli probability θ from repeated samples of size $n = 50$. The three Bayesian intervals correspond to the three different priors: i) the uniform Beta(1,1), ii) Jeffreys' prior Beta(1/2, 1/2) presented in Section 4.8 and iii) the zero imaginary prior sample size prior Beta(ϵ, ϵ) with $\epsilon \rightarrow 0$. The plots in the right column zooms in on a coverage between [0.9, 1] for visibility. The target coverage of 95% is marked out with a red line.

age profile gets closer and closer to the horizontal target coverage of 0.95 as the sample size increases.



Even though Bayesian credible intervals should not be expected to have the correct frequentist coverage for all values of θ , they do have the following **Bayesian coverage property**:

The *average* coverage probability of a $100q\%$ Bayesian credible interval is exactly q when the parameter θ is sampled from the prior that was used to construct the interval.

This is illustrated in Table 2.1 where we see that the Bayesian equal tail interval constructed using the prior $\theta \sim \text{Beta}(5, 45)$ gives the correct coverage for $n = 50$ when averaging over parameter values from $\text{Beta}(5, 45)$. The intervals constructed using a $\text{Beta}(5, 5)$ prior will undercover badly when parameters are drawn from $\text{Beta}(5, 45)$. The non-informative uniform prior $\text{Beta}(1, 1)$ gives nearly correct average coverage even when parameters comes from a different prior since it has close to correct coverage for all θ as was seen in Figure 2.18.

	Wald	Wilson	$\text{Beta}(1,1)$	$\text{Beta}(5,45)$	$\text{Beta}(5,5)$
Coverage	0.900	0.954	0.953	0.951	0.672
Expected length	0.155	0.163	0.162	0.114	0.182

The last row of Table 2.1 shows the expected interval length, with the expectation computed with respect to the joint distribution of θ and the data, $s = 0, 1, \dots, n$. When using the same prior $\text{Beta}(5, 45)$ as the one from which the parameters are drawn from, the expected interval length form the Bayesian interval is much shorter, while obtaining the same coverage as for example the Wilson interval. The expected interval length when using the $\text{Beta}(5, 5)$ is larger than frequentist intervals. So when using a prior that is at least in partial agreement with the data generating process the Bayesian credible intervals tend to be short and have good frequentist coverage.

2.7 Bayesian learning and the likelihood principle

The Bernoulli example will be used to demonstrate an important property of Bayesian learning. Consider the following three experi-

Figure 2.19: Coverage probabilities from repeated samples of different sizes n of Bayesian equal tail and HPD intervals when using an informative $\text{Beta}(5, 45)$ prior (left) or $\text{Beta}(45, 5)$ prior (right).

Bayesian coverage property

Table 2.1: Average coverage probabilities for different 95% intervals when the parameter θ is drawn from $\text{Beta}(5, 45)$. The three last columns correspond to three different priors used when constructing the intervals. Note the correct coverage of the Bayesian interval based on the same $\text{Beta}(5, 45)$ prior used to generate the parameters in the averaging.

ments, all resulting in s successes in n trials:

- **Experiment 1:** sample data from $X_1, \dots, X_n | \theta \sim \text{Bern}(\theta)$, where n is a predetermined number of trials.
Stored data: the outcome in each trial: x_1, \dots, x_n .
- **Experiment 2:** sample data from $X_1, \dots, X_n | \theta \sim \text{Bern}(\theta)$, where n is a predetermined number of trials.
Stored data: the number of trials n and the total number of successes: $s = \sum_{i=1}^n x_i$.
- **Experiment 3:** sample data from $X_i | \theta \sim \text{Bern}(\theta)$ until exactly s , a predetermined number of successes, have been obtained.
Stored data: the number of trials, n , until s successes have been obtained.

The above three experiments show that we need to be careful in defining exactly *which* data to use in the likelihood function. We know from before that the likelihood from Experiment 1 is

$$p(x_1, \dots, x_n | \theta) = \theta^s (1 - \theta)^{n-s}. \quad (2.19)$$

In the second experiment we only get to observe that there were s successes in n trials, but the exact sequence x_1, \dots, x_n is not recorded. So the data is here represented as the outcome of a random variable $S = \sum_{i=1}^n X_i \sim \text{Binom}(n, \theta)$. The likelihood for experiment 2 is therefore given by the binomial distribution

$$p(s) = \binom{n}{s} \theta^s (1 - \theta)^{n-s}. \quad (2.20)$$

This is different from the likelihood in Experiment 1 since the outcome $S = s$ can be obtained from several different observed data sequences x_1, \dots, x_n , each with exactly s successes. The exact number of such possible sequences is given by the binomial factor $\binom{n}{s}$.

Finally, the random variable in Experiment 3 is the number of performed trials, N , which follows the **negative binomial distribution**; we use a capital letter for the sample size, which is random in this experiment. The likelihood from Experiment 3 is therefore

$$p(n) = \binom{n-1}{s-1} \theta^s (1 - \theta)^{n-s}. \quad (2.21)$$

The factor $\binom{n-1}{s-1}$ counts the number of ways we can order the $s-1$ successes in the first $n-1$ trials; we know that the n th trial must have been a success since the experiment terminated after n trials. Note that there are several versions of the negative binomial distribution depending on whether we count the number of trials or the number of failures until s successes.

negative binomial distribution

Now, the likelihood functions in (2.19)-(2.21) differ only by a constant that does not depend on θ , i.e. the likelihoods are proportional. The likelihood for the j th experiment can therefore be written as $c_j f(\theta)$, where $f(\theta) = \theta^s(1-\theta)^{n-s}$, $c_1 = 1$, $c_2 = \binom{n}{s}$ and $c_3 = \binom{n-1}{s-1}$. The posterior distribution of θ from the j th experiment is then by

(1.7)

$$p_j(\theta|x_1, \dots, x_n) = \frac{c_j f(\theta)p(\theta)}{\int c_j f(\theta)p(\theta)d\theta} = \frac{f(\theta)p(\theta)}{\int f(\theta)p(\theta)d\theta}.$$

The posterior distribution for θ is therefore the same in all three experiments. It is now obvious that Bayesian inference always satisfies the following likelihood principle.

Definition. *Likelihood principle.* Two experiments that result in (proportionally) equal likelihood functions should give the same inferences.

Likelihood principle

Informally, the likelihood principle says that all relevant information in an experiment about θ is contained in the likelihood function. The importance of the likelihood principle is that it can be mathematically derived from two simpler principles that everyone holds as self evident. Hence the word *should* in the principle; see Casella and Berger (2002, ch. 6.2) for a discussion of this famous **Birnbaum's theorem**.

Birnbaum's theorem

Many frequentist methods violate the likelihood principle. The maximum likelihood *estimate* is easily seen to be $\hat{\theta}_{MLE} = s/n$ for all three experiments for a given data set. However, the sampling variability of the maximum likelihood *estimator*, $V(\hat{\theta}_{MLE})$, will be different in Experiment 3 from that in Experiment 1 and 2. This is a consequence of the estimator being S/n in Experiment 1 and 2, but s/N in Experiment 3; note the difference in random variables (capital letters) in these estimators.

In summary, Bayesian inference *conditions on the observed data* and does not rely on repeated sampling properties. The data only enters through the likelihood function and Bayesian inference respects the likelihood principle.

2.8 Exponential Family and Sufficiency*

This section presents the concept of sufficient statistics and the exponential family of distributions, with particular emphasis on their role in Bayesian learning. While these concepts are very important in statistics, this starred section can be skipped at first reading, but should be read before the generalized linear models in Chapter 8, where the exponential family plays a prominent role.

Sufficient statistics

In all models covered so far in this book, the dataset, (x_1, \dots, x_n) , has only entered the likelihood through some low-dimensional summary statistic; for example the number of successes $s = \sum_{i=1}^n x_i$ in the Bernoulli model, the sample mean $\bar{x} = n^{-1} \sum_{i=1}^n x_i$ in the Gaussian model, and the sum of counts, $\sum_{i=1}^n x_i$, in the Poisson model. Note that we did not choose this data reduction, it just turned out that the likelihood only depended on the summarizing statistic; the statistic captured all the relevant information in the sample. In all of the above examples, the statistic was one-dimensional. In other models more than a single dimension is needed to compress the dataset, and we let the vector-valued function $\mathbf{t}(x_1, \dots, x_n) \rightarrow \mathbb{R}^k$ denote the statistic in general, where k is the dimension of reduction.

The following definition captures the idea that a statistic may contain *all* relevant information in the data about a parameter θ .

Definition. *Sufficient statistic.* A statistic $\mathbf{t}(X_1, \dots, X_n)$ is sufficient for θ if the conditional distribution of the sample X_1, \dots, X_n given the value of the statistic $\mathbf{t}(X_1, \dots, X_n)$ does not depend on θ .

Sufficient statistic

The sufficiency of a statistic can be checked by the following lemma; see Casella and Berger (2002) for a proof.

Lemma 1. Factorization criterion. A statistic $t(x_1, \dots, x_n)$ is sufficient for a parameter θ if and only if the likelihood can be factorized as

Factorization criterion

$$p(x_1, \dots, x_n | \theta) = h(x_1, \dots, x_n) f(\mathbf{t}(x_1, \dots, x_n); \theta), \quad (2.22)$$

where $h(x_1, \dots, x_n)$ does not depend on θ and $f(\mathbf{t}; \theta)$ is a function of the data only through the sufficient statistic $\mathbf{t}(x_1, \dots, x_n)$.

The idea behind sufficient statistics is so appealing that it is often formulated as a desired inference principle similar to the likelihood principle presented in Section 2.7.

Definition. *Sufficiency principle.* If $\mathbf{t}(X_1, \dots, X_n)$ is a sufficient statistic for θ then any inference about θ should depend on the sample x_1, \dots, x_n only through the value $\mathbf{t}(x_1, \dots, x_n)$.

Sufficiency principle

Theorem 1. Bayesian learning satisfies the sufficiency principle.

Proof. If $\mathbf{t}(x_1, \dots, x_n)$ is a sufficient statistic for θ then by Lemma 1

$$\begin{aligned} p(\theta | x_1, \dots, x_n) &= \frac{p(x_1, \dots, x_n | \theta) p(\theta)}{\int p(x_1, \dots, x_n | \theta) p(\theta) d\theta} \\ &= \frac{h(x_1, \dots, x_n) f(\mathbf{t}(x_1, \dots, x_n); \theta) p(\theta)}{\int h(x_1, \dots, x_n) f(\mathbf{t}(x_1, \dots, x_n); \theta) p(\theta) d\theta} \\ &= \frac{f(\mathbf{t}(x_1, \dots, x_n); \theta) p(\theta)}{\int f(\mathbf{t}(x_1, \dots, x_n); \theta) p(\theta) d\theta}, \end{aligned}$$

which only depends on the data through the sufficient statistic $\mathbf{t}(x_1, \dots, x_n)$. □

Exponential family

All models considered so far are part of the large and important exponential family of distributions. A random variable X follows a distribution in the (one-parameter) **exponential family** if its density can be written in the form

$$p(x|\theta) = h(x) \exp\left(\eta(\theta)t(x) - A(\theta)\right), \text{ for } x \in \mathcal{X}, \quad (2.23)$$

where $h(x)$ is a function of only x and $A(\theta)$ is a function of only θ . The support \mathcal{X} is not allowed to depend on θ , so that for example the $\text{Uniform}(0, \theta)$ distribution does not belong to the exponential family. The function $\eta(\theta)$ is called the **natural parameter** and is an invertible transformation of the parameter θ . Here are some examples.

exponential family

natural parameter

EXAMPLE: POISSON DISTRIBUTION. The $\text{Pois}(\theta)$ distribution can be rewritten as follows

$$p(x|\theta) = \frac{\theta^x e^{-\theta}}{x!} = \frac{e^{x \ln \theta} e^{-\theta}}{x!} = \frac{1}{x!} \exp(x \ln \theta - \theta),$$

which is in the exponential family with $h(x) = (x!)^{-1}$, $A(\theta) = \theta$, $\eta(\theta) = \ln \theta$ and $t(x) = x$. Note in particular that the natural parameter is the logarithm of the Poisson mean, $\eta(\theta) = \ln \theta$.

EXAMPLE: BERNOULLI DISTRIBUTION. The $\text{Bern}(\theta)$ distribution can also be written as an exponential family:

$$p(x|\theta) = \theta^x (1-\theta)^{1-x} = \left(\frac{\theta}{1-\theta}\right)^x (1-\theta) = \exp\left(\eta(\theta)x - A(\theta)\right),$$

where $\eta(\theta) = \ln(\frac{\theta}{1-\theta})$, $A(\theta) = \ln(\frac{1}{1-\theta})$, $t(x) = x$ and $h(x) = 1$. The natural parameter for the Bernoulli distribution is therefore the log-odds, $\ln(\frac{\theta}{1-\theta})$.

The normal distribution and many other distributions can similarly be shown to belong to the exponential family; but not all do, the Student-t distribution presented in the next chapter (see Box 3.3) is an example of a commonly used distribution that does not belong to the exponential family. We will use $\text{ExpFam}(\theta)$ as a generic notation for a distribution in the exponential family, leaving the specific $h(x)$, $A(\theta)$, $\eta(\theta)$ and $t(x)$ functions implicit.

The likelihood function for iid data from an $\text{ExpFam}(\theta)$ distribution is

$$p(x_1, \dots, x_n|\theta) = \left[\prod_{i=1}^n h(x_i) \right] \exp\left(\eta(\theta) \sum_{i=1}^n t(x_i) - nA(\theta)\right). \quad (2.24)$$

Lemma 1 can be directly used to show that $\sum_{i=1}^n t(x_i)$ is a sufficient statistic for θ . In the next chapter well will see a multiparameter version of the exponential family with a vector of k sufficient statistics.

The Pitman–Koopman–Darmois theorem (Bernardo and Smith, 2009) proves that among distributions whose support does not depend on θ , only the distributions in the exponential family have sufficient statistics of fixed dimension, i.e the dimension k does not depend on the size of the data, n (or at least is bounded).

The exponential family has several other attractive properties (Sundberg, 2019). One property of particular interest here is that a conjugate prior always exists for models in the exponential family. In fact, the following family of priors is conjugate to the exponential family likelihood in (2.24)

$$p(\theta) = H(\tau_0, \nu_0) \exp \left(\eta(\theta)\tau_0 - \nu_0 A(\theta) \right), \quad (2.25)$$

where $H(\tau_0, \nu_0)$ is the normalizing constant. Note that this prior has two hyperparameter τ_0 and ν_0 that need to be set by the user. We will use the symbol $\theta \sim \text{ExpFamConj}(\tau_0, \nu_0)$ for this prior distribution, where it must be remembered that the form of the prior depends on which specific exponential family member the prior is conjugate to, i.e. it depends on $\eta(\theta)$ and $A(\theta)$.

EXAMPLE: BERNoulli MODEL. It was shown above that $\eta(\theta) = \ln(\frac{\theta}{1-\theta})$ and $A(\theta) = \ln(\frac{1}{1-\theta})$, for Bernoulli data. The prior in (2.25) is therefore

$$\begin{aligned} p(\theta) &\propto \exp \left(\eta(\theta)\tau_0 - \nu_0 A(\theta) \right) \\ &= \exp \left(\ln \left(\frac{\theta}{1-\theta} \right) \tau_0 - \nu_0 \ln \left(\frac{1}{1-\theta} \right) \right) \\ &\propto \theta^{\tau_0} (1-\theta)^{\nu_0 - \tau_0}, \end{aligned}$$

which is proportional to the Beta($\tau_0, \nu_0 - \tau_0$) distribution. The parametrization in (2.25) is therefore interpreted as the information from an imaginary prior sample of τ_0 success in ν_0 trials. The Beta(α, β) prior used in Section 2.1 have prior hyperparameters with a different meaning where the imaginary prior sample consists of α successes and β failures in $\alpha + \beta$ trials.

The posterior distribution for θ in the exponential family with a conjugate prior is obtained by multiplying the likelihood in (2.24) with prior (2.25)

$$p(\theta|x_1, \dots, x_n) \propto \exp \left[\eta(\theta) \left(\tau_0 + \sum_{i=1}^n t(x_i) \right) - (\nu_0 + n) A(\theta) \right],$$

which is of the form ExpFamConj, but with updated hyperparameters: $\tau_0 \Rightarrow \tau_0 + \sum_{i=1}^n t(x_i)$ and $\nu_0 \Rightarrow \nu_0 + n$. We summarize this in

Conjugate analysis - Exponential family data

Model: $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{ExpFam}(\theta)$

Prior: $\theta \sim \text{ExpFamConj}(\tau_0, \nu_0)$

Posterior: $\theta|x_1, \dots, x_n \sim \text{ExpFamConj}(\tau_0 + \sum_{i=1}^n t(x_i), \nu_0 + n)$

Box 2.10: Bayesian updating for iid exponential family data with a conjugate prior.

Box 2.10.

This result shows that we can think quite generally about ν_0 as the (imaginary) prior sample size and τ_0 as the prior data compressed by the sufficient statistic. For example, in the Poisson model the information in the conjugate prior equals a prior sample of ν_0 data points with a mean count of τ_0/ν_0 .

EXERCISES

Exercise 2.1

Let $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} \text{Expon}(\theta)$ be iid exponentially distributed data. Show that the Gamma distribution is the conjugate prior for this model.

Exercise 2.2

Let $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} \text{Geom}(\theta)$ be iid from a geometric distribution with parameter $0 < \theta < 1$. Show that the Beta distribution is the conjugate prior for this model.

Exercise 2.3

The dataset `lung` in the R package `survival` contains data on 228 patients with advanced lung cancer. We will here analyze the survival time of the patient in days (`time`). The variable `status` is a binary variable with `status = 1` if the survival time of the patient is censored (patient was still alive at the end of the study) and `status = 2` if the survival time was uncensored (patient was dead before the end of the study).

In this exercise we will only analyze the uncensored patients; Exercise 2.8 below asks you to analyze all patients. Assume that the survival times X_1, \dots, X_n of the uncensored patients are iid $\text{Exp}(\theta)$ distributed. Use the conjugate prior $\theta \sim \text{Gamma}(\alpha = 3, \beta = 300)$, which can be shown to imply that the expected survival time

Exponential distribution

$X \sim \text{Expon}(\theta)$ for $X \in (0, \infty)$

$$p(x) = \theta e^{-\theta x}$$

$$\mathbb{E}(X) = 1/\theta$$

$$\mathbb{V}(X) = 1/\theta^2$$

Box 2.11: The exponential distribution.

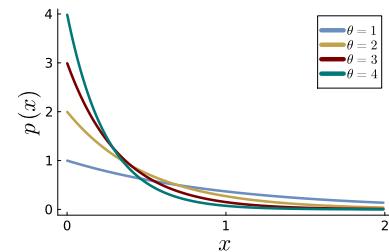


Figure 2.20: Some Exponential distributions.

$\mathbb{E}(X|\theta) = 1/\theta$ for this population is around 200 days. Plot the prior and posterior densities for θ over a suitable grid of θ -values.

Exercise 2.4

Let X_1, \dots, X_n be an iid sample from a distribution with density function

$$p(x) \propto \theta^2 x \exp(-x\theta) \quad \text{for } x > 0 \text{ and } \theta > 0.$$

Find the conjugate prior for this distribution and derive the posterior distribution from an iid sample x_1, \dots, x_n .

Exercise 2.5

(a) Let x_1, \dots, x_{10} be a sample with mean $\bar{x} = 1.873$. Assume the model $X_1, \dots, X_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, 1)$ and the prior $\theta \sim N(0, 5)$. Compute the posterior distribution of θ .

(b) You now get hold of a second sample $y_1, \dots, y_{10} | \theta \stackrel{\text{iid}}{\sim} N(\theta, 2)$, where θ is the same quantity as in (a) but the measurements have a larger variance. The sample mean in this second sample is $\bar{y} = 0.582$. Compute the posterior distribution of θ using both samples (the x 's and the y 's) under the assumption that the two samples are independent.

Hint: batch learning.

(c) You finally obtain a third sample $z_1, \dots, z_{10} | \theta \stackrel{\text{iid}}{\sim} N(\theta, 3)$, with mean $\bar{z} = 1.221$. Unfortunately, the measuring device for this latter sample was defective and any measurement above 3 was recorded as exactly 3. There were two such measurements. Give an expression for the unnormalized posterior distribution (likelihood \times prior) for θ based on all three samples (x, y and z). If you have a computer available you may plot this unnormalized posterior over a grid of θ values.

Hint: the posterior distribution is not normal anymore when the measurements are truncated at 3.

Exercise 2.6

Derive the posterior distribution for the normal model with a normal prior in Box 2.5. *Hint: complete the square.*

Exercise 2.7

(a) Let $X_1, \dots, X_n | \theta \sim \text{Uniform}(\theta - 1/2, \theta + 1/2)$. The estimator $\hat{\theta} = \bar{X}$ of θ is unbiased. Calculate for the sampling variance of $\hat{\theta}$.

Geometric distribution

$X \sim \text{Geom}(\theta)$, for $X \in \{0, 1, 2, \dots\}$

$$p(x) = (1 - \theta)^x \theta$$

$$\mathbb{E}(X) = \frac{1 - \theta}{\theta}$$

$$\mathbb{V}(X) = \frac{1 - \theta}{\theta^2}$$

Box 2.12: Properties of the geometric distribution.

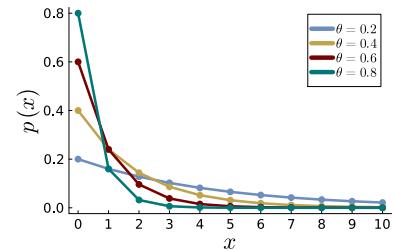


Figure 2.21: Some Geometric distributions.

- (b) Derive the posterior distribution for θ assuming a uniform prior distribution.

Hint: once you have observed some data, some values for θ are no longer possible.

- (c) Assume that you have observed three data observations: $x_1 = 1.1, x_2 = 2.09, x_3 = 1.4$. What would a frequentist conclude about θ ? What would a Bayesian conclude? Discuss.

Exercise 2.8

Exercise 2.3 modelled the survival times of uncensored lung cancer patients with an iid exponential model. In this exercise we will extend that analysis to include also the censored patients, using the same prior as in Exercise 2.3. Plot the prior and posterior densities for θ over a suitable grid of θ -values.

Hint: The posterior is no longer tractable due to contributions of the censored patients to the likelihood. For the censored patients we only know that they lived *at least* the number of days recorded in the dataset. The likelihood contribution $p(x_c|\theta)$ for the c th censored patient with recorded time x_c is therefore $p(X \geq x_c|\theta) = e^{-\theta x_c}$, which follows from the distribution function of the exponential distribution $p(X \leq x|\theta) = 1 - e^{-\theta x}$.

Exercise 2.9

Show that the $N(\mu, 1)$ distribution belongs to the exponential family.

3 Multi-parameter models

3.1 Joint posterior distributions

Most models have more than one parameter, and many models are incredibly rich in parameters. Datasets are increasing rapidly in size which makes it possible to estimate increasingly more complex models. To explore how Bayesian methods can be used in multiparameter models we first return in this chapter to the iid $N(\theta, \sigma^2)$, but now in the more realistic setting where both θ and σ^2 are unknown parameters. In later chapters we will tackle regression and classification models where each covariate (input) x_k affects the response (output) y through a regression coefficient β_k ; hence in a regression with K covariates we have K regression coefficients β_1, \dots, β_K .

Consider a general probability model $p(x_1, \dots, x_n | \theta_1, \dots, \theta_K)$ with K parameters for a dataset x_1, \dots, x_n ; for example the iid normal model where $\theta_1 = \mu$ and $\theta_2 = \sigma^2$. Bayesian learning proceeds exactly as with a single parameter, except that the prior and posterior distributions are now both multidimensional joint distributions. Figure 3.1 gives an illustration of a bivariate ($K = 2$) normal distribution.

Using Bayes' theorem in proportional form, the **joint posterior distribution** $p(\theta_1, \dots, \theta_K | x_1, \dots, x_n)$ is given by

$$p(\theta_1, \dots, \theta_K | x_1, \dots, x_n) \propto p(x_1, \dots, x_n | \theta_1, \dots, \theta_K) p(\theta_1, \dots, \theta_K),$$

where $p(\theta_1, \dots, \theta_K)$ is a multidimensional prior distribution and $p(x_1, \dots, x_n | \theta_1, \dots, \theta_K)$ is the likelihood function; Note that the likelihood function is now a **likelihood surface** in the sense that it is a function of several parameters, $\theta_1, \dots, \theta_K$.

To keep the notation simpler we often use vector notation and write $\boldsymbol{\theta} \equiv (\theta_1, \dots, \theta_n)$ and $\mathbf{x} \equiv (x_1, \dots, x_n)$. The multivariate Bayes' theorem can then be expressed as

$$p(\boldsymbol{\theta} | \mathbf{x}) \propto p(\mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta}). \quad (3.1)$$

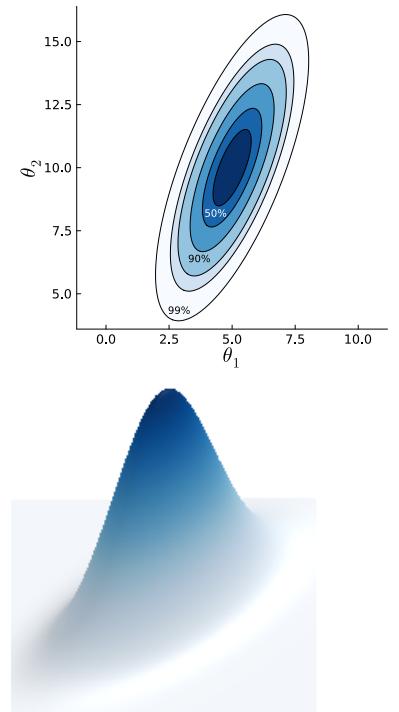


Figure 3.1: Contour plot (top) and surface plot (bottom) of the bivariate normal distribution. The contour levels contain 25, 50, 75, 90, 95 and 99% of the probability mass, respectively.

joint posterior distribution

likelihood surface

3.2 Marginalization

The joint posterior distribution $p(\theta|x)$ contains all posterior information about θ , but is obviously hard to visualize in the same way as we did for single-parameter models. In many cases we are also most interested in a subset of parameters, and the other parameters are only needed to model the data well but are of no real interest. Such parameters are just a nuisance when presenting inferences and are therefore often called **nuisance parameters**. Getting rid of nuisance parameters is very difficult in a non-Bayesian setting, for example when using maximum likelihood estimation. So what is the Bayesian solution to this dilemma?

Nuisance parameters can be handled in a very natural way in a Bayesian approach since the posterior distribution is a probability distribution for θ . We can therefore just integrate out, or marginalize out, the nuisance parameters just as in ordinary probability calculus; this is often called **marginalization**. Take a simple example where $\theta = (\theta_1, \theta_2)$ and assume that the parameter of interest is θ_1 whereas θ_2 is considered a nuisance parameter; θ_1 could for example be the mean of iid Gaussian model and θ_2 the variance. The marginal posterior of θ_1 is then

$$p(\theta_1) = \int p(\theta_1, \theta_2) d\theta_2,$$

where the integration is over the full support of θ_2 . Figure 3.2 illustrates the marginalization concept. Using the decomposition $p(\theta_1, \theta_2) = p(\theta_1|\theta_2)p(\theta_2)$ we can alternatively express this as

$$p(\theta_1) = \int p(\theta_1|\theta_2)p(\theta_2) d\theta_2,$$

which shows that marginalization is achieved by averaging over the values of θ_2 with weights given by $p(\theta_2)$.

More generally, with more than two parameters, partition the elements of θ into two vectors, θ_a and θ_b . The marginal posterior of the subvector θ_a is then obtained by marginalizing out the remaining parameters in θ_b from the joint posterior

$$p(\theta_a) = \int \cdots \int p(\theta_a, \theta_b) d\theta_b. \quad (3.2)$$

We will see examples of marginalization in the following sections.

3.3 Gaussian data with unknown variance

The previous chapter analyzed iid normal data $x_1, \dots, x_n | \theta, \sigma^2 \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$ under the usually unrealistic assumption that σ^2 is known. Let us now tackle the case where both parameters are unknown. It

nuisance parameters

marginalization

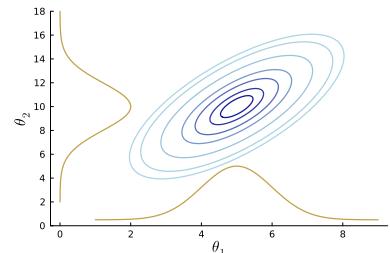


Figure 3.2: Contour plot of the bivariate normal distribution in Figure 3.1 along with the marginal distributions plotted against each axis.

turns out that the conjugate prior for this model has dependence between θ and σ , so we will describe the prior using the decomposition $p(\theta|\sigma^2)p(\sigma^2)$ as follows

$$\theta|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0) \quad (3.3)$$

$$\sigma^2 \sim \text{Inv}-\chi^2(\nu_0, \sigma_0^2). \quad (3.4)$$

The marginal conjugate prior for σ^2 involves a new distribution, the **scaled inverse chi-squared distribution**, denoted by $\text{Inv}-\chi^2(\nu, \tau^2)$ in general; see Box 3.1 and Figure 3.3. This distribution is a specific parametrization of the **inverse Gamma distribution**. The name comes from the characterization

$$X \sim \chi_\nu^2 \Rightarrow Y = \nu\tau^2 \frac{1}{X} \sim \text{Inv}-\chi^2(\nu, \tau^2),$$

so that a $\text{Inv}-\chi^2(\nu, \tau^2)$ variable is an inverted χ_ν^2 variable scaled by $\nu\tau^2$. Note from Box 3.1 that the parameter τ^2 is close to the mean when ν is large. The mode is $\nu\tau^2/(\nu + 2)$, so τ^2 is somewhere between the mode and the mean. We will therefore call τ^2 the location of $\text{Inv}-\chi^2(\nu, \tau^2)$, or sometimes just sloppily as "our best guess".

The conjugate prior in (3.3) is specified via the four prior hyperparameters:

- μ_0 - the prior mean for θ
- κ_0 - the number of prior data observations for θ
- σ_0^2 - the prior location of σ^2
- ν_0 - the prior degrees of freedom for σ^2 .

Note that, similar to the conjugate prior for the exponential family, we are only *interpreting* κ_0 as the number of prior observations. The prior may not actually be based on previous data, but the information in the prior $\theta|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0)$ has the equivalent strength of an *imaginary* prior sample of κ_0 observations from a normal distribution with variance σ^2 . The hyperparameter ν_0 plays the same role for σ^2 .

Box 3.2 shows that the posterior is indeed in the same form as the prior in (3.3), as required for a conjugate prior. There is a lot of greek letters in Box 3.2, but note that the same sort of intuition applies here as in the case with a known variance in Chapter 2:

- the posterior mean μ_n is a weighted average of the data mean \bar{x} and the prior mean μ_0
- the weight on the data $w = n/(\kappa_0 + n)$ is close to one when either the data is informative (large n) or the prior is weak (small κ_0)

scaled inverse chi-squared distribution

inverse Gamma distribution

Inv- χ^2 distribution

$$X \sim \text{Inv}-\chi^2(\nu, \tau^2), X \in (0, \infty)$$

$$p(x) = \frac{(\tau^2\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \frac{\exp\left(-\frac{\nu\tau^2}{2x}\right)}{x^{1+\nu/2}}$$

$$\mathbb{E}(X) = \frac{\nu}{\nu-2}\tau^2$$

$$\mathbb{V}(X) = \frac{2\nu^2\tau^4}{(\nu-2)^2(\nu-4)}$$

Box 3.1: The Inv- χ^2 distribution.

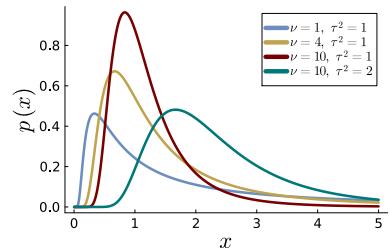


Figure 3.3: Some Inv- χ^2 distributions.

Gaussian iid data with conjugate prior

Model: $x_1, \dots, x_n | \theta, \sigma^2 \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$

Prior: $\theta | \sigma^2 \sim N(\mu_0, \sigma^2 / \kappa_0)$
 $\sigma^2 \sim \text{Inv-}\chi^2(v_0, \sigma_0^2)$

Posterior: $\theta | \sigma^2, \mathbf{x} \sim N(\mu_n, \sigma^2 / \kappa_n)$
 $\sigma^2 | \mathbf{x} \sim \text{Inv-}\chi^2(v_n, \sigma_n^2)$

$$\mu_n = w\bar{x} + (1-w)\mu_0$$

$$w = \frac{n}{\kappa_0 + n}$$

$$\kappa_n = \kappa_0 + n$$

$$v_n = v_0 + n$$

$$\nu_n \sigma_n^2 = v_0 \sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_0 + n} (\bar{x} - \mu_0)^2$$

$$\text{where } \bar{x} = n^{-1} \sum_{i=1}^n x_i, (n-1)s^2 = \sum_{i=1}^n (x_i - \bar{x})^2$$

Marginal: $\theta | \mathbf{x} \sim t(\mu_n, \sigma_n^2 / \kappa_n, v_n)$

Box 3.2: Bayesian updating for the iid Gaussian model with unknown mean and variance using the conjugate prior.

- the data variance σ^2 does not appear in w , as it did when the variance was known. The reason for this difference is that the prior variance for θ is scaled by σ^2 in the conjugate prior, and σ^2 therefore cancels out in w .
- the posterior sample size $\kappa_n = \kappa_0 + n$ is the sum of the number of prior observations κ_0 and the sample size n .

Interest centers mainly on the average download speed, so we would like to obtain the *marginal* posterior distribution of θ . This distribution can be derived by marginalizing out the nuisance parameter σ^2 from the joint posterior

$$p(\theta | x_1, \dots, x_n) = \int p(\theta | \sigma^2, x_1, \dots, x_n) p(\sigma^2 | x_1, \dots, x_n) d\sigma^2,$$

where $p(\theta | \sigma^2, x_1, \dots, x_n)$ and $p(\sigma^2 | x_1, \dots, x_n)$ are given in Box 3.2. In Exercise 3.2 you are asked to show that the marginal posterior of θ is a **student-t distribution**; see Box 3.3 and 3.4 for a definition and properties. Specifically, the marginal posterior of θ is

$$\theta | x_1, \dots, x_n \sim t(\mu_n, \sigma_n^2 / \kappa_n, v_n), \quad (3.5)$$

where $\mu_n, \sigma_n^2, \kappa_n$ and v_n are all defined as in Box 3.2. Note that also the marginal *prior* for θ follows a student-t distribution of the form (3.5), but with hyperparameters naturally subscripted by 0 instead of n .

Student-t distribution

$X \sim t(\mu, \sigma, \nu)$ for $X \in (-\infty, \infty)$

$$p(x) = \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2) \sqrt{\pi \nu \sigma^2}} \times \left(1 + \frac{1}{\nu} \left(\frac{x-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$$

$$\mathbb{E}(X) = \mu \text{ if } \nu > 1$$

$$\mathbb{V}(X) = \sigma^2 \frac{\nu}{\nu-2} \text{ if } \nu > 2$$

Box 3.3: The student-t distribution.

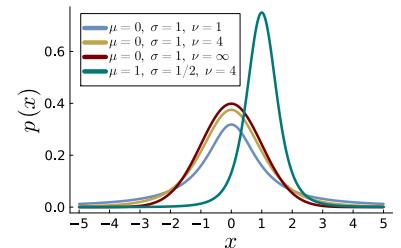


Figure 3.4: Some Student-t distributions.

student-t distribution

EXAMPLE: INTERNET SPEED DATA. Let us return to the example with the $n = 5$ download speeds with a mean of $\bar{x} = 15.998$ Mbit/s from Chapter 2. This time we assume that also σ^2 , the variability of the measurements from the speed testing service, is unknown. I will use the prior hyperparameters $\mu_0 = 20$, $\kappa_0 = 1$, $v_0 = 5$ and $\sigma_0^2 = 5^2$, which agrees in location with my previous prior when σ^2 was assumed known at $\sigma^2 = 5^2$; setting $v_0 = 5$ gives a prior equal to the yellow distribution in the right graph of Figure 3.6, which I find sensible.

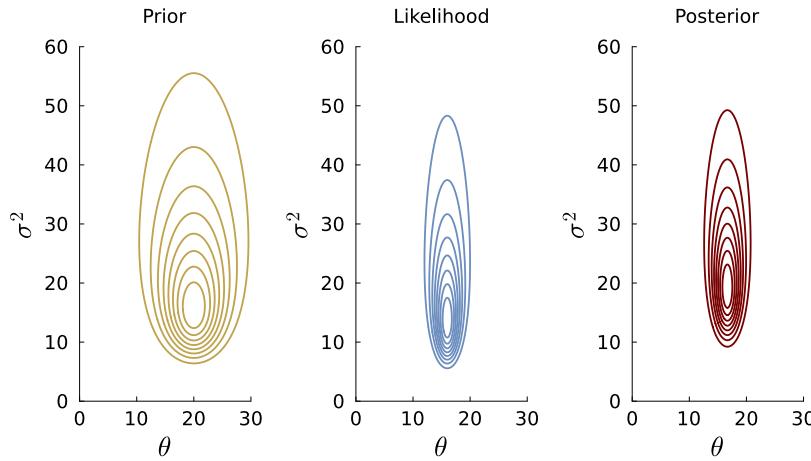


Figure 3.5: Bayesian updating for the internet speed data in the iid Normal model. Contours of joint distributions of θ and σ^2 .

Figure 3.5 displays contours of the joint prior, likelihood and posterior for θ and σ^2 ; the posterior is more concentrated than the prior, especially for θ . The marginal priors and posterior for the two parameters are shown in Figure 3.6. The data have made both marginal posteriors more concentrated, but less so for σ^2 since we do not learn so much about a variance from only $n = 5$ observations. The probability of at least 20 Mbit download speed has decreased from the prior probability of 0.5 to 0.066 in the posterior.

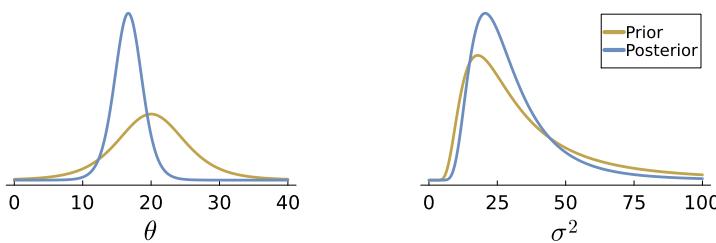


Figure 3.6: Marginal posteriors for the internet speed data in the iid Normal model.

3.4 A first look at Monte Carlo simulation

The iid Gaussian model with conjugate prior is an example of a model where we can obtain both the joint and the marginal posteriors in analytical form. This is rarely the case in more complex models or when non-conjugate priors are used. The idea with Monte Carlo methods is to simulate **posterior draws** of θ from $p(\theta|x_1, \dots, x_n)$ and approximate the posterior by for example a histogram. We will have much more to say about this in Chapters 9 and 10 where powerful simulation algorithms are presented, but we will already here introduce the most basic Monte Carlo simulation method.

posterior draws

Posterior simulation - iid Gaussian with conjugate prior.

```
Input: data  $x = (x_1, \dots, x_n)$ 
        number of posterior draws  $m$ .
compute  $\mu_n, \sigma_n^2, \kappa_n$  and  $\nu_n$  using Box 3.2.
for  $i$  in  $1:m$  do
     $\sigma^2 \leftarrow \text{rINVCHI2}(\nu_n, \sigma_n^2)$ 
     $\theta \leftarrow \text{rNORMAL}(\mu_n, \sigma^2 / \kappa_n)$ 
end
Output:  $m$  draws for  $\theta$  and  $\sigma^2$  from joint posterior.
```

```
Function  $\text{rINVCHI2}(\nu, \tau^2)$ 
     $x \leftarrow \text{rCHI2}(\nu)$ 
     $y \leftarrow \nu \tau^2 / x$ 
    return  $y$ 
```

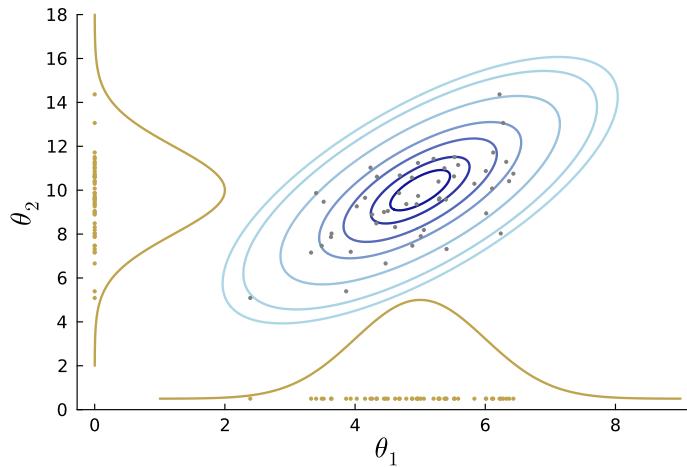
Box 3.4: Algorithm for posterior simulation for the iid Normal model with conjugate prior. The `rNORMAL` and `rCHI2` random number generators are assumed to be part of the standard library. The variable σ^2 is highlighted in orange to indicate that the most recent draw of σ^2 is used in the call to the `rNORMAL` function.

The algorithm in Box 3.4 gives pseudo-code for simulating from the $p(\theta, \sigma^2 | \mathbf{x})$ in the iid normal model by iteratively simulating from $p(\sigma^2 | \mathbf{x})$ followed by simulation from $p(\theta | \sigma^2, \mathbf{x})$. Note how this involves using the most recently simulated value of σ^2 when simulating θ . The algorithm includes the subfunction `rINVCHI2(ν_n, σ_n^2)` to draw from the Inv- χ^2 distribution. The algorithm implicitly assumes that the standard library of your programming language includes random number generators `rCHI2(ν)` and `rNORMAL($\mu_n, \sigma^2 / \kappa_n$)` for the χ^2 and normal distributions, respectively.

draw	θ	σ^2	σ/θ	$\theta \geq 20$
1	18.165	18.451	0.236	0
2	20.431	29.943	0.267	1
3	15.565	29.094	0.346	0
:	:	:	:	:
10,000	16.400	21.668	0.283	0
Mean	16.645	30.813	0.330	0.066

EXAMPLE: INTERNET SPEED DATA

Let us now simulate from the posterior of θ and σ^2 in the Internet speed data. The second and third columns in Table 3.1 show the output from generating $m = 10,000$ joint posterior draws with the algorithm in Box 3.4.



One attractive feature of simulating from the joint posterior distribution is that all marginal posterior distributions are directly obtained by just selecting the column for the parameter in question; tedious integration is replaced by plotting a histogram of the selected column. This is illustrated in Figure 3.7.

Figure 3.8 shows the marginals for the internet speed data example obtained from simulation; the figure also plots the analytical marginal posteriors, which happen to be known in this simple example.

The histograms of the simulated draws in Figure 3.8 are clearly approximating the posteriors extremely well. Monte Carlo simulation is theoretically known to be **simulation consistent** in the sense that we are guaranteed to get arbitrary close to the true posterior if we simulate a large number of draws. For example, the sample mean of the draws will converge to the true posterior expectation $E(\theta|x)$ in large simulations. Formally, if we let $\theta^{(i)}$ denote the i th posterior draw of

Table 3.1: Posterior simulation output for the Internet speed dataset with computed functions of the parameters.

Figure 3.7: Illustrating marginalization by selection. The figure plots the contours of a joint distribution with the marginal distributions overlaid as orange curves. The gray points are 100 draws from joint distribution and the orange points are projections of the gray points on the two axes. The orange points corre-

Convergence in probability

A sequence of random variables X_1, \dots, X_n **converges in probability to a constant c** , if and only if for any $\epsilon > 0$

$$\Pr(|X_n - c| > \epsilon) \rightarrow 0 \text{ as } n \rightarrow \infty.$$

We then write $X_n \xrightarrow{P} c$.

X_1, \dots, X_n **converges in probability to a random variable X** if and only if for any $\epsilon > 0$

$$\Pr(|X_n - X| > \epsilon) \rightarrow 0 \text{ as } n \rightarrow \infty.$$

We write $X_n \xrightarrow{P} X$.

Box 3.5: Convergence in probability.

Law of large numbers

Let X_1, X_2, \dots be iid random variables with finite mean μ . Then

$$\bar{X}_n \xrightarrow{P} \mu \text{ as } n \rightarrow \infty,$$

where \xrightarrow{P} denotes convergence in probability.

There is also a strong law of large numbers based on an alternative notion of probabilistic

any of the parameters in a model, this result can be expressed as

$$\bar{\theta}_{1:m} \equiv \frac{1}{m} \sum_{i=1}^m \theta^{(i)} \xrightarrow{p} \mathbb{E}(\theta|\mathbf{x}) \text{ as } m \rightarrow \infty,$$

where \xrightarrow{p} denotes **convergence in probability**, see Box 3.5. This result is a version of the **law of large numbers**, see Box 3.6 and [this observable notebook](#). The left side of Figure 3.9 illustrates this convergence by plotting the posterior mean estimates $\bar{\theta}_{1:m}$ for increasing m ; note that the figure shows these cumulative estimates only up to $m = 1000$.

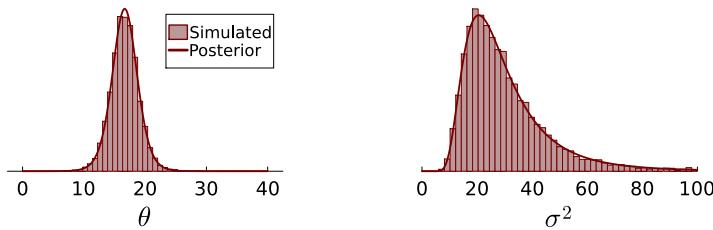


Figure 3.8: Histogram of simulated marginal posteriors for the internet speed data with analytical marginal posterior densities overlayed.

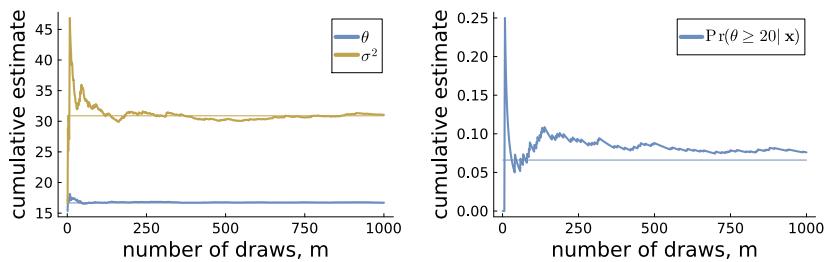


Figure 3.9: Convergence of the Monte Carlo estimate of the posterior expectation of θ and σ^2 (left) and $\Pr(\theta \geq 20|\mathbf{x})$ (right). The analytical posterior results are displayed as thin horizontal lines.

The **central limit theorem** (see Box 3.8 and [this observable notebook](#)) can be used to prove that $\bar{\theta}_{1:m}$ **converges in distribution** (Box 3.7) to a normal distribution. Hence, the following approximation of the posterior estimate $\bar{\theta}_{1:m}$ is accurate when m is large:

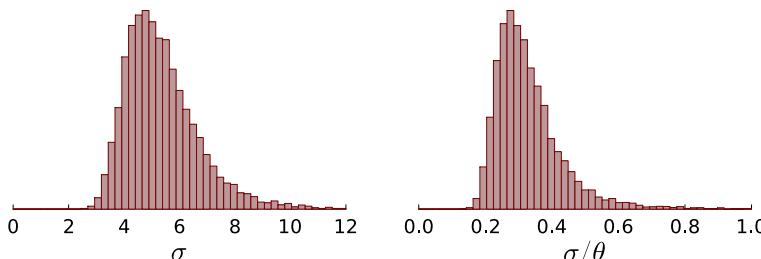
$$\bar{\theta}_{1:m}|\mathbf{x} \sim N\left(\mathbb{E}(\theta|\mathbf{x}), \frac{\mathbb{V}(\theta|\mathbf{x})}{m}\right), \quad (3.6)$$

where $\mathbb{V}(\theta|\mathbf{x})$ is the posterior variance of θ ; note that we get the usual reduction in variance that comes from taking averages of m draws, i.e. the variance of $\bar{\theta}_{1:m}$ decreases with m . The result in (3.6) can be used to determine the required number of draws m needed for a given estimation precision. A multivariate version of the central limit theorem can be used to prove a similar result to (3.6) when θ is a

vector; an interesting aspect is that $\text{Cov}(\bar{\theta}_{1:m})$ (a covariance matrix in the multiparameter case) still decreases at the rate $1/m$, regardless of the dimension of θ .

It is often the case that the quantities of interest are functions $f(\theta)$ of the parameters; for example the **coefficient of variation** σ/θ in the iid normal model. Even when the posterior for the model parameters θ is available analytically, deriving the posterior for $f(\theta)$ involves tedious multidimensional change-of-variables calculations. Here is a second attractive property of simulation: the posterior for $f(\theta)$ can be directly obtained from a posterior sample of θ by simply computing the function $f(\theta)$ for each posterior draw. Provided the posterior variance of $f(\theta)$ exists, a central limit theorem of the form (3.6) exists also in this case, with the expected value and variance replaced by those of $f(\theta)$.

To illustrate how simulation immediately provides inference for any function of the parameters, Table 3.1 contains a fourth column named σ/θ with the computed coefficient of variation for each draw. We can now just plot a histogram of this new column to approximate the marginal posterior of the function $f(\theta, \sigma^2) = \sigma/\theta$. The results are presented in the right part of Figure 3.10; the left part of the figure shows the results for the standard deviation $f(\theta, \sigma^2) = \sqrt{\sigma^2}$.



The final column of Table 3.1 is a binary variable that records if θ was at least 20, i.e. it computes the indicator function $f(\theta, \sigma^2) = I(\theta \geq 20)$. The marginal posterior probability $\Pr(\theta \geq 20 | \mathbf{x})$ is then easily approximated by the mean of the final column; the right side of Figure 3.9 illustrates the Monte Carlo convergence of this estimate.

3.5 Multinomial data

Categorical data have observations that belong to one of C discrete classes. A computer bug can for example be allocated to C developing teams; an item sold in an auction may reported as: 'defective', 'normal quality', or 'new'; a continuous variable like age can be recorded in age intervals: 0–18, 19–28, 29–49, 50–64 and 65+, which

[-4cm]

Convergence in distribution

A sequence of random variables X_1, \dots, X_n converges in distribution to the random variable X , if and only if

$$F_n(x) \rightarrow F(x) \text{ as } n \rightarrow \infty,$$

for all x where $F(\cdot)$ is continuous, where $F_n(x)$ and $F(x)$ are the cumulative distribution functions (cdf) of X_n and X , respectively.

We then write $X_n \xrightarrow{d} X$.

Box 3.7: Convergence in distribution.

coefficient of variation

Central limit theorem (CLT)

Let X_1, X_2, \dots be iid random variables with finite mean μ and variance σ^2 . Then

Figure 3.10: Histograms of simulated marginal posteriors for σ/\sqrt{n} (left) and the coefficient of variation σ/θ (right) for the internet speed data. The CLT denotes convergence in distribution.

$$\bar{X}_n \xrightarrow{d} N(\mu, \sigma^2/n) \text{ as } n \rightarrow \infty.$$

Box 3.8: The central limit theorem.

Categorical data

would then also be a categorical variable. The categories in the latter two situations are examples of **ordinal data** where the categories have a natural order. There are special models for ordinal data which we will not cover in this chapter; here we will consider categorical data without natural order. Categorical variables are often called **multi-class** in the machine learning literature.

A multi-class random variable X is often written in **one-hot encoding** as $\mathbf{x} = (x_1, \dots, x_C)$ where $X = c$ is encoded as $x_c = 1$ and $x_j = 0$ for $j \neq c$; hence when $C = 3$, $\mathbf{x} = (0, 1, 0)$ means that the observation belongs to the second class. The categorical random variable $X|\theta \sim \text{Cat}(\theta_1, \dots, \theta_C)$ has probability distribution

$$p(x) = \theta_1^{x_1} \cdots \theta_C^{x_C}, \quad (3.7)$$

where (x_1, \dots, x_C) is the one-hot encoding of x , $0 < \theta_c < 1$ is the probability of class c and $\sum_{c=1}^C \theta_c = 1$. Note how Bernoulli data is the special case with $C = 2$ categories ‘success’ and ‘failure’, so that the $\text{Cat}(\theta_1, \dots, \theta_C)$ distribution generalizes the Bernoulli distribution to the case $C > 2$. Figure 3.11 is an example of $\text{Cat}(\theta_1, \dots, \theta_C)$ for $C = 4$.

We saw in Section 1.3 that counting the number of successes s in n binary Bernoulli trials gave rise to $S \sim \text{Binomial}(n, \theta)$ data. In the same way we can count the number of observations in category c for $c = 1, \dots, C$ in multi-class data. This gives data as a count vector $\mathbf{y} = (y_1, \dots, y_C)$ where y_c is the number of observations in category c in $n = \sum_{c=1}^C y_c$ ‘trials’. Here is an example:

MOBILE PHONE SURVEY DATA. A survey was conducted among $n = 513$ mobile phone users. Among other questions, the participants were asked: ‘What kind of mobile phone do you mainly use?’ with the four options:

1. iPhone
2. Android
3. Windows
4. Other/Don’t know

The number of responses in the four categories were: $\mathbf{y} = (180, 230, 62, 41)$.

The **multinomial distribution** generalizes the binomial distribution to $C > 2$ categories; its main properties are summarized in Box 3.9. The Binomial distribution with x successes in n trials with probability θ in Box 1.2 is the special case with $C = 2$ categories, which is seen by defining $\theta_1 = \theta$, $\theta_2 = 1 - \theta$, $y_1 = x$, $y_2 = n - x$, and noting that

$$\frac{n!}{y_1!y_2!} = \frac{n!}{x!(n-x)!} = \binom{n}{x}. \quad (3.8)$$

ordinal data

multi-class

one-hot encoding

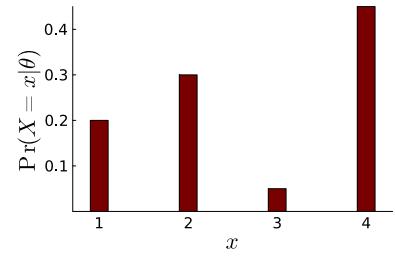


Figure 3.11: Categorical distribution with probabilities $\theta = (0.20, 0.30, 0.05, 0.45)$.

multinomial distribution

Multinomial distribution

$(Y_1, \dots, Y_C) \sim \text{MultiNomial}(n, \theta)$ where $\sum_{c=1}^C Y_c = n$, $\theta = (\theta_1, \dots, \theta_C)$ and $\sum_c \theta_c = 1$.

$$p(\mathbf{y}) = \frac{n!}{y_1! \cdots y_C!} \theta_1^{y_1} \cdots \theta_C^{y_C}$$

$$\mathbb{E}(Y_c) = n\theta_c$$

$$\mathbb{V}(Y_c) = n\theta_c(1 - \theta_c)$$

Box 3.9: The multinomial distribution.

The multinomial distribution is a multivariate distribution with convenient marginalization properties. For example, if we group the counts in one or more categories - for example turning the mobile phone dataset into three categories by merging 'Windows' and 'Other' - the distribution remains multinomial. The probability of a merged category is simply the sum of the probabilities of the merged categories. Hence

$$(y_1, y_2, y_3 + y_4) \sim \text{Multinomial}(\theta_1, \theta_2, \theta_3 + \theta_4).$$

In particular, merging to only two categories - for example 'iPhone' and 'not iPhone' - gives a binomial distribution where the probability of success (iPhone) is θ_1 and the probability of failure (not iPhone) is $\theta_2 + \theta_3 + \theta_4$.

A Bayesian analysis of multinomial data requires a prior distribution for the model parameters, $\theta = (\theta_1, \dots, \theta_C)$. Since each θ_c is a probability, the first distribution that comes to mind may be a Beta distribution; the Beta distribution is not appropriate here however since it does not enforce the constraint that the probabilities sum to one. Hence, the parameter space of the multinomial distribution is the **unit simplex**, i.e. the set $\theta = (\theta_1, \dots, \theta_C) : 0 < \theta_c < 1$ and $\sum_c \theta_c = 1$. Luckily, there is a very nice distribution on the unit simplex, the Dirichlet distribution, summarized in Box 3.10.

The **Dirichlet distribution** is specified with the prior hyperparameters $\alpha_c > 0$, see Figure 3.12 for some examples. The *relative* sizes of the elements in α determine the prior means for elements of θ . For example, setting $\alpha_1 = \dots = \alpha_C = 1.5$, as in the upper left graph of Figure 3.12, gives equal prior mean for all categories: $\mathbb{E}(\theta_c) = 1/C$ for all c . The *absolute* size of α , measured by $\alpha_+ = \sum_{c=1}^C \alpha_c$, is inversely related to the variance, see Box 3.10; hence, the prior hyperparameters $\alpha = (1.5, \dots, 1.5)$ and $\alpha = (5, \dots, 5)$ in the upper part of Figure 3.12 have the same mean, but the latter has smaller variance. Finally, the bottom part of Figure 3.12 shows examples where the prior mean is different over the categories.

The $\text{Dirichlet}(1, \dots, 1)$ has a constant density and is therefore the **uniform distribution on the unit simplex**; this generalizes the result that $\text{Beta}(1, 1)$ is uniform on the unit interval $[0, 1]$. Finally, when $\alpha_c < 1$, the Dirichlet density becomes 'bathtub' shaped with probability mass piling up against the edges of the unit simplex.

The Dirichlet distribution is conjugate to the multinomial likeli-

Dirichlet distribution

$\theta | \alpha \sim \text{Dirichlet}(\alpha)$ where
 $\theta = (\theta_1, \dots, \theta_C)$, $\sum_c \theta_c = 1$,
 $\alpha = (\alpha_1, \dots, \alpha_C)$ and $\alpha_c > 0$.

$$\begin{aligned} p(\theta) &= k \cdot \theta_1^{\alpha_1-1} \cdots \theta_C^{\alpha_C-1} \\ k &= \frac{\Gamma(\sum_{c=1}^C \alpha_c)}{\prod_{c=1}^C \Gamma(\alpha_c - 1)}. \\ \mathbb{E}(\theta_c) &= \frac{\alpha_c}{\sum_{j=1}^C \alpha_j} \\ \mathbb{V}(\theta_c) &= \frac{\mathbb{E}(\theta_c)(1 - \mathbb{E}(\theta_c))}{1 + \alpha_+} \\ \alpha_+ &= \sum_{c=1}^C \alpha_c. \end{aligned}$$

Marginal distributions:

$$\theta_c \sim \text{Beta}(\alpha_c, \alpha_+ - \alpha_c).$$

Box 3.10: The Dirichlet distribution.

unit simplex

Dirichlet distribution

uniform distribution on the unit simplex

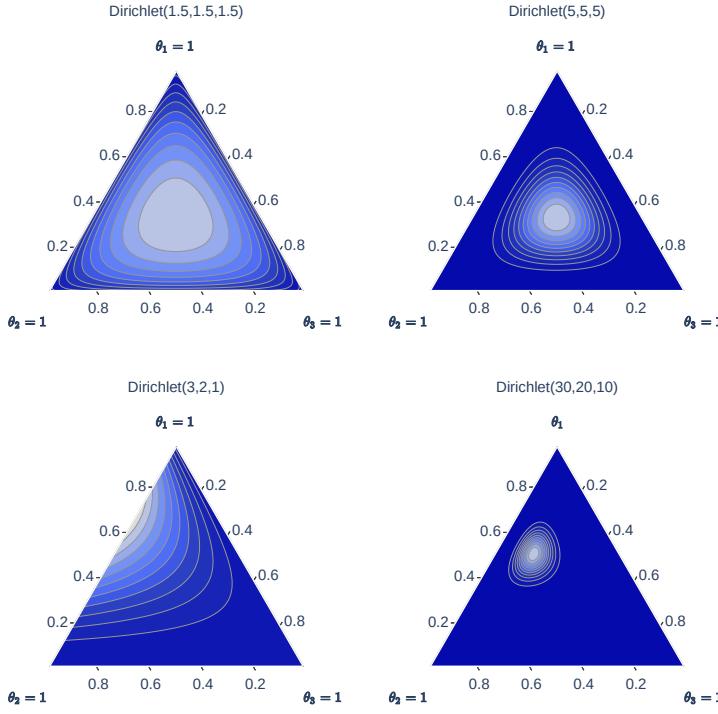


Figure 3.12: Probability density functions for some Dirichlet distributions for $\theta = (\theta_1, \theta_2, \theta_3)$. Lighter color means higher density. The corners of the simplex represent $\theta_1 = 1$, $\theta_2 = 1$ and $\theta_3 = 1$ respectively, as denoted in the figures.

hood which is easily seen by computing the posterior

$$p(\theta|y) \propto p(y|\theta)p(\theta) \quad (3.9)$$

$$= \frac{n!}{y_1! \cdots y_C!} \theta_1^{y_1} \cdots \theta_C^{y_C} \cdot \frac{\Gamma(\sum_{c=1}^C \alpha_c)}{\prod_{c=1}^C \Gamma(\alpha_c - 1)} \theta_1^{\alpha_1-1} \cdots \theta_C^{\alpha_C-1} \quad (3.10)$$

$$\propto \theta_1^{\alpha_1+y_1-1} \cdots \theta_C^{\alpha_C+y_C-1}, \quad (3.11)$$

which is proportional to the $\text{Dirichlet}(\alpha_1 + y_1, \dots, \alpha_C + y_C)$ density. This is a convenient result: the posterior is simply obtained by adding the data count y_c to the prior hyperparameter α_c in each category. This parallels and generalizes the binary case where a $\text{Beta}(\alpha, \beta)$ prior was updated to a posterior by adding the number of successes s to α and the number of failures f to β . Box 3.11 summarizes the prior-to-posterior updating for multinomial data with a Dirichlet prior.

MOBILE PHONE SURVEY DATA We are now ready to analyze the four market shares $\theta_1, \dots, \theta_4$ in the mobile phone data. We will determine the prior hyperparameters in the Dirichlet prior using data from a similar survey from four years ago. The proportions in the four categories back then were: 30%, 30%, 20% and 20%. This was a large survey, but since time has passed and user patterns most likely have

Multinomial data with Dirichlet prior

Model: $\mathbf{y}|\boldsymbol{\theta} \sim \text{Multinomial}(\boldsymbol{\theta})$, where
 $\mathbf{y} = (y_1, \dots, y_C)$ are counts in C categories
 $\boldsymbol{\theta} = (\theta_1, \dots, \theta_C)$ are category probabilities.
Prior: $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, for $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_C)$
Posterior: $\boldsymbol{\theta}|\mathbf{y} \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{y})$

Box 3.11: Bayesian updating for multinomial data with the Dirichlet prior.

Posterior simulation - Multinomial data, Dirichlet prior.

Input: data $\mathbf{y} = (y_1, \dots, y_C)$
prior hyperparameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_C)$
the number of posterior draws m .

for i in $1:m$ **do**
| $\boldsymbol{\theta} \leftarrow \text{RDIRICHLET}(\boldsymbol{\alpha} + \mathbf{y})$
end

Output: m posterior draws of $\boldsymbol{\theta} = (\theta_1, \dots, \theta_C)$.

Function $\text{RDIRICHLET}(\boldsymbol{\alpha})$
for c in $1:C$ **do**
| $\mathbf{z}[c] \leftarrow \text{RGAMMA}(\boldsymbol{\alpha}[c], 1)$
end
return $\mathbf{z}/\text{SUM}(\mathbf{z})$

Box 3.12: Algorithm for posterior simulation for the multinomial model with the conjugate Dirichlet prior. The RGAMMA random number generator is assumed to be part of the standard library.

changed, I value the information in this older survey as being equivalent to a survey with only 50 participants. This gives us the prior:

$$(\theta_1, \dots, \theta_4) \sim \text{Dirichlet}(\alpha_1 = 15, \alpha_2 = 15, \alpha_3 = 10, \alpha_4 = 10)$$

Note that $\mathbb{E}(\theta_1) = 15/50 = 0.3$ and so on, so the prior mean is set equal to the proportions from the older survey. Also, $\sum_{c=1}^4 \alpha_c = 50$, so the prior information is equivalent to a survey based on 50 respondents, as required.

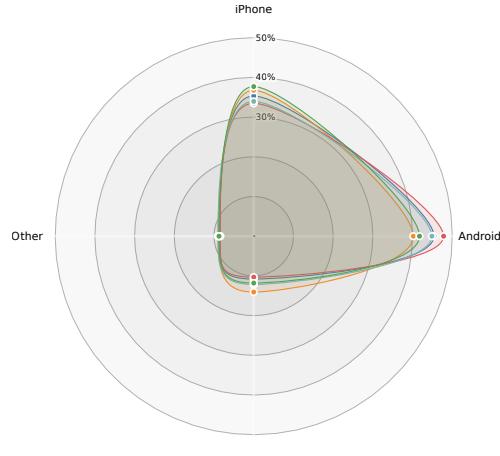


Figure 3.13: Radar chart illustrating five draws from the joint posterior of the market shares for the mobile phone survey data.

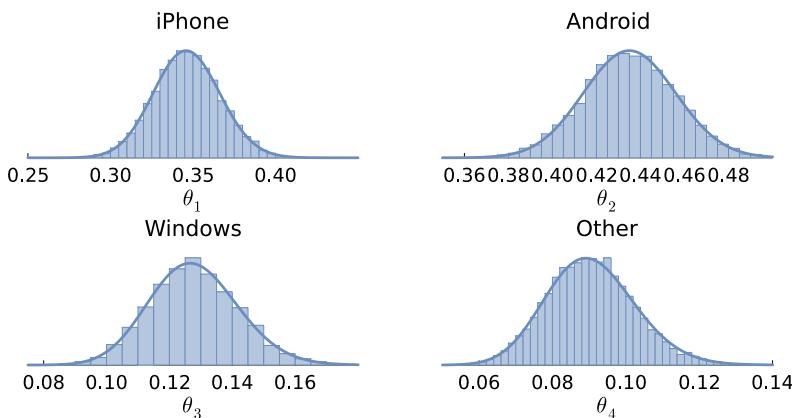


Figure 3.14: Marginal posteriors of the market shares for the mobile phone survey data. Simulated (histogram) draws and analytical density functions (solid curves).

The joint posterior distribution of all four shares is by Box 3.11 equal to

$$(\theta_1, \dots, \theta_4) | \mathbf{y} \sim \text{Dirichlet}(15 + 180, 15 + 230, 10 + 62, 10 + 41)$$

Figure 3.13 illustrates five draws from the posterior distribution in a so called radar or spider chart; it is clear that the posterior is quite

draw	θ_1	θ_2	θ_3	θ_4	θ_2 largest
1	0.338	0.446	0.130	0.086	1
2	0.332	0.457	0.124	0.086	1
3	0.325	0.442	0.136	0.094	1
⋮	⋮	⋮	⋮	⋮	⋮
10,000	0.343	0.443	0.132	0.081	1
Mean	0.346	0.435	0.127	0.090	0.991

informative with little variation from draw to draw. The marginal posteriors are plotted in Figure 3.14 as histograms from Monte Carlo simulation (see the algorithm in Box 3.12) and the analytical posteriors from Box 3.11 are overlayed.

Figure 3.14 indicates that Android may have the largest market share with a posterior mean around 0.44 versus iPhones posterior mean of 0.35. Computing the probability that Android has the largest market share involves integrating the joint posterior $\theta|y \sim \text{Dirichlet}(\alpha + y)$ over the region $\{\theta : \theta_2 > \max(\theta_1, \theta_3, \theta_4)\}$, a tedious calculation. The probability is however easily computed by simulation by recording for each posterior θ draw if the condition $\theta_2 > \max(\theta_1, \theta_3, \theta_4)$ is satisfied; see Table 3.2, which shows that

$$\Pr(\text{Andriod has largest market share}|y) \approx 0.991.$$

We are almost certain that Android is the most popular mobile phone in the population targeted by the survey.

3.6 Multivariate normal data with known covariance

This section considers the iid **multivariate normal distribution** model for a p -dimensional data vector x :

$$x_1, \dots, x_n | \theta, \Sigma \stackrel{\text{iid}}{\sim} N(\theta, \Sigma), \quad (3.12)$$

where θ is the p -dimensional mean vector and Σ is a $p \times p$ positive definite covariance matrix. We will here take Σ to be known and derive the posterior for θ .

Presenting a Bayesian analysis of this model here gives us a chance to meet the important multivariate normal distribution and its properties relatively early in the book; see Box 3.13 for the density and properties, and Figure 3.15 for contour plots of some example densities.

The likelihood for the multivariate model in (3.12) is the product of the individual densities for each vector observation x_i

$$p(x_1, \dots, x_n | \theta, \Sigma) \propto |\Sigma|^{-n/2} \exp\left(-\frac{1}{2} \sum_{i=1}^n (x_i - \theta)^\top \Sigma^{-1} (x_i - \theta)\right),$$

A vector version of the argument leading up to (2.9) in the univariate case can be used to show that the likelihood can be written as the

Table 3.2: Posterior simulation output for the multinomial model applied to the mobile phone survey data. The last column is a computed binary indicator for the event that Android has the largest market share, i.e. if $\theta_2 > \max(\theta_1, \theta_3, \theta_4)$.

Multivariate normal

$x|\mu, \Sigma \sim N(\mu, \Sigma)$ where $x \in \mathbb{R}^p$, $\mu \in \mathbb{R}^p$ and Σ is a $p \times p$ positive definite covariance matrix.

$$p(x) = |2\pi\Sigma|^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1} (x - \mu)\right)$$

$$\mathbb{E}(x) = \mu$$

$$\mathbb{V}(x) = \Sigma$$

Define the decomposition

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

and similarly for μ and Σ

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

Marginal distributions:

$$x_k \sim N(\mu_k, \sigma_k^2)$$

$$x_1 \sim N(\mu_1, \Sigma_{11})$$

Conditional distributions:

$$x_1|x_2 \sim N(\tilde{\mu}_1, \tilde{\Sigma}_1)$$

where

$$\tilde{\mu}_1 = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

$$\tilde{\Sigma}_1 = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Box 3.13: The multivariate normal distribution.

multivariate normal distribution

exponential of a quadratic (form):

$$p(\mathbf{x}_1, \dots, \mathbf{x}_n | \boldsymbol{\theta}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-n/2} \exp\left(-\frac{n}{2}(\boldsymbol{\theta} - \bar{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \bar{\mathbf{x}})\right), \quad (3.13)$$

where $\bar{\mathbf{x}} = n^{-1} \sum_{i=1}^n \mathbf{x}_i$ is the usual sample mean vector. When $\boldsymbol{\Sigma}$ is known, the term $|\boldsymbol{\Sigma}|^{-n/2}$ can be absorbed into the proportionality constant, but we have kept it here since we will use the same equation later when $\boldsymbol{\Sigma}$ is unknown.

Not too surprisingly, the multivariate normal prior

$$\boldsymbol{\theta} \sim N(\boldsymbol{\theta}_0, \boldsymbol{\Lambda}_0),$$

turns out to be conjugate for this model. The posterior can be derived by multiplying together the likelihood in (3.13) with the prior and completing the quadratic forms in the exponentials; see Box 3.14 for a general result on quadratic form completion. The posterior can then be shown to indeed be a multivariate normal:

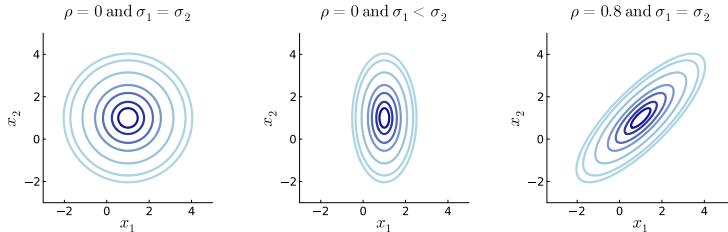
$$\boldsymbol{\theta} | \mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\Sigma} \sim N(\boldsymbol{\theta}_n, \boldsymbol{\Lambda}_n),$$

where

$$\begin{aligned}\boldsymbol{\theta}_n &= (\boldsymbol{\Lambda}_0^{-1} + n\boldsymbol{\Sigma}^{-1})^{-1}(\boldsymbol{\Lambda}_0^{-1}\boldsymbol{\theta}_0 + n\boldsymbol{\Sigma}^{-1}\bar{\mathbf{x}}) \\ \boldsymbol{\Lambda}_n^{-1} &= \boldsymbol{\Lambda}_0^{-1} + n\boldsymbol{\Sigma}^{-1}.\end{aligned}$$

Letting $\boldsymbol{\Lambda}_0^{-1} \rightarrow \mathbf{0}$ (in the matrix sense that all elements approaches zero) we obtain a noninformative uniform prior $p(\boldsymbol{\theta}) \propto c$ and the posterior becomes

$$\boldsymbol{\theta} | \mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\Sigma} \sim N(\bar{\mathbf{x}}, n^{-1}\boldsymbol{\Sigma}).$$



Completing quadratic forms

This formula shows how to combine two quadratic forms in a vector of interest \mathbf{x} , to a single quadratic form in \mathbf{x} plus two constant terms that do not depend on \mathbf{x} :

$$\begin{aligned}(\mathbf{x} - \mathbf{a})^\top \mathbf{A}(\mathbf{x} - \mathbf{a}) + (\mathbf{x} - \mathbf{b})^\top \mathbf{B}(\mathbf{x} - \mathbf{b}) \\ = (\mathbf{x} - \mathbf{d})^\top \mathbf{D}(\mathbf{x} - \mathbf{d}) \\ + (\mathbf{d} - \mathbf{a})^\top \mathbf{A}(\mathbf{d} - \mathbf{a}) \\ + (\mathbf{d} - \mathbf{b})^\top \mathbf{B}(\mathbf{d} - \mathbf{b}),\end{aligned}$$

where

$$\mathbf{D} = \mathbf{A} + \mathbf{B} \text{ and } \mathbf{d} = \mathbf{D}^{-1}(\mathbf{A}\mathbf{a} + \mathbf{B}\mathbf{b}).$$

Box 3.14: Completing quadratic forms.

Figure 3.15: Contour plots of some bivariate normal distributions with correlation ρ .

3.7 Multivariate normal data with unknown covariance*

Let us now treat the same multivariate normal model

$$\mathbf{x}_1, \dots, \mathbf{x}_n | \boldsymbol{\theta}, \boldsymbol{\Sigma} \stackrel{\text{iid}}{\sim} N(\boldsymbol{\theta}, \boldsymbol{\Sigma}), \quad (3.14)$$

but with both the mean vector $\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma}$ as unknowns. The tricky part is the prior distribution for the covariance

matrix Σ since we need a distribution with support over the set of $p \times p$ positive definite matrices (see Appendix A.2). Fortunately, such a prior exists and is of rather pleasant form. Let us first remind ourselves about the conjugate prior for the univariate normal model:

$$\theta | \sigma^2 \sim N(\mu_0, \sigma^2 / \kappa_0) \quad (3.15)$$

$$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2), \quad (3.16)$$

where we recall that the conjugate prior for θ had to be conditional on σ^2 , and κ_0 and ν_0 played the role of imaginary sample sizes.

The conjugate prior for the multivariate case has the same structure:

$$\theta | \Sigma \sim N(\theta_0, \Sigma / \kappa_0) \quad (3.17)$$

$$\Sigma \sim \text{IW}(\nu_0, \nu_0 \Sigma_0), \quad (3.18)$$

where we again see that the prior for the mean vector θ is conditional on the covariance matrix Σ with the scalar κ_0 still playing the role of an imaginary sample size. The prior for the covariance matrix Σ is called the **inverse Wishart distribution** and is defined in Box 3.15. The product $\nu_0 \Sigma_0$ as the second parameter in the multivariate case does not seem to mimic the second parameter σ_0^2 in the univariate case, but this is just an artifact of how the inverse Wishart is usually parameterized. The exact form of the density is less important at this stage, but the following things should be noted:

- the inverse Wishart distribution is a distribution over the set of $p \times p$ positive definite matrices
- the inverse Wishart distribution generalizes the scaled inv- χ^2 distribution typically used as a prior for a variance to the multivariate case with a covariance matrix
- the mean of $\text{IW}(\nu_0, \nu_0 \Sigma_0)$ distribution is

$$\mathbb{E}(\Sigma) = \frac{\nu_0}{\nu_0 - p - 1} \Sigma_0,$$

which exists if $\nu_0 > p + 1$. Hence, similarly to the scaled inv- χ^2 distribution, the hyperparameter matrix Σ_0 can be interpreted as our best prior guess for the covariance matrix (it is between the mean and the mode of the distribution).

- the degrees of freedom ν_0 determines the spread of the prior distribution, with larger values of ν_0 corresponding to a more concentrated prior distribution around Σ_0 .

The posterior distribution is summarized in Box 3.16 without proof; the normal-inverse Wishart prior is indeed the conjugate prior

Inverse Wishart

$\mathbf{S} | \nu, \Psi \sim \text{IW}(\nu, \Psi)$ where \mathbf{S} and Ψ have support in $\text{PD}_{p \times p}$, the space of positive definite matrices, and $\nu > p - 1$ is a scalar.

$$p(\mathbf{S}) = \frac{|\Psi|^{\nu/2}}{2^{\nu p/2} \Gamma_p(\nu/2)} |\mathbf{S}|^{-(\nu+p+1)/2} \times \exp\left(-\frac{1}{2} \text{tr}(\Psi \mathbf{S}^{-1})\right),$$

where the trace operator $\text{tr}(\mathbf{A})$ returns the sum of all diagonal elements of the square matrix \mathbf{A} and

$$\Gamma_p(a) = \pi^{p(p-1)/4} \prod_{j=1}^p \Gamma\left(a + \frac{1-j}{2}\right)$$

is the multivariate Gamma function.

$$\mathbb{E}(\mathbf{S}) = \frac{\Psi}{\nu - p - 1} \text{ if } \nu > p + 1$$

Box 3.15: The inverse Wishart distribution.

inverse Wishart distribution

for the multivariate normal model since the posterior also belongs to the same normal-inverse Wishart distribution as the prior. The reader should compare the multivariate case in Box 3.16 with the univariate case in Box 3.2.

Multivariate normal data with Normal-Inverse Wishart prior

Model: $\mathbf{x}_1, \dots, \mathbf{x}_n | \boldsymbol{\theta}, \boldsymbol{\Sigma} \stackrel{\text{iid}}{\sim} N(\boldsymbol{\theta}, \boldsymbol{\Sigma})$, where \mathbf{x}_i are p -vectors

Prior: $\boldsymbol{\theta} | \boldsymbol{\Sigma} \sim N(\boldsymbol{\theta}_0, \boldsymbol{\Sigma} / \kappa_0)$

$\boldsymbol{\Sigma} \sim IW(\nu_0, \nu_0 \boldsymbol{\Sigma}_0)$

Posterior: $\boldsymbol{\theta} | \mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\Sigma} \sim N(\boldsymbol{\theta}_n, \boldsymbol{\Sigma} / \kappa_n)$

$\boldsymbol{\Sigma} | \mathbf{x}_1, \dots, \mathbf{x}_n \sim IW(\nu_n, \nu_n \boldsymbol{\Sigma}_n)$

$$\boldsymbol{\theta}_n = w \cdot \bar{\mathbf{x}} + (1 - w) \cdot \boldsymbol{\theta}_0$$

$$w = \frac{n}{\nu_0 + n}$$

$$\kappa_n = \kappa_0 + n$$

$$\nu_n = \nu_0 + n$$

$$\begin{aligned} \nu_n \boldsymbol{\Sigma}_n &= \nu_0 \boldsymbol{\Sigma}_0 + \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\ &\quad + \frac{\kappa_0 n}{\kappa_n} (\bar{\mathbf{x}} - \boldsymbol{\theta}_0)(\bar{\mathbf{x}} - \boldsymbol{\theta}_0)^T \end{aligned}$$

Box 3.16: Bayesian updating for multivariate normal data with the normal-inverse Wishart prior.

3.8 Likelihood and Information

We will end this chapter by defining some useful measures of how much information a dataset carries about the parameters in a model. Recall from the spam data example in Chapter 2 that the likelihood became more and more peaked around the maximum likelihood estimate (MLE) as the sample size increased. This suggests that the information in a dataset can be measured by how peaked the likelihood is around its mode. This section formalizes this idea using the mathematical concept of Taylor approximations. If you are not familiar with the Taylor approximation of a function, pause your reading and go to Section A.3 in the mathematical Appendix.

Let $\frac{\partial f(x)}{\partial x}|_{x=\hat{x}}$ denote the derivative of the function $f(x)$ evaluated at $x = \hat{x}$. A Taylor expansion of the log-likelihood around the MLE $\hat{\theta}$ gives

$$\begin{aligned} \ln p(\mathbf{x} | \boldsymbol{\theta}) &= \ln p(\mathbf{x} | \hat{\boldsymbol{\theta}}) + \frac{\partial \ln p(\mathbf{x} | \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \\ &\quad + \frac{1}{2!} \frac{\partial^2 \ln p(\mathbf{x} | \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^2 + \dots \end{aligned}$$

where the higher order terms indicated by \dots can be shown to be

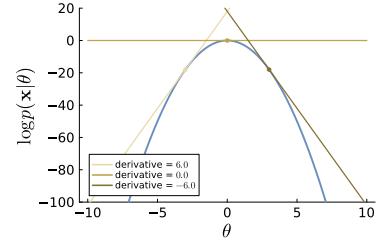


Figure 3.16: A peaked log likelihood function where the derivative changes a lot between the three points: $\theta \in \{-3, 0, 3\}$. The second derivative at the mode $\theta = 0$ is -4 .

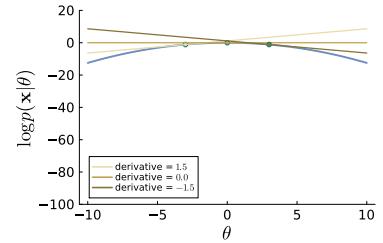


Figure 3.17: A flat log likelihood function where the derivative changes little between the three points: $\theta \in \{-3, 0, 3\}$. The second derivative at the mode $\theta = 0$ is -0.25 .

small in large samples. From the definition of the MLE we know that

$$\frac{\partial \ln p(\theta|x)}{\partial \theta}|_{\theta=\hat{\theta}} = 0$$

We therefore have the following approximation of the likelihood in large samples

$$p(x|\theta) \approx p(x|\hat{\theta}) \exp\left(-\frac{1}{2}J_x(\hat{\theta})(\theta - \hat{\theta})^2\right)$$

where we have defined

$$J_x(\hat{\theta}) = -\frac{\partial^2 \ln p(x|\theta)}{\partial \theta^2}|_{\theta=\hat{\theta}}.$$

Hence, the likelihood function will be proportional to the $N(\hat{\theta}, J_x^{-1}(\hat{\theta}))$ density in large samples. The quantity $J_x(\hat{\theta})$ is clearly the precision in the likelihood and is a natural measure of the information in the data x about the parameter θ :

Definition (Observed information). *The observed information in a sample $x = (x_1, \dots, x_n)$ is defined as*

$$J_x(\hat{\theta}) = -\frac{\partial^2 \ln p(x|\theta)}{\partial \theta^2}|_{\theta=\hat{\theta}} \quad (3.19)$$

Recall from calculus that the second derivative measures how fast the first derivative changes, so $J_x(\hat{\theta})$ measures how peaked the log-likelihood is around the maximum; this is illustrated in Figures 3.16 and 3.17. The negative sign in the definition makes sure the information is always positive, since we know from calculus that the second derivative is negative at the maximum. Note that the observed information is here defined based on *sample* with n observations $x = (x_1, \dots, x_n)^\top$, whereas some textbooks will refer to the observed information based on a single observation x_i . It is easy to see that for iid data we have $J_x(\theta) = nJ_{x_i}(\theta)$, where $J_{x_i}(\theta)$ is the observed information based on a single observation x_i at θ .

The observed information $J_x(\hat{\theta})$ varies from sample to sample. The average, or expected, information is called the Fisher information:

Definition (Fisher information). *The Fisher information is the expected information over all possible samples from the model*

$$I(\theta) = \mathbb{E}_{x|\theta}(J_x(\hat{\theta})), \quad (3.20)$$

where $J_{\theta,x} = -\frac{\partial^2 \ln p(x|\theta)}{\partial \theta^2}$ is the information at θ .

Hence the observed and Fisher information measure different things and are appropriate in different contexts. The observed information is a measure of how much information a *particular* sample

observed information

Fisher information

carries about the parameter, and is therefore the right quantity once you have observed a dataset. The Fisher information is instead a measure of how much information the model is *expected* to provide about the parameter, before you have collected data; the Fisher information is therefore useful in experimental design. Since Bayesian inference conditions on the observed data, we will be mostly concerned with the observed information, but the Fisher information will make an entrance when we discuss invariant priors in Section 4.8.

The observed and Fisher information can be extended to the multiparameter case as follows.

Definition (Observed information in the multiparameter case). *The observed information matrix in a sample $\mathbf{x} = (x_1, \dots, x_n)^\top$ from the model $p(\mathbf{x}|\boldsymbol{\theta})$ with a p -dimensional parameter vector $\boldsymbol{\theta}$ is defined as*

$$J_{\mathbf{x}}(\hat{\boldsymbol{\theta}}) = -\frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \quad (3.21)$$

where $\frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}$ is the $p \times p$ matrix of second derivatives.

Definition (Fisher information in the multiparameter case). *The Fisher information matrix is the expected information matrix over all possible samples from the model*

$$I(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}|\boldsymbol{\theta}}(J_{\mathbf{x}}(\boldsymbol{\theta})). \quad (3.22)$$

The matrix $\frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}$ in (3.21) may be a little intimidating. Writing out its elements explicitly in the case of two parameters, $\boldsymbol{\theta} = (\theta_1, \theta_2)$,

$$\frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} = \begin{pmatrix} \frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1^2} & \frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} \\ \frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_2^2} \end{pmatrix},$$

we see that calculating $\frac{\partial^2 \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}$ is no harder than calculating a single second derivative, there are just more of them. Luckily, we will learn in Chapter 8 that we can often let the computer do this job for us.

observed information matrix

Fisher information matrix

EXERCISES

Exercise 3.1

Let $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$, where θ is assumed known. Show that the Inv- χ^2 distribution is a conjugate prior for σ^2 .

Exercise 3.2

Log-Normal distribution

$$X \sim \text{LN}(\mu, \sigma^2)$$

Support: $X \in (0, \infty)$

$$p(x) = \frac{\exp(-\frac{1}{2\sigma^2}(\log(x) - \mu)^2)}{x\sqrt{2\pi\sigma^2}}$$

$$\mathbb{E}(X) = \exp(\mu + \sigma^2/2)$$

$$\mathbb{V}(X) = (\exp(\sigma^2) - 1)\exp(2\mu + \sigma^2)$$

and μ is the median of X .

If $Y \sim N(\mu, \sigma^2)$ then
 $\log Y \sim \text{LN}(\mu, \sigma^2)$.

Derive the marginal posterior of θ in (3.5) for the iid Gaussian model $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$.

Exercise 3.3

The monthly income (in thousands Swedish Krona) of ten randomly selected persons are: 14, 25, 45, 25, 30, 33, 19, 50, 34 and 67. The log-normal distribution (see Box 3.17 and 3.18) is a commonly used model for income distributions. Let $y_1, \dots, y_n | \mu, \sigma^2 \stackrel{\text{iid}}{\sim} \text{LN}(\mu, \sigma^2)$, where $\mu = \log(33)$ is assumed to be known but σ^2 is unknown with non-informative prior $p(\sigma^2) \propto 1/\sigma^2$.

- a) Show that posterior for σ^2 given μ is the $\text{Inv}-\chi^2(n, \tau^2)$ distribution, where

$$\tau^2 = \frac{\sum_{i=1}^n (\log y_i - \mu)^2}{n}.$$

- b) Simulate 10,000 draws from the posterior of σ^2 (assuming $\mu = \log(33)$) and compare graphically with the theoretical $\text{Inv}-\chi^2(n, \tau^2)$ posterior distribution.
- c) A commonly used measure of income inequality is the Gini coefficient, $0 \leq G \leq 1$, where $G = 0$ is complete income equality, and $G = 1$ means complete income inequality. It can be shown that $G = 2\Phi(\sigma/\sqrt{2}) - 1$ when incomes follow a $\text{LN}(\mu, \sigma^2)$ distribution, where $\Phi(z)$ is the cumulative distribution function (CDF) for the standard normal distribution. Use the posterior draws in a) to compute the posterior distribution of the Gini coefficient G .
- d) Use the posterior draws from c) to compute a 95% equal tail credible interval and a 95% Highest Posterior Density (HPD) interval for G . Compare the two intervals. To compute the HPD interval you will need an estimate of the posterior density for G ; a common approach is to use a kernel density estimator.

Box 3.17: The log-normal distribution.

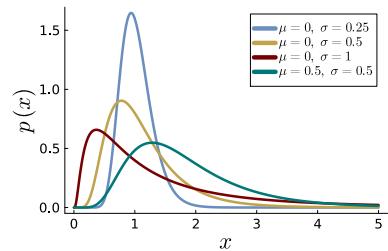


Figure 3.18: Some log-normal distributions.

4 *Priors*

The secret sauce of Bayesian learning is the prior. Only with a prior can we turn a likelihood function into a probability distribution for the unknown parameters, and subsequently use this posterior distribution for decision making. Priors make it possible to fuse information from a variety of different sources. This chapter discusses different types of prior information and how they can be combined in a given model. We will return to the issue of prior elicitation in later chapters when we perform more serious modelling.

There are situations where one may want to use as little prior information as possible, or at least use a prior where the information added is transparent to everyone involved. This can be the case when there is not enough time or effort to carefully determine a prior; we therefore want to make sure that the prior is not greatly affecting the results. Another situation where a noninformative prior may be desired is when reporting scientific results to an unknown audience with potentially rather different prior opinions. The ideal would be to present the posterior distribution for a variety of different priors to contrast the different views and to examine the possibility of a subjective consensus. This is challenging however, particularly when the model contains many parameters and data are only weakly informative. Sections 4.7 and 4.8 presents several ‘non-informative’ priors that may be appealing in such circumstances.

4.1 *Time series*

A time series model will be used to illustrate some ways in which priors can be specified. Time series data have **dependent observations**, and models for such data are therefore necessarily more complex; it is however worthwhile to spend a little time on this topic in this chapter as the particular model presented here will be used many times in this book.

A **time series** is a realization of a **stochastic process** observed over discrete number of time periods, here denoted by $t = 1, 2, \dots, T$. Time series are one of the most commonly occurring data types and are

dependent observations

time series

stochastic process

destined to play a large role in the future as time-stamped data are now collected by many electronic devices and at a rapid pace. Figure 4.1 shows a time series of Swedish inflation, Figure 4.2 displays the daily number of rides with a bike sharing company, and Figure 4.3 illustrates a time series of electroencephalography (EEG) recordings of electrical activity at one brain location. Many time series consist of multivariate measurements at every time period, for example EEG recordings taken simultaneously at multiple locations on the brain, or meteorological data collected at different geographical locations.

The **autoregressive model** of order p is a time series model of the form

$$y_t = \mu + \phi_1(y_{t-1} - \mu) + \dots + \phi_p(y_{t-p} - \mu) + \varepsilon_t, \quad \varepsilon_t \stackrel{\text{iid}}{\sim} N(0, \sigma^2), \quad (4.1)$$

where y_{t-k} is the **k th lagged value** of time series and ε_t are the disturbances, or innovations, that drives the process. Hence, an AR(p) process models today's value y_t as a linear function of the values at the p most recent days y_{t-1}, \dots, y_{t-p} plus a random disturbance ε_t . The time series may equally well be observed on another frequency than daily, for example monthly, with lags being past months. The effect of the k th lags is captured by the AR coefficients ϕ_k .

The AR(p) process in (4.1) is in **steady-state form** where the parameter μ is the unconditional mean $E(y_t)$ of the process. We assume that the AR(p) process is **stationary**, meaning that the mean $E(y_t)$ and variance $V(y_t)$ remain unchanged over time. Moreover, the covariance between any two time points $Cov(y_t, y_s)$ in a stationary process is fully determined by the time distance $|t - s|$ between the observations. The assumption of a constant mean may seem restrictive, but this often means stationary around a deterministic time trend. The unconditional mean μ is important since long horizon forecasts are guaranteed to end up at μ when the process is stationary, i.e.

$$E(y_{T+h} | y_{1:T}) \rightarrow \mu \text{ as } h \rightarrow \infty,$$

where $y_{1:T}$ are all historical data available at the time of the forecast $t = T$. The convergence usually happens rather fast in applications; see Figure 4.4 where an AR(1) model estimated by maximum likelihood is used to predict Swedish inflation for the coming 60 months.

In later chapters we will learn how to obtain the joint posterior of all parameter $p(\mu, \phi_1, \dots, \phi_p, \sigma^2 | \mathbf{y})$ by approximation or simulation. In this chapter we will only worry about how to elicit a joint prior distribution for all model parameters $p(\mu, \phi_1, \dots, \phi_p, \sigma^2)$. We make the simplifying assumption that all parameters are independent a priori; this is most likely not our true beliefs since properties like stationarity involves all ϕ parameters, but it is nevertheless what is most often used in applications. We will also ignore the restrictions

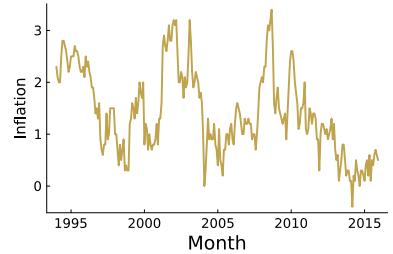


Figure 4.1: Swedish inflation 1995-2016 - annualized monthly observations.

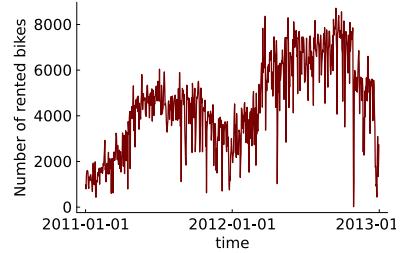


Figure 4.2: Daily number of rides with a bike sharing company.

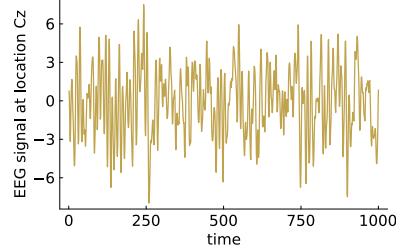


Figure 4.3: EEG recordings of electrical activity at one brain scalp location.

autoregressive model

lagged value

steady-state form

stationary

on ϕ_1, \dots, ϕ_p needed to guarantee stationarity when designing the prior. Such restrictions can be imposed by simply truncating the parameter space of ϕ_1, \dots, ϕ_p to the stationary region. We will walk through a number of methods for prior elicitation and use different methods for different parameters.

4.2 Past or other data

Bayes' theorem dictates that we are not allowed to use the same data in the likelihood and in the prior, i.e. no double dipping of the data if you want the posterior to correctly quantify the uncertainty. It is however allowed to use **past data** for specifying the prior as long as that data are not used in the likelihood; for example, fitting the time series model to data on Swedish inflation data *before* 1995 and using those estimates as the prior mean. Since older data can be from a different economic regime, one would probably use a fairly large prior variance to reflect that the estimates from older data are not necessarily close to the estimates on new data; this is similar to how an older survey was used for the Dirichlet prior in the mobile phone survey data in Section 3.5.

We may base our prior on estimates of the model's parameters from **other data**, e.g. inflation data from other countries during the same time period 1995 – 2016. Other countries are certainly different from Sweden, but still relevant, especially data from similar countries.

4.3 Expert opinion

The ML estimate of the mean of the time series is $\hat{\mu}_{MLE} = 1.409$, which constrains the mean forecasts at longer horizon to end up at 1.409; see Figure 4.4. This is lower than the Central Bank of Sweden's inflation target at 2%. We can use this form of expert opinion as a $\mu \sim N(2, \tau_0^2)$ prior with a small prior variance τ_0^2 , if we trust the central bank experts. Prior information on the steady-state has been shown to improve forecasting performance for a number of economic variables; see Villani (2009).

Prior elicitation of the experts were made on a quantity that was well understood by central bank economists, the long-run behavior of inflation. The challenge is to elicit prior beliefs from experts on quantities that the expert understands well. This will often involve observable quantities, like inflation, rather than abstract parameters in statistical models. The process is often iterative where model consequences from the initially given expert opinion are presented to the expert, who then adjusts the initial opinion. Eliciting expert opinions

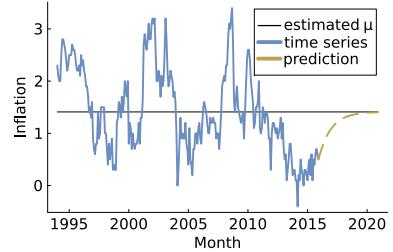


Figure 4.4: Swedish inflation 1995–2016 with 60 months ahead mean prediction in dashed orange.

past data

other data

is a large area in itself, with help from cognitive science to account for the biases and shortcomings that are unfortunately part of being a human; O'Hagan (2019) gives an introduction to the area of expert elicitation and discusses some protocols that aim to minimize these biases.

4.4 Structured regularization priors

An important type of prior beliefs are priors that regularize, or shrink, parameter-rich models. **Regularization priors** are particularly popular in machine learning for probabilistically restricting complex models that would otherwise easily overfit the data. There will be many examples of regularization priors later in the book, but we can get a first understanding of the concept from a commonly used prior for the autoregressive parameters ϕ_1, \dots, ϕ_p in the AR process. A regularization prior on ϕ_1, \dots, ϕ_p makes it possible to use a large **lag length** p even on shorter time series. The prior embodies the idea that the magnitude of the ϕ_k are likely to be smaller for larger k , as in the following prior:

$$\phi_k | \sigma^2 \sim N\left(\mu_k, \sigma^2 \frac{\tau^2}{k^2}\right), \quad (4.2)$$

where $\mu_k = 0$ for all k except for the first lag where $\mu_1 = 0.8$, for example. This centers the prior on the AR(1) process with coefficient $\phi_1 = 0.8$, a reasonable prior guess in the case of Swedish inflation. The reason for scaling the prior variance of all ϕ_k by the error variance σ^2 is that the prior then becomes conjugate conditional on μ (compare with (3.3) in Section 3.3), which will turn out to be useful when we devise an algorithm for posterior simulation in Chapter 9.

The hyperparameter τ is the prior standard deviation of ϕ_1 . The hyperparameter τ is called the **global shrinkage** since it has the effect of shrinking *all* ϕ_k toward their prior means; this is the same effect as the prior standard deviation τ_0 had in the iid normal model in Chapter 2 where the posterior mean μ_n was shrunk toward the prior mean μ_0 via the weight w . Finally, the regularization part of the prior is that the factor $1/k^2$ reduces the prior variance of ϕ_k for longer lags, that is for larger k . Since the prior means for ϕ_k are zero for $k > 1$, this means that longer lags are more heavily shrunk toward zero. The idea here is that longer lags are more likely to be redundant a priori, and their ϕ_k will only be sizeable in the posterior if the data strongly suggest so.

Priors can more generally be used to incorporate **smoothness beliefs**. For example, we will later analyze nonlinear regression models where a response variable y is functionally related to an explanatory

Regularization priors

lag length

global shrinkage

smoothness beliefs

variable x via some function $f(x)$. Rather than assuming a restrictive functional form, most commonly linear, we often want $f(\cdot)$ to be flexible enough to adapt to almost any shape. However, our prior beliefs may still be that $f(\cdot)$ is smooth; Figure 4.5 shows examples of priors for function with wiggly and smooth beliefs. The parameter space here is the abstract space of functions, as will be explained in Chapter 15. We will in later chapters see many examples of quite elegant use of priors to impose smoothness without loosing desired flexibility. A well designed smoothness prior tames the flexibility in the right way and thereby helps to avoid overfitting the data. Note however that a regularization prior still represents subjective beliefs; my prior beliefs regarding the function $f(\cdot)$ puts higher prior probability on the smooth functions in the bottom part of Figure 4.5 than on the wiggly functions shown in the top part of the figure. This then *implies* a posterior that favors smoother functions, unless the data strongly suggest otherwise.

4.5 Hierarchical priors

The structure of the presented regularization prior for the AR(p) process is attractive, but it may be hard to specify an exact value for the global shrinkage τ . The solution is simple: if something is unknown to you, put a prior on it. This gives rise to the following **hierarchical prior** on the AR coefficients

$$p(\phi_1, \dots, \phi_p, \tau^2 | \sigma^2) = p(\phi_1 | \tau^2, \sigma^2) \cdots p(\phi_p | \tau^2, \sigma^2) p(\tau^2 | \sigma^2),$$

where each $p(\phi_k | \tau^2, \sigma^2)$ is the previous $N\left(\mu_k, \sigma^2 \frac{\tau^2}{k^2}\right)$, with independence now only conditionally on τ^2 , and $p(\tau^2 | \sigma^2)$ is the marginal prior for the unknown prior hyperparameter τ^2 . The joint posterior $p(\mu, \phi_1, \dots, \phi_p, \sigma^2, \tau^2 | y)$ involves the now unknown τ^2 , so data will also inform us about τ^2 . Since τ^2 is a variance parameter, the prior $\tau^2 \sim \text{Inv-}\chi^2(\nu_0, \tau_0^2)$ is a natural choice. We still need to specify τ_0^2 our 'best guess' for τ^2 and the uncertainty via ν_0 , but the posterior is often considerably less sensitive to these prior hyperparameters further down the hierarchy, as will be demonstrated in a similar context in Chapter 12.

4.6 Elicitation through prior predictive distributions

So far we have mainly discussed how to elicit prior beliefs directly on the model parameters. This can be challenging when the expert in the subject area does not have a solid understanding of probability theory; the greek letters that statisticians love to use as model parameters may not make any sense to the expert. It is often much more

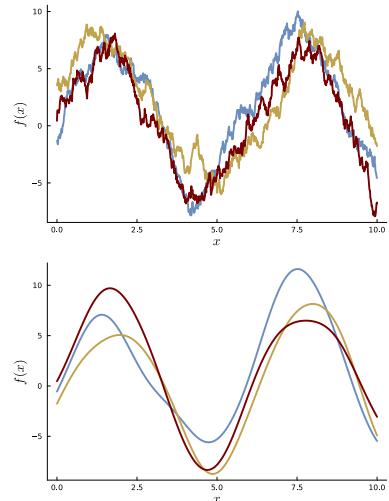


Figure 4.5: Three simulated draws from a prior over functions without smoothness beliefs (top) and with smooth beliefs (bottom).
hierarchical prior

intuitive for a non-statistical expert to specify prior beliefs for the *data*, which usually have a concrete meaning in the real world. The statistician can then use that information to back out the prior beliefs about the model parameters. The key principle here is that:

A good prior for the model parameters should imply a data distribution that agrees with the expert's beliefs about the data.

The relevant data distribution to use here is the **prior predictive distribution**

$$p(x) = \int p(x|\theta)p(\theta)d\theta, \quad (4.3)$$

prior predictive distribution

which is the data distribution of the chosen model $p(x|\theta)$ averaged (marginalized) over the parameter values from the prior. Before collecting any data, the prior predictive distribution $p(x)$ encodes the prior beliefs about the data for a given model and prior.

Let us illustrate this with the Poisson(θ) model for the number of bidders in eBay auctions from Chapter 2. There we used the prior $\theta \sim \text{Gamma}(\alpha, \beta)$ with $\alpha = 2$ and $\beta = 1/2$, which I argued agreed with my prior beliefs about the expected number of bidders, θ , in a randomly selected eBay auction. Having browsed the results from many auctions in the past, I also believe that around 2% of all auctions attract more than 15 bidders. This is a prior predictive statement about the *data* : $\Pr(X > 15) = 0.02$, where X is the number of bidders in an auction. The prior predictive distribution for this model and prior combination can be derived by computing the integral

$$p(x|\alpha, \beta) = \int \text{Poisson}(x|\theta)\text{Gamma}(\theta|\alpha, \beta)d\theta, \quad (4.4)$$

where $\text{Poisson}(x|\theta)$ is the Poisson probability mass function and $\text{Gamma}(\theta|\alpha, \beta)$ is the Gamma density. Note that we have explicitly written out that the prior predictive distribution depends on α and β . We will compute an integral of this type in Chapter 6; a special case of that result shows that the prior predictive distribution from a $\text{Poisson}(x|\theta)$ model with a $\theta \sim \text{Gamma}(\alpha, \beta)$ prior is a negative binomial distribution:

$$X \sim \text{NegBin}\left(\alpha, \frac{\beta}{\beta + 1}\right). \quad (4.5)$$

With the initial $\text{Gamma}(2, 1/2)$ prior used in Chapter 2, the implied prior predictive distribution is therefore $\text{NegBin}(2, 1/3)$, from which we can compute that $\Pr(X > 15) \approx 0.01$; this does not agree with our prior beliefs $\Pr(X > 15) = 0.02$, and we should therefore refine our prior. Formally, we need to solve for α and β in the following

system of equations:

$$\mathbb{E}(X) = \frac{\alpha}{\beta} = 4 \quad (4.6)$$

$$\Pr(X > 15|\alpha, \beta) = 0.02, \quad (4.7)$$

where the mean $\mathbb{E}(X) = \alpha/\beta$ follows from the properties of the negative binomial distribution in Box 6.1, and we have explicitly written out that the prior predictive probability in (4.7) depends on α and β . Solving (4.6) for α gives $\alpha = 4\beta$. Inserting this into (4.7) we can solve the equation $\Pr(X > 15|\alpha = 4\beta, \beta) = 0.02$ for β . This cannot be solved analytically, but we can use numerical methods to find that $\beta \approx 0.322$, and $\alpha = 4\beta \approx 1.289$ gives the desired prior predictive distribution. The left graph of Figure 4.6 shows the initial prior from Chapter 2 and the refined prior; the right graph shows the prior predictive distributions for the number of bidders from the two priors. This [interactive widget](#) lets you experiment with how α and β in the Gamma prior affects the prior predictive distribution.

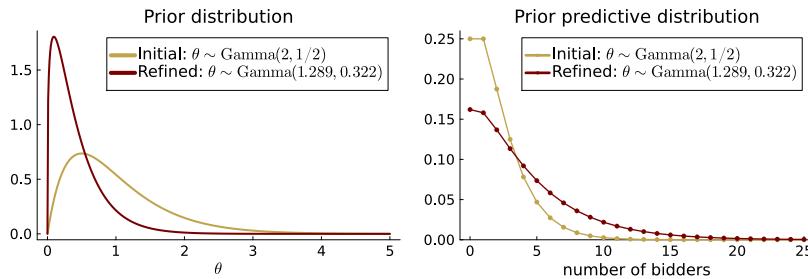


Figure 4.6: Two $\text{Gamma}(\alpha, \beta)$ priors (left) and their prior predictive distributions (right) for the number of bidders in the eBay data in a Poisson model.

In more complex models, we typically cannot compute the prior predictive distribution in closed form. In such cases we can use simulation to obtain the prior predictive distribution. This is done by simulating a large number of parameter values from the prior $p(\theta)$, and for each parameter value simulating a data point from the model $p(x|\theta)$. The prior predictive distribution is then the distribution of the simulated data points. We return to this simulation idea when Bayesian prediction is presented in Chapter 6. Regardless of how you elicit a prior, it is good practice to check that the prior predictive distribution agrees with your prior beliefs about the data. When the data is multi-dimensional, or complex in other ways, one can check the prior predictive distribution for some carefully chosen low-dimensional data summaries; which summaries to choose depends on the particular problem at hand, and is an art in itself. It is often surprisingly useful to simulate just a few realizations from the prior predictive distribution to get a feel for what the prior implies for the data; at a minimum this can be an effective way to detect priors.

with unwanted consequences for the data. As an example, this [interactive widget](#) lets you simulate time series realizations from the prior predictive distribution of an AR(p) process with the structured regularization prior described earlier in this chapter.

4.7 Noninformative priors

It is often convenient to use a prior with relatively little information, at least for some model parameters. Eliciting priors takes effort and we sometimes prefer to specify priors for some parameters with a little less care than other key parameters. The data may also be known to be highly informative on some model parameters and the prior will therefore anyway be overruled by the likelihood. In short, it can be convenient to give some parameters a noninformative prior. A noninformative prior is a bit of a misnomer since any prior carries some information; see [Irony and Singpurwalla \(1997\)](#) for transcribed car dialogue among Bayesian statisticians about this topic. Consider for example the iid Bernoulli(θ) where $\theta \in [0, 1]$. The Uniform(0, 1) distribution is a candidate for a noninformative prior since it assigns the same density to every possible value of θ . There are at least two arguments against this seemingly natural idea.

First, recall that the posterior from a $\theta \sim \text{Beta}(\alpha, \beta)$ prior is $\theta|x \sim \text{Beta}(\alpha + s, \beta + f)$. This means that the prior carries the information equivalent to a prior sample of α successes and β failures. Since the Uniform(0, 1) distribution is the Beta(1, 1) distribution, the uniform prior is equivalent to a prior sample of $n = 2$ trials with one success and one failure; this is clearly *some* information. An alternative definition of a noninformative prior is the **zero sample prior** $\text{Beta}(\epsilon, \epsilon)$ where $\epsilon \downarrow 0$, i.e. ϵ is a tiny number; the posterior is then $\text{Beta}(s, f)$. The idea of the zero sample prior carries directly over the conjugate analysis for exponential family models presented in Box 2.10 by letting v_0 and τ_0 go to zero.

zero sample prior

A second argument against a uniform density as noninformative is that uniformity is typically not preserved when θ is transformed to an alternative parametrization $\phi = g(\theta)$, where $g(\cdot)$ is a one-to-one transformation; for example $g(\theta) = \log(\theta/(1 - \theta))$, the log-odds transformation of the Bernoulli success probability θ . To see this we use the results on transformations of random variables in Box 4.1 to obtain

$$p_\phi(\phi) = p_\theta(g^{-1}(\phi)) \left| \frac{\partial g^{-1}(\phi)}{\partial \phi} \right| = 1 \cdot \frac{e^\phi}{(1 + e^\phi)^2},$$

since $p_\theta(\theta)$ is uniform and the inverse transformation is $g^{-1}(\phi) = e^\phi / (1 + e^\phi)$. Hence, a uniform distribution for θ does not imply a uniform distribution on the log-odds. The next section presents rules

for constructing priors that are guaranteed to be invariant to one-to-one transformations of the model parameter.

4.8 Invariant priors

As we saw in the previous section, a prior which is uniform in one parametrization is usually not uniform in another parametrization; the uniform distribution is not an **invariant prior** for θ in the Bernoulli model. Jeffreys' rule is a method for constructing priors that are guaranteed to be invariant to any one-to-one transformation of the parameter.

Definition (Jeffreys' rule). *Jeffreys' prior for a parameter vector θ in a model $p(\mathbf{x}|\theta)$ is of the form*

$$p(\theta) = |I(\theta)|^{1/2}, \quad (4.8)$$

where $I(\theta)$ is the Fisher information matrix and $|\cdot|$ denotes the matrix determinant.

We will for simplicity concentrate on the one-parameter version $p(\theta) = I(\theta)^{1/2}$ in this section. It can be proved that Jeffreys' prior is invariant to reparametrization (Migon et al., 2014), which was physicist Harold Jeffreys' original motivation for the rule (Jeffreys, 1998). Invariance means that the following two ways to obtain a prior for θ give identical results:

(A) apply Jeffreys' rule directly in the θ -parametrization to obtain

$$p_\theta(\theta) = I(\theta)^{1/2}.$$

(B) apply Jeffreys' rule in the ϕ -parametrization to first obtain

$$p_\phi(\phi) = I(\phi)^{1/2},$$

and then transform to $p_\theta(\theta)$ by the variable transformation formula in Box 4.1

$$p_\theta(\theta) = p_\phi(\phi(\theta)) \left| \frac{d\phi(\theta)}{d\theta} \right| = I(\phi(\theta))^{1/2} \left| \frac{d\phi(\theta)}{d\theta} \right|.$$

EXAMPLE: JEFFREYS' PRIOR FOR BERNOULLI TRIALS. Consider once again the iid Bernoulli model

$$x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\theta),$$

with likelihood $\ln p(\mathbf{x}|\theta) = s \ln \theta + f \ln(1 - \theta)$, where as before $s = \sum_{i=1}^n x_i$ is the number of successes and $f = n - s$ is the number of

Transforming variables

Let $X \sim f_X(x)$ and

$$Y = g(X)$$

an invertible monotonically increasing or decreasing transformation with continuous derivative and inverse transformation

$$X = g^{-1}(Y).$$

The density of Y is then

$$f_Y(y) = f_X(g^{-1}(y)) \left| \frac{d}{dy} g^{-1}(y) \right|$$

Box 4.1: The change-of-variable formula.

invariant prior

Jeffreys' prior

failures. The first and second derivative of the log-likelihood are

$$\begin{aligned}\frac{d \log p(\mathbf{x}|\theta)}{d\theta} &= \frac{s}{\theta} - \frac{f}{(1-\theta)} \\ \frac{d^2 \log p(\mathbf{x}|\theta)}{d\theta^2} &= -\frac{s}{\theta^2} - \frac{f}{(1-\theta)^2}\end{aligned}$$

so that the Fisher information is (using lowercase letters for the random variable s and f)

$$I(\theta) = \frac{E_{\mathbf{x}|\theta}(s)}{\theta^2} + \frac{E_{\mathbf{x}|\theta}(f)}{(1-\theta)^2} = \frac{n\theta}{\theta^2} + \frac{n(1-\theta)}{(1-\theta)^2} = \frac{n}{\theta(1-\theta)}$$

Thus, the Jeffreys prior is

$$p(\theta) = I(\theta)^{1/2} \propto \theta^{-1/2}(1-\theta)^{-1/2} \propto \text{Beta}(1/2, 1/2). \quad (4.9)$$

Hence Jeffreys' prior lies between the zero imaginary sample prior $\text{Beta}(\epsilon, \epsilon)$ and the uniform $\text{Beta}(1, 1)$. This derivation corresponds to Route A above. Exercise 1 shows that the same $\theta \sim \text{Beta}(1/2, 1/2)$ prior is obtained by taking Route B.

EXAMPLE: JEFFREYS' PRIOR FOR A GAUSSIAN VARIANCE. Consider the model $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$. Let us also assume that θ is known and we use Jeffreys' rule to obtain the invariant prior for σ^2 . The log-likelihood is

$$\log p(\mathbf{x}|\sigma^2) = -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\sigma^2) - \frac{\sum_{i=1}^n (x_i - \theta)^2}{2\sigma^2}$$

with first and second derivative

$$\begin{aligned}\frac{\partial}{\partial \sigma^2} \log p(\mathbf{x}|\sigma^2) &= -\frac{1}{2\sigma^2} + \frac{\sum_{i=1}^n (x_i - \theta)^2}{2(\sigma^2)^2} \\ \frac{\partial^2}{\partial (\sigma^2)^2} \log p(\mathbf{x}|\sigma^2) &= \frac{1}{2(\sigma^2)^2} - \frac{\sum_{i=1}^n (x_i - \theta)^2}{(\sigma^2)^3}.\end{aligned}$$

Since $\mathbb{E}_{\mathbf{x}} \sum_{i=1}^n (x_i - \theta)^2 = \sum_{i=1}^n \mathbb{E}_{x_i} (x_i - \theta)^2 = n\sigma^2$ we have

$$I(\sigma^2) = -\frac{1}{2(\sigma^2)^2} + \frac{n\sigma^2}{(\sigma^2)^3} = -\frac{1}{2(\sigma^2)^2} + \frac{n}{(\sigma^2)^2} = \frac{n-1/2}{(\sigma^2)^2},$$

so Jeffreys' prior for the variance is

$$p(\sigma^2) = I(\sigma^2)^{1/2} \propto \frac{1}{\sigma^2},$$

which also implies that Jeffreys' prior for standard deviation is $p(\sigma) \propto \frac{1}{\sigma}$ by the variable transformation formula in Box 4.1 and the invariance of the Jeffreys prior. Since

$$\int_0^\infty \frac{1}{\sigma} d\sigma = \infty$$

Jeffreys' rule gives an **improper prior** in this case, i.e. not a proper density since its integral diverges. Improper priors are somewhat strange, but can be successfully used in practice if the posterior density is known to be proper, i.e. has a finite integral over the whole parameter space. The $1/\sigma$ form of Jeffreys' prior may seem peculiar as it seemingly favors small values for σ . One way of understanding this prior is that it corresponds to a uniform distribution on $\log \sigma \in \mathbb{R}$. In the case where both θ and σ^2 are unknown, the multiparameter version of Jeffreys' rule shows that Jeffreys' prior for σ is still $1/\sigma$ and the prior for θ is uniform.

improper prior

In Chapter 2 we explored the frequentist long run coverage of Bayesian credible intervals for the Bernoulli parameter θ . It was shown that Jeffreys' prior, Beta(1/2, 1/2), came close to the target coverage q for most values of θ . We also briefly discussed that Bayesian intervals have correct coverage in large samples for any prior, and that the actual coverage is $q + c \cdot n^{-1/2}$ for some constant $c > 0$. For Jeffreys' prior the convergence to target coverage is $q + \tilde{c} \cdot n^{-1}$ for some constant \tilde{c} , for any model. That is, Jeffreys' prior gives a quicker convergence to the target coverage q .

Jeffreys' rule has a serious drawback: it violates the likelihood principle; see Section 2.7. The reason is that Jeffreys' rule is based on the Fisher information, which is an expectation with respect to the sampling distribution $p(\mathbf{x}|\theta)$. Exercise 2 asks you to derive Jeffreys' prior for binary data obtained by negative binomial sampling, instead of Bernoulli trials. This exercise shows that Jeffreys' prior for the success probability θ is not the Beta(1/2, 1/2) that we obtained for Bernoulli trials.

Probably the most promising so called Objective Bayes approach is the **reference prior** proposed by José Bernardo based on information arguments. It is motivated as a non-informative prior useful for scientific reporting where one wants to present posterior results to a wide audience using a single well understood prior. The reference prior is invariant to one-to-one transformations and is in fact equal to Jeffreys' prior when the usual regularity conditions for likelihood inference apply. The reference prior is more general however, and avoids some of the problems that have been found with Jeffreys' rule; see [Bernardo and Smith \(2009\)](#) for a comprehensive introduction to reference priors.

reference prior

EXERCISES 4.1

1. Show that using Jeffreys' rule to obtain a prior for the log odds $\phi \equiv \log \theta / (1 - \theta)$ in Bernoulli trials implies the same Beta(1/2, 1/2) prior for θ (i.e. that Route A and B in the text give the same prior).

2. Derive Jeffreys' prior for the success probability θ in the negative binomial model for a dataset where n trials were needed to obtain a predetermined s number of successes. Compare with the Jeffreys prior derived for the Bernoulli model in the text. Discuss the implication for the likelihood principle.
3. Let $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} \text{Expon}(\theta)$.
 - a) Show that Jeffreys' prior is $p(\theta) \propto 1/\theta$. Is it proper?
 - b) Derive the posterior of θ for Jeffreys' prior. Is it proper?
 - c) Motivate the particular form of the Jeffreys prior as non-informative.

NOTEBOOKS 4.2

1. See the notebook priors.

5 Linear Regression

Regression models are the most important of all statistical models as they appear as a component in nearly any situation where an output variable y is modeled as a function of a set of input variables x_1, \dots, x_p . The variable y can for example be the salary of a person that we are trying to explain using information on that person's age (recorded by the continuous variable x_1) and sex (recorded by the binary variable x_2). The input variables are often called **covariates**, **predictors** or **features**, and the output variable is most commonly termed the **response variable** or target variable.

In Chapter 8 we will see regression models for a binary response variable, for example a variable $y \in \{0, 1\}$ that records if a person is employed ($y = 0$) or unemployed ($y = 1$). We will also encounter regression models for response variables of other data types, for example counts, where y may record the number of tickets sold to an event or the number of faulty products produced on any given day by a manufacturing machine. Regression is also the basis for deep neural networks where a linear combination of covariates are passed through several nonlinear activation functions before finally being linked to the response variable.

covariates
features
response variable

5.1 The linear Gaussian regression model

The basic **linear Gaussian regression model** is

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2) \quad \text{for } i = 1, \dots, n, \quad (5.1)$$

where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$ is the vector of covariates for the i th observation in the dataset, \top denotes vector transpose and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ is the vector of **regression coefficients**. The β_j are called **weights** in the machine learning literature and are therefore frequently denoted by w_j . The first element of each \mathbf{x}_i is typically 1 so that β_1 is the **intercept** term; the intercept β_1 is, rather confusingly, called the **bias** in machine learning. Finally, the model is said to be **homoscedastic** since the error variance σ^2 is the same (homo-

linear Gaussian regression model

regression coefficients
weights
intercept
bias
homoscedastic

means same or identical in Greek) for all observations. The case with heteroscedastic errors, $\mathbb{V}(\varepsilon_i) = \sigma_i^2$, will be presented later in the book.

It is convenient to stack all n response observations in a vector $\mathbf{y} = (y_1, \dots, y_n)^\top$ and the covariate observations vectors as rows in the $n \times p$ covariate matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$. The linear Gaussian regression model can then be expressed as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \quad \varepsilon \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2 I_n), \quad (5.2)$$

where ε is a vector with all the ε_i and $\mathcal{N}(0, \sigma^2 I_n)$ is the multivariate normal distribution with diagonal covariance matrix $\sigma^2 I_n$ and I_n is the identity matrix; the simple diagonal structure of $\text{Cov}(\varepsilon)$ reflects the assumption that the ε_i are independent with the same variance. The reader who is not very familiar with vectors and matrices is encouraged to read Appendix A.2 and check that the **matrix-vector product** $\mathbf{X}\boldsymbol{\beta}$ is a vector of length n with the i th element being $\mathbf{x}_i^\top \boldsymbol{\beta}$.

matrix-vector product

5.2 Maximum likelihood

The likelihood for the linear regression model with homoscedastic Gaussian errors is given by the following multivariate normal distribution

$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 I_n), \quad (5.3)$$

where we note that the covariates \mathbf{X} are assumed fixed so the likelihood is the distribution of only the response \mathbf{y} .

The **least squares estimator** $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ is well known to minimize the sum of squared residuals

$$Q(\boldsymbol{\beta}) \equiv (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

When the errors are homoscedastic Gaussian, $\hat{\boldsymbol{\beta}}$ is also the MLE since the log-likelihood from (5.3) is a constant plus $-(1/2\sigma^2)Q(\boldsymbol{\beta})$; hence, minimizing the sum of squared residuals $Q(\boldsymbol{\beta})$ is the same as maximizing the likelihood $p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X})$.

The sampling distribution of the MLE is easily obtained since $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ is a linear function of \mathbf{y} and $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ is a constant matrix. Since $\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 I_n)$, the frequentist sampling distribution of $\hat{\boldsymbol{\beta}}$ is obtained by applying the result in Box 5.1 with $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$, $\Sigma = \sigma^2 I_n$ and $\mathbf{A} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}$ to obtain

$$\hat{\boldsymbol{\beta}}|\boldsymbol{\beta}, \sigma^2, \mathbf{X} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}). \quad (5.4)$$

The MLE can then be used to compute **fitted values** $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ or for making predictions for new covariate observations, see Chapter 6.

The result in (5.4) shows that the MLE is unbiased for $\boldsymbol{\beta}$, that is, $\mathbb{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$; using the MLE guarantees that your estimate of $\boldsymbol{\beta}$ is

least squares estimator
residuals

Linear transformation of Gaussians

Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ be multivariate Gaussian in p dimensions and \mathbf{A} a constant full rank $m \times p$ matrix. Then

$$\mathbf{Ax} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\Sigma\mathbf{A}^\top).$$

Particularly, for $m = 1$ and $\mathbf{A} = (a_1, \dots, a_p)^\top$, a row vector, we get that linear combinations $\sum_{j=1}^p a_j x_j$ of Gaussian variables are Gaussian.

Box 5.1: Linear transformation of Gaussians.

fitted values

correct on average over all possible datasets of size n from the data generating process in (5.2).

The MLE for σ^2 can be shown to be $\hat{\sigma}^2 \equiv (\mathbf{y} - \mathbf{X}\hat{\beta})^\top(\mathbf{y} - \mathbf{X}\hat{\beta})/n$. The estimator $\hat{\sigma}^2$ is however biased for σ^2 , and the following unbiased estimator is typically used instead

$$s^2 \equiv \frac{(\mathbf{y} - \mathbf{X}\hat{\beta})^\top(\mathbf{y} - \mathbf{X}\hat{\beta})}{n - p}.$$

Let us now consider what happens to the MLE $\hat{\beta}$ when we rescale the covariates, i.e. when we multiply each covariate x_j with a constant c_j . Such transformations are very common, for example changing the measurement unit of x_j from meters to centimeters, which corresponds to multiplying all observations in x_j by $c_j = 100$. Now, the regression coefficient β_j measures how much the expected value of the response variable changes if you change x_j by one unit. We would therefore like that the rescaling $x_j \rightarrow c_j \cdot x_j$ of our covariate brings about a corresponding inverse scaling of the estimate: $\hat{\beta}_j \rightarrow (1/c_j) \cdot \hat{\beta}_j$. This desirable **equivariance** property holds for the MLE $\hat{\beta}$, which we will now show.

We will prove a more general equivariance result for the MLE where we linearly transform the whole vector of p covariates by a $p \times p$ invertible transformation matrix \mathbf{A} . The scaling of individual covariates is then the special case where $\mathbf{A} = \text{Diag}(1, \dots, c_j, \dots, 1)$ is a diagonal matrix with ones on the diagonal except in the j th position. So, let us consider what happens to the MLE under the general invertible transformation $\mathbf{x} \rightarrow \mathbf{Ax}$. The matrix of transformed covariates can then be written $\mathbf{X}_A = \mathbf{XA}^\top$, since the i th row of \mathbf{X} is the transpose of the column vector \mathbf{x}_i . The MLE for the coefficients β_A in the transformed covariate model is then easily obtained from the formula of the MLE and a little linear algebra:

$$\begin{aligned}\hat{\beta}_A &= (\mathbf{X}_A^\top \mathbf{X}_A)^{-1} \mathbf{X}_A^\top \mathbf{y} = (\mathbf{A} \mathbf{X}^\top \mathbf{X} \mathbf{A}^\top)^{-1} \mathbf{A} \mathbf{X}^\top \mathbf{y} \\ &= (\mathbf{A}^\top)^{-1} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{A}^{-1} \mathbf{A} \mathbf{X}^\top \mathbf{y} = (\mathbf{A}^\top)^{-1} \hat{\beta}.\end{aligned}$$

Hence, the MLE with transformed covariates is the inversely transformed MLE in with the original model: $\hat{\beta}_A = (\mathbf{A}^\top)^{-1} \hat{\beta}$. To further convince ourselves that this is sensible we can compute the fitted values in the model with transformed covariates

$$\hat{\mathbf{y}}_A = \mathbf{X}_A \hat{\beta}_A = \mathbf{XA}^\top (\mathbf{A}^\top)^{-1} \hat{\beta} = \mathbf{X}\hat{\beta} = \hat{\mathbf{y}},$$

which shows that the fit is not affected by transforming the covariates when maximum likelihood is used to estimate β .

5.3 Non-informative prior

We will start with the invariant Jeffreys prior (see Section 4.8) which can be shown to be

$$p(\beta, \sigma^2) \propto \frac{1}{\sigma^2},$$

i.e. an improper uniform distribution for β independently of σ^2 ; note that σ^2 has the same $1/\sigma^2$ prior as in the iid normal model derived in Section 4.8.

The joint posterior for β and σ^2 is given by Bayes' theorem as

$$\begin{aligned} p(\beta, \sigma^2 | \mathbf{y}) &\propto p(\mathbf{y} | \beta, \sigma^2) p(\beta, \sigma^2) \propto N(\mathbf{y} | \mathbf{X}\beta, \sigma^2 I_n) \cdot \frac{1}{\sigma^2} \\ &= |2\pi\sigma^2 I_n|^{-1/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) \right\} \cdot \frac{1}{\sigma^2}, \end{aligned} \quad (5.5)$$

where the conditioning on the fixed covariates \mathbf{X} is suppressed to simplify the notation. Now, $(\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta)$ can be rewritten using the MLE $\hat{\beta}$ as

$$(\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) = (\mathbf{y} - \mathbf{X}\hat{\beta})^\top (\mathbf{y} - \mathbf{X}\hat{\beta}) + (\beta - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X} (\beta - \hat{\beta}), \quad (5.6)$$

which can be directly verified by substituting the definition of $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$. Recall from linear algebra that the determinant of a diagonal matrix is the product of its diagonal elements, so $|2\pi\sigma^2 I_n| = (2\pi\sigma^2)^n \propto (\sigma^2)^n$. Using this result and (5.6) in (5.5) we obtain the posterior

$$\begin{aligned} p(\beta, \sigma^2 | \mathbf{y}) &\propto (\sigma^2)^{-(n+2)/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\hat{\beta})^\top (\mathbf{y} - \mathbf{X}\hat{\beta}) \right\} \\ &\quad \cdot \exp \left\{ -\frac{1}{2\sigma^2} (\beta - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X} (\beta - \hat{\beta}) \right\} \end{aligned} \quad (5.7)$$

The posterior is most transparent if we use the decomposition of the joint posterior

$$p(\beta, \sigma^2 | \mathbf{y}) = p(\beta | \sigma^2, \mathbf{y}) p(\sigma^2 | \mathbf{y}).$$

Focusing first on $p(\beta | \sigma^2, \mathbf{y}, \mathbf{X})$ we only need to be concerned with the last factor in (5.7) as it is the only part that depends on β ; note that $\hat{\beta}$ only depends on the data. We immediately recognize this last factor as proportional to the multivariate normal density, so

$$\beta | \sigma^2, \mathbf{y} \sim N(\hat{\beta}, \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}).$$

The marginal posterior of σ^2 is obtained by integrating out β in (5.7)

$$\begin{aligned}
p(\sigma^2 | \mathbf{y}) &= \int p(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}) d\boldsymbol{\beta} \\
&\propto (\sigma^2)^{-(n+2)/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \right\} \\
&\quad \cdot \int \exp \left\{ -\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^\top \mathbf{X}^\top \mathbf{X} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \right\} d\boldsymbol{\beta} \\
&\propto (\sigma^2)^{-(n+2)/2} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \right\} (\sigma^2)^{p/2},
\end{aligned}$$

where the last proportionality comes from the fact that

$$\int \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} d\mathbf{x} = |2\pi\Sigma|^{1/2}$$

for any p -vectors \mathbf{x} and $\boldsymbol{\mu}$, and positive definite matrix Σ since we know that the $N(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$ density integrates to one over \mathbb{R}^p . The marginal posterior for σ^2 is therefore

$$p(\sigma^2 | \mathbf{y}) \propto (\sigma^2)^{-[1+(n-p)/2]} \exp \left\{ -\frac{1}{2\sigma^2} (n-p)s^2 \right\}, \quad (5.8)$$

which can be recognized as proportional to the $\text{Inv}-\chi^2(n-p, s^2)$ density.

Linear Gaussian regression with non-informative prior

Model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \stackrel{iid}{\sim} N(0, \sigma^2 I_n)$

Prior: $p(\boldsymbol{\beta}, \sigma^2) \propto 1/\sigma^2$

Posterior: $\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X} \sim N(\hat{\boldsymbol{\beta}}, \sigma^2(\mathbf{X}^\top \mathbf{X})^{-1})$
 $\sigma^2 | \mathbf{y}, \mathbf{X} \sim \text{Inv}-\chi^2(n-p, s^2)$

where $\hat{\boldsymbol{\beta}} \equiv (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ and $s^2 \equiv (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) / (n-p)$.

Box 5.2: Bayesian updating for the linear Gaussian regression with non-informative prior.

We summarize the prior-to-posterior updating in linear Gaussian regression with a noninformative prior in Box 5.2. Note that since we have used a noninformative prior the posterior mean of $\boldsymbol{\beta}$ is exactly the MLE and the posterior covariance matrix of $\boldsymbol{\beta}$ is the same as the sampling covariance matrix of the MLE. The *interpretation* of the Bayesian results Box 5.2 are very different though; the Bayesian posterior is still a distribution for the unknown parameters conditional on the observed dataset. We can make great use of this distribution in prediction and decision making, as we will see in the next chapter.

5.4 Conjugate prior

Let us now turn to the more interesting case with a conjugate prior for the linear Gaussian regression. Recall that the conjugate prior for the iid Normal model $x_1, \dots, x_n | \theta, \sigma^2 \sim N(\theta, \sigma^2)$ was of the form $p(\theta, \sigma^2) = p(\theta|\sigma^2)p(\sigma^2)$ where

$$\begin{aligned}\theta|\sigma^2 &\sim N(\mu_0, \sigma^2/\kappa_0) \\ \sigma^2 &\sim \text{Inv}-\chi^2(\nu_0, \sigma_0^2).\end{aligned}$$

The conjugate prior in linear regression is very similar

$$\beta|\sigma^2 \sim N(\mu_0, \sigma^2 \Omega_0^{-1}) \quad (5.9)$$

$$\sigma^2 \sim \text{Inv}-\chi^2(\nu_0, \sigma_0^2), \quad (5.10)$$

with the prior sample size κ_0 replaced by the $p \times p$ precision matrix Ω_0 .

Linear Gaussian regression with conjugate prior

Model: $\mathbf{y} = \mathbf{X}\beta + \varepsilon, \quad \varepsilon \stackrel{iid}{\sim} N(0, \sigma^2 I_n)$

Prior: $\beta|\sigma^2 \sim N(\mu_0, \sigma^2 \Omega_0^{-1})$
 $\sigma^2 \sim \text{Inv}-\chi^2(\nu_0, \sigma_0^2)$

Posterior: $\beta|\sigma^2, \mathbf{y}, \mathbf{X} \sim N(\mu_n, \sigma^2 \Omega_n^{-1})$
 $\sigma^2|\mathbf{y}, \mathbf{X} \sim \text{Inv}-\chi^2(\nu_n, \sigma_n^2)$
 $\beta|\mathbf{y} \sim t(\mu_n, \sigma_n^2 \Omega_n^{-1}, \nu_n)$

where

$$\begin{aligned}\Omega_n &= \mathbf{X}^\top \mathbf{X} + \Omega_0, \\ \mu_n &= \Omega_n^{-1} (\mathbf{X}^\top \mathbf{X} \hat{\beta} + \Omega_0 \mu_0), \quad \hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \\ \nu_n &= \nu_0 + n, \\ \nu_n \sigma_n^2 &= \nu_0 \sigma_0^2 + (n - p)s^2 + (\mu_n - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X} (\mu_n - \hat{\beta}) \\ &\quad + (\mu_n - \mu_0)^\top \Omega_0 (\mu_n - \mu_0), \\ s^2 &= (\mathbf{y} - \mathbf{X} \hat{\beta})^\top (\mathbf{y} - \mathbf{X} \hat{\beta}) / (n - p).\end{aligned}$$

Box 5.3: Bayesian updating for the linear Gaussian regression with conjugate prior.

A detailed elicitation of the matrix Ω_0 can be demanding. We will have more to say about Ω_0 in Chapter 12, where the simple choice $\Omega_0 = \kappa_0 I_p$ will be discussed in more detail. This prior assumes that the regression coefficients are a priori independent since $\sigma^2 \Omega_0^{-1}$ is diagonal. Prior independence does often not reflect true prior beliefs, but is convenient since we do not have to specify all prior correlations between parameters. Note also that this prior will be

Multivariate student-t

$\mathbf{x}|\mu, \Sigma, \nu \sim t_\nu(\mu, \Sigma)$ where $\mathbf{x} \in \mathbb{R}^p, \mu \in \mathbb{R}^p, \Sigma$ is a $p \times p$ covariance matrix and $\nu > 0$ are the degrees of freedom.

$$\begin{aligned}p(\mathbf{x}) &= \frac{\Gamma((\nu + p)/2)}{\Gamma(\nu/2)(\nu\pi)^{p/2} |\Sigma|^{1/2}} \\ &\times \left(1 + \frac{1}{\nu} (\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\right)^{-(\nu+p)/2}\end{aligned}$$

$$\mathbb{E}(\mathbf{x}) = \mu \text{ if } \nu > 1$$

$$\mathbb{V}(\mathbf{x}) = \frac{\nu}{\nu - 2} \Sigma \text{ if } \nu > 2$$

Marginal distributions:

$$x_k \sim t_\nu(\mu_k, \sigma_k^2)$$

$$\mathbf{x}_1 \sim t_\nu(\mu_1, \Sigma_{11})$$

Conditional distributions:

$$\mathbf{x}_1 | \mathbf{x}_2 \sim t_{\nu+p_2}(\tilde{\mu}_1, c(\mathbf{x}_2) \cdot \tilde{\Sigma}_1)$$

where

$$\tilde{\mu}_1 = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (\mathbf{x}_2 - \mu_2)$$

$$\tilde{\Sigma}_1 = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$

$$c(\mathbf{x}_2) = \frac{\nu + d(\mathbf{x}_2)}{\nu + p_2}$$

$$d(\mathbf{x}_2) = (\mathbf{x}_2 - \mu_2)^\top \Sigma_{22}^{-1} (\mathbf{x}_2 - \mu_2).$$

Box 5.4: The multivariate student-t distribution.

updated with data with the effect that the parameters are dependent in the posterior distribution. One further thing to note is that this prior assumes the *same* prior precision κ_0 for all β coefficients. This is only sensible if the covariates are roughly on the same scale, i.e. have similar mean and standard deviations. Similarly, in contrast to the MLE, this prior gives a posterior which is not equivariant with respect to scaling of the covariates. The problem is that we are insisting on using the *same* prior for the regression coefficient after the transformation, thereby ignoring that the interpretation of β is directly dependent on the scaling of covariates; hence, the prior really should change after rescaling the covariates. This can be achieved by inversely transforming the prior, but a simpler solution is to standardize the covariates before the analysis, for example to have mean zero and unit variance.

A commonly used prior distribution for linear regression is Zellner's prior where $\Omega_0 = \frac{\kappa_0}{n}(\mathbf{X}^\top \mathbf{X})$. Note here that we are using covariate data to formulate a prior, which seems to go against the requirement a prior should not depend on data. However, covariates are typically assumed to be known in regression analysis and can then actually be used when formulating a prior; Zellner's prior does not depend on response data, and the prior therefore contains the information about β *before* observing y . One way to understand the particular form of Zellner's prior is that its prior covariance matrix is $\frac{n}{\kappa_0}\sigma^2(\mathbf{X}^\top \mathbf{X})^{-1}$, which is a scaled version of the sampling covariance matrix of the MLE, $\mathbb{V}(\hat{\beta}) = \sigma^2(\mathbf{X}^\top \mathbf{X})^{-1}$. Zellner's prior therefore automatically adjusts to the potentially different scales of the covariates and can be shown to give a posterior which is equivariant. The covariance matrix in Zellner's prior can more generally be defined as a scaled version of the Fisher information, i.e. the prior information is proportional to the expected information from a sample of size n . By setting $\kappa_0 = 1$, Zellner's prior therefore becomes a noninformative **unit information prior** with information content equal to the expected information from just a single observation. By setting $\kappa_0 = n$, the prior contains as much information as the expected information from a sample of size n .

Box 5.3 shows that the prior in (5.9) is indeed a conjugate prior. Box 5.3 also gives the marginal posterior of β as a **multivariate student-t distribution**, see Box 5.4. The proofs of these results are given at the end of this chapter.

UNIVERSITY SALARIES DATA. The **salaries dataset**, described in the book [Fox and Weisberg \(2019\)](#) and made available as the data frame `Salaries` in the R package `carData`, contains salaries for $n = 397$ university professors. The professors have three different ranks

unit information prior

salaries dataset

(Assistant, Associate and Full professor) and work in two different disciplines (A and B). The number of years since the PhD degree (academic age) is thought to be an important determinant of salaries. Table 5.1 summarizes the data.

Since salaries are positive and often skewed, we follow the usual convention of taking the natural logarithm of salaries as the response variable to make them more normal. Figure 5.1 plots the response variable `logsalary` against `phdage`, the year since the PhD degree normalized so that `phdage= 0` is a fresh PhD graduate and `phdage= 1` for the professor with the highest academic age in the dataset. The relationship seems to be nonlinear with salaries first rapidly increasing with `phdage` and then possibly decreasing toward the end of the career; note however that the data are **cross-sectional** where each observation is a unique professor, not **longitudinal** where persons are measured at several points in time. The nonlinearity will be modelled by using also the square of `phdage` as a covariate. Some of the nonlinearities also seem to disappear when we control for the rank, see the graph on the top right in Figure 5.1.

variable	description	data type	values	comment
<code>logsalary</code>	$\log(\text{salary})$	continuous	$(-\infty, \infty)$	
<code>phdage</code>	years as PhD	continuous	$[0, 1]$	normalized
<code>rank</code>	prof rank	categorical	Asst., Assoc., Prof.	
<code>sex</code>	sex	binary	$[M, F]$	$M = 1$
<code>discipline</code>	discipline	binary	$\{A, B\}$	$A = 1$

cross-sectional
longitudinal

Table 5.1: Summary of the university salaries data.

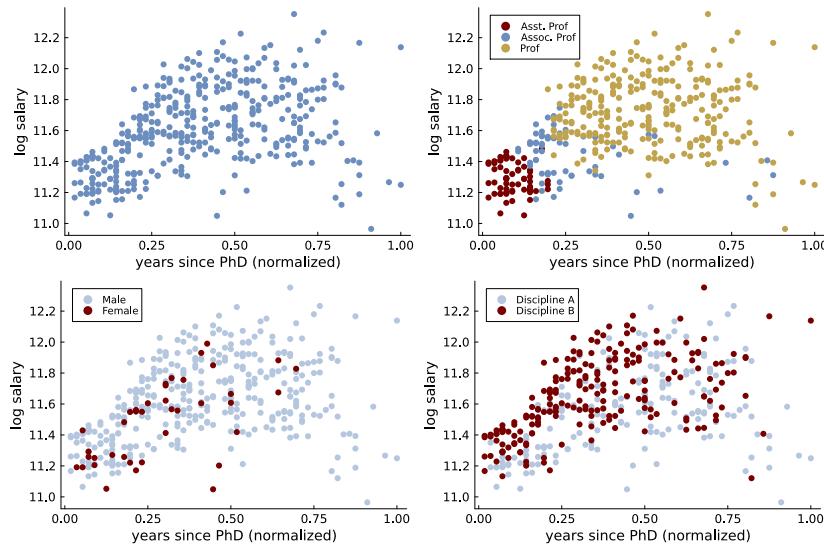


Figure 5.1: University salaries data. Scatterplot of `logsalary` against `phdage` (topleft), color-coded by rank (top right), sex (bottom left) and discipline (bottom right). See Table 5.1 for variable definitions.

Datasets typically contain categorical covariates that needs to be recoded into several binary variables, often called **dummy variables** in statistics and **one-hot encoding** in machine learning. The usual practice is to code a categorical variable with K different values, or

dummy variables
one-hot encoding

levels, into K binary variables, where an observation in category k is recorded as 1 in the k th binary variable and 0 in the other variables. For example, the variable `rank` is A for assistant professors, B for associate professors, and C for full professors. This variable is coded into $K = 3$ new binary variables: `rank1`, `rank2`, and `rank3` where, for example, an observation for an associate professor is coded as 1 in `rank2` and 0 in `rank1` and `rank3`.

Using all K binary variables as covariates in a regression model introduces an exact linear dependence, or exact **multicollinearity**, between the covariates: the sum of the K covariates is exactly one for any observation. This causes problems in the estimation of the regression coefficients and standard practise is therefore to use only $K - 1$ of the binary covariates. We will always drop the binary variable for the first category, which is then the **reference category**. The β coefficient for each of the $K - 1$ included covariates now measures the additional effect of the category *over and above* the effect in the reference category. The effect of the reference category ends up in the intercept since all of the $K - 1$ included covariates are zero for observations in the reference category.

The model for $y = \text{logsalary}$ is then

$$\begin{aligned} \text{logsalary} = & \beta_0 + \beta_1 \cdot \text{phdage} + \beta_2 \cdot \text{phdagesqr} + \beta_3 \cdot \text{rank2} \\ & + \beta_4 \cdot \text{rank3} + \beta_5 \cdot \text{sex} + \beta_6 \cdot \text{discipline} + \varepsilon, \end{aligned}$$

where `phdagesqr` is the square of `phdage`, `sex` and `discipline` are each 0-1 coded variables where `sex=1` for males and `discipline=1` for discipline A , respectively. The errors ε are iid $N(0, \sigma^2)$.

I will first elicit a prior for σ^2 and then for β . My prior for σ^2 is $\text{Inv}-\chi^2(\nu_0 = 10, \sigma_0^2 = 0.3^2)$ and is plotted in Figure 5.2. I came up with this prior by first looking up online that the median salary for associate professors (middle rank) in the US is around \$80,000. Since we will assume that `logsalary` is normally distributed, the salary on the original scale follows a **log-normal distribution**, see Box 3.17. I plotted the implied log-normal distribution of salaries, $\text{LN}(80000, \sigma_0^2)$, for some different values of σ_0^2 . The log-normal distribution for salary given $\sigma_0^2 = 0.3^2$ is shown to the left in Figure 5.3, where the blue line marks out the salary spread as given by the difference between the 10% and 90% percentiles. This agrees rather well with my prior beliefs about the salary spread and $\sigma_0^2 = 0.3^2$ therefore seems reasonable. To determine ν_0 , I compute the same measure of salary spread for 100,000 draws from the $\text{Inv}-\chi^2(\nu_0, \sigma_0^2 = 0.3^2)$ prior for some different value of ν_0 . The result for $\nu_0 = 10$ to the right in Figure 5.3 agrees with my prior beliefs: the spread could be as low as \$50,000, but also as much as \$150,000; I am not very familiar with US salaries.

I will use Zellner's prior $\beta|\sigma^2 \sim N(\mu_0, \sigma^2 \Omega_0^{-1})$ with $\Omega_0 =$

multicollinearity

reference category

log-normal distribution

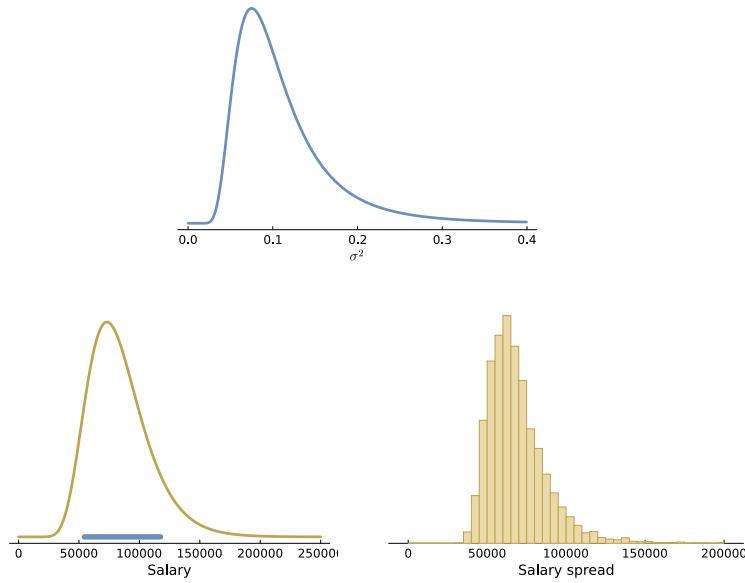


Figure 5.2: Prior for σ^2 in university salaries data.

$\frac{\kappa_0}{n}(\mathbf{X}^\top \mathbf{X})$, and experiment with κ_0 to see the effect of this prior hyperparameter.

The prior mean of β is set to $\mu_0 = (b_0, b_1, b_2, 0, 0, 0)$. This prior implies that the most probable model a priori is the simplified model

$$\text{logsalary} = b_0 + b_1 \cdot \text{phdage} + b_2 \cdot \text{phdagesqr} + \epsilon,$$

and we can determine values for b_0 , b_1 , b_2 and κ_0 that are sensible given our knowledge of university wages. I set $b_0 = \log(70,000)$ so that the median salary for a newly graduated professor ($\text{phdage}=0$) is \$70,000, i.e. \$10,000 below the median salary for middle rank professors found in my online search. The coefficient on phdage is set to $b_1 = 2$ and $b_2 = -1.5$ is used for phdagesqr ; these values imply a median salary of middle age professors ($\text{phdage}=0.5$) around $\exp(\log(70,000) + 2 \cdot 0.5 - 1.5 \cdot 0.5^2) \approx \$130,777$ and a median salary for the oldest professors ($\text{phdage}=1$) around \$115,410.

	mean	std	lower95	upper95
intercept	11.20	0.03	11.13	11.26
phdage	1.36	0.20	0.96	1.75
phdagesqr	-1.11	0.19	-1.48	-0.74
rank2	0.04	0.03	-0.02	0.11
rank3	0.17	0.04	0.10	0.25
sex	0.03	0.02	-0.02	0.07
discipline	-0.07	0.01	-0.09	-0.04
σ	0.20	0.01	0.19	0.21

It remains to determine κ_0 which determines the precision in the prior for β . One way to determine κ_0 is to simulate from the prior for different values of κ_0 and determine if the simulated prior agrees with our prior beliefs. Figure 5.4 explores the prior by simulation for

Figure 5.3: Prior elicitation for σ^2 in university salaries data.

Left: Implied log-normal distribution of salaries from assuming a median salary of 80,000 and $\sigma_0^2 = 0.3^2$; the blue line marks out the wage spread as measured by the difference between the 90% and 10% salary percentiles.

Right: implied prior distribution on the wage spread from the $\text{Inv-}\chi^2(\nu_0 = 10, \sigma_0^2 = 0.3^2)$ prior.

Table 5.2: Summary of the posterior distribution for the regression for the salaries data. The summaries for the regression coefficients were computed analytically from their marginal student- t posterior. The summary for σ was computed by taking the square root transformation of 10,000 posterior draws of σ^2 .

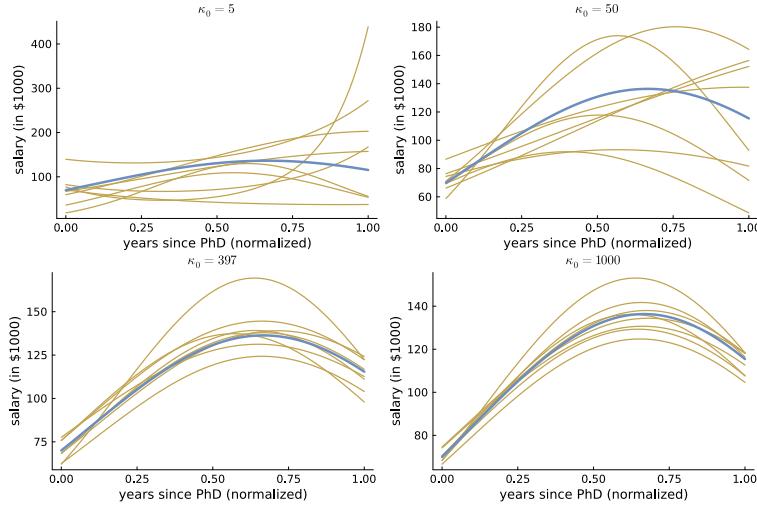


Figure 5.4: Implied relationship between salary and phdage from 10 simulations from the prior for β for four different κ_0 values. The thick blue line is the median salary at the prior mean $\mu_0 = (\log(70,000), 2, -1.5, 0, 0, 0, 0)$.

four different κ_0 : 5, 50, 397 and 1000 by plotting the implied median salary curve over phdage for each of ten β simulated from the prior. Note that the sample size $n = 397$, so $\kappa_0 = 397$ gives the same weight to the prior and the likelihood. The think curve is the median salary at prior mean $\mu_0 = (b_0, b_1, b_2, 0, 0, 0, 0)$ and the thinner lighter curves are the median salary curves for the prior draws of β . The smallest κ_0 gives too much uncertainty about the relationship between salary and phdage and the largest κ_0 implies a prior that contains more information than I actually have about US academic wages. $\kappa_0 = 397$ seems like an appropriate value for my prior beliefs and I will continue the posterior analysis with this prior.

Table 5.2 presents a summary of the posterior distribution for the prior with $\kappa_0 = 397$. We see that all β coefficients except for rank2 and sex have 95% posterior intervals that do not include zero and can therefore be said to be important ("significant") in the Bayesian analysis. Hence, associate professors (rank2=1) can not be shown to have higher salaries than assistant professors, but full professors are likely to have higher salaries than assistant professors (for the same academic age and other covariates). Figure 5.5 shows the marginal posteriors for the regression coefficients for both $\kappa_0 = 397$ and $\kappa_0 = 50$; the maximum likelihood (ML) estimate is also marked out with a green dot; the choice of κ_0 has some effect on the posteriors, which is expected since the sample size is fairly small ($n = 397$). Finally, Figure 5.6 displays the posterior distribution of the salary for female professors in Discipline B at different phdage for the three ranks; there is a clear decrease in salary in the last quarter of the career.

BIKE SHARE DATA. The **bike share dataset** collected by Fanaee-T and Gama (2013) and made available in the UCI repository¹ records the

bike share dataset

¹ <https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset>

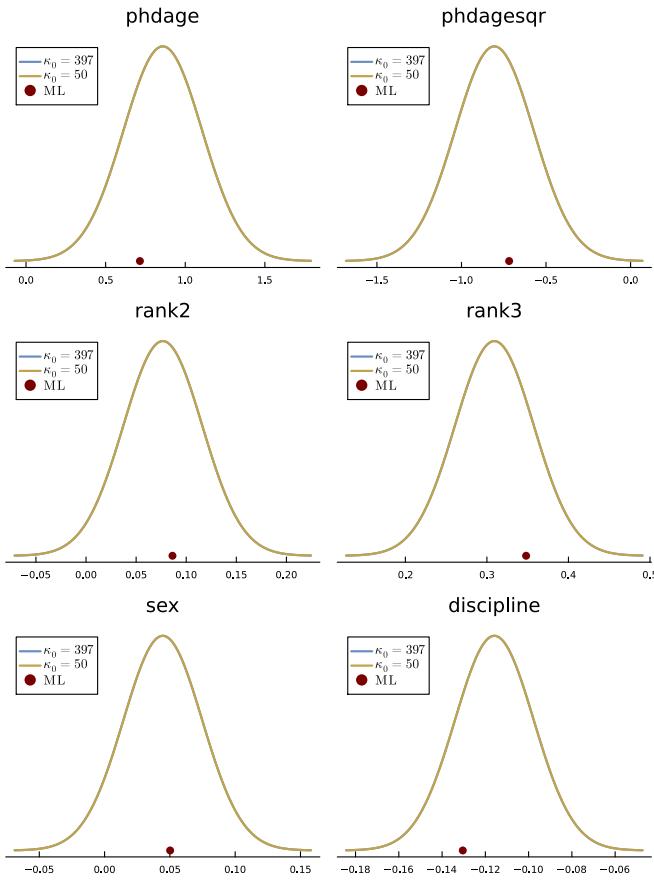


Figure 5.5: Marginal posterior densities for the regression coefficients in linear Gaussian regression fitted to the salaries data with two different priors. The maximum likelihood (ML) estimate is marked out with a red dot. See Table 5.1 for variable definitions.

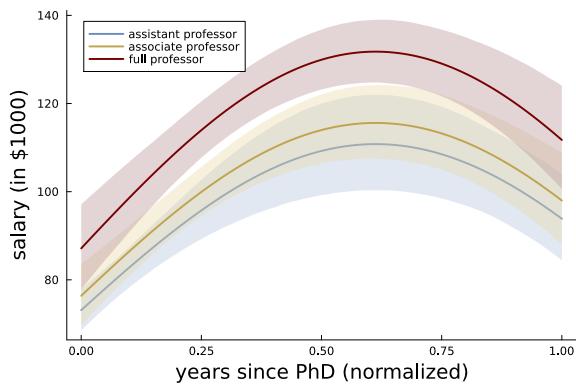


Figure 5.6: Posterior distribution (mean + 95% pointwise intervals) of the salary for female professors in Discipline B at different phdage for the three ranks.

number of daily rides with the bike share company [capital bikeshare](#). The dataset contains the number of daily bike rides on 731 days during the two years 2011 and 2012 and a number of variables that may affect the demand for bikes, e.g. weather conditions, day of the week and holidays; Table 5.3 summarizes the dataset. Figure 5.7 plots the time series of daily rides.

We will ignore the time series nature of `nrides` in this chapter and model it by regression; in the next chapter on prediction the model will be extended with time series aspects. The variable `nrides` are count data, but we will nevertheless model it by a linear Gaussian regression since large counts are often approximately Gaussian; regression models for count data will be introduced in Chapter 8.

variable	description	data type	values	comment
<code>nrides</code>	number of rides	counts	{0, 1, ...}	min= 22, max= 8714
<code>feeltemp</code>	perceived temp	continuous	[0, 1]	min= 0.07, max= 0.85
<code>hum</code>	humidity	continuous	[0, 1]	min= 0.00, max= 0.98
<code>wind</code>	wind speed	continuous	[0, 1]	min= 0.02, max= 0.51
<code>year</code>	year	binary	{0, 1}	year 2011 = 0
<code>season</code>	season	categorical	{1, 2, 3, 4}	winter → fall
<code>weather</code>	weather	ordinal	{1, 2, 3}	clear → rain/snow
<code>weekday</code>	day of week	categorical	{0, 1, ..., 6}	sunday → saturday
<code>holiday</code>	holiday	binary	{0, 1}	holiday = 1

Table 5.3: Summary of the bike share data.

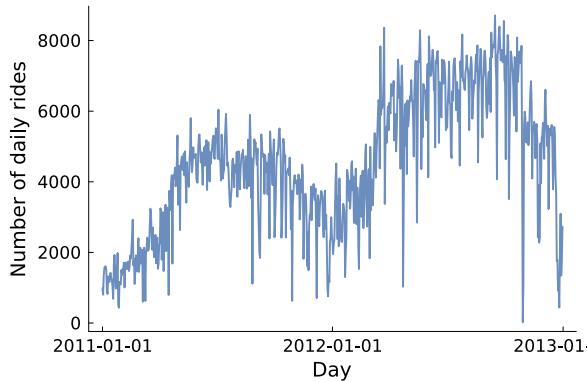


Figure 5.7: Time series plot of `nrides` in the Bike share data.

The dataset contains several categorical covariates which again need to be one-hot encoded into several binary variables. For example, the variable `season` is coded into the $K = 3$ new binary variables: `season2`, `season3`, and `season4`, i.e. the first season (winter) is the reference category.

Figure 5.8 shows scatterplots of `nrides` against the most important continuous covariate, the perceived temperature `feeltemp`. It is clear that `feeltemp` can only explain a smaller portion of the rather sizeable variability in `nrides`. The relationship between `nrides` and `feeltemp` seems slightly nonlinear: there is less biking on the hottest

days, but it is hard to tell when plotting against only one covariate as the decrease in rides at high temperatures may be explained by other covariates, and we choose not to add higher order polynomial terms here. There are also some days with extremely low number of rides; these **outliers** correspond to hurricanes and will be more discussed when we revisit this example in Chapter 6.

outliers

Figure 5.8 also shows the effect of some of the categorical variables by color coding the observations with respect to the levels: rainy weather accounts for some of the low `nrides` observations, and fall (`season = 4`) seems to have more biking than winter (`season=1`) for the same temperature.

I use Zellner's unit information prior for simplicity by setting $\kappa_0 = n = 731$. The prior mean μ_0 for β is set to the zero vector with the exception of the intercept which is 1000 to reflect a rough guess of the number of rides on a day where all covariates are hypothetically zero (a very cold, dry and clear winter Sunday with no wind). I set $\sigma_0^2 = 1000^2$ as a rough guess of σ^2 , with $v_0 = 5$ so that my prior information about σ^2 is only worth five observations.

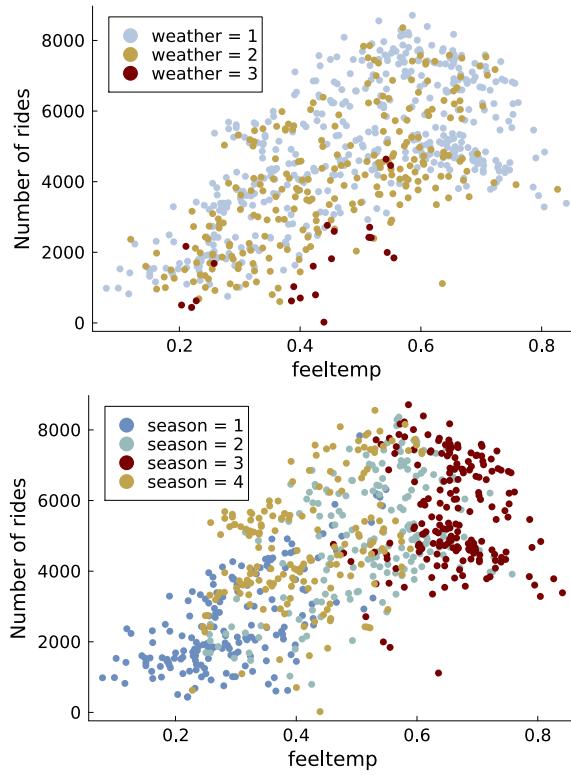


Figure 5.8: Bike share data. Scatterplots of `nrides` against `feeltemp`, color-coded by weather (top), season (bottom). See Table 5.3 for variable definitions.

An observable notebook for this example is available by clicking on the banner in the margin.

BIKE NOTEBOOK

TODO! RESIDUAL ANALYSIS. ADD LAGS. Pointer to prediction

	mean	std	2.5%	97.5%
intercept	1142.26	242.44	666.94	1617.57
feeltemp	5477.32	340.49	4809.79	6144.84
hum	-1245.12	301.81	-1836.83	-653.41
wind	-2494.02	435.24	-3347.32	-1640.72
year	2021.15	62.66	1898.30	2144.01
season2	1173.01	114.54	948.45	1397.58
season3	966.57	147.43	677.53	1255.61
season4	1541.81	98.33	1349.03	1734.58
weather2	-447.70	82.83	-610.09	-285.32
weather3	-1945.19	211.88	-2360.58	-1529.79
weekday1	203.28	118.65	-29.34	435.91
weekday2	298.03	115.94	70.73	525.34
weekday3	377.65	116.18	149.88	605.43
weekday4	392.76	116.15	165.04	620.47
weekday5	454.84	116.13	227.16	682.53
weekday6	446.26	115.54	219.75	672.77
holiday	-630.00	193.07	-1008.52	-251.48
σ	835.00	21.85	793.74	871.65

Table 5.4: Summary of the posterior distribution for the regression for the bike share data. The summaries for the regression coefficients were computed analytically from their marginal student- t posterior. The summary for σ was computed by taking the square root transformation of 10,000 posterior draws of σ^2 .

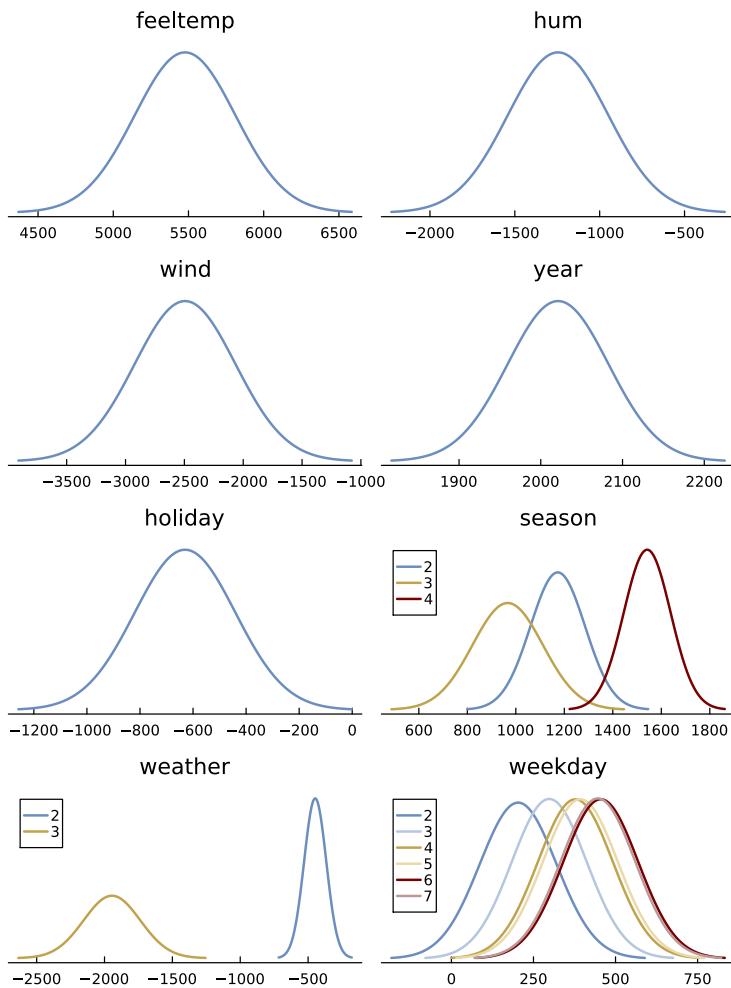


Figure 5.9: Marginal posterior densities for the regression coefficients in linear Gaussian regression fitted to the bike share data. See Table 5.3 for variable definitions.

BIKE PRIOR -> POST

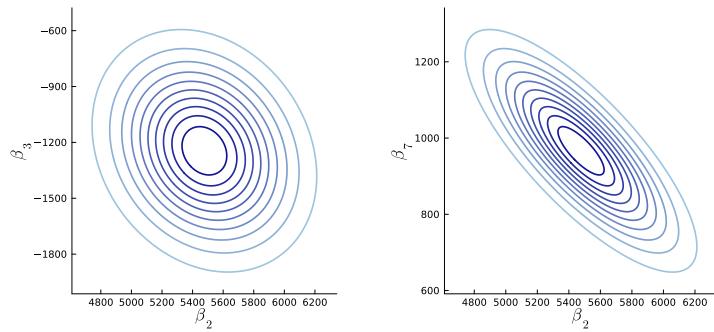


Figure 5.10: Bivariate student- t posterior densities for the regression coefficients on `feeltemp` and `hum` (left), and `feeltemp` and `season3` (right) in the bike share data. See Table 5.3 for variable definitions.

chapter.

PROOFS 5.1

This section derives the posterior distribution for linear regression with a conjugate prior in Box 5.3. The joint posterior is

$$\begin{aligned}
p(\beta, \sigma^2 | \mathbf{y}) &\propto p(\mathbf{y} | \beta, \sigma^2) p(\beta, \sigma^2) \\
&\propto |2\pi\sigma^2 I_n|^{-1/2} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta)\right) \\
&\times |2\pi\sigma^2 \Omega_0^{-1}|^{-1/2} \exp\left(-\frac{1}{2\sigma^2} (\beta - \mu_0)^\top \Omega_0 (\beta - \mu_0)\right) \\
&\times (\sigma^2)^{-(v_0/2+1)} \exp\left(-\frac{1}{2\sigma^2} v_0 \sigma_0^2\right) \\
&\propto (\sigma^2)^{-((v_0+n+p)/2+1)} \exp\left(-\frac{1}{2\sigma^2} (v_0 \sigma_0^2 + (n-p)s^2)\right) \\
&\times \exp\left(-\frac{1}{2\sigma^2} ((\beta - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X}(\beta - \hat{\beta}) + (\beta - \mu_0)^\top \Omega_0 (\beta - \mu_0))\right),
\end{aligned}$$

where $s^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^\top (\mathbf{y} - \mathbf{X}\hat{\beta}) / (n-p)$ as before. Completing the squares in the exponents using the result in Box 3.14 gives

$$\begin{aligned}
&(\beta - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X}(\beta - \hat{\beta}) + (\beta - \mu_0)^\top \Omega_0 (\beta - \mu_0) = \\
&(\beta - \mu_n)^\top \Omega_n (\beta - \mu_n) + (\mu_n - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X}(\mu_n - \hat{\beta}) + (\mu_n - \mu_0)^\top \Omega_0 (\mu_n - \mu_0),
\end{aligned}$$

where $\mu_n = \Omega_n^{-1}(\mathbf{X}^\top \mathbf{X}\hat{\beta} + \Omega_0\mu_0)$. Note that only the first of the three quadratic forms in the last expression depends on β , since $\hat{\beta}$ is just a function of the data. Hence, the last two quadratic forms can be absorbed into the normalizing constant and we have

$$\begin{aligned}
p(\beta, \sigma^2 | \mathbf{y}) &\propto (\sigma^2)^{-((v_n+p)/2+1)} \exp\left(-\frac{v_n \sigma_n^2}{2\sigma^2}\right) \\
&\times \exp\left(-\frac{1}{2\sigma^2} (\beta - \mu_n)^\top \Omega_n (\beta - \mu_n)\right) \tag{5.11}
\end{aligned}$$

where $v_n = v_0 + n$ and $v_n \sigma_n^2 = v_0 \sigma_0^2 + (n-p)s^2 + (\mu_n - \hat{\beta})^\top \mathbf{X}^\top \mathbf{X}(\mu_n - \hat{\beta}) + (\mu_n - \mu_0)^\top \Omega_0 (\mu_n - \mu_0)$. Now,

$$\begin{aligned}
p(\beta, \sigma^2 | \mathbf{y}) &\propto (\sigma^2)^{-((v_n+p)/2+1)} \exp\left(-\frac{v_n \sigma_n^2}{2\sigma^2}\right) |2\pi\sigma^2 \Omega_n^{-1}|^{1/2} \\
&\times |2\pi\sigma^2 \Omega_n^{-1}|^{-1/2} \exp\left(-\frac{1}{2\sigma^2} (\beta - \mu_n)^\top \Omega_n (\beta - \mu_n)\right) \\
&\propto (\sigma^2)^{-(v_n/2+1)} \exp\left(-\frac{v_n \sigma_n^2}{2\sigma^2}\right) \\
&\times |2\pi\sigma^2 \Omega_n^{-1}|^{-1/2} \exp\left(-\frac{1}{2\sigma^2} (\beta - \mu_n)^\top \Omega_n (\beta - \mu_n)\right).
\end{aligned}$$

From the second factor we see that $\beta | \sigma^2, \mathbf{y} \sim N(\mu_n, \sigma^2 \Omega_n^{-1})$ and from the first factor that $\sigma^2 | \mathbf{y} \sim \text{Inv-}\chi^2(v_n, \sigma_n^2)$.

The marginal posterior of β is obtained by integrating $p(\beta, \sigma^2 | \mathbf{y})$ in (5.11)

with respect to σ^2 using properties of the $\text{Inv-}\chi^2$ distribution

$$\begin{aligned} p(\beta | \mathbf{y}) &\propto \int (\sigma^2)^{-((\nu_n+p)/2+1)} \times \exp\left(-\frac{1}{2\sigma^2}(\nu_n\sigma_n^2 + (\beta - \mu_n)^\top \Omega_n(\beta - \mu_n))\right) d\sigma^2 \\ &\propto \left((\nu_n\sigma_n^2 + (\beta - \mu_n)^\top \Omega_n(\beta - \mu_n))/2\right)^{-(\nu_n+p)/2} \\ &\propto \left(1 + \frac{1}{\nu_n}(\beta - \mu_n)^\top \sigma^{-2}\Omega_n(\beta - \mu_n)\right)^{-(\nu_n+p)/2} \end{aligned}$$

which is proportional to the multivariate student- t density

$$\beta | \mathbf{y} \sim t(\mu_n, \sigma_n^2 \Omega_n^{-1}, \nu_n).$$

EXERCISES 5.2

1. This is the first problem.
2. This is the second problem.

NOTEBOOKS 5.3

1. See the notebook [regression](#).

6 Prediction and Decision making

In the previous chapters we studied Bayesian learning of unknown model parameters across a variety of models. Models are a simplified view of the real world, where the parameters often encode some aspects of the mechanism that generated the data. Sometimes parameter learning is the primary goal, for example quantifying how a covariate affects a response in a regression model. In other cases it is only a step toward predicting unknown quantities. We may learn about a regression coefficient in order to predict the effect of a treatment for that individual. Such predictions may ultimately be used to make decisions, for example whether to give a patient a certain treatment or not. In this chapter we will more fully start to see the advantages of the Bayesian approach, in particular how useful it is to have a full probability distribution for the unknowns conditional on the observed data.

6.1 Bayesian prediction

We often want a prediction for an unknown quantity. That unknown quantity can be future yet unobserved value x_t of a time series, or the response observation for a person y given that person's covariate values \mathbf{x} in a regression problem; for example the effect of some medical treatment for a person with some given characteristics such as age, weight, smoking and exercise habits.

We will use the tilde (\sim) symbol to make explicit that a variable is the aim for prediction. Hence, \tilde{y} is for example the regression response that we want to predict based on observed covariates $\tilde{\mathbf{x}}$ for a given subject; in a time series problem we let \tilde{y}_{T+h} denote the time series h time periods in the future relative to the time T where the prediction is made.

Having already observed n training data points, $\mathbf{y} = (y_1, \dots, y_n)$, we now want a prediction of a new observation \tilde{y} . Consider first an iid model for the data: $y_i|\theta \stackrel{iid}{\sim} p(y|\theta)$. The Bayesian **predictive distribution** is the distribution of the unknown \tilde{y} given the known

predictive distribution

training data \mathbf{y} :

$$p(\tilde{y}|\mathbf{y}) = \int_{\theta} p(\tilde{y}|\theta)p(\theta|\mathbf{y})d\theta, \quad (6.1)$$

where $p(\theta|\mathbf{y})$ is the posterior distribution for the model parameters θ . The predictive distribution is therefore a weighted average of the model distribution $p(\tilde{y}|\theta)$ with respect to θ , with the posterior density $p(\theta|\mathbf{y})$ as weights.

The predictive distribution in (6.1) can be summarized by a **point prediction**, for example the predictive mean $\mathbb{E}(\tilde{y}|\mathbf{y})$, and predictive variance $\mathbb{V}(\tilde{y}|\mathbf{y})$ or by a 95% **predictive interval**, just as we summarized the posterior distribution for a parameter θ . But it is important to remember that the Bayesian approach gives a complete probability distribution for the unknown \tilde{y} , not just a point prediction and variance. As we will see, the predicted value can even be a vector in which case the predictive distribution $p(\tilde{\mathbf{y}}|\mathbf{y})$ is a multivariate distribution.

point prediction
predictive interval

When observations are not necessarily iid, for example in time series problems, we have the slightly more general form

$$p(\tilde{y}|\mathbf{y}) = \int_{\theta} p(\tilde{y}|\theta, \mathbf{y})p(\theta|\mathbf{y})d\theta, \quad (6.2)$$

where the distribution of the predicted value \tilde{y} now depends on all the training data \mathbf{y} . In many cases it is enough to condition on just a few of the training data points. For example, in the AR(p) process

$$y_t = \mu + \sum_{k=1}^p \phi_k(y_{t-k} - \mu) + \varepsilon, \quad (6.3)$$

we only need to condition on the p values preceding the time period we want to predict; the AR(p) is said to be a **Markov process** of order p , as explained later in this chapter.

The following subsections presents a series of prediction examples with varying degree of complexity.

Prediction for iid Poisson data

Consider the iid Poisson model

$$x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} \text{Pois}(\theta) \quad (6.4)$$

with the conjugate prior $\theta \sim \text{Gamma}(\alpha, \beta)$. We know from Section 2.4 (see Box 2.9) that the posterior is a Gamma distribution

$$\theta | x_1, \dots, x_n \sim \text{Gamma}(\alpha_n, \beta_n), \quad (6.5)$$

where we have defined the short hand symbols $\alpha_n = \alpha + \sum_{i=1}^n x_i$ and $\beta_n = \beta + n$ to reduce the notation clutter.

The predictive distribution for a new observation \tilde{x} is then

$$\begin{aligned}
 p(\tilde{x}|x_1, \dots, x_n) &= \int p(\tilde{x}|\theta)p(\theta|x_1, \dots, x_n)d\theta \\
 &= \int \frac{\theta^{\tilde{x}}e^{-\theta}}{\tilde{x}!} \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \theta^{\alpha_n-1} e^{-\beta_n\theta} d\theta \\
 &= \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)\tilde{x}!} \int \theta^{\tilde{x}+\alpha_n-1} e^{-(\beta_n+1)\theta} d\theta \\
 &= \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)\tilde{x}!} \frac{\Gamma(\tilde{x}+\alpha_n)}{(\beta_n+1)^{\tilde{x}+\alpha_n}} \\
 &= \frac{\Gamma(\tilde{x}+\alpha_n)}{\Gamma(\alpha_n)\tilde{x}!} \left(\frac{1}{\beta_n+1}\right)^{\tilde{x}} \left(\frac{\beta_n}{\beta_n+1}\right)^{\alpha_n} \\
 &= \frac{\Gamma(\tilde{x}+\alpha_n)}{\Gamma(\alpha_n)\tilde{x}!} \left(1 - \frac{\beta_n}{\beta_n+1}\right)^{\tilde{x}} \left(\frac{\beta_n}{\beta_n+1}\right)^{\alpha_n}, \quad (6.6)
 \end{aligned}$$

where the integral is computed from the normalization constant of the Gamma distribution, i.e. using that (with α as $\tilde{x} + \alpha_n$ and β as $\beta_n + 1$)

$$\int \theta^{\alpha-1} e^{-\beta\theta} d\theta = \frac{\Gamma(\alpha)}{\beta^\alpha},$$

for any positive α and β . The distribution in (6.6) can be recognized as the negative binomial distribution in Box 6.1 with $r = \alpha_n$ and $\theta = \beta_n/(\beta_n + 1)$. Note that the negative binomial variable in Box 6.1 counts the number of successes x before the r th failure occurs, whereas the negative binomial variable in Chapter 2 counted the total number of trials until a certain number of successes. In summary, the predictive distribution for a new observation \tilde{x} in the Poisson model with a $\text{Gamma}(\alpha, \beta)$ prior is

$$\tilde{x}|x_1, \dots, x_n \sim \text{NegBin}\left(\alpha + \sum_{i=1}^n x_i, \frac{\beta + n}{\beta + n + 1}\right). \quad (6.7)$$

Using the formula for the mean of a Negative binomial distribution in Box 6.1, the predictive mean is

$$\mathbb{E}(\tilde{x}|x_1, \dots, x_n) = \frac{\alpha_n}{\beta_n} = \frac{\alpha + \sum_{i=1}^n x_i}{\beta + n};$$

when α and β approaches zero, the prior becomes non-informative, and the predictive mean approaches the sample mean $\bar{x} = \sum x_i / n$. The predictive distribution for the number of bidders in a new coin auction on eBay with low (left) and high (right) reservation price is shown in Figure 6.2.

Prediction in normal model with known variance

My streaming service becomes unreliable and buffers at speeds below 5Mbit/sec. I am therefore particularly interested in this 'catastrophic' event happening tonight while watching my favourite

Negative binomial distribution

$X \sim \text{NegBin}(r, \theta)$
Support: $X \in \{0, 1, \dots\}$

$$\begin{aligned}
 p(x) &= \frac{\Gamma(x+r)}{\Gamma(r)x!} (1-\theta)^x \theta^r \\
 \mathbb{E}(X) &= \frac{r(1-\theta)}{\theta} \\
 \mathbb{V}(X) &= \frac{r(1-\theta)}{\theta^2}
 \end{aligned}$$

Box 6.1: The Negative binomial distribution.

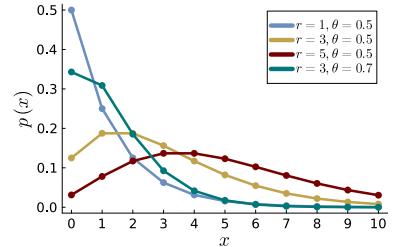


Figure 6.1: Some Negative binomial distributions.

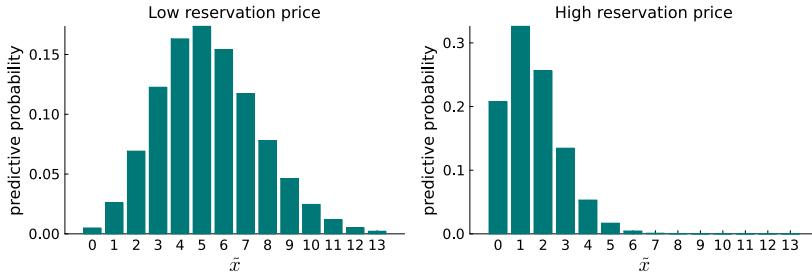


Figure 6.2: Predictive distribution for the number of bidders in an eBay coin auctions with low (left) and high (right) reservation price.

movie. Finding the probability of a *single* measurement lower than 5MBit/sec is an exercise in prediction.

The Gaussian model $\tilde{y} \sim N(\theta, \sigma^2)$ for my internet speed can be trivially expressed as $\tilde{y} = \theta + \tilde{\epsilon}$, where $\tilde{\epsilon} \sim N(0, \sigma^2)$. Since we already know that the posterior for θ is $N(\mu_n, \tau_n^2)$ we see that \tilde{y} is the sum of two Gaussian variables, and the predictive distribution for \tilde{y} is therefore also Gaussian (Box 5.1). To obtain the mean and variance of this predictive distribution it is helpful to first condition on θ and then ‘undo’ the conditioning by integrating with respect to the posterior for θ . This two-step approach of computing the mean and variance of random variables by first conditioning on another random variable are called the iteration laws; specifically the **law of iterated expectation** and the **law of total variance**. Box 6.2 gives these laws in the case of two generic random variables X and Y as typically presented in introductory probability textbooks. Box 6.3 are the exact same laws but written in the context of computing the marginal posterior mean and variance for a parameter. Note the use of subscripts on expectations to explicitly denote which distribution the expectation is taken with respect to; for example

$$\mathbb{E}_{\theta|y}(\theta) \equiv \int \theta p(\theta|y) d\theta.$$

The predictive mean of \tilde{y} can now be computed by first computing the mean given θ

$$\mathbb{E}_{\tilde{y}|y,\theta}(\tilde{y}) = \theta$$

and then undo the conditioning in the second step by taking the posterior expectation

$$\mathbb{E}(\tilde{y}|y) = \mathbb{E}_{\theta|y}(\theta) = \mu_n,$$

since μ_n is by definition the posterior mean of θ . The predictive variance is similarly given by the law of total variance as

$$\begin{aligned}\mathbb{V}(\tilde{y}|y) &= \mathbb{E}_{\theta|y}[\mathbb{V}_{\tilde{y}|y,\theta}(\tilde{y})] + \mathbb{V}_{\theta|y}[\mathbb{E}_{\tilde{y}|y,\theta}(\tilde{y})] \\ &= \mathbb{E}_{\theta|y}(\sigma^2) + \mathbb{V}_{\theta|y}(\theta) \\ &= \sigma^2 + \tau_n^2.\end{aligned}$$

law of iterated expectation

law of total variance

Iteration laws

Law of iterated expectation:

$$\mathbb{E}_X(X) = \mathbb{E}_Y(\mathbb{E}_{X|Y}(X))$$

Law of total variance:

$$\begin{aligned}\mathbb{V}_X(X) &= \mathbb{E}_Y(\mathbb{V}_{X|Y}(X)) \\ &\quad + \mathbb{V}_Y(\mathbb{E}_{X|Y}(X))\end{aligned}$$

Box 6.2: Law of iterated expectations and law of total variance.

Iteration laws for Bayes

Marginal posterior mean:

$$\mathbb{E}_{\theta_1|y}(\theta_1) = \mathbb{E}_{\theta_2|y}(\mathbb{E}_{\theta_1|\theta_2,y}(\theta_1))$$

Marginal posterior variance:

$$\begin{aligned}\mathbb{V}_{\theta_1|y}(\theta_1) &= \mathbb{E}_{\theta_2|y}(\mathbb{V}_{\theta_1|\theta_2,y}(\theta_1)) \\ &\quad + \mathbb{V}_{\theta_2|y}(\mathbb{E}_{\theta_1|\theta_2,y}(\theta_1))\end{aligned}$$

Box 6.3: Iteration laws applied to compute marginal posterior moments given some data y .

Hence, the posterior predictive distribution is

$$\tilde{y}|\mathbf{y} \sim N(\mu_n, \sigma^2 + \tau_n^2).$$

The predictive variance is the sum of the model variance σ^2 and the posterior variance of θ , τ_n^2 , which represents the parameter uncertainty from not knowing θ when we make the prediction. The model variance σ^2 comes from each observation not being completely predictable even if the $N(\theta, \sigma^2)$ model was entirely known. The parameter uncertainty will disappear with more training data since $\tau_n^2 \rightarrow 0$ as $n \rightarrow \infty$. These two sources of predictive uncertainty appear at least implicitly in all models, and their relative importance depends on the size of the training sample, the fit and complexity of the model.

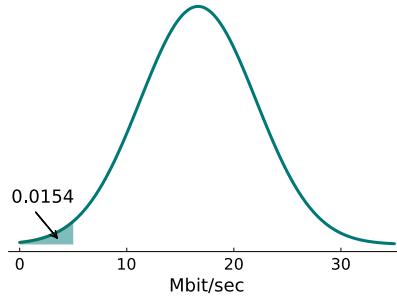


Figure 6.3 plots the predictive distribution for the internet download speed example with $n = 5$ training observations, and marks out the probability of interest, $\Pr(\tilde{y} < 5 | y_1, \dots, y_5) \approx 0.0154$.

Prediction in linear regression

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \stackrel{iid}{\sim} N(0, \sigma^2 I_n). \quad (6.8)$$

We have already in Chapter 5 learned how to use a training dataset (\mathbf{y}, \mathbf{X}) with n observations to compute the posterior for the conjugate prior:

$$\begin{aligned} \sigma^2 | \mathbf{X}, \mathbf{y} &\sim \text{Inv}-\chi^2(\nu_n, \sigma_n^2) \\ \boldsymbol{\beta} | \sigma^2, \mathbf{X}, \mathbf{y} &\sim N(\mu_n, \sigma^2 \Omega_n^{-1}) \end{aligned}$$

Interest now centers on predicting the response $\tilde{\mathbf{y}}$ for \tilde{n} new observations using the $\tilde{n} \times p$ covariate matrix $\tilde{\mathbf{X}}$; the most common case is when $\tilde{n} = 1$ so that we predict a single response \tilde{y} using a vector of covariates $\tilde{\mathbf{x}}$ for that observation. The joint posterior predictive distribution for all \tilde{n} elements of $\tilde{\mathbf{y}}$ is

$$p(\tilde{\mathbf{y}} | \tilde{\mathbf{X}}, \mathbf{X}, \mathbf{y}) = \iint p(\tilde{\mathbf{y}} | \tilde{\mathbf{X}}, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}, \sigma^2 | \mathbf{X}, \mathbf{y}) d\boldsymbol{\beta} d\sigma^2. \quad (6.9)$$

Figure 6.3: Predictive density for the internet download speed after observing $n = 5$. The probability of less than 5Mbit/sec download speed is marked out by the orange region.

INTERNET SPEED PREDICTION

I have here implicitly used some conditional independencies to reduce the notational clutter. We can for example write $p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \beta, \sigma^2)$ instead of the longer $p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{X}, \mathbf{y}, \beta, \sigma^2)$, since $\tilde{\mathbf{y}}$ is independent of the training data \mathbf{X}, \mathbf{y} conditional on the parameters β, σ^2 ; that is, given the true parameter values, there is no additional information in the training data that is useful for predicting $\tilde{\mathbf{y}}$.

The predictive distribution in (6.9) can be derived in two steps:

- i) integrate out β to get $p(\tilde{\mathbf{y}}|\sigma^2, \tilde{\mathbf{X}}, \mathbf{X}, \mathbf{y}) = \int p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \beta, \sigma^2)p(\beta|\sigma^2, \mathbf{y})d\beta$
- ii) integrate out σ^2 to obtain $p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{X}, \mathbf{y}) = \int p(\tilde{\mathbf{y}}|\sigma^2, \tilde{\mathbf{X}}, \mathbf{X}, \mathbf{y})p(\sigma^2|\mathbf{y})d\sigma^2$.

These two steps are derived in the Proof section at the end of the chapter. Box 6.4 summarizes the end result: the joint predictive distribution of all test responses of $\tilde{\mathbf{y}}$ is a multivariate student- t . In the case with a single observation in the test set with covariate vector $\tilde{\mathbf{x}}$, the predictive distribution for the scalar \tilde{y} is a univariate student- t distribution

$$\tilde{y}|\tilde{\mathbf{x}}, \mathbf{y}, \mathbf{X} \sim t\left(\tilde{\mathbf{x}}\boldsymbol{\mu}_n, \sigma_n^2(1 + \tilde{\mathbf{x}}^\top \boldsymbol{\Omega}_n^{-1} \tilde{\mathbf{x}}), \nu_n\right).$$

Predictive density conjugate Gaussian linear regression

Model: $\mathbf{y} = \mathbf{X}\beta + \varepsilon, \quad \varepsilon \stackrel{iid}{\sim} N(0, \sigma^2 I_n)$

Posterior: $\beta|\sigma^2, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\mu}_n, \sigma^2 \boldsymbol{\Omega}_n^{-1})$
 $\sigma^2|\mathbf{y}, \mathbf{X} \sim \text{Inv-}\chi^2(\nu_n, \sigma_n^2)$

Predictive density for \tilde{n} observations with covariate matrix $\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{y}, \mathbf{X} \sim t\left(\tilde{\mathbf{x}}\boldsymbol{\mu}_n, \sigma_n^2(I_{\tilde{n}} + \tilde{\mathbf{X}}\boldsymbol{\Omega}_n^{-1}\tilde{\mathbf{X}}^\top), \nu_n\right)$$

where the posterior hyperparameters $\boldsymbol{\mu}_n$, $\boldsymbol{\Omega}_n$, σ_n^2 and ν_n defined in Box 5.3.

Box 6.4: Predictive density in Gaussian linear regression with a conjugate prior.

The result in Box 6.4 also shows that the predictive distribution includes uncertainty from two sources:

- i) *observation noise* $\tilde{\varepsilon}$, represented by the term $\sigma_n^2 I_{\tilde{n}}$, and
- ii) *parameter uncertainty*, represented by the term $\sigma_n^2(\tilde{\mathbf{X}}\boldsymbol{\Omega}_n^{-1}\tilde{\mathbf{X}}^\top)$.

To see that the latter term is the uncertainty that comes from not knowing the parameters, note that the prediction of $\tilde{\mathbf{y}}$ conditional on the parameters is given by $\tilde{\mathbf{X}}\beta$. This explains why the predictive variance is a quadratic form in $\tilde{\mathbf{X}}$; see the proof at the end of the

chapter for a more precise explanation. The parameter uncertainty will vanish with large training samples since it can be shown that $\Omega_n^{-1} \xrightarrow{p} \mathbf{0}$ and $\sigma_n^2 \xrightarrow{p} \sigma^2$ as $n \rightarrow \infty$, under the common assumption that $n^{-1}\mathbf{X}^\top\mathbf{X}$ converges to a constant non-singular matrix. In Chapter 14 we will see how the posterior predictive distribution can also incorporate model uncertainty, and in Section 14.6 how to handle the uncertainty in the choice of covariates in regression and classification.

TODO! Make predictions for bike share data. Add lag to improve predictions.

Time series prediction with an autoregressive process

Imagine that you have the task of predicting the future development of a time series, for example forecasting the Swedish inflation in the coming 12 quarters. Not only would you like to have a mean prediction, but also some notation of predictive uncertainty.

A popular model for macroeconomic time series forecasting is the autoregressive process with p lags, AR(p), introduced in Chapter 4:

$$y_t = \mu + \phi_1(y_{t-1} - \mu) + \dots + \phi_p(y_{t-p} - \mu) + \varepsilon_t, \quad \varepsilon_t \stackrel{\text{iid}}{\sim} N(0, \sigma^2), \quad (6.10)$$

where y_t is the time series observed at time t , y_{t-k} is the k th lagged value of the time series and ε_t are future shocks to the time series.

Having observed training data $\mathbf{y}_{1:T} \equiv (y_1, \dots, y_T)$ up to time T , we now want the joint predictive density of the time series in the h coming time periods $\tilde{\mathbf{y}}_{T+1:T+h} \equiv (\tilde{y}_{T+1}, \dots, \tilde{y}_{T+h})$. This predictive density can as usual be written as an integral with respect to the posterior distribution,

$$p(\tilde{\mathbf{y}}_{T+1:T+h} | \mathbf{y}_{1:T}) = \int p(\tilde{\mathbf{y}}_{T+1:T+h} | \mathbf{y}_{1:T}, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{y}_{1:T}) d\boldsymbol{\theta},$$

where $\boldsymbol{\theta} = (\mu, \phi_1, \dots, \phi_p, \sigma^2)$ is the vector of parameters in the AR(p) process and $p(\boldsymbol{\theta} | \mathbf{y}_{1:T})$ is the posterior distribution of the parameters based on the training data.

We can simulate from the predictive distribution $p(\tilde{\mathbf{y}}_{T+1:T+h} | \mathbf{y}_{1:T})$ by repeating the following two steps for $i = 1, \dots, m$:

- simulate a posterior parameter draw $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta} | \mathbf{y}_{1:T})$
- simulate a h -steps-ahead realization path $\tilde{\mathbf{y}}_{T+1:T+h}^{(i)}$ from the model $p(\tilde{\mathbf{y}}_{T+1:T+h} | \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(i)})$ conditional on $\boldsymbol{\theta}^{(i)}$.

The first step above will be described in Chapter 9. The second step is implemented using the usual sequential decomposition of a

Markov process

A discrete-time stochastic process X_1, X_2, \dots is said to be **first-order Markov** if

$$\Pr(X_{n+1} | \mathbf{X}_{1:n}) = \Pr(X_{n+1} | X_n),$$

i.e. if the distribution of future values are independent of the past, conditional on the most recent value.

A process is **p th order Markov** if the distribution of future values are independent of the past, conditional on the p most recent values.

A Markov process in discrete time is also called a **Markov Chain**.

joint distribution

$$\begin{aligned} p(\tilde{\mathbf{y}}_{T+1:T+h} | \mathbf{y}_{1:T}, \boldsymbol{\theta}) = & p(\tilde{\mathbf{y}}_{T+1} | \mathbf{y}_{1:T}, \boldsymbol{\theta}) p(\tilde{\mathbf{y}}_{T+2} | \mathbf{y}_{1:T+1}, \boldsymbol{\theta}) \\ & \cdots p(\tilde{\mathbf{y}}_{T+h} | \mathbf{y}_{1:T+h-1}, \boldsymbol{\theta}). \end{aligned} \quad (6.11)$$

We can simulate from each term in (6.11) forward in time, i.e. from left to right, by iterating on (6.10) with a new simulated future shock, ε_{T+j} injected at each time step. Since the AR(p) process is a **Markov process** of order p (see Box 6.5) it is sufficient to condition on the p most recent time observations in each term instead of the full training sample $\mathbf{y}_{1:T}$. Note also that with exception of $p(\tilde{\mathbf{y}}_{T+1} | \mathbf{y}_{1:T}, \boldsymbol{\theta})$, all terms in (6.11) conditions on future, yet unobserved values, which have been simulated in earlier time steps. The algorithm is detailed in Box 6.6 where this is made explicit by highlighting such data points in orange. Using this algorithm with $m = 10,000$ draws produces the $h = 12$ -steps-ahead predictive distribution for Swedish inflation in Figure 6.4.

Markov process

Predictive distribution - AR process.

```

Input: time series  $\mathbf{y}_{1:T} = (y_1, \dots, y_T)$   

        number of predictive draws  $m$ .  

        forecast horizon  $h$ .  

for  $i$  in  $1:m$  do  

     $\mu, \phi_1, \dots, \phi_p, \sigma \leftarrow \text{rPOSTERIORAR}(\mathbf{y}_{1:T}, \text{Prior})$   

     $\varepsilon_{T+1} \leftarrow \text{rNORM}(0, \sigma)$   

     $\tilde{y}_{T+1} \leftarrow \mu + \phi_1(y_T - \mu) + \dots + \phi_p(y_{T+1-p} - \mu) + \varepsilon_{T+1}$   

     $\varepsilon_{T+2} \leftarrow \text{rNORM}(0, \sigma)$   

     $\tilde{y}_{T+2} \leftarrow \mu + \phi_1(\tilde{y}_{T+1} - \mu) + \dots + \phi_p(y_{T+2-p} - \mu) + \varepsilon_{T+2}$   

     $\vdots$   

     $\varepsilon_{T+h} \leftarrow \text{rNORM}(0, \sigma)$   

     $\tilde{y}_{T+h} \leftarrow \mu + \phi_1(\tilde{y}_{T+h-1} - \mu) + \dots + \phi_p(\tilde{y}_{T+h-p} - \mu) + \varepsilon_{T+h}$   

end  

Output:  $m$  draws from the joint predictive density:  

 $p(\tilde{\mathbf{y}}_{T+1}, \dots, \tilde{\mathbf{y}}_{T+h} | \mathbf{y}_{1:T})$ .
```

Box 6.6: Algorithm for simulating from the joint h -step-ahead predictive distribution of an AR process. The function `rPOSTERIORAR()` uses Gibbs sampling and will be presented in Chapter 9. The terms in orange font are future values used in the prediction which have been simulated in earlier time steps.

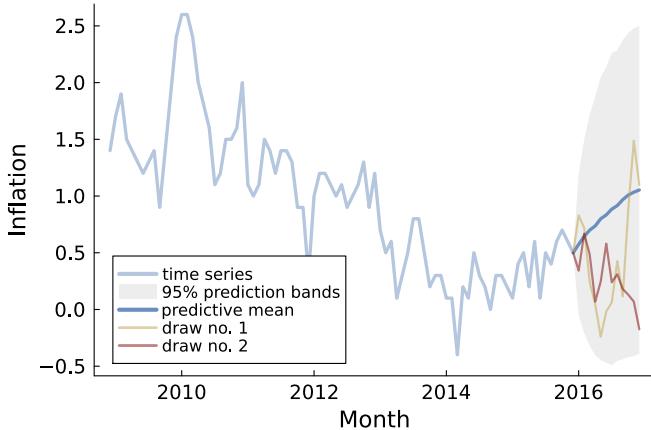


Figure 6.4: Predictive distribution $h = 12$ steps ahead for Swedish inflation represented by a mean prediction in dark blue and 95% predictive intervals as the gray region. Two of the $m = 10,000$ simulated paths from the algorithm in Box 6.6 are marked out.

6.2 Bayesian decisions

Predictions play a major role in modern statistical analysis and machine learning, but the final aim is often **decision making under uncertainty**, with the predictive distribution as an essential component. This is obvious in AI applications, where self-driving cars or automatic stock trading apps need to constantly make decisions to reach pre-determined goals.

One can argue that decisions are nearly always the final aim, even when this is not as apparent as for automatic AI systems. Consider for example data from a clinical trial where the interest is to quantify the reduction in blood pressure from a given dose of beta-blocker medicine. A first idea would be to **infer** the regression coefficient β in linear regression of blood pressure (y) on the covariates dosage (x) and to check if the value $\beta = 0$ (no effect) is included in a 95% HPD credible interval.

A more interesting goal is **predicting** the blood pressure reduction for a given dosage, particularly if additional subject covariates, such as age, sex, exercise habits etc, are used in the regression model to obtain personalized predictions.

The ultimate goal however is to **make a decision** if a particular patient should be given the medicine. To answer this question we clearly need personalized predictions of the blood pressure reduction and its subsequent effect in reducing the probability of stroke, but also a valuation of the cost and the risk of potential side effects of taking the medicine. This section will introduce the Bayesian framework for making such decisions under uncertainty.

Actions and Utility

Let $a \in \mathcal{A}$ be an **action** in a set \mathcal{A} of possible actions. Let $\theta \in \Theta$

decision making under uncertainty

action

represent an unknown quantity. The consequences of choosing action a when θ turns out to be θ is quantified by a **utility function** $u(a, \theta)$. The utility function is subjective since the consequences of the actions typically vary from person to person. Table 6.1 presents the utility of different action-unknown pairs in an example with a discrete set of actions and a discrete set of possible values for θ .

	θ_1	θ_2	\dots	θ_K
a_1	$u(a_1, \theta_1)$	$u(a_1, \theta_2)$	\dots	$u(a_1, \theta_K)$
a_2	$u(a_2, \theta_1)$	$u(a_2, \theta_2)$	\dots	$u(a_2, \theta_K)$
\vdots	\vdots	\vdots		\vdots
a_J	$u(a_J, \theta_1)$	$u(a_J, \theta_2)$	\dots	$u(a_J, \theta_K)$

Table 6.2 presents a toy decision problem where the choice is between bringing or not bringing an umbrella with you today. The consequences of this decision depend on the weather during the day. The best outcome is when you have chosen not to bring the umbrella and it turns out to be a sunny day. The worst outcome is when it rains and you left your umbrella at home.

	Rain	Sun
No umbrella	-50	50
Umbrella	10	30

utility function

Table 6.1: Utility table.

Here are some more interesting decision problems.

SURGERY. A surgeon needs to decide if a delicate surgery should be performed ($a = 1$) or not ($a = 0$). The surgery can be successful ($\theta = 1$) or lead to severe complications ($\theta = 0$). The probability of a successful operation can be computed based on the patient's characteristics. The utility function may be difficult to assess, but should involve the consequences for the patient as well as the cost of the operation. This is an example with discrete \mathcal{A} and Θ .

CENTRAL BANK'S INTEREST RATE DECISIONS. A central bank with an explicit inflation target needs to continually decide the level of their steering rate (a) to simultaneously reach a pre-determined target level for future inflation (θ_1) and to reduce future unemployment (θ_2). A simplified utility function could be

$$u(a, \theta) = \omega (\theta_1(a) - \bar{\theta}_1)^2 - (1 - \omega)\theta_2(a),$$

where $\theta = (\theta_1, \theta_2)$, $\bar{\theta}_1$ is the inflation target, ω is the weight of the inflation target relative to the unemployment, and both unknowns θ_1 and θ_2 are functions of the central bank's steering rate, a . Here the set of actions \mathcal{A} can be considered discrete (steering rate changes are in quarter percentage units) and Θ is two-dimensional and continuous.

Table 6.2: Utility table.

PRICE REDUCTION ON ELECTRIC CARS. A government wants to give a price deduction on purchases of environmentally friendly electric cars (a) in an attempt to minimize future global warming. This is a complex decision problem with many unknowns. The government may settle for the intermediate goal of maximizing the expected utility from the CO₂ reduction from the price deduction (θ), net of the monetary cost of the deduction. Both \mathcal{A} and Θ are continuous spaces here.

FIRMS' STOCKING DECISIONS. Deciding how much of a product to keep in stock is a balancing act where too much stock is costly in storage, and too little stock runs the risk of not being able to deliver on time. Let a be the number of items in stock, θ the unknown number of items demanded by the customers in the coming period and p the set price for the product. A utility function for the firm may have the form

$$u(a, \theta) = \begin{cases} p \cdot \theta - c_1(a - \theta) & \text{if } a \geq \theta \\ p \cdot a - c_2(\theta - a)^2 & \text{if } a < \theta, \end{cases}$$

where c_1 and c_2 are positive constants. In the first case, too much stock was kept ($a \geq \theta$) and the utility is the profit, i.e. revenue $p \cdot \theta$ minus stocking costs for unsold items (c_1 each). In the second case, the firm kept too small stock, can only sell a units and suffers a reputation cost of not being a trustworthy firm that delivers on time. The reputation cost is considered to be quadratic in the number of undelivered items (many people complaining on social media etc).

Maximizing expected utility

There have been a large number of heuristic decision rules proposed in the literature. As an example, one such rule is the **maximin rule**: choose the action that gives the highest utility if the worst possible outcome of θ happens. In the umbrella example in Table 6.2 we see that the maximin decision is to always carry an umbrella since the worst utility for this choice is 10 (it rains) whereas if you choose not to carry an umbrella, the utility could be as low as -50 if it rains. The problem with the minimax rule, and many other heuristics, is that it completely ignores the probability of rain. Always bringing an umbrella may be a decent rule for rainy Bergen in Norway, but not for sunny California.

The Bayesian solution to a decision problem is instead based on the **posterior expected utility** of an action

$$\bar{u}(a) \equiv \mathbb{E}_{\theta|x} [u(a, \theta)] = \int u(a, \theta) p(\theta|x) d\theta, \quad (6.12)$$

from which the **optimal Bayesian decision** is to choose the action

maximin rule

posterior expected utility

optimal Bayesian decision

$a \in \mathcal{A}$ that maximizes posterior expected utility:

$$a^* = \arg \max_{a \in \mathcal{A}} \bar{u}(a). \quad (6.13)$$

The Bayesian decision rule is naturally based on averaging over the unknown θ with respect to your best quantification of uncertainty, the posterior distribution; break the Bayesian eggs and you too can enjoy a Bayesian omelette.

Figure 6.5 illustrates the optimal Bayesian decision in the umbrella toy decision problem in Table 6.2. Note how the probability for rain must be at least 0.25 for the Bayesian to make the same decision as the constantly umbrella carrying pessimist following the maximin rule.

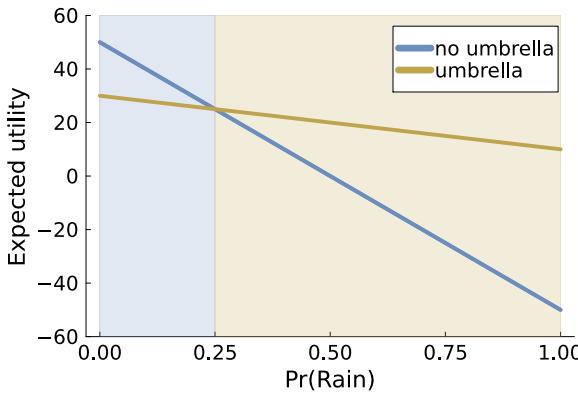


Figure 6.5: Expected utility of bringing an umbrella as function of the probability of rain. The shaded regions mark out the action that maximizes expected utility.

An interesting feature of the Bayesian theory is that it implies the **separation principle**, i.e. that inference and decision problems can and should be kept separate:

separation principle

1. first learn a posterior distribution for the unknown state of the world θ and then
2. set up a utility function $u(a, \theta)$ that values the consequence of actions $a \in \mathcal{A}$, to finally
3. choose the optimal action that maximizes posterior expected utility $\bar{u}(a)$.

Finding the optimal Bayesian decision involves computing the integral in (6.12), which is often analytically intractable. A simple approach is to compute the integral by Monte Carlo integration

$$\bar{u}(a) \equiv \mathbb{E}_{\theta|x} [u(a, \theta)] \approx m^{-1} \sum_{i=1}^m u(a, \theta^{(i)}), \quad (6.14)$$

where $\theta^{(1)}, \dots, \theta^{(m)} \sim p(\theta|x)$ are posterior draws. Expression (6.14) can be optimized numerically, see Chapter 8, to find the approximate Bayes decision a^* .

Point estimate as a decision problem

Chapter 2 presented ways of summarizing a posterior distribution by a measure of posterior location, e.g. the posterior mean, median or mode. Choosing between these location measures is a decision problem where the action a is the **point estimate** of the unknown parameter θ . Reporting the estimate a when the unknown is really θ gives utility $u(a, \theta)$. For example, with a **quadratic utility** $u(a, \theta) = -(a - \theta)^2$, the optimal decision is to summarize the posterior distribution $p(\theta|x)$ with the posterior mean, $\mathbb{E}(\theta|x)$. To see this, note that the negative posterior expected utility is

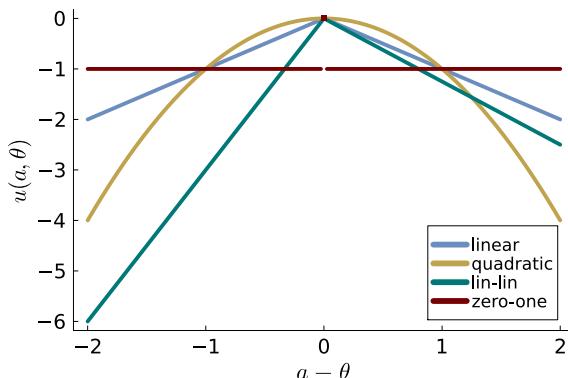
$$\mathbb{E}_{\theta|x}(a - \theta)^2 = \mathbb{E}_{\theta|x}(a - \mathbb{E}(\theta|x) - (\theta - \mathbb{E}(\theta|x)))^2 = (a - \mathbb{E}(\theta|x))^2 + \mathbb{V}(\theta|x),$$

since the cross-term is zero by the fact that $\mathbb{E}_{\theta|x}(\theta - \mathbb{E}(\theta|x)) = 0$. Maximizing the posterior expected utility is the same as minimizing $\mathbb{E}(a - \theta)^2$. Hence, since $\mathbb{V}(\theta|x)$ does not depend on a , the posterior mean $a = \mathbb{E}(\theta|x)$ is the optimal estimate for the quadratic utility function.

Similarly, one can show that the posterior median is optimal under the **linear utility** $u(a, \theta) = -|a - \theta|$. The posterior mode, the θ value with the highest posterior density, seems like a sensible summary, but actually corresponds to the rather peculiar **zero-one utility**

$$u(a, \theta) = \begin{cases} 0 & \text{if } a = \theta \\ -1 & \text{if } a \neq \theta. \end{cases}$$

The zero-one utility hence gives a constant loss (negative utility) regardless of the size of the estimation error $a - \theta$, except when the estimate is spot on.



The linear, quadratic and zero-one utility are all symmetric in the error $a - \theta$. The following so called **lin-lin utility** function values

point estimate

quadratic utility

linear utility

zero-one utility

Figure 6.6: Utility functions for point estimation as a function of estimation error $a - \theta$. The lin-lin utility has $c_1 = 3$ and $c_2 = 1.25$.

lin-lin utility

over- and underestimation differently.

$$u(a, \theta) = \begin{cases} -c_1|a - \theta| & \text{if } a \leq \theta \\ -c_2|a - \theta| & \text{if } a > \theta. \end{cases}$$

where c_1 and c_2 are positive constants. A lin-lin loss is for example appropriate for budget spending prediction, where underestimation is worse than overestimation. The optimal estimate under lin-lin loss can be shown to be the $c_1/(c_1 + c_2) \cdot 100\%$ percentile of the posterior distribution $p(\theta|\mathbf{x})$, i.e. the value that has exactly $c_1/(c_1 + c_2)$ of the probability mass to the left. For example, with $c_1 = 9$ and $c_2 = 1$, i.e. the loss from underestimation is 9 times larger than for overestimation, the optimal estimate is the 90% percentile of $p(\theta|\mathbf{x})$.

The four presented utility functions are plotted in Figure 6.6 as function of the estimation error $a - \theta$.

percentile

PROOFS 6.1

This section derives the predictive distribution for linear regression with a conjugate prior in Box 6.4.

Since $\tilde{\mathbf{y}} = \tilde{\mathbf{X}}\beta + \tilde{\varepsilon}$, $\tilde{\mathbf{X}}$ is assumed known and β and $\tilde{\varepsilon}$ are both normal, we immediately see that $p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \sigma^2, \mathbf{y})$ is multivariate normal with

$$\begin{aligned} \mathbb{E}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \sigma^2) &= \mathbb{E}(\tilde{\mathbf{X}}\beta) + \mathbb{E}(\tilde{\varepsilon}) = \tilde{\mathbf{X}}\mu_n + 0 \\ \mathbb{V}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \sigma^2) &= \mathbb{V}(\tilde{\mathbf{X}}\beta) + \mathbb{V}(\tilde{\varepsilon}) = \tilde{\mathbf{X}}\sigma^2\Omega_n^{-1}\tilde{\mathbf{X}}^\top + \sigma^2I_{\tilde{n}} = \sigma^2\tilde{\Sigma}, \end{aligned}$$

where $\tilde{\Sigma} = I_{\tilde{n}} + \tilde{\mathbf{X}}\Omega_n^{-1}\tilde{\mathbf{X}}^\top$; note that the expectation and variances are with respect to the posterior $p(\beta|\sigma^2, \mathbf{y})$. Hence,

$$\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{y}, \sigma^2 \sim N(\tilde{\mathbf{X}}\mu_n, \sigma^2\tilde{\Sigma}).$$

Now, since $\sigma^2|\mathbf{y} \sim \text{Inv-}\chi^2(\nu_n, \sigma_n^2)$, we have

$$\begin{aligned} p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{y}) &= \int p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \sigma^2, \mathbf{y})p(\sigma^2|\mathbf{y})d\sigma^2 \\ &= \int |2\pi\sigma^2\tilde{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)^\top\tilde{\Sigma}^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)\right) \\ &\quad \times \frac{(\nu_n\sigma_n^2/2)^{\nu_n/2}}{\Gamma(\nu_n/2)}(\sigma^2)^{-(\nu_n/2+1)} \exp\left(-\frac{\nu_n\sigma_n^2}{2\sigma^2}\right)d\sigma^2 \\ &= |2\pi\tilde{\Sigma}|^{-1/2} \frac{(\nu_n\sigma_n^2/2)^{\nu_n/2}}{\Gamma(\nu_n/2)} \\ &\quad \times \int (\sigma^2)^{-(\nu_n+\tilde{n}/2+1)} \exp\left(-\frac{\nu_n\sigma_n^2 + a(\mathbf{y})}{2\sigma^2}\right)d\sigma^2 \\ &= (2\pi)^{-\tilde{n}/2} |\tilde{\Sigma}|^{-1/2} \frac{(\nu_n\sigma_n^2/2)^{(\nu_n+\tilde{n})/2} \Gamma((\nu_n+\tilde{n})/2)}{((\nu_n\sigma_n^2 + a(\mathbf{y}))/2)^{(\nu_n+\tilde{n})/2} \Gamma(\nu_n/2)} \end{aligned}$$

where $a(\tilde{\mathbf{y}}) = (\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)^\top\tilde{\Sigma}^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)$, and the last equality follows from the integrand being proportional to a Inv- χ^2 distribution. The density above can with a little bit of simple algebra be written as

$$\begin{aligned} p(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}, \mathbf{y}) &= \frac{\Gamma((\nu_n+\tilde{n})/2)}{\Gamma(\nu_n/2)(\pi\nu_n)^{\tilde{n}/2}|\sigma_n^2\tilde{\Sigma}|^{1/2}} \\ &\quad \times \left(1 + \frac{1}{\nu_n}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)^\top(\sigma_n^2\tilde{\Sigma})^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mu_n)\right)^{-(\nu_n+\tilde{n})/2}, \end{aligned}$$

which can be recognized as the density of a multivariate student- t distribution.

EXERCISES 6.2

1. (a) Let $x_1, \dots, x_n \stackrel{iid}{\sim} \text{Bern}(\theta)$, with a $\text{Beta}(\alpha, \beta)$ prior for θ . Derive the predictive distribution for x_{n+1} .
- (b) You need to decide if you bring your umbrella during your daily walk. It has rained on two days during the last ten days, and you assess those ten days to be representative of the weather today, the 11th day. Your utility for the action-state combinations are given in the table below. Assume a $\text{Beta}(1, 1)$ prior for θ . Compute the Bayesian decision.
- (c) How sensitive is your decision in (b) to the changes in the prior hyperparameters, α and β ?
2. Let $x_1, \dots, x_n | \theta \stackrel{iid}{\sim} \text{Expon}(\theta)$. Derive the predictive distribution for a new observation \tilde{x}_{n+1} .
3. (a) Let x_i be the number of sales of a product on month i . Let $x_1, \dots, x_n \stackrel{iid}{\sim} N(\theta, \sigma^2)$ be the (approximate) distribution for the sales, and let $\theta \sim N(200, 50^2)$ a priori. Assume that $\sigma^2 = 25^2$ and that we have observed $n = 5$ and $\bar{x} = 320.4$. Compute the predictive distribution for x_6 .
- (b) The company has the choice of performing a marketing campaign for their product. The marketing campaign costs 300 and is believed to increase sales by 20% compared to when no campaign is performed. The company sells the product for $p = 10$ dollar and the cost of producing the product is $q = 5$ dollar. There are no fixed production costs. Assume that the company's utility is described by $U(y) = 1 - \exp(-y/1000)$, where y is the total profit from sales in the next month. Should the company perform the marketing campaign? Hint: the expected value of the exponential function of a normal random variable $S \sim N(\mu, \sigma^2)$ is $\mathbb{E}(\exp(S)) = \exp(\mu + \sigma^2/2)$.

NOTEBOOKS 6.3

1. See the notebook [Prediction and Decision](#).

7 Normal posterior approximation

7.1 Intractable posterior and approximation

So far in this book we have analyzed models where we could always find a conjugate prior. The posterior then belongs to the same distributional family and updating the prior with new data is straightforward, often by adding some summary of the data to the prior hyperparameters. Unfortunately, in many models we simply cannot find a conjugate prior, or even a prior that gives the posterior in a mathematically tractable form. Here is an example.

BETA DISTRIBUTION AS A MODEL FOR PROPORTIONS. Many problems involve data in the form of proportions, i.e. values in the unit interval $[0, 1]$. Some examples of data in the form of proportions are financial debt ratios for firms and the proportion of a certain substance in a container. The $\text{Beta}(\alpha, \beta)$ distribution can be used to model such data. Note that we are here using the Beta distribution as a model for the data, not as a prior for Bernoulli probability, and the interest is in the posterior distribution for the Beta parameters α and β . Assuming the model $x_1, \dots, x_n | \alpha, \beta \stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, \beta)$, the likelihood function is

$$p(x_1, \dots, x_n | \alpha, \beta) = \prod_{i=1}^n \frac{1}{B(\alpha, \beta)} x_i^{\alpha-1} (1 - x_i)^{\beta-1} \quad (7.1)$$

$$= \frac{1}{B(\alpha, \beta)^n} \left(\prod_{i=1}^n x_i \right)^{\alpha-1} \left(\prod_{i=1}^n (1 - x_i) \right)^{\beta-1}, \quad (7.2)$$

where $B(\alpha, \beta)$ is as usual the Beta function. Since the model parameters α and β are partly located inside the fairly complicated Beta function, it is hard to see how one can find a prior that would make the posterior belong to a known distributional family.

The previous example is common: we can compute the unnormalized posterior $p(\theta | \mathbf{x}) \propto p(\mathbf{x} | \theta)p(\theta)$ for any values of the parameters θ , but we cannot recognize the posterior as belonging to a

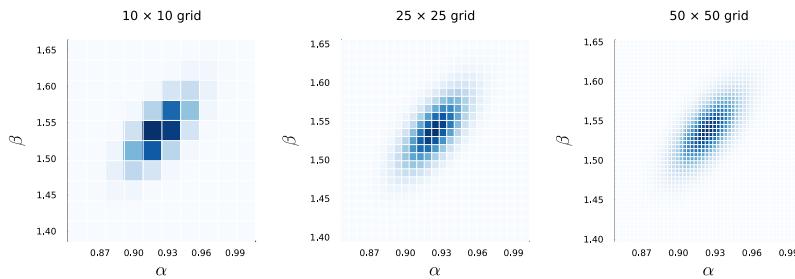
known distribution, and we cannot compute the normalizing constant $\int p(\mathbf{x}|\theta)p(\theta)d\theta$ in closed form. We then say that our problem has an **intractable posterior**. There are three main ways to proceed whenever the posterior distribution is intractable.

First, a brute force solution is to evaluate the unnormalized posterior $p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta)$ over a rectangular grid of θ values

$$(\theta_1, \theta_2, \dots, \theta_p), \text{ for } \theta_1 \in \{\theta_1^{(1)}, \dots, \theta_1^{(N)}\}, \dots, \theta_p \in \{\theta_p^{(1)}, \dots, \theta_p^{(N)}\},$$

and use numerical integration to compute the normalizing constant. The problem with this approach is that the number of grid points is N^p , which grows exponentially with the number of parameters in θ ; if 100 grid points are needed to get accurate numerical results in one dimension, then $100^2 = 10000$ grid points are needed in two dimensions, 1 million points in three dimensions and so on.

Figure 7.1 illustrates the gridding technique in the two-dimensional posterior $p(\alpha, \beta|\mathbf{x})$ from the iid Beta model using the Firm leverage dataset that we will encounter later in this chapter. Exercise 1 asks you use the gridding technique in one and two dimensions.



intractable posterior

Figure 7.1: Evaluating the posterior $p(\alpha, \beta|\mathbf{x})$ in the Firm Leverage dataset over a two-dimensional grid for different grid sizes.

Second, we can explore the posterior distribution by simulation, as we have already seen in Chapters 3 and 5. In those chapters the posterior was tractable, and we used simulation to compute the posterior distribution of transformations of the parameters. We will see in later chapters how powerful simulation algorithms can be used to simulate from intractable posteriors.

A third approach to dealing with intractable posteriors is to approximate the posterior with a simpler tractable distribution. There are several general approaches to approximating a posterior distribution, for example the variational inference method presented in Chapter 11. This chapter presents a simple but often very accurate normal approximation of the posterior, and explains why the normal distribution is a particularly accurate approximation when the dataset is large. The method is shown to have great appeal for practical work since it can be automated on a computer with little effort.

7.2 Taylor approximation of the posterior

Section A.3 in the mathematical appendix states that the K th order Taylor approximation of a function $f(x)$ around the expansion point $x = a$ is

$$f(x) \approx \sum_{k=0}^K \frac{f^{(k)}(a)}{k!} (x - a)^k, \quad (7.3)$$

where $f^{(k)}(a)$ is the k th derivative of $f(x)$ evaluated at $x = a$. The 0th order derivative is just the function itself $f^{(0)}(x) = f(x)$ and $0! = 1$.

Taylor approximations are local approximations that are tailored to the function $f(x)$ around the point $x = a$, and are therefore accurate in a region around $x = a$. Since the posterior distribution will be concentrated in a small region around its mode whenever we have large datasets, we can expect a Taylor approximation of the posterior to be accurate in large datasets. This also suggests that $a = \tilde{\theta}$, where $\tilde{\theta}$ is the posterior mode, is a natural expansion point for the approximation.

To give a first illustration of using the Taylor approximation for posterior approximation, consider a posterior distribution of the form

$$p(\theta|y) \propto \exp(-\exp(\theta/\kappa_0)(\theta - \bar{y})^2), \quad (7.4)$$

where κ_0 is a prior hyperparameter and \bar{y} is the sample mean, and $\theta \in (-\infty, \infty)$. Let us not worry about what kind of model, prior and data gave rise to the posterior in (7.4), just note that the posterior is *not* normal. We will do a Taylor approximation of the *logarithm* of the posterior distribution, $\log p(\theta|y) \propto -\exp(\theta/\kappa_0)(\theta - \bar{y})^2$. The reason for expanding the *log* posterior is that it can often be approximated well with a low order Taylor approximation (see Theorem 2 below). Using the product rule, the first derivative of the log posterior is

$$\frac{\partial \log p(\theta|y)}{\partial \theta} = -\frac{1}{\kappa_0} \exp\left(\frac{\theta}{\kappa_0}\right) (\theta - \bar{y})^2 - \exp\left(\frac{\theta}{\kappa_0}\right) 2(\theta - \bar{y}),$$

which is clearly zero at $\theta = \bar{y}$, so \bar{y} is the posterior mode. The Taylor approximation will therefore be around $\theta = \bar{y}$. The second derivative is

$$\begin{aligned} \frac{\partial^2 \log p(\theta|y)}{\partial \theta^2} = & -\frac{1}{\kappa_0^2} \exp\left(\frac{\theta}{\kappa_0}\right) (\theta - \bar{y})^2 - \frac{2}{\kappa_0} \exp\left(\frac{\theta}{\kappa_0}\right) (\theta - \bar{y}) \\ & - \frac{2}{\kappa_0} \exp\left(\frac{\theta}{\kappa_0}\right) (\theta - \bar{y}) - 2 \exp\left(\frac{\theta}{\kappa_0}\right), \end{aligned}$$

which is $-2 \exp(\bar{y}/\kappa_0) < 0$ at $\theta = \bar{y}$. We can continue in the same fashion to compute the third and fourth derivative, to finally obtain

a fourth order Taylor approximation of the log posterior by inserting the derivatives in (7.3):

$$\begin{aligned}\log p(\theta|\mathbf{y}) \approx & -\exp(\bar{y}/\kappa_0)(\theta - \bar{y})^2 - \frac{\exp(\bar{y}/\kappa_0)}{\kappa_0}(\theta - \bar{y})^3 \\ & - \frac{\exp(\bar{y}/\kappa_0)}{2\kappa_0^2}(\theta - \bar{y})^4.\end{aligned}$$

The graph to the left in Figure 7.2 shows the Taylor approximation of $\log p(\theta|\mathbf{y})$ for the case $\bar{y} = 2$ and $\kappa_0 = 20$. The approximation improves as we increase the polynomial order, and the fourth order approximation is very accurate for all $\theta \in [-10, 10]$. The second order approximation seems to be too crude, the approximation error is large for all θ outside of the interval $(-1, 5)$. However, and this is the important part, the posterior is negligible outside of the interval $(-1, 5)$, so we really do not care if the approximation is poor there. This is shown in the graph to the right in Figure 7.2, which plots the posterior and the implied Taylor approximation on the original scale $p(\theta|\mathbf{y}) \propto \exp(\log p(\theta|\mathbf{y}))$. Even the second order approximation is more or less perfect on the original scale over the the interval $(-1, 5)$ where the posterior density has essentially all its mass.

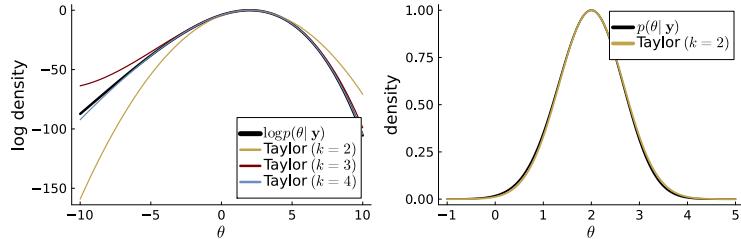


Figure 7.2: Taylor approximation of the posterior distribution $p(\theta|\mathbf{y}) = \exp(-\exp(\bar{y}/\kappa_0)(x - \bar{y})^2)$ around $\theta = \bar{y} = 2$. The figure on the left show Taylor approximations of the log posterior for different polynomial orders. The figure on the right shows the implied second order approximation of the posterior on the original scale.

7.3 Normal posterior approximation and large sample asymptotics

Note that the second order Taylor approximation of the log posterior in the previous example implies the posterior approximation

$$p(\theta|\mathbf{y}) \approx \exp\left(-2\exp(\bar{y}/\kappa_0)(\theta - \bar{y})^2\right),$$

which can be recognized as the normal distribution

$$\theta|\mathbf{y} \sim N\left(\bar{y}, \frac{1}{4\exp(\bar{y}/\kappa_0)}\right).$$

In fact, a posterior based on a second order Taylor approximation of the log posterior is always a normal distribution. This can be seen as

follows. A second order approximation of the log posterior is

$$\begin{aligned}\log p(\theta|\mathbf{y}) &\approx \log p(\tilde{\theta}|\mathbf{y}) + \frac{\partial \log p(\theta|\mathbf{y})}{\partial \theta}|_{\theta=\tilde{\theta}}(\theta - \tilde{\theta}) \\ &\quad + \frac{1}{2} \frac{\partial^2 \log p(\theta|\mathbf{y})}{\partial \theta^2}|_{\theta=\tilde{\theta}}(\theta - \tilde{\theta})^2\end{aligned}\quad (7.5)$$

where the first order term is zero since

$$\frac{\partial \log p(\theta|\mathbf{y})}{\partial \theta}|_{\theta=\tilde{\theta}} = 0$$

from the definition of the posterior mode. Hence, taking exponentials on both sides of (7.5) gives

$$\begin{aligned}p(\theta|\mathbf{y}) &\approx \exp(\log p(\tilde{\theta}|\mathbf{y})) \exp\left(\frac{1}{2} \frac{\partial^2 \log p(\theta|\mathbf{y})}{\partial \theta^2}|_{\theta=\tilde{\theta}}(\theta - \tilde{\theta})^2\right) \\ &\propto \exp\left(\frac{1}{2} \frac{\partial^2 \log p(\theta|\mathbf{y})}{\partial \theta^2}|_{\theta=\tilde{\theta}}(\theta - \tilde{\theta})^2\right)\end{aligned}$$

since $\exp(\log p(\tilde{\theta}|\mathbf{y}))$ does not depend on θ (the mode $\tilde{\theta}$ is just a number for a given dataset). Extending the definition of observed likelihood information in Section 3.8 to the *observed posterior information*

$$J_y(\tilde{\theta}) = -\frac{\partial^2 \ln p(\theta|\mathbf{y})}{\partial \theta^2}|_{\theta=\tilde{\theta}},$$

we have the approximation

$$p(\theta|\mathbf{y}) \approx \exp\left(-\frac{1}{2} J_y(\tilde{\theta})(\theta - \tilde{\theta})^2\right).$$

Hence, we have the following normal posterior approximation

$$\theta|\mathbf{y} \stackrel{\text{a}}{\sim} N\left(\tilde{\theta}, J_y^{-1}(\tilde{\theta})\right), \quad (7.6)$$

where the symbol $\stackrel{\text{a}}{\sim}$ denotes "is approximately distributed as".

The following theorem shows that this normal posterior approximation will become more and more accurate with the size of the dataset.

Theorem 2 (large sample normality of posterior). *The posterior distribution of θ conditional on data $\mathbf{y} = (y_1, \dots, y_n)$ converges to a normal distribution in large samples:*

$$J_y^{1/2}(\tilde{\theta})(\theta - \tilde{\theta}) | \mathbf{y} \xrightarrow{d} N(0, 1), \text{ as } n \rightarrow \infty,$$

where $\tilde{\theta}$ is the posterior mode and

$$J_y(\tilde{\theta}) = -\frac{\partial^2 \ln p(\theta|\mathbf{y})}{\partial \theta^2}|_{\theta=\tilde{\theta}}$$

is the observed posterior information at $\tilde{\theta}$.

The result in Theorem 2 is often called the **Bernstein-von Mises theorem** after the persons that proved a similar result. The result requires some regularity conditions, for example that the posterior distribution becomes concentrated in a small neighborhood around the posterior mode as the sample size grows large; see [Bernardo and Smith \(2009\)](#) for some details. These conditions are met in most models used in practise, but we will give an example where they are not, and the result in Theorem 2 fails to hold.

Bernstein-von Mises theorem

Theorem 2 does not tell us how large n must be for the approximation to be accurate, and this will be model specific. But the convergence happens quickly in many problems and the normal approximation is often accurate enough for practical applications.

It is important to note that it sufficient to use the proportional form $p(\theta|y) \propto p(y|\theta)p(\theta)$ of the posterior to derive the normal approximation in (7.6) since the missing normalizing constant $c = 1/\int(p(y|\theta)p(\theta)d\theta)$ will become additive on the log scale

$$\log p(\theta|y) = \log c + \log p(y|\theta) + \log p(\theta)$$

and will therefore not affect the derivatives of the log posterior.

The normal approximation in (7.6) is based on the posterior information, i.e. the derivative of the log posterior, at the posterior mode. Since the likelihood will dominate the prior in large samples, we can also base a posterior approximation on the likelihood information $-\frac{\partial^2 \ln p(y|\theta)}{\partial \theta^2}|_{\theta=\hat{\theta}}$ at the maximum likelihood estimate (MLE) $\hat{\theta}$. In large samples, the two approximations will be very close.

NORMAL APPROXIMATION OF A GAMMA POSTERIOR. Consider the iid Poisson model $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} \text{Pois}(\theta)$ with the conjugate $\theta \sim \text{Gamma}(\alpha, \beta)$ prior. Here we know that the posterior is the $\text{Gamma}(\alpha + \sum y_i, \beta + n)$ distribution, so there is no need to approximate it. Let us however as an exercise derive a normal approximation of this posterior and compare it with the exact posterior. The log posterior density is

$$\log p(\theta|x) \propto (\alpha + \sum x_i - 1) \log \theta - \theta(\beta + n)$$

with first derivative

$$\frac{\partial \log p(\theta|x)}{\partial \theta} = \frac{\alpha + \sum x_i - 1}{\theta} - (\beta + n).$$

Setting the first derivative to zero and solving for θ give the posterior mode

$$\tilde{\theta} = \frac{\alpha + \sum x_i - 1}{\beta + n}. \quad (7.7)$$

The second derivative at the mode $\tilde{\theta}$ is

$$\frac{\partial^2 \ln p(\theta|\mathbf{x})}{\partial \theta^2} \Big|_{\theta=\tilde{\theta}} = -\frac{\alpha + \sum x_i - 1}{\left(\frac{\alpha + \sum x_i - 1}{\beta + n}\right)^2} = -\frac{(\beta + n)^2}{\alpha + \sum x_i - 1},$$

which is negative for all θ if $\alpha + \sum x_i > 1$, i.e. if at least one observation is non-zero or if $\alpha > 1$, so this is essentially always satisfied; hence $\tilde{\theta}$ in (7.7) is really the mode.

In summary, the normal posterior approximation is

$$\theta|\mathbf{x} \sim N\left(\frac{\alpha + \sum x_i - 1}{\beta + n}, \frac{(\beta + n)^2}{\alpha + \sum x_i - 1}\right).$$

Figure 7.3 displays the normal approximation and the true posterior for the number of bidders in the eBay data with increasing sample sizes from the first $n = 5$ observations in the data to the first $n = 50$ observations. The normal approximation is crude for the smallest sample size, but already with $n = 10$ it is rather accurate and at $n = 50$ it is nearly perfect.

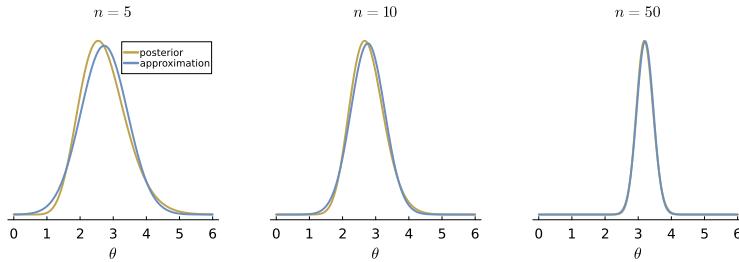


Figure 7.3: Normal approximation of the Gamma posterior in the Poisson model for the number of bidders in the eBay data for different sample sizes.

Normal posterior approximation

The posterior can in large samples be approximated by

$$\theta|\mathbf{y} \stackrel{a}{\sim} N(\tilde{\theta}, J_y^{-1}(\tilde{\theta}))$$

where $\tilde{\theta}$ is the posterior mode and

$$J_y(\tilde{\theta}) = -\frac{\partial^2 \ln p(\mathbf{y}|\theta)p(\theta)}{\partial \theta \partial \theta^\top} \Big|_{\theta=\tilde{\theta}}$$

is the $d \times d$ observed posterior information matrix at $\tilde{\theta}$.

Box 7.1: Multivariate normal approximation of a posterior distribution.

The posterior normality in large samples in Theorem 2 also holds when the model parameter is a vector, hence motivating the multi-

variate normal approximation of the posterior for a d -dimensional parameter vector θ in Box 7.1. Here is a two-dimensional example.

BETA DISTRIBUTION AS A MODEL FOR PROPORTIONS. Let us return to the Beta distribution as a model for proportions

$$X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{Beta}(\alpha, \beta).$$

We will derive a bivariate normal approximation to the posterior distribution $p(\alpha, \beta | \mathbf{x})$, following the recipe in Box 7.1. We therefore need to compute the posterior mode and the observed information matrix at the mode, and our first business is therefore to obtain the gradient (vector of first derivatives) and Hessian (matrix of second and cross derivatives) of the log likelihood. From (7.1) the log-likelihood $\log p(\mathbf{x} | \alpha, \beta)$ function is

$$-n \log B(\alpha, \beta) + (\alpha - 1) \sum_{i=1}^n \log x_i + (\beta - 1) \sum_{i=1}^n \log(1 - x_i), \quad (7.8)$$

recalling that the Beta function is $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$. Now, defining $\theta = (\alpha, \beta)^\top$, the gradient of the log-likelihood $\partial \log p(\mathbf{x} | \theta) / \partial \theta$ is the 2-dimensional vector

$$\begin{pmatrix} \frac{\partial \log p(\mathbf{x} | \alpha, \beta)}{\partial \alpha} \\ \frac{\partial \log p(\mathbf{x} | \alpha, \beta)}{\partial \beta} \end{pmatrix} = \begin{pmatrix} -n(\psi^{(0)}(\alpha) - \psi^{(0)}(\alpha + \beta)) + \sum_{i=1}^n \log x_i \\ -n(\psi^{(0)}(\beta) - \psi^{(0)}(\alpha + \beta)) + \sum_{i=1}^n \log(1 - x_i) \end{pmatrix}$$

where $\psi^{(0)}(z) \equiv \frac{d}{dz} \log \Gamma(z)$ is the **digamma function** which is available in most statistical software. Setting the gradient to zero gives us a system of two nonlinear equations in α and β that can be solved numerically to obtain the posterior mode $\tilde{\theta} = (\tilde{\alpha}, \tilde{\beta})^\top$. Finally, the Hessian matrix is the 2×2 matrix

$$\begin{aligned} \frac{\partial^2 \ln p(\mathbf{x} | \theta) p(\theta)}{\partial \theta \partial \theta^\top} &= \begin{pmatrix} \frac{\partial^2 \log p(\mathbf{x} | \alpha, \beta)}{\partial \alpha^2} & \frac{\partial^2 \log p(\mathbf{x} | \alpha, \beta)}{\partial \alpha \partial \beta} \\ \frac{\partial^2 \log p(\mathbf{x} | \alpha, \beta)}{\partial \alpha \partial \beta} & \frac{\partial^2 \log p(\mathbf{x} | \alpha, \beta)}{\partial \beta^2} \end{pmatrix} \\ &= -n \begin{pmatrix} \psi^{(1)}(\alpha) - \psi^{(1)}(\alpha + \beta) & -\psi^{(1)}(\alpha + \beta) \\ -\psi^{(1)}(\alpha + \beta) & \psi^{(1)}(\beta) - \psi^{(1)}(\alpha + \beta) \end{pmatrix} \end{aligned}$$

where $\psi^{(1)}(z) \equiv \frac{d^2}{dz^2} \log \Gamma(z)$ is the **trigamma function**. The cryptic superscripts on the digamma and trigamma functions comes from them being special cases of the **polygamma function** of order k , defined as $\psi^{(k)}(z) \equiv \frac{\partial^{k+1}}{\partial z^{k+1}} \log \Gamma(z)$, which is also a common name used in software.

We illustrate the normal posterior approximation by analyzing the Firm Leverage dataset in Rajan and Zingales (1995) containing the proportion leverage = totalDebt/(totalDebt + equity) for 4405

digamma function

trigamma function

polygamma function

American non-financial firms in the year 1992. We use an exponential prior with rate $\lambda = 1$ for both α and β . The left panel in Figure 7.4 shows the true posterior computed on a two-dimensional grid of α and β values, and the right panel shows that the normal approximation is highly accurate. The left panel of Figure 7.5 shows that the model fits the data well, except possibly at the very smallest proportions. The right panel of Figure 7.5 displays the posterior distribution of the mean proportion $E(X) = \alpha/(\alpha + \beta)$ in the Beta model, computed by simulating 10,000 draws from the bivariate normal posterior approximation of $p(\alpha, \beta|x)$ and computing $\alpha/(\alpha + \beta)$ for each draw.

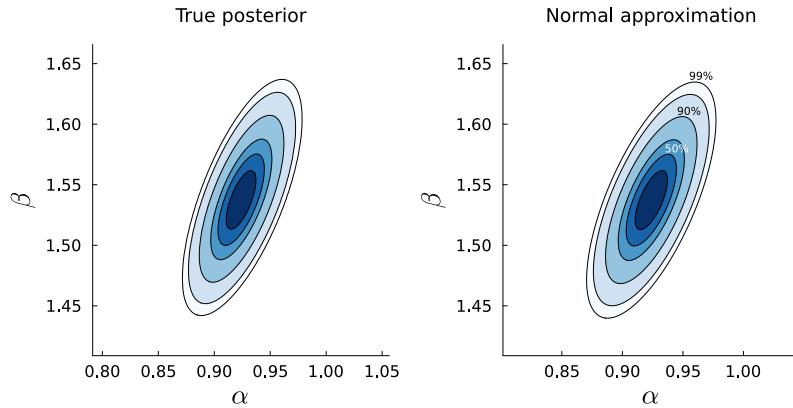
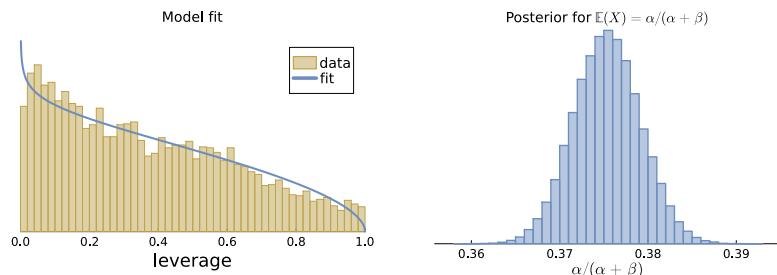


Figure 7.4: Contour plot of posterior $p(\alpha, \beta|x)$ in the Firm Leverage dataset computed over a two-dimensional grid (left) and its normal approximation (right).



APPROXIMATING THE POSTERIOR IN A STUDENT- t MODEL. This example gives an illustration where one of the conditions of the Bernstein-von Mises theorem is violated and the posterior is not asymptotically normal.

Let us first start with an example where asymptotic normality holds. Consider data coming from a standard student- t distribution with $\nu = 4$ degrees of freedom, i.e. $y_1, \dots, y_n \stackrel{\text{iid}}{\sim} t(0, 1, \nu = 4)$. The same issues will appear for models with unknown location and scale in the student- t distribution. We will fit the model $y_1, \dots, y_n | \nu \stackrel{\text{iid}}{\sim}$

Figure 7.5: Bayesian inference in the Firm Leverage dataset. Left: Histogram of the variable leverage and the predictive density from the Beta model with the parameters α and β integrated out with the normal posterior approximation. Right: Posterior distribution of the mean proportion $E(X) = \alpha/(\alpha + \beta)$ in the Beta model approximated by sampling 10,000 from the normal approximation of the parameter $p(\alpha, \beta|x)$ and computing $\alpha/(\alpha + \beta)$ for each draw.

$t(0, 1, \nu_n)$ to the data. The posterior $p(\nu | y_1, \dots, y_n)$ is intractable for any prior. We will first use a non-informative uniform prior over $\nu \in (0, \infty)$ to expose problems in the likelihood for this model, and then add a more informative prior. Figure 7.6 shows that when the data comes from a $t(0, 1, \nu = 4)$ distribution, the normal posterior approximation improves as we increase the sample size, as suggested by the Bernstein-von Mises theorem.

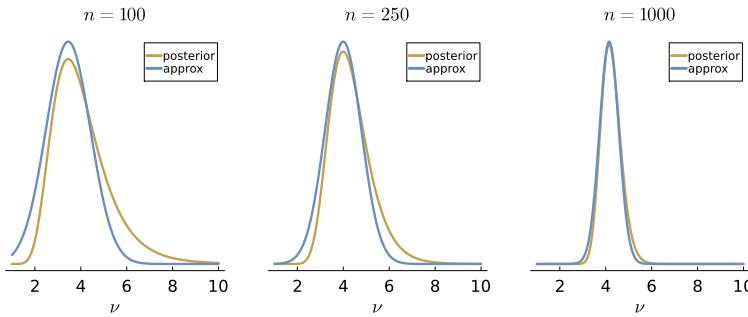


Figure 7.6: Normal posterior approximation for the degrees of freedom in the $t(0, 1, \nu)$ model fitted to iid data from the $t(0, 1, \nu = 4)$ distribution. The posterior clearly tends toward normality as the sample size, n , increases.

Assume now that the data is generated from a $N(0, 1)$ distribution, but we still fit a $t(0, 1, \nu)$ model to the data. Note that the $N(0, 1)$ data generating process is a student- t distribution where $\nu \rightarrow \infty$. Figure 7.7 shows that the normal posterior approximation is a disaster here. The reason is that the likelihood is essentially flat for all $\nu > 50$ since, for example, the $t(0, 1, \nu = 50)$ and $t(0, 1, \nu = 100)$ distributions are more or less identical models as both are very close to the Normal model, i.e. $\nu \rightarrow \infty$. The problem here is that the true parameter value ($\nu = \infty$) is at the boundary of the parameter space and all posterior mass will therefore ‘pile up at infinity’ for large sample sizes. This violates the so called steepness assumption needed for the Bernstein-von Mises theorem (see [Bernardo and Smith \(2009\)](#)), and the posterior does not tend to a normal distribution in large samples.

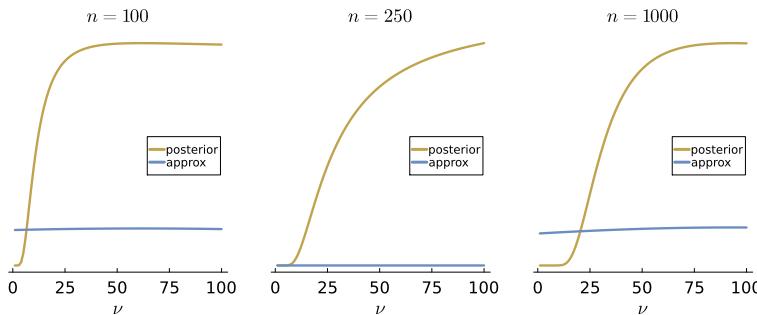
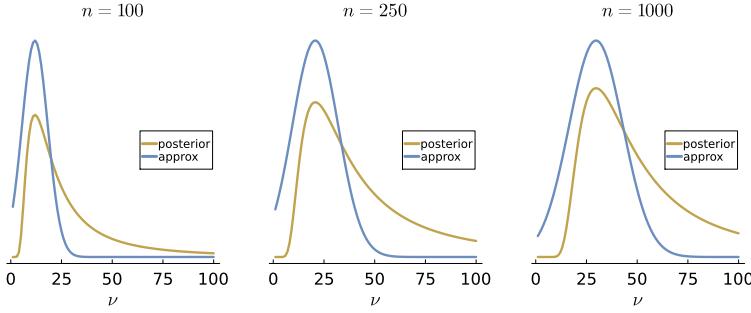


Figure 7.8 shows the posterior when using a $\nu \sim \text{Inv-}\chi^2(5, 10)$ prior. The prior adds curvature to flat regions of the likelihood, but

Figure 7.7: Normal posterior approximation for the degrees of freedom in the $t(0, 1, \nu)$ model fitted to iid data from the $N(0, 1)$ distribution. The prior for ν is uniform over $(0, \infty)$ to highlight properties of the likelihood. The posterior does not tend to normality even at large sample size.



even for $n = 1000$ is the normal approximation very poor. Moreover, the prior will be dominated by the likelihood as the sample size increases further, so the same problems we saw when using the flat prior will reappear as we add more and more data.

7.4 Computing the normal posterior approximation numerically

Computing the matrix of second derivatives needed in the normal approximation in Box 7.1 can be tedious. Moreover, the posterior mode may not be available in closed form since the system of equations

$$\frac{\partial \log p(\theta|y)}{\partial \theta} \Big|_{\theta=\tilde{\theta}} = 0 \quad (7.9)$$

is often non-linear without analytical solution. This is for example the case in the logistic regression model.

We will now explain how a computer with numerical optimization routines can be used to automatically find the posterior mode $\tilde{\theta}$ and the observed information matrix. It will be sufficient to code up the log-likelihood function and the log prior, and then let the computer do all the tedious differentiation and equation solving.

First, we can use Newton's method to solve the system of equations in 7.9 for the posterior mode. **Newton's method** starts with an initial value $\theta^{(0)}$ and iterates for $t = 1, 2, \dots$ until convergence:

$$\theta^{(t)} = \theta^{(t-1)} - H(\theta^{(t-1)})^{-1} g(\theta^{(t-1)}),$$

where $g(\theta^{(t-1)})$ is the gradient and $H(\theta^{(t-1)})$ the Hessian matrix of $\log p(\theta|y)$ at the previous parameter value $\theta^{(t-1)}$.

Newton's method requires the gradient and Hessian of the log posterior, which can be obtained in most modern programming languages by **automatic differentiation**. Automatic differentiation is a technique that applies the chain rule for differentiation from Calculus in clever algorithmic ways to produce derivatives that are both numerically accurate and fast to compute, even for functions with many inputs. Alternatively, there are optimizers like the BFGS algorithm

Figure 7.8: Normal posterior approximation for the degrees of freedom in the $t(0, 1, \nu)$ model fitted to iid data from the $N(0, 1)$ distribution. The prior $\nu \sim \text{Inv-}\chi^2(5, 10)$ gives some curvature to the posterior, but the normal approximation is poor even at $n = 1000$.

Newton's method

automatic differentiation

that returns both the posterior mode $\tilde{\theta}$ and the Hessian (observed information), where the Hessian matrix is iteratively built up during the iterations of the algorithm. The bottom line is that there are many ways to obtain all we need for the normal approximation in Box 7.1 using a computer. All we need to code is a function that computes $\log p(\mathbf{y}|\theta) + \log p(\theta)$ for any value of θ for a fixed dataset \mathbf{y} . Note again that we only need to code the proportional form of Bayes' theorem (on the log scale), since the normalizing constant does not depend on θ and will therefore not affect the optimization for the posterior mode or the observed information matrix. Box ?? shows the Julia code for the numerical approximation of the joint posterior for the Beta model $p(\alpha, \beta|\mathbf{x})$ for the variable $\mathbf{x}=\text{leverage}$ in the Firm Leverage dataset.

```
# Setting up the log posterior function
function logPostBeta(α, β, x, α₀, β₀)
    logLik = sum(logpdf(Beta(α, β), x))
    logPrior = logpdf(Exponential(α₀), α) + logpdf(Exponential(β₀), β)
    return logLik + logPrior
end

# Normal approximation using numerical optimization and autodiff
α₀ = β₀ = 1
θ₀ = [1.0, 1.0]
logpost(θ) = logPostBeta(θ[1], θ[2], x, α₀, β₀)
optres = maximize(logpost, θ₀, autodiff = :forward)
θ̂ = Optim.maximizer(optres)
H(θ) = ForwardDiff.hessian(logpost, θ)
Ω = Symmetric(-inv(H(θ̂)))
```

Box 7.2: Julia code for the log posterior function used to approximate the joint posterior for the Beta model $p(\alpha, \beta|\mathbf{x})$ with a multivariate normal distribution $\theta \sim N(\tilde{\theta}, \Omega)$, where $\theta = (\alpha, \beta)^T$, $\tilde{\theta}$ is the posterior mode, $\Omega = -H^{-1}$ and H is the 2×2 Hessian matrix evaluated at $\tilde{\theta}$, computed with automatic differentiation using the ForwardDiff.jl package.

7.5 Reparametrization

It is often useful to approximate the posterior in another parameterization than the one used when expressing the model. Consider for example a model where the parameter θ is strictly positive, such as when $\theta = \sigma^2$ is a variance parameter. Instead of approximating the posterior distribution $p(\theta|\mathbf{x})$ by a normal distribution, we can reparametrize the model in terms of a new parameter $\phi = g(\theta)$, which is simply the original parameter transformed by the function $g(\theta)$. For a strictly positive parameter $\theta > 0$ we can choose the log

transformation $\phi = \log(\theta)$, where $-\infty < \phi < \infty$. In general we choose a transformation such that the new parameter is unrestricted. Transforming the parameter usually brings three advantages when approximating the posterior:

- the new parameter space for ϕ is unrestricted and therefore agrees with support of the normal approximate distribution,
- we can use algorithms for *unconstrained* optimization to find the posterior mode $\tilde{\phi}$,
- the posterior distribution $p(\phi|x)$ in the transformed space may be closer to normal compared to the original posterior $p(\theta|x)$.

Finding a suitable transformation is not always easy, but a good first choice is the log transformation $\phi = \log(\theta)$ for positive parameters, and the logit transformation $\phi = \log(\theta/(1 - \theta))$ for parameters θ in the unit interval $(0, 1)$.

Let us write $p_\theta(\theta|x) \propto p_\theta(x|\theta)p_\theta(\theta)$ for the posterior in the original parameterization, where we have used the more elaborate notation with subscripts to explicitly denote the parameter argument. The posterior in the new parameterization is similarly denoted by $p_\phi(\phi|x)$. Now, if the transformation $\phi = g(\theta)$ is monotone and continuously differentiable with inverse transformation $\theta = g^{-1}(\phi)$, then we can express the posterior for ϕ in terms of the original $p_\theta(\theta|x)$ posterior using the change-of-variable formula:

$$\begin{aligned} p_\phi(\phi|x) &= p_\theta(g^{-1}(\phi)|x) \left| \frac{dg^{-1}(\phi)}{d\phi} \right| \\ &\propto p_\theta(x|g^{-1}(\phi)) p_\theta(g^{-1}(\phi)) |x| \left| \frac{dg^{-1}(\phi)}{d\phi} \right|, \end{aligned}$$

where the last factor is the absolute value of the so called *Jacobian* of the transformation. Hence, the log posterior for ϕ can be expressed

$$\log p_\phi(\phi|x) \propto \log p_\theta(x|g^{-1}(\phi)) + \log p_\theta(g^{-1}(\phi)) + \log \left| \frac{dg^{-1}(\phi)}{d\phi} \right|,$$

where the proportionality \propto now means that we are missing an *additive* constant; this constant term is not important when approximating the posterior distribution since it will disappear when taking derivatives with respect to ϕ .

We can now perform the numerical optimization to obtain a normal approximation of the posterior $p_\phi(\phi|x)$ in exactly the same way as for θ , using our previous code for the likelihood $p_\theta(x|\theta)$ and prior $p_\theta(\theta)$ functions with respect to θ , we just need to evaluate those functions at $\theta = g^{-1}(\phi)$ and to remember to add the log of the absolute value of the Jacobian $\log \left| \frac{dg^{-1}(\phi)}{d\phi} \right|$ to the log posterior function.

NORMAL APPROXIMATION OF A GAMMA POSTERIOR - REPARAMETERIZATION

Let us return to the Gamma posterior approximation in the Poisson model for the number of bidders in the eBay data, this time using the log transformation $\phi = \log \theta$ to reparametrize the model. Note that the inverse transformation is $\theta = g^{-1}(\phi) = \exp(\phi)$ and the absolute value of the Jacobian is $|\frac{dg^{-1}(\phi)}{d\phi}| = \exp(\phi)$. The log posterior density for ϕ is

$$\log p(\phi|x) \propto \left(\alpha + \sum_{i=1}^n x_i - 1 \right) \phi - \exp(\phi)(\beta + n) + \phi,$$

where the last term ϕ is the log absolute Jacobian. The first derivative is

$$\frac{\partial \log p(\phi|x)}{\partial \phi} = \left(\alpha + \sum_{i=1}^n x_i - 1 \right) - \exp(\phi)(\beta + n) + 1.$$

Setting the first derivative to zero and solving for ϕ gives the posterior mode

$$\tilde{\phi} = \log \left(\frac{\alpha + \sum_{i=1}^n x_i}{\beta + n} \right). \quad (7.10)$$

The second derivative at the mode $\tilde{\phi}$ is

$$\frac{\partial^2 \ln p(\phi|x)}{\partial \phi^2} \Big|_{\phi=\tilde{\phi}} = -\exp(\tilde{\phi})(\beta + n) = -\left(\alpha + \sum_{i=1}^n x_i \right),$$

which is negative for all ϕ if $\alpha > 0$, so $\tilde{\phi}$ is really the mode. In summary, the normal posterior approximation for ϕ is

$$\phi|x \sim N \left(\log \frac{\alpha + \sum_{i=1}^n x_i}{\beta + n}, \frac{1}{\alpha + \sum_{i=1}^n x_i} \right).$$

This can directly converted back to the original $\theta = \exp(\phi)$ scale since we know that if a random variable X is normally distributed $X \sim N(\mu, \sigma^2)$, then $Y = \exp(X)$ is log-normally distributed $Y \sim \text{LogNormal}(\mu, \sigma^2)$. Hence, we have implicitly derived a Log-Normal approximation for the posterior of θ

$$\theta|x \sim \text{LogNormal} \left(\log \frac{\alpha + \sum_{i=1}^n x_i}{\beta + n}, \frac{1}{\alpha + \sum_{i=1}^n x_i} \right).$$

BETA MODEL FOR PROPORTIONS - REPARAMETERIZATION

Let us return to the Beta model for the proportion `leverage=totalDebt/(totalDebt + equity)` in the Firm Leverage data, this time using a log transformation of both parameters $\phi_1 = \log \alpha$ and $\phi_2 = \log \beta$ to reparametrize the model. This is a bivariate transformation $\boldsymbol{\phi} = \mathbf{g}(\boldsymbol{\theta}) = (g_1(\theta_1), g_2(\theta_2))^\top$,

where $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top = (\alpha, \beta)^\top$

$$\begin{aligned}\phi_1 &= \log \theta_1 \\ \phi_2 &= \log \theta_2\end{aligned}\tag{7.11}$$

with inverse transformation $\boldsymbol{\theta} = g^{-1}(\boldsymbol{\phi})$ given by the two equations

$$\begin{aligned}\theta_1 &= \exp(\phi_1) \\ \theta_2 &= \exp(\phi_2).\end{aligned}\tag{7.12}$$

We need a multivariate change-of-variables formula, which is given in Box 7.3. The derivative of the inverse transformation in the univariate case is now replaced by a Jacobian matrix in the multivariate case

$$\frac{\partial g^{-1}(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \begin{pmatrix} \frac{\partial \theta_1}{\partial \phi_1} & \frac{\partial \theta_1}{\partial \phi_2} \\ \frac{\partial \theta_2}{\partial \phi_1} & \frac{\partial \theta_2}{\partial \phi_2} \end{pmatrix} = \begin{pmatrix} \exp(\phi_1) & 0 \\ 0 & \exp(\phi_2) \end{pmatrix}.$$

Since this Jacobian matrix is a diagonal matrix, the determinant is simply the product of the diagonal elements: $\exp(\phi_1) \exp(\phi_2) = \exp(\phi_1 + \phi_2)$. Hence, the log absolute Jacobian is $\phi_1 + \phi_2$. Note that the Jacobian matrix will always be a diagonal matrix whenever each parameter in the model is transformed separately, which is often the case. Box ?? shows how the Julia code for the log posterior function `logPostBeta` from the original parameterization in Box ?? can be re-used by applying the inverse transformation and adding the log absolute Jacobian $\phi_1 + \phi_2$.

```
# Normal approximation using numerical optimization and autodiff
α₀ = β₀ = 1
φ₀ = [0.0, 0.0]
logpost(ϕ) = logPostBeta(exp(ϕ[1]), exp(ϕ[2]), x, α₀, β₀) + sum(ϕ)
optres = maximize(logpost, φ₀, autodiff = :forward)
ξ = Optim.maximizer(optres)
H(ϕ) = ForwardDiff.hessian(logpost, ϕ)
Ω_ϕ = Symmetric(-inv(H(ξ)))
```

Transforming variables - multivariate

Let $\mathbf{X} \sim p_x(x)$ be a continuous multivariate random vector and $\mathbf{Y} = g(\mathbf{X})$, where $g(\cdot)$ is a multidimensional one-to-one continuously differentiable transformation with inverse $\mathbf{x} = g^{-1}(\mathbf{y})$. The density of \mathbf{Y} is then

$$p_y(y) = p_x(g^{-1}(y)) \left| \frac{\partial}{\partial y} g^{-1}(y) \right|,$$

where $\frac{\partial}{\partial y} g^{-1}(y)$ is the Jacobian matrix and $|A|$ denotes the absolute value of the determinant of the matrix A .

Box 7.3: Transformation of multivariate random variables.

Box 7.4: Julia code for the reparametrized log posterior function used to approximate the joint posterior for the Beta model $p(\alpha, \beta | \mathbf{x})$ with a multivariate normal distribution. Note how the log posterior function `logPostBeta` from Box ?? is re-used by immediately inserting the inverse transformations and then adding the log absolute Jacobian $\phi_1 + \phi_2$.

EXERCISES 7.1

1. The wind direction was measured once a month at a given location. The measurements for the first ten months were

$$\mathbf{y} = (-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02),$$

recorded in radians $-\pi \leq y_i \leq \pi$ with South located at zero radians; see Figure 7.5. Assume that these data points are independent observations following the **von Mises** distribution for directional data (see Box 7.5)

$$y_1, \dots, y_n | \mu, \kappa \stackrel{\text{iid}}{\sim} \text{VM}(\mu, \kappa).$$

- (a) Assume that μ is known to be 2.39, and $\kappa \sim \text{Expon}(\theta = 1)$ a priori. Plot the posterior distribution of κ over a fine grid of κ values.
- (b) Use numerical optimization to approximate the posterior distribution of κ and plot the approximation in the same graph.
- (c) Assume now that both μ and κ are unknown. Plot the bivariate posterior $p(\mu, \kappa | \mathbf{y})$ over a two-dimensional grid of (μ, κ) pairs.
- (d) Use numerical optimization to approximate the bivariate posterior distribution $p(\mu, \kappa | \mathbf{y})$.

2. Next problem!

von Mises

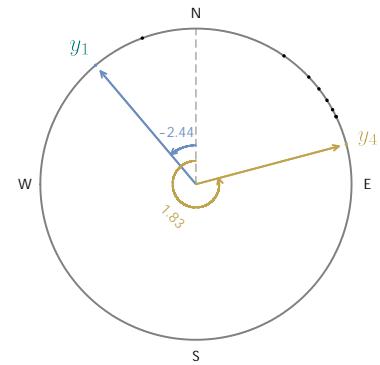


Figure 7.9: Wind direction data in radians $y \in [-\pi, \pi]$, with South at $y = 0$ radians.

[0.0cm]

Von Mises distribution

$$X \sim \text{VM}(\mu, \kappa) \text{ for } X \in [-\pi, \pi]$$

is a common distribution for directional data.

$$p(x) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}$$

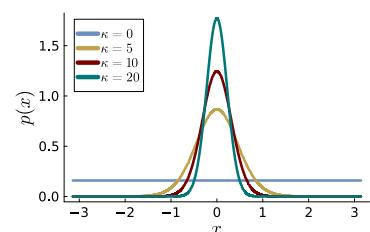
$$\mathbb{E}(X) = \mu$$

$$\mathbb{V}(X) = 1 - I_1(\kappa)/I_0(\kappa),$$

where $I_\nu(\kappa)$ is the modified Bessel function of the first kind of order ν , implemented as `besseli` in many programming languages.

The variance can be shown to be decreasing in κ , so κ is the precision of the distribution.

Box 7.5: Von Mises distribution.



8 Classification and Generalized regression

Chapter 5 presented Bayesian learning and prediction for a *continuous* response variable y explained by a set of covariates \mathbf{x} ; the covariates may be binary, categorical or continuous. Classification is instead when we are modeling a *binary* or *categorical response variable* as a function of covariates \mathbf{x} . This distinction in terminology between regression and classification is not always upheld, for example one of the more commonly used classification models is named logistic regression.

Modeling non-continuous response variables requires other models than the Gaussian linear regression model, and this chapter will present Bayesian learning, prediction and decision making for such models.

The likelihood function can be complex in classification problems, and the posterior distribution is therefore often mathematically intractable. We will therefore use the normal posterior approximation presented in Chapter 7, and later chapters will present simulation-based methods to explore the posterior in models used for classification.

8.1 Classification problems

Binary classification

Many interesting problems involve modeling and predicting a **binary response variable** $y \in \{0,1\}$. The coding of the two different values need not be 0/1, but can equally well be True/False, or something application specific such as Heads/Tails in coin tossing; the coding $y \in \{-1,1\}$ is common in machine learning. A binary variable is said to have two possible **classes**, for example the two classes Heads and Tails in coin tossing. It is also common to distinguish the two classes by the generic labels **positive class** and **negative class**, where positive does not necessarily mean positive in the usual sense, but may for example indicate the presence of a disease. Here are some examples of binary classification problems.

binary response variable

classes

positive class

negative class

EXAMPLE: SPAM PREDICTION. You want to build a spam filter that can determine if a newly arrived email is **spam** (perhaps coded as $y = 1$) or **ham** (coded as $y = 0$). The spam prediction can use covariates based on the processed text in the email. For example, dummy variables that indicate the presence of certain trigger words for spam, or covariates based on the number of \$-signs or CAPITAL LETTERS in the given email; see Figure 8.1.

EXAMPLE: INTERNET AUCTION SALE. What determines if an internet auction ends up in a sale? This can be analyzed by collecting data on many past auctions and recording the response variable **sold** ($y = 1$) and **not sold** ($y = 0$) for each auction. To aid in the classification one can also collect information (covariates) about each auction, for example the seller's reservation price, the feedback/review score of the seller, and measures of the auctioned object's condition determined from the seller's text description, or visual inspection of the posted images by a human.

The aim in binary classification problems is the probability of the positive class, $\Pr(y = 1|\mathbf{x})$, conditional on a set of covariates \mathbf{x} . We then of course immediately get the probability of the negative class $\Pr(y = 0|\mathbf{x}) = 1 - \Pr(y = 1|\mathbf{x})$. The importance of $\Pr(y = 1|\mathbf{x})$ in prediction problems should be clear: computing $\Pr(y = 1|\tilde{\mathbf{x}})$ for a new observation $\tilde{\mathbf{x}}$ gives the probability of the positive class, which can be directly used for Bayesian decision making. For example, consider a planned auction where we can use the reservation price, seller and object information as covariates to compute the probability that the object will be sold. This probability can be used by the seller to make a decision of whether or not to put the object up for auction, or for determining a more suitable reservation price that increases the sale probability.

Multi-class classification

Many problems involve more than two classes. **Multi-class classification** has a response variable y that belongs to exactly one of C possible classes or categories. We have seen such categorical data before in Section 3.5 where a Bayesian analysis of multinomial data with a Dirichlet prior was presented. Here we will model the class probabilities $\Pr(y = c|\mathbf{x})$, $c \in \{1, \dots, C\}$, conditional on a set of covariates \mathbf{x} .

EXAMPLE: MARKETING BRAND PREDICTION. Customers can often choose from several competing brands when shopping. Marketers

Dear Professor Villani
I am writing regarding your recently published paper 'A new approach to Bayesian statistics' which I read with much interest. I was particularly intrigued by the ...

Sir or Madam
I HAVE GRAT NEWS FOR YOU!!!
You win \$1,345,566,666 !!!
Please send **bank** details.

Figure 8.1: Indicators for spam in red text and ham in blue text.

Multi-class classification

can build multi-class classification models to predict the brand choice ($y \in \{1, \dots, C\}$) of a customer from covariates (\mathbf{x}) constructed from personal data (age, income, residence area, sex) and characteristics of the product (price, price of competing brands, placement).

EXAMPLE: IMAGE CLASSIFICATION. An image consists of a large number of pixels, where each pixel is color-coded according to some color system; for example RGB where each pixel is described by a three-dimensional vector with numbers ranging from 0–255. Each RGB vector gives the composition of red (first element), green (second element) and blue (third element) colors in the pixel. Consider a dataset of images where each observation is an image with a label (y) that describes the category of the image ("dog", "cat", "human", "car", "train" etc.) and three covariates for each pixel in the image. A self-driving car robot is using multi-class classification to determine the category of an object from camera images, and other sensors.

8.2 Logistic regression

We will use the notation $\Pr(y = y^* | \mathbf{x})$, to denote the probability that the binary class variable y takes the value $y^* \in \{0, 1\}$. **Logistic regression** assumes that the responses y_1, \dots, y_n are independent conditional on the covariates and the probability of the positive class is modelled by

$$\Pr(y = 1 | \mathbf{x}, \boldsymbol{\beta}) = \frac{\exp(\mathbf{x}^\top \boldsymbol{\beta})}{1 + \exp(\mathbf{x}^\top \boldsymbol{\beta})}, \quad (8.1)$$

where \mathbf{x} is a p -dimensional vector with covariates and $\boldsymbol{\beta}$ is the vector of regression coefficients. A common alternative form is obtained by multiplying both the numerator and denominator of (8.1) by $\exp(-\mathbf{x}^\top \boldsymbol{\beta})$ to obtain

$$\Pr(y = 1 | \mathbf{x}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(-\mathbf{x}^\top \boldsymbol{\beta})}. \quad (8.2)$$

The function $f(x) = 1/(1 + \exp(-x))$ is called the **logistic function** (see Figure 8.2), hence the name logistic regression. Logistic regression is similar to the usual linear regression in that the linear combination $\mathbf{x}^\top \boldsymbol{\beta}$ is the connection between the covariates \mathbf{x} and the response y . The role of the logistic function is to 'squash' $\mathbf{x}^\top \boldsymbol{\beta}$ so that the end result is a number between 0 and 1, which is required here since we are modeling a probability, $0 \leq \Pr(y = 1 | \mathbf{x}) \leq 1$.

There are many other squashing functions that can be used instead of the logistic, for example the distribution function (CDF) $\Phi(z)$ of the standard normal distribution, which gives rise to the popular **Probit regression** model

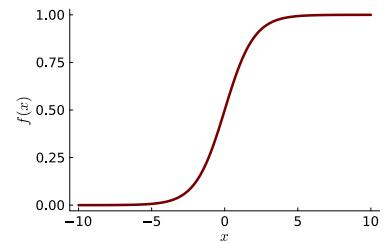


Figure 8.2: The logistic function $f(x) = 1/(1 + e^{-x})$.

Logistic regression

logistic function

Probit regression

$$\Pr(y = 1|\mathbf{x}, \boldsymbol{\beta}) = \Phi(\mathbf{x}^\top \boldsymbol{\beta}). \quad (8.3)$$

The fact that $\Phi(z)$ is a CDF guarantees that $0 \leq \Pr(Y = y|\mathbf{x}) \leq 1$. We will return to this model later in the book.

The parameters in the logistic regression model are most easily interpreted in odds form. To see this, note first that the complementary probability is

$$\Pr(y = 0|\mathbf{x}, \boldsymbol{\beta}) = 1 - \Pr(y = 1|\mathbf{x}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(\mathbf{x}^\top \boldsymbol{\beta})}, \quad (8.4)$$

and the odds of the positive class compared to the negative class is therefore

$$\text{Odds}(\mathbf{x}) \equiv \frac{\Pr(y = 1|\mathbf{x})}{\Pr(y = 0|\mathbf{x})} = \exp(\mathbf{x}^\top \boldsymbol{\beta}). \quad (8.5)$$

Consider now how changing the j th covariate by one unit affects the odds. In particular we will look at the **odds ratio** (OR) for the j th covariate

$$\text{OR}_j = \frac{\text{Odds}(\mathbf{x} = (x_1, \dots, x_j + 1, \dots, x_p))}{\text{Odds}(\mathbf{x} = (x_1, \dots, x_j, \dots, x_p))} = \exp(\beta_j). \quad (8.6)$$

The important fact about an odds ratio from logistic regression is that it does not depend on the value of the covariates \mathbf{x} . A value of $\exp(\beta_j)$ of 1.01 has the interpretation that the odds for the positive class increases by 1% whenever x_j increases by one unit, regardless of the value for the other covariates, or the value of x_j before the unit change. For this reason, it is common to report inferences for $\exp(\beta_j)$ rather than β_j .

Finally, note that log odds is a linear function of the covariates

$$\text{LogOdds}(\mathbf{x}) \equiv \log \frac{\Pr(y = 1|\mathbf{x})}{\Pr(y = 0|\mathbf{x})} = \mathbf{x}^\top \boldsymbol{\beta}. \quad (8.7)$$

The logistic regression is therefore often said to be a linear model, even though the probability of the response is clearly a nonlinear function of the covariates. One important implication of the linearity in the log odds is that the decision boundaries that separate the two classes are linear. The logistic regression model is therefore not suitable for classification problems where the classes are not linearly separable. The Gaussian process extension of the logistic regression presented in Chapter ?? is an interesting nonlinear alternative, with the drawback of having a more complex interpretation and more demanding numerical computations for obtaining the posterior distribution.

Bayesian inference for logistic regression

Assume that the response observations y_1, \dots, y_n are independent conditional on the covariates. As in the regression case, it is com-

mon to assume that the covariates are known. Define $\theta(\beta, \mathbf{x}) = 1/(1 + \exp(-\mathbf{x}^\top \beta))$ as the success probability for an observation with covariate vector \mathbf{x} . The likelihood function for the logistic regression is then a product of Bernoulli distributions

$$p(y_1, \dots, y_n | \mathbf{x}_1, \dots, \mathbf{x}_n, \beta) = \prod_{i=1}^n \theta(\beta, \mathbf{x}_i)^{y_i} (1 - \theta(\beta, \mathbf{x}_i))^{1-y_i}, \quad (8.8)$$

just like the likelihood for Bernoulli trials in Chapter 2. The difference is that the success probabilities $\theta(\beta, \mathbf{x}_i)$ are not constant here across observations, but instead vary with the covariates \mathbf{x}_i in a way determined by the logistic regression model.

Now that we have seen the connection to the likelihood for the Bernoulli model, let us write the likelihood for the logistic regression more compactly as

$$p(\mathbf{y} | \mathbf{X}, \beta) = \prod_{i=1}^n \left(\frac{\exp(\mathbf{x}_i^\top \beta)}{1 + \exp(\mathbf{x}_i^\top \beta)} \right)^{y_i} \left(\frac{1}{1 + \exp(\mathbf{x}_i^\top \beta)} \right)^{1-y_i} \quad (8.9)$$

$$= \exp \left(\sum_{i=1}^n y_i \mathbf{x}_i^\top \beta \right) \prod_{i=1}^n \left(\frac{1}{1 + \exp(\mathbf{x}_i^\top \beta)} \right). \quad (8.10)$$

The posterior distribution of β is then

$$p(\beta | \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} | \mathbf{X}, \beta) p(\beta), \quad (8.11)$$

for some prior distribution $p(\beta)$. Assuming for example a multivariate normal prior $\beta \sim N(\mu, \Omega)$, we obtain the posterior

$$\exp \left(\sum_{i=1}^n y_i \mathbf{x}_i^\top \beta \right) \prod_{i=1}^n \left(\frac{1}{1 + \exp(\mathbf{x}_i^\top \beta)} \right) \exp \left(-\frac{1}{2} (\beta - \mu)^\top \Omega^{-1} (\beta - \mu) \right). \quad (8.12)$$

Unfortunately, the posterior in (8.12) is not a multivariate normal distribution, or any other known distribution. The posterior distribution for the logistic regression is intractable.

A natural approach is to use the posterior approximation in Chapter 7. Figure 8.3 gives a complete code for obtaining the normal posterior approximation in Box 7.1 for the logistic regression model with a multivariate normal prior for β . Note that the log-likelihood of the logistic regression can be written

$$\log(\mathbf{y} | \beta, \mathbf{X}) = \sum_{i=1}^n y_i \mathbf{x}_i^\top \beta - \sum_{i=1}^n \log(1 + \exp(\mathbf{x}_i^\top \beta)).$$

The code in Figure 8.3 uses the maximum likelihood estimate as the initial value β_0 for β . This is a good choice for fast convergence of the iterative optimization algorithm. A more rough estimate, or even setting $\beta_0 = \mathbf{0}$, is often sufficient for convergence. The exceptions

```

# 0. Loading packages
using Plots, Distributions, GLM, LinearAlgebra, Optim, ForwardDiff

# 1. Setting up the log posterior function
"""
    logisticreg_logpost(β, y, X, μ, Σ)
log posterior for the logistic regression model
    Pr(y=1|x) = 1/(1 + exp(-x*β))
with the prior
    β ~ N(μ,Σ).
"""
function logisticreg_logpost(β, y, X, μ, Σ)
    loglik = sum( y.(X*β) .- log.(1 .+ exp.(X*β)) )
    logprior = logpdf(MvNormal(μ, Σ), β)
    return(loglik + logprior)
end

# 2. Generate data from logistic regression with β = [1,-1,1,-1]
n = 100
p = 4
X = [ones(n,1) randn(n,p-1)]
β = [1, -1, 1, -1]
probs = 1 ./ (1 .+ exp.(-X*β))
y = rand.(Bernoulli.(probs))

# 3. Set up prior
μ = zeros(p)
Σ = 10*I(p)

# 4. Initial value for the optimization
glmfit = glm(X, y, Bernoulli(), LogitLink()) # find MLE.
β₀ = coef(glmfit) # initial values from MLE.

# 5. Run optimizer with automatic differentiation to find mode and Hessian.
optres = maximize(β -> logisticreg_logpost(β, y, X, μ, Σ), β₀, autodiff = :forward)
βmode = Optim.optimizer(optres)

# 6. Compute Hessian to get posterior covariance matrix approximation
H(β) = ForwardDiff.hessian(β -> logisticreg_logpost(β, y, X, μ, Σ), β)
Ω_β = Symmetric(-inv(H(βmode))) # This is J^{-1}

# 7. Simulate from normal posterior approximation and compute odds ratios
βsim = rand(MvNormal(βmode, Ω_β), 10000)'
oddsratio = exp.(βsim) # 10000 × 4 matrix with draws of exp(β_j) in jth column.

```

Figure 8.3: Numerical optimization to find normal posterior approximation for the logistic regression model in Julia. The broadcasting operator in Julia is denoted by the dot (.) so that $\exp(x)$ is the vector that applies the exponential function to each element of the vector x . Similarly, $a .- b$ is the elementwise difference of the two vectors a and b . The Bernoulli distribution is computed using the `logpdf` function from the `Distributions.jl` package. The `optimize` function is from the `Optim.jl` package.

are when the log posterior is a complex surface with multiple modes, extended ridges or other difficulties. Note that the gradient vector

$$\frac{\partial \log[p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})p(\boldsymbol{\beta})]}{\partial \boldsymbol{\beta}}$$

and Hessian matrix

$$\frac{\partial^2 \log[p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})p(\boldsymbol{\beta})]}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}$$

are computed by automatic differentiation in Figure 8.3 using the Julia package `ForwardDiff.jl`. The gradient and Hessian for the logistic regression model are actually quite simple to derive and can be given as explicit arguments to the `maximize` function. We nevertheless use automatic differentiation in the code to illustrate the generality of the normal approximation approach in Figure 8.3 also when the gradient and Hessian are much more tedious to derive. Note also that we only need to implement a function that computes the log-likelihood and log prior, the rest is handled by the software.

The last two lines of code in Figure 8.3 illustrates the important point that the normal approximation is easy to simulate from, and the generated posterior draws can be used to compute the posterior distribution of any transformation of the parameters, exactly as we did in Chapter 3. Similarly, the posterior draws can be used to simulate from the predictive distribution, as we did in Chapter 6. Given that the posterior approximation is accurate enough, the normal approximation obtained by numerical optimization followed by posterior simulation from the approximate posterior is clearly a general and highly useful approach to Bayesian learning, prediction and decision making.

APPLIED LOGISTIC REGRESSION - WHO SURVIVED THE TITANIC?

On April 15, 1912, the RMS Titanic sank on her maiden voyage after colliding with an iceberg. The catastrophe resulted in the death of 1502 persons among the 2224 persons onboard. The `titanic` dataset is a subset of the `titanic dataset` in the Kaggle repository¹, with missing values imputed by regression models with the other variables as covariates. The dataset consists of 887 passengers out of which 342 persons survived. Several variables such as age, ticket class and number of relatives are available to explain the binary response `Survived`. Table 8.1 gives a summary of the data. We will here model the survival probability with a logistic regression on an intercept and the three covariates: `age`, `sex` and `class`; the `class` variable is turned into a binary variable with `class=1` for first class and `class=0` for second and third class.

Let the prior be $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Omega})$. To determine suitable values for $\boldsymbol{\mu}$ and $\boldsymbol{\Omega}$, let us derive the implied prior for the more interpretable

`titanic dataset`

¹ <https://www.kaggle.com/c/titanic/>

variable	description	data type	values	comment
survived	survived	binary	{0,1}	survived=1 for survived
class	ticket class	ordinal	{1,2,3}	class=1 for first class
sex	sex	binary	{0,1}	sex=1 for females
age	age	continuous	[0,∞]	range = [0.42, 80]
sibling/spouse	number aboard	counts	{0,1,...}	range = [0,8]
parent/child	number aboard	counts	{0,1,...}	range = [0,6]
fare	fare in \$	continuous	[0,∞]	range = [14.45, 512.33]

Table 8.1: Summary of the titanic data.

survival odds $\Pr(y = 1|x)/\Pr(y = 0|x)$, which we have seen is $\exp(x^\top \beta)$ in the logistic regression. If $\beta \sim N(\mu, \Omega)$, then $x^\top \beta \sim N(x^\top \mu, x^\top \Omega x)$, which means that $\exp(x^\top \mu)$ follows the log-normal distribution

$$\exp(x^\top \beta) \sim LN(x^\top \mu, x^\top \Omega x).$$

This means in particular that the prior median for the survival odds is $\exp(x^\top \mu)$. We will here set

$$\mu = (-1, -1/80, 1, 1)^\top.$$

The prior mean for the intercept $\mu_1 = -1$ was chosen so that survival probability of $\Pr(y = 1|x = 0) \approx 0.269$ was deemed reasonable for newborn (age=0) boy (sex=0) not traveling in first class (class=0).

The prior mean for the coefficient on age $\mu_2 = -1/80$ implies that the survival odds decrease with a multiplicative factor of $\exp(-1/80) \approx 0.988$ for each year so that the survival odds of for example an 80-year-old is roughly a third of a newborn's odds ($\exp(-80/80) \approx 0.368$). The prior means for sex and class are set so that both of these factors increase the survival odds by a factor $\exp(1) \approx 2.718$.

It remains to determine the prior variance around the mean. We will use

$$\Omega = \begin{pmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 1/(80^2) & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

which assumes prior independence between the elements in β , for simplicity. The prior variances are chosen so that the implied prior distributions for the survival odds of some selected ages, the variable sex and the variable class agree with my prior beliefs, see Figure 8.4.

Figure 8.5 shows the marginal posteriors for the elements in β from the normal approximation as dark blue lines. Note that these marginal posteriors are for the survival odds $\exp(\beta_j)$, which follow a log-normal distribution when the posterior for β is approximated by a multivariate normal distribution. The histograms in Figure 8.5 are from 100,000 posterior draws using the Hamiltonian Monte Carlo (HMC) method presented later in Chapter 10. The histograms from

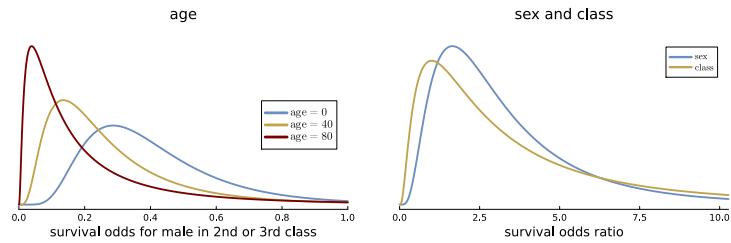


Figure 8.4: Implied prior distributions for the titanic data.

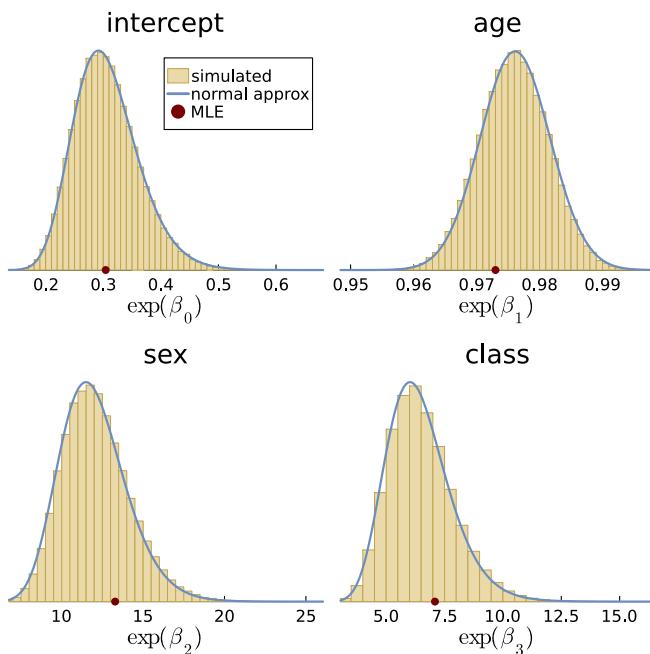


Figure 8.5: Marginal posterior distributions of the odds ratios from the normal approximate posterior for β compared to 100,000 draws simulated from the true posterior using the Hamiltonian Monte Carlo method.

HMC sampling can for practical purposes be taken to represent the exact posterior without approximation, and Figure 8.5 therefore shows that the normal approximation for β , or equivalently, the log-normal approximation for $\exp(\beta_j)$ is extremely accurate here. The maximum likelihood (MLE) estimates are shown as reference.

To investigate how robust the posterior is to changes in the prior, Figure 8.6 compares the above results to those from a noninformative regularization prior $\beta \sim N(\mathbf{0}, 10^2 I_p)$, where I_p is the $p \times p$ identity matrix; see Chapter 12 for more on regularization priors. Since the dataset is only moderately large, the informative prior has some effect on the posterior, although not excessive.

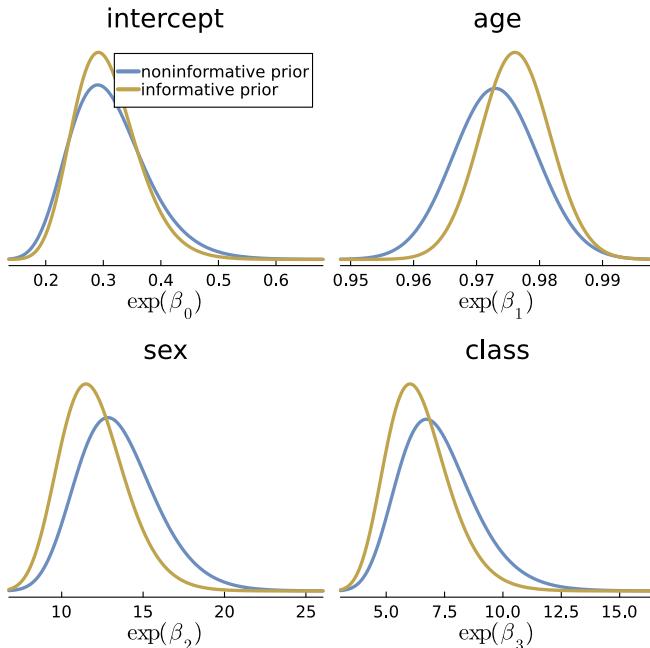


Figure 8.6: Marginal posterior distributions of the odds ratios from the normal approximate posterior for β using the informative prior. The marginal posteriors from the noninformative prior are given as a reference.

8.3 Multi-class logistic regression

The direct extension of the logistic regression model to the multi-class case is

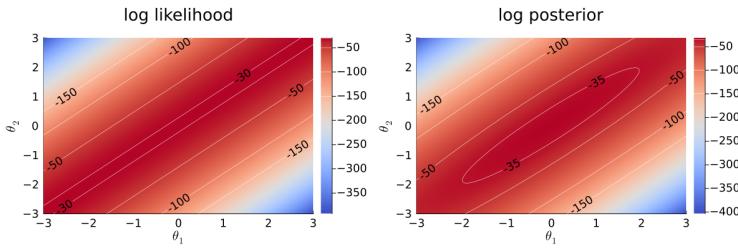
$$\Pr(y = c | \mathbf{x}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_C) = \frac{\exp(\mathbf{x}^\top \boldsymbol{\beta}_c)}{\sum_{j=1}^C \exp(\mathbf{x}^\top \boldsymbol{\beta}_j)}, \quad (8.13)$$

where one immediately can see that $\sum_{c=1}^C \Pr(y = c | \mathbf{x}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_C) = 1$, as required. Note that each class has its own vector of regression coefficients, $\boldsymbol{\beta}_c, c = 1, \dots, C$.

A problem with the model in (8.13) is that it is non-identified. A probabilistic model $p(\mathbf{x}|\theta)$ is said to be **non-identified** if there

non-identified

are multiple sets of parameter values $\theta_1, \dots, \theta_q$ that imply identical probability distributions for the data. Non-identified models are problematic since the likelihood cannot discriminate between these different parameter values: for *any* dataset \mathbf{x} from the data generating process we have $p(\mathbf{x}|\theta_1) = \dots = p(\mathbf{x}|\theta_q)$. There can even be an infinite number of parameter values with identical $p(\mathbf{x}|\theta)$, for example all linear combinations $\mathbf{a}^\top \theta = c$, for some vector \mathbf{a} and constant c . The left graph in Figure 8.7 plots the log-likelihood function for the toy model $x_1, \dots, x_n | \theta_1, \theta_2 \stackrel{\text{iid}}{\sim} N(\theta_1 - \theta_2, 1)$, which is non-identified since for any pair (θ_1, θ_2) where $\theta_1 = \theta_2$ we obtain exactly the same probability distribution $N(0, \sigma^2)$ for the data. Hence for *any* dataset x_1, \dots, x_n we have for example $p(x_1, \dots, x_n | \theta_1 = 1, \theta_2 = 1) = p(x_1, \dots, x_n | \theta_1 = 10, \theta_2 = 10)$.



The multi-class logistic regression in (8.13) is non-identified since adding a vector \mathbf{a} to all β_c does not affect the class probabilities:

$$\Pr(y = c | \mathbf{x}, \beta_1 + \mathbf{a}, \dots, \beta_C + \mathbf{a}) = \frac{\exp(\mathbf{x}^\top (\beta_c + \mathbf{a}))}{\sum_{j=1}^C \exp(\mathbf{x}^\top (\beta_j + \mathbf{a}))} \quad (8.14)$$

$$= \frac{\exp(\mathbf{x}^\top \beta_c)}{\sum_{j=1}^C \exp(\mathbf{x}^\top \beta_j)}, \quad (8.15)$$

as \mathbf{a} cancels out in the numerator and denominator. Hence, the likelihood attains exactly the same value for any \mathbf{a} , and the model is non-identified. Luckily, the model can be identified by setting $\beta_c = 0$ for one of the classes. This class is then referred to as the **reference class**; we will set the last class to zero, i.e. $\beta_C = 0$. The reason why this restriction identifies the model is that it makes sure that only $\mathbf{a} = 0$ is allowed, since any non-zero \mathbf{a} violates the restriction $\beta_C = 0$. Looking back now to the binary logistic regression we can understand why there is only one β in that model, despite there being two classes: we implicitly set the negative class to the reference class with zero regression coefficients.

Many machine learning libraries do not use zero restrictions to identify the multi-class model, and instead use a regularization prior to identify the model. This is illustrated in the right graph in Figure 8.7 where the likelihood function is combined with a $N(0, 1)$ prior

Figure 8.7: Illustrating non-identification in the model $x_1, \dots, x_n | \theta_1, \theta_2 \stackrel{\text{iid}}{\sim} N(\theta_1 - \theta_2, 1)$. The left graph plots the log-likelihood as a heatmap with overlayed contour lines. The log-likelihood attains the same value along each contour line. The log-likelihood cannot discriminate between the parameter combinations along a given line. The right graph shows how a prior "solves" the non-identification by combining the likelihood with a $N(0, 1)$ prior for each parameter; the parameter combinations along the previous lines no longer have the same posterior density reference class values.

for each parameter. The contour curves are no longer lines since the prior is now adding information that helps to discriminate between parameter value pairs with identical likelihoods; the prior can be said to identify the model. Using a prior to cover up an identification problem in the model is typically not a great idea, but can be useful when it is difficult to impose identifying restrictions.

The role and interpretation of the multi-class parameters can be understood by the log odds comparing the classes pairwise:

$$\text{LogOdds}_{c,k}(\mathbf{x}) \equiv \log \frac{\Pr(y = c|\mathbf{x})}{\Pr(y = k|\mathbf{x})} = \mathbf{x}^\top (\boldsymbol{\beta}_c - \boldsymbol{\beta}_k). \quad (8.16)$$

The log odds between any pair of classes is hence linear in the difference of the classes' parameter vectors. An increase in a covariate x_j with a larger coefficient in class c compared to class k would therefore increase the probability for class c *compared to* class k . Setting $\boldsymbol{\beta}_C = 0$ for identification makes it particularly easy to interpret the $\boldsymbol{\beta}_c$ coefficients by comparing them against the reference class

$$\text{LogOdds}_{c,C}(\mathbf{x}) \equiv \log \frac{\Pr(y = c|\mathbf{x})}{\Pr(y = C|\mathbf{x})} = \mathbf{x}^\top \boldsymbol{\beta}_c.$$

8.4 Poisson regression

Model and likelihood

The normal posterior approximation technique can be directly applied to many other interesting regression and classification models. As an example, we consider the **Poisson regression** model for count data y conditional on a vector of covariates \mathbf{x} . The model is specified as

$$\begin{aligned} y_i | \mathbf{x}_i &\stackrel{\text{indep}}{\sim} \text{Pois}(\lambda_i) \\ \lambda_i &= \exp(\mathbf{x}_i^\top \boldsymbol{\beta}), \end{aligned} \quad (8.17)$$

Poisson regression

for $i = 1, \dots, n$, where the observations $y_i | \mathbf{x}_i$ are assumed to be conditionally independent, as in the Gaussian linear regression model. We could have written the model as a one-liner

$$y_i | \mathbf{x}_i \stackrel{\text{indep}}{\sim} \text{Pois}(\lambda_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})),$$

but we prefer the formulation in (8.17) as it will make it easier to extend the model later. There are a couple of things to note here. First, the mean λ_i in the Poisson distribution is a *conditional* mean, so it varies from observation to observation, depending on the observation's covariate vector \mathbf{x}_i ; this why this is a Poisson *regression* model. Second, the Poisson mean $\lambda_i = \mathbb{E}(y_i | \mathbf{x}_i) = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$ is modeled via

the exponential function to make sure that the mean is always positive, as required in the Poisson distribution; this allows us to have β unrestricted in \mathbb{R}^p . Since $\log \mathbb{E}(y|\mathbf{x}) = \mathbf{x}^\top \beta$, we usually express this by saying that the model in (8.17) has a log *link function*.

The likelihood for the Poisson regression model is obtained by multiplying n Poisson distributions (since the $y_i|\mathbf{x}_i$ are assumed conditionally independent), each with its own mean $\lambda_i = \exp(\mathbf{x}_i^\top \beta)$:

$$p(\mathbf{y}|\beta, \mathbf{X}) = \prod_{i=1}^n \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!} = \prod_{i=1}^n \frac{\exp(\mathbf{x}_i^\top \beta)^{y_i} e^{-\exp(\mathbf{x}_i^\top \beta)}}{y_i!}. \quad (8.18)$$

Let us for simplicity assume a multivariate normal prior $\beta \sim N(\mathbf{0}, \tau^2 I_p)$, but any prior can be used in the following. The posterior distribution is proportional to the likelihood times the prior

$$p(\beta|\mathbf{y}, \mathbf{X}) \propto \prod_{i=1}^n \frac{\exp(\mathbf{x}_i^\top \beta)^{y_i} e^{-\exp(\mathbf{x}_i^\top \beta)}}{y_i!} \exp\left(-\frac{1}{2\tau^2} \beta^\top \beta\right). \quad (8.19)$$

The expression (8.19) is simple to evaluate for any β , but is not of a known distributional form in β . Moreover, there is no known prior that would make the posterior for the Poisson regression model tractable. However, the normal posterior approximation technique is easy to apply: all we need to do is to replace the unnormalized log posterior for the logistic regression in Figure 8.3 with the logarithm of (8.19). This code is shown in Figure 8.8, where we can make use of the Distributions.jl package in Julia to compute the log of the pdf for the Poisson distribution in vectorized form. Similar functions exist in R and Python.

```
"""
poisreg_logpost(β, y, X, μ, Σ)

log posterior for the Poisson regression model
    y|x ~ Pois(exp(x'β))
with the prior
    β ~ N(μ,Σ).
"""

function poisreg_logpost(β, y, X, μ, Σ)
    loglik = sum(logpdf(Poisson.(exp.(X*β)), y) )
    logprior = logpdf(MvNormal(μ, Σ), β)
    return(loglik + logprior)
end
```

Figure 8.8: Julia code for the log posterior function use to approximate the joint posterior for the Poisson regression model $p(\beta|\mathbf{y}, \mathbf{X})$ with a multivariate normal distribution. The Poisson and MvNormal distributions are imported from the Distributions.jl package.

Interpreting the Poisson regression coefficients

Results from a Poisson regression are usually presented for the transformation $\exp(\beta_j)$ instead of the original β_j parameters. The reason is that $\exp(\beta_j)$ are interpretable multiplicative factors, similarly to the odds ratios in logistic regression. To see this, note that we can express the Poisson mean for an arbitrary observation using the usual rules for exponentials (see margin for a reminder)

$$\lambda = \exp(\beta_0 + x_1\beta_1 + \dots + x_p\beta_p) = \exp(\beta_0) \exp(\beta_1)^{x_1} \cdots \exp(\beta_p)^{x_p} \quad (8.20)$$

The factor $\exp(\beta_0)$ is the baseline mean when all covariates are zero. The factor $\exp(\beta_j)$ is the multiplicative effect of increasing the covariate x_j by one unit, for example, $\exp(\beta_j) = 1.1$ means a 10% increase in the mean of y as x_j increases by one unit, while $\exp(\beta_j) = 0.6$ means a 40% decrease in the mean of y . Similar to the odds ratios in logistic regression, we can formally equate each $\exp(\beta_j)$ to a mean ratio, which is often termed the **incidence rate ratio** (IRR) or relative risk

$$\text{IRR}_j = \frac{\lambda(\mathbf{x} = (x_1, \dots, x_j + 1, \dots, x_p))}{\lambda(\mathbf{x} = (x_1, \dots, x_j, \dots, x_p))} = \exp(\beta_j), \quad (8.21)$$

where the only difference between the numerator and denominator is that x_j is increased by one unit. Hence, the IRR is the multiplicative effect of increasing x_j by one unit while keeping all other covariates unchanged.

Poisson regression for modeling the number bids in eBay auctions

As an application of Poisson regression we will return to the eBay auction data from Section 2.4, this time using a Poisson regression where the number of bids (y) is explained by the auction's starting price and other covariates. The starting price is the lowest price that the seller is willing to accept and is expected to affect the number of bids; a too high starting price may discourage bidder from placing a bid. The other covariates are properties of the seller and of the auctioned item. Table 8.2 gives the details of the dataset.

We approximate the posterior distribution $p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$ by a multivariate normal distribution using the numerical optimization technique described earlier in this chapter. The solid lines in Figure 8.9 are the marginal posterior from a normal approximation of the Poisson regression model for the number of bids. The figure plots the marginal posteriors for the incidence rate ratios (IRR), i.e. the transformation $\exp(\beta_j)$, as explained above. The marginal posterior densities for the IRRs in Figure 8.9 is obtained by the following three-step process:

$$\begin{aligned}\exp(a+b) &= \exp(a)\exp(b) \\ \exp(cd) &= \exp(c)^d\end{aligned}$$

incidence rate ratio

variable	description	data type	original range
nbids	number of bids	counts	[0,12]
bookvalue	coin's book value	continuous	[7.5, 399.5]
startprice	seller's reservation price / book value	continuous	[0, 1.702]
minblemish	minor blemish	binary	[0, 1]
majblemish	major blemish	binary	[0, 1]
negfeedback	large negative feedback score	binary	[0, 1]
powerseller	large quantity seller	binary	[0, 1]
verified	verified seller on ebay	binary	[0, 1]
sealed	unopened package	binary	[0, 1]

Table 8.2: Summary of the ebay data. The variable `bookvalue` is used in logarithms and both $\log(\text{bookvalue})$ and `startprice` are transformed to have zero mean before the modelling.

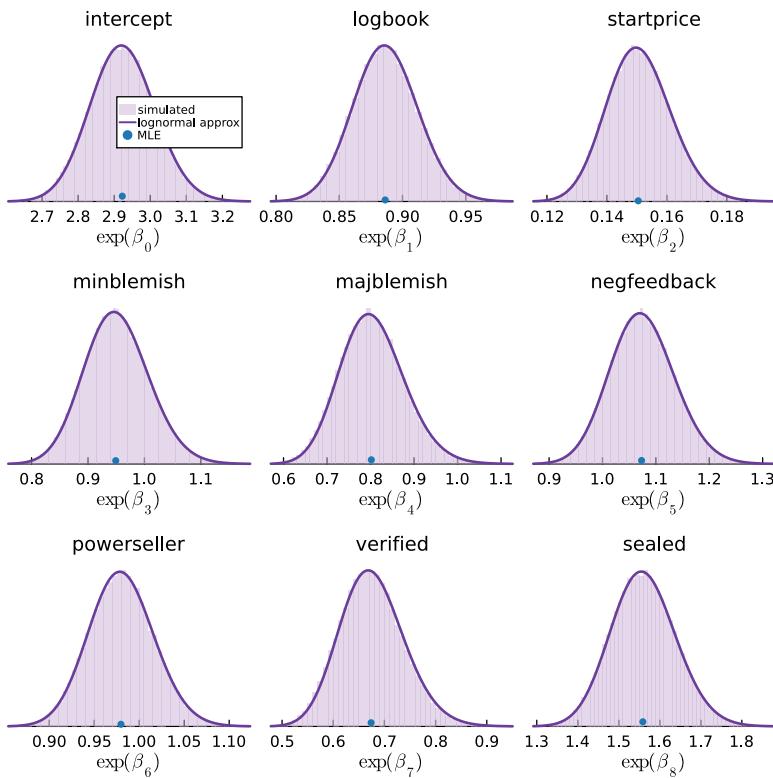


Figure 8.9: Normal approximation (solid lines) of the marginal posterior distributions of the mean ratio $\exp(\mathbf{x}^\top \boldsymbol{\beta})$ for the covariates in the Poisson regression model for the number of bids in the eBay auction data. The covariates `logbook` and `startprice` are demeaned, and the prior variance is set to $\tau^2 = 10^2$. The histograms are based on 100,000 HMC samples from the posterior and therefore represents the exact marginal posteriors without approximation.

1. obtain a multivariate approximation $\beta|\mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by optimizing the joint posterior for β , where $\boldsymbol{\mu}$ is the posterior mode of β and $\boldsymbol{\Sigma}$ is the negative inverse Hessian at the mode.
2. obtain the univariate marginal posterior for each $\beta_j|\mathbf{y}, \mathbf{X} \sim N(\mu_j, \sigma_j^2)$, by just selecting μ_j as the j th element of $\boldsymbol{\mu}$, and σ_j^2 as the j th diagonal element of $\boldsymbol{\Sigma}$.
3. return the approximate posterior of $\exp(\beta_j)$ as log-normally distributed $\beta_j \sim \text{LogNormal}(\mu_j, \sigma_j^2)$.

The final step follows from the very definition of the log-normal distribution (see Box 3.17): a random variable X is log-normal with parameters μ_j and σ_j^2 if $\log(X)$ is normally distributed with mean μ_j and variance σ_j^2 .

Figure 8.9 also show histograms are based on 100,000 samples from the posterior (using Hamiltonian Monte Carlo, HMC), which you will learn about later in Chapter 10) and can therefore be taken to represent the exact marginal posteriors without approximation.

The HMC sampling is from $p(\beta|\mathbf{y}, \mathbf{X})$, from which the marginal posteriors for $\exp(\beta_j)$ can be obtained by simply selecting the draws for β_j and transforming each draw with the exponential function. The marginal posteriors of $\exp(\beta_j)$ from the (log)normal approximation are very close to the exact marginal posteriors from HMC sampling.

The marginal posteriors in Figure 8.9 show that increasing `logbook` by one unit reduces the number of bids by approximately 12% ($1 - 0.88$) and increasing `startprice` by one unit reduces the number of bids by approximately 85% ($1 - 0.15$), and these effects are rather precisely determined (the posteriors are fairly tight); recall however from the definition of `startprice` that a unit increase in `startprice` corresponds to a dramatic increase in the starting price from \$0 to the full book value of the object. As expected, a major blemish tends to reduce the number of bids more than a minor one. Interestingly, a large negative feedback score for the seller does not seem to affect the number of bids: a 95% equal tail credible interval (0.978, 1.177) for $\exp(\beta_5)$ includes the value $\exp(\beta_5) = 1$.

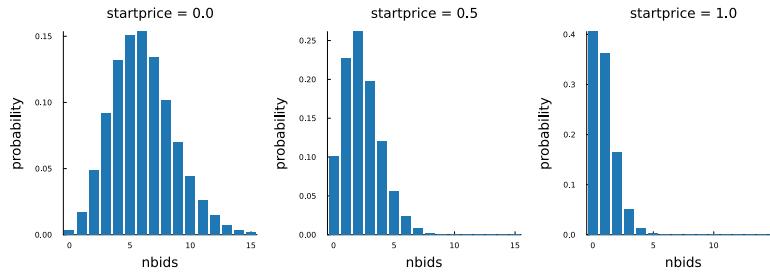
Finally, let us apply the Poisson regression to predict the number of bids in a hypothetical auction for a coin valued at \$100 in book value, with a major blemish, sold in a sealed package by a verified powerseller with no large negative feedback. The seller is interested in how the `startprice` affects the predictive distribution for the number of bids, and considers three `startprice` scenarios: 0, 0.5 (half the book value) and 1 (the full book value). The predictive distribution from a Poisson regression model is not tractable mathematically, but can be easily obtained by simulation, following the general recipe presented in Chapter 6:

1. simulate N draws $\beta^{(1)}, \dots, \beta^{(N)}$ from the multivariate normal approximation of the posterior $p(\beta|\mathbf{y}, \mathbf{X})$
2. for each posterior draw $\beta^{(k)}$, simulate a prediction from

$$y_{\star}^{(k)} \mid \beta^{(k)}, \mathbf{x}_{\star} \sim \text{Poisson}(\exp(\mathbf{x}_{\star}^{\top} \beta^{(k)})),$$

where \mathbf{x}_{\star} is the vector of covariates for the test observation (including the `startprice` considered).

Figure 8.10 presents histograms of the simulated predictive distribution of the number of bids for each `startprice` scenario. The predictive distribution is clearly strongly affected by the `startprice` with the distribution increasingly more concentrated on low number of bids as `startprice` increases.



8.5 Generalized linear models and beyond

Generalized linear models provide a generalization of the logistic regression for binary data and the Poisson regression for count data to any distribution in the exponential family. One example would be Beta regression where proportion data are modelled by the $\text{Beta}(\alpha, \beta)$ distribution with the mean of the distribution linked to covariates.

One version of the **generalized linear models (GLM)** family is

$$\begin{aligned} y_i | \mathbf{x}_i &\stackrel{\text{indep}}{\sim} \text{ExpFamily}(\mu_i) \\ g(\mu_i) &= \mathbf{x}_i^{\top} \boldsymbol{\beta}, \end{aligned} \tag{8.22}$$

where we use a slightly different notation for the exponential family with the conditional mean $\mu = \mathbb{E}(y|\mathbf{x})$ as the argument instead of the parameter θ . A key assumption in GLMs is that the conditional mean transformed by the **link function** $g()$ is assumed to be a linear function of the covariates. The linear combination $\mathbf{x}_i^{\top} \boldsymbol{\beta}$ is termed the **linear predictor** in the GLM literature. The logistic regression is a GLM with the Bernoulli distribution as the exponential family member and the log odds as link function, since for a binary variable $\mathbb{E}(y|\mathbf{x}) = \Pr(y = 1|\mathbf{x})$. Poisson regression is a GLM with the Poisson

Figure 8.10: Predictive distributions for a test auction selling a coin with a \$100 book value, with a major blemish, sold in a sealed package by a verified powerseller with no large negative feedback. Each graph corresponds to a choice of `startprice`. The predictive distributions are obtained by simulation from the normal approximation of the joint posterior distribution.

generalized linear models

GLM

link function

linear predictor

distribution and the log link. Gaussian linear regression is a GLM with a Gaussian distribution and the identity function $g(\mu) = \mu$ as a link. The posterior distribution for β in GLMs are almost always intractable (the Gaussian linear regression is an exception), but the normal approximation method is simple to implement. General expressions for the gradient and Hessian matrix for GLMs are available in many textbooks, but automatic differentiation is an otherwise attractive alternative.

The GLM model in (8.22) can be further extended by replacing the linear predictor $\mathbf{x}_i^\top \beta$ with a nonlinear function of the covariates. Polynomials or splines (see Chapter 12) are useful here, with the Gaussian processes (Chapter 15) as an interesting flexible alternative. Functions that are nonlinear in both the covariates \mathbf{x} and the parameters β can also be used for further expressiveness in the mean, e.g. deep neural networks. There is in principle nothing that stops us from using a normal posterior approximation for such models, but the approximation may be inaccurate and numerically costly in models with highly non-Gaussian high-dimensional posteriors.

There are two main reasons why the GLM class of regression models are considered important. First, the maximum likelihood estimator and its covariance matrix can be obtained from the same iteratively reweighted least squares algorithm for all GLM models; this made it possible to write general purpose software for GLMs. Second, the exponential family is relatively easy to work with theoretically, so many properties of this model class are known. The first reason was important in the 1970's, when GLMs were introduced, and in the following couple of decades, but with the advent of powerful computers, general optimization algorithms and automatic differentiation, this is less important today.

While the class of GLMs includes an unexpectedly wide array of models, many useful models are not GLMs. For example, regression models with distributions outside of the exponential family are not GLMs; the student- t distribution often used for robust regression is one example. Another important class of models not covered by GLMs are models where the covariates enter the response distribution through other quantities than the mean. Consider for example the **heteroscedastic** Gaussian linear regression

$$\begin{aligned} y_i &= \mathbf{x}_i^\top \beta + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_i^2) \\ \sigma_i^2 &= \exp(\mathbf{x}_i^\top \gamma) \end{aligned}$$

where we note that the variance is no longer constant (homoscedastic), but a function of the covariates with its own set of regression coefficients in the vector γ ; we are again using the exponential function to ensure that all variances σ_i^2 are positive. This model therefore

heteroscedastic

has two regression functions: one for the mean $\mathbb{E}(y_i|\mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$ and one for the variance $\mathbb{V}(y_i|\mathbf{x}_i) = \exp(\mathbf{x}_i^\top \boldsymbol{\gamma})$. It is no more complicated to use different sets of covariates in the two regression functions.

This useful model is not a GLM, but the same normal approximation technique can be applied to approximate the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\gamma}|\mathbf{y}, \mathbf{X})$. The computer does not care if you have one or two regression functions, as long as you can code up the log-likelihood and prior so that it can optimize the log posterior and return the posterior mode and Hessian needed for the normal approximation. Note that most computer optimization packages require the optimization to over a single parameter vector, so just stack the parameters in one long (column) vector: $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \boldsymbol{\gamma}^\top)^\top$. This implementation is left as an exercise to the reader.

Let us consider an extension of the Poisson regression which is not a GLM: the **negative binomial regression**

$$y_i|\mathbf{x}_i \stackrel{\text{indep}}{\sim} \text{NegBin}\left(\psi, p_i = \frac{\psi}{\psi + \lambda_i}\right) \quad (8.23)$$

$$\lambda_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta}) \quad (8.24)$$

where $\psi > 0$ is the so called **over-dispersion** parameter. To see that this a generalization of the Poisson note that from the mean and variance of the model in (8.23) is

$$\mathbb{E}(y_i|\mathbf{x}_i) = \lambda_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$$

$$\mathbb{V}(y_i|\mathbf{x}_i) = \lambda_i \left(1 + \frac{\lambda_i}{\psi}\right)$$

This shows that the negative binomial has the same mean regression as the Poisson regression model, but without the Poisson restriction that the mean and variance has to be equal. The variance of the negative binomial regression is always larger than the mean, how much larger is determined by the over-dispersion parameter ψ . As $\psi \rightarrow \infty$ we get back to the Poisson restriction $\mathbb{E}(y_i|\mathbf{x}_i) = \mathbb{V}(y_i|\mathbf{x}_i) = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$. Even more, it can be shown that the negative binomial distribution converges to the Poisson distribution when $\psi \rightarrow \infty$. When ψ is known, the negative binomial distribution does belong to the exponential family and the negative binomial regression is a GLM. But in the more common case when ψ is unknown and needs to be estimated, the model is not a GLM.

We can nevertheless approximate the posterior $p(\boldsymbol{\beta}, \psi|\mathbf{y}, \mathbf{X})$ by defining the extended parameter vector $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \psi)$ and applying the same optimization approach as before. The over-dispersion parameter ψ has to be positive however, so we define $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \tilde{\psi})$, where $\tilde{\psi} = \log(\psi)$. This has two advantages: i) the parameter used in the optimization $\tilde{\psi}$ is unrestricted so no need for constrained optimization, ii) the log transformation has a tendency to make the posterior

negative binomial regression

over-dispersion

more normal for strictly positive parameters like the over-dispersion or variance parameters. The latter advantage is not a mathematical fact, but has been observed empirically for many models used in practical work. The approximate normal marginal posterior for $\tilde{\psi}$ can then be transformed back to the original parameter ψ , either mathematically by the change-of-variable formula, or by simulating from the approximated posterior of $\tilde{\psi}$ and computing $\psi = \exp(\tilde{\psi})$ for each draw. Note that a normal prior $\log \psi \sim N(\mu_0, \sigma_0^2)$ implies a $\psi \sim \text{LogNormal}(\mu_0, \sigma_0^2)$ on the original scale.

```
"""
negbinreg_logpost(beta, y, X, mu, Sigma)

log posterior for the negative regression model
y|x ~ NegativeBinomial(psi, p = psi/(psi + lambda))
lambda = exp(x'beta)
beta ~ N(mu, Sigma)
psi ~ LogNormal(mu_0, sigma_0^2)

function negbinreg_logpost(theta, y, X, mu, Sigma, mu_0, sigma_0^2)
    beta = theta[1:(length(theta)-1)]
    logpsi = theta[length(theta)]
    psi = exp(logpsi)
    lambda = exp.(X*beta)
    loglik = sum(logpdf.(NegativeBinomial.(psi, psi ./ (psi .+ lambda)), y) )
    logprior = logpdf(MvNormal(mu, Sigma), beta) + logpdf(LogNormal(mu_0, sigma_0^2), psi)
    return(loglik + logprior)
end
"""


```

Box 8.1: Julia code for the log posterior function use to approximate the joint posterior for the negative binomial regression model $p(\beta, \log \psi | \mathbf{y}, \mathbf{X})$ with a multivariate normal distribution. Note how all parameters are stacked in the vector $\theta = (\beta^\top, \log \psi)^\top$ and that the over-dispersion parameter ψ is reparametrized in logarithm.

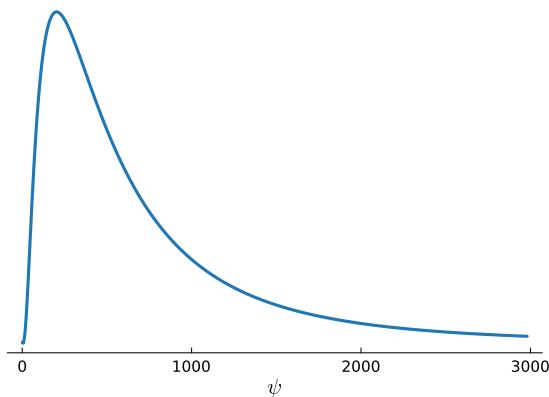


Figure 8.11: Marginal posterior $p(\psi | \mathbf{y}, \mathbf{X})$ for the over-dispersion parameter in a negative binomial regression model fitted to the ebay auction data. The joint posterior $p(\beta, \log \psi | \mathbf{y}, \mathbf{X})$ is approximated by a multivariate normal distribution using optimization.

A negative binomial regression is fitted to the ebay auction data that was previously analyzed with a Poisson regression model in Section 8.4. The posterior is obtained by the above outlined multivariate normal approximation of $p(\beta, \log \psi | \mathbf{y}, \mathbf{X})$; see Box ?? for the Julia code for the log posterior. As before, we can then obtain the marginal posterior $p(\log \psi | \mathbf{y}, \mathbf{X})$ as univariate normal from which it follows that $p(\psi | \mathbf{y}, \mathbf{X})$ is log-normal, all under the original normal approximation. Figure 8.11 shows that this marginal posterior for over-dispersion parameter ψ for the ebay data concentrates on very large values. Since when $\psi \rightarrow \infty$ we obtain the Poisson regression model, this strongly indicates that there is no over-dispersion after accounting for the covariates, and the simpler Poisson regression model is appropriate for this dataset.

As an example where over-dispersion is present we consider fitting the above model again, but this time using only a single covariate `logbook` in addition to the intercept. Figure 8.12 shows that the marginal posterior for ψ is now centered around 4.5, indicating some over-dispersion in the data. The marginal posterior of the intercept β_0 and the slope β_1 for `logbook` are relatively similar to the corresponding posteriors in the Poisson regression.

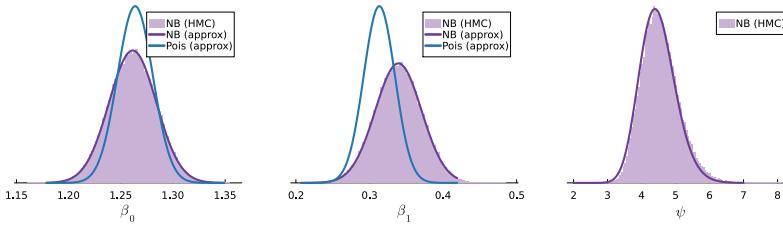


Figure 8.12: Marginal posteriors for the marginal posteriors of the Poisson and negative binomial (NB) fitted to the ebay auction data with an intercept and a single covariate `logbook`. The histograms are obtained from posterior sampling with HMC while the solid lines are from a normal approximation of the joint posterior.

8.6 Bayesian discriminant analysis and Naive Bayes

TODO! This section is very incomplete.

Logistic regression is a so called **discriminative model** where the class probabilities $\Pr(Y = y | \mathbf{x})$ are directly modelled using the logistic function. This is in contrast to a **generative model** where the class probabilities are modeled more implicitly using Bayes' theorem

$$\Pr(Y = y | \mathbf{x}) \propto \Pr(\mathbf{x} | Y = y) \cdot \Pr(Y = y), \quad (8.25)$$

where $\Pr(Y = y)$ is the prior probability of the class and $\Pr(\mathbf{x} | Y = y)$ is the **class-conditional distribution** of the covariates \mathbf{x} . The prior probability is usually relatively simple to determine, it can for example be computed as the fraction of observations in class c in the data,

discriminative model

generative model

class-conditional distribution

or by the Bayesian analysis of Bernoulli data with a Beta prior presented in Section 2.1. The class-conditional distributions $\Pr(\mathbf{x}|Y = y)$ for $y = 0$ and $y = 1$ are usually more difficult as they are the joint distributions of all covariates \mathbf{x} for each of the two classes.

If all covariates are continuous with values over the whole real line, perhaps after suitable transformations, a natural first model is a multivariate Gaussian model for each class

$$\mathbf{x}|y, \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y \sim N(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \quad y \in \{0, 1\}. \quad (8.26)$$

The parameters of the negative class $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ can be estimated from all the covariate observations in the negative class in the dataset, and the parameters of the positive class, $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$, can similarly be learned from the positive cases. This Gaussian model gives rise to the popular Quadratic Discriminant Analysis (QDA) procedure. If we restrict the two classes to have the same covariance matrix $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1$ we obtain Linear Discriminant Analysis (LDA); see (Lindholm et al., 2022) for details and applications.

Defining $\omega_0 = \Pr(Y = 0)$ and $\omega_1 = \Pr(Y = 1)$, the unknown model parameters are $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1, \omega_0$ and ω_1 . The predictive distribution is as usual obtained by integrating out the parameters with respect to the posterior distribution $p(\boldsymbol{\mu}_{\tilde{y}}, \boldsymbol{\Sigma}_{\tilde{y}}, \omega_{\tilde{y}}|\mathbf{y}, \mathbf{X})$, where \mathbf{y} and \mathbf{X} is the training data. Formally, we write

$$p(\tilde{y}|\tilde{\mathbf{x}}) \propto \omega_{\tilde{y}} \int N(\tilde{\mathbf{x}}|\boldsymbol{\mu}_{\tilde{y}}, \boldsymbol{\Sigma}_{\tilde{y}}) \cdot p(\boldsymbol{\mu}_{\tilde{y}}, \boldsymbol{\Sigma}_{\tilde{y}}, \omega_{\tilde{y}}|\mathbf{y}, \mathbf{X}) d\boldsymbol{\mu}_{\tilde{y}} d\boldsymbol{\Sigma}_{\tilde{y}} d\omega_{\tilde{y}}, \quad (8.27)$$

for $\tilde{y} = 0$ and $\tilde{y} = 1$. Here $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal density function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at \mathbf{x} .

1. Give expression for $p(\tilde{y}|\tilde{\mathbf{x}})$.
2. Apply it to Palmer penguins data with Bill length and Flipper length as covariates. Plot decision boundaries.
3. Naive Bayes.

EXERCISES 8.1

1. Consider the heteroscedastic linear regression model described in the text

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_i^2 = \exp(\mathbf{x}_i^\top \boldsymbol{\gamma})).$$

- a) Write a computer program (a function!) that simulates data from this model, taking the parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$, and sample size n as arguments (function inputs). The covariates can be simulated from a $N(0, 1)$ distribution. Generate a dataset of size

$n = 200$ with a single covariate using the parameter values $\beta = (1, 1)^\top$ and $\gamma = (-0.5, 0.5)^\top$. Make a scatter plot of the data.

b) Use optimization to compute a normal approximation of the posterior distribution $p(\beta, \gamma | \mathbf{y}, \mathbf{X})$, where \mathbf{y} and \mathbf{X} is the simulated data from the previous exercise. Simulate 10,000 samples from this approximation and make histograms to approximate the marginal posterior for each parameter in β and γ .

NOTEBOOKS 8.2

1. See the notebook [Classification](#).

9 Gibbs sampling

Gibbs sampling is a general iterative method for simulating from complex multivariate distributions. It can in principle be applied to any multivariate distribution, not necessarily a Bayesian posterior distribution, but we will here consider the Bayesian application where the aim is to sample from a joint posterior distribution $p(\theta_1, \dots, \theta_p | \mathbf{y})$.

Part of the appeal of Gibbs sampling is that we can often augment the data with additional auxiliary variables in a way that makes Gibbs sampling very easy to implement in a robust fashion. We will see several examples of this **data augmentation** approach in this chapter, for example when we design sampling algorithms for the probit and logistic regression models for binary response data.

9.1 The Gibbs sampling algorithm

Gibbs sampling simulates from a multivariate distribution by iteratively simulating each parameter from its so called full conditional posterior distribution. The **full conditional posterior** for the parameter θ_j is

$$p(\theta_j | \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_p, \mathbf{y}),$$

where we condition on *all* other parameters *except* θ_j ; this is the meaning of the word *full* conditional posterior. A common notation for the full conditional posterior is $p(\theta_j | \boldsymbol{\theta}_{-j}, \mathbf{y})$, where $\boldsymbol{\theta}_{-j}$ is the vector of all parameters except θ_j .

Starting from a set of initial values $\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_p^{(0)}$, Gibbs sampling iterates over the parameters, simulating a new draw from the parameter's full conditional posterior, conditioned on the most recent draw available for the other parameters. The algorithm is illustrated in Box 9.1 where parameters highlighted in orange indicates that the parameter has been updated in the current iteration of the algorithm.

Gibbs sampling is a member of the family of Markov Chain Monte Carlo (MCMC) algorithms, where Markov Chains are used to simulate from multivariate distributions. We will explain MCMC more

full conditional posterior

Gibbs sampling

Input: initial values $\theta_2^{(0)}, \dots, \theta_p^{(0)}$
 number of posterior draws m .
for i in $1:m$ **do**

$$\left| \begin{array}{l} \theta_1 \sim p(\theta_1 | \theta_2^{(i-1)}, \theta_3^{(i-1)}, \dots, \theta_p^{(i-1)}, \mathbf{y}) \\ \theta_2 \sim p(\theta_2 | \theta_1^{(i)}, \theta_3^{(i-1)}, \dots, \theta_p^{(i-1)}, \mathbf{y}) \\ \vdots \\ \theta_p \sim p(\theta_p | \theta_1^{(i)}, \theta_2^{(i)}, \dots, \theta_{p-1}^{(i)}, \mathbf{y}) \end{array} \right.$$

end
Output: m autocorrelated draws for $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^\top$
 that converge in distribution to the joint
 posterior $p(\theta_1, \dots, \theta_p | \mathbf{y})$.

Box 9.1: Gibbs sampling algorithm for sampling from a joint posterior distribution $p(\theta_1, \dots, \theta_p | \mathbf{y})$. Parameters highlighted in orange indicates that the parameter has been updated in the current iteration of the algorithm.

fully in the next chapter; for the moment, five related things about MCMC algorithms are important for understanding Gibbs sampling:

- samples from an MCMC algorithm are *autocorrelated* over the iterations, meaning that successive draws of $\boldsymbol{\theta}$ are dependent on each other.
- even if the MCMC draws are autocorrelated, they will nevertheless *converge in distribution* to the target posterior distribution

$$\theta^{(1)}, \dots, \theta^{(m)} \xrightarrow{d} p(\boldsymbol{\theta} | \mathbf{y}) \text{ as } m \rightarrow \infty.$$

This means for example that histograms of the draws will look more and more like the posterior distribution as we keep on simulating draws with MCMC. This holds for both marginal and joint posterior distributions.

- the convergence to the target posterior happens *for any initial values* used to start up the sampling algorithm.
- a version of the central limit theorem for dependent variables can be used to establish that the sample mean of the draws can be well

approximated by a normal distribution if we sample long enough. Informally, where $\bar{\theta}_{1:m}$ is the sample mean of m MCMC draws:

$$\bar{\theta}_{1:m} \xrightarrow{\text{approx}} N\left(\mathbb{E}(\theta|y), V(\bar{\theta}_{1:m})\right) \text{ for large } m$$

- MCMC sampling tends to be *less efficient* than iid sampling from the joint posterior, which means that we have to sample more draws to obtain the same posterior approximation accuracy as that obtained from iid sampling.

As an example consider simulating from a bivariate normal distribution $\theta \sim N(\mu, \Sigma)$, with mean vector $\mu = (\mu_1, \mu_2)^\top$ and covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

where ρ is the correlation between θ_1 and θ_2 . In this case there is really no need for Gibbs sampling at all since we can simulate from a p -dimensional multivariate normal distribution $\theta \sim N(\mu, \Sigma)$ by simulating a vector $z = (z_1, \dots, z_p)^\top$ and with independent standard univariate normal variables and setting

$$\theta = \mu + Lz, \quad (9.1)$$

where L is the $p \times p$ lower triangular Cholesky factor in the matrix decomposition $\Sigma = LL^\top$; see Appendix A.2 for details. We will nevertheless show how Gibbs sampling is implemented for the bivariate normal distribution as an illustration and compare its efficiency to direct iid sampling. Box 9.2 gives the Gibbs sampling algorithm for a bivariate normal distribution target in pseudo code, and Box 9.3 provides a Julia implementation of the general case with a multivariate normal distribution target.

Figure 9.1 plots the draws (points) from a bivariate normal target distribution (contours) for different correlations in the target distribution. Both iid sampling (left column) and Gibbs sampling (right column) are shown. It is clear that Gibbs sampling's coordinate-wise nature makes it explore the target distribution very slowly when the parameters are highly correlated. This is particularly clear in the case with $\rho = 0.99$ where the target distribution is strongly 'cigar shaped' and it takes a long time for Gibbs sampling to travel across the cigar. In contrast, iid sampling can of course freely move from one end of the cigar to the other from one iteration to the next since there is no dependence on the previous draw.

As mentioned above, the major disadvantage of Gibbs sampling (and MCMC more generally) is the draws are autocorrelated. This

Gibbs sampling from a bivariate normal

Input: initial value $\theta_2^{(0)}$
 number of posterior draws m .
for i in $1:m$ **do**

$$\left| \begin{array}{l} \theta_1^{(i)} | \theta_2 \sim N\left(\mu_1 + \rho \frac{\sigma_1}{\sigma_2} (\theta_2^{(i-1)} - \mu_2), \sigma_1^2 (1 - \rho)^2\right) \\ \theta_2^{(i)} | \theta_1 \sim N\left(\mu_2 + \rho \frac{\sigma_2}{\sigma_1} (\theta_1^{(i)} - \mu_1), \sigma_2^2 (1 - \rho)^2\right) \end{array} \right.$$
end
Output: m autocorrelated draws for $\theta = (\theta_1, \theta_2)^\top$ that
 converge in distribution to the bivariate normal
 distribution $\theta \sim N(\mu, \Sigma)$, where $\mu = (\mu_1, \mu_2)^\top$
 and

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{pmatrix}$$

Box 9.2: Gibbs sampling algorithm for sampling from the bivariate normal distribution of $\theta = (\theta_1, \theta_2)^\top$, i.e. $\theta \sim N(\mu, \Sigma)$. Parameters highlighted in orange indicates that the parameter has been updated in the current iteration of the algorithm.

```
using Distributions, LinearAlgebra, InvertedIndices

function GibbsMvNormal(μ, Σ, m, θ₀)
    # Pre-computing stuff what doesn't change in the Gibbs sampler iterations.
    p = length(μ)
    σ = zeros(p)          # Conditional variances
    β = zeros(p-1,p)      # "Regression coefficients" for reg on other coordinates
    for j = 1:p
        β[:,j] = Σ[Not(j),Not(j)]\Σ[Not(j),j]
        σ[j] = √(Σ[j,j]-Σ[j,Not(j)]·β[:,j])
    end

    # Gibbs sampling iterations
    postDraws = zeros(m,p)
    θ = θ₀
    for i = 1:m
        for j = 1:p
            θ[j] = rand(Normal(μ[j] + β[:,j]·(θ[Not(j)]-μ[Not(j)]), σ[j]))
            postDraws[i,j] = θ[j]
        end
    end
    return postDraws
end

julia> gibbsDraws = GibbsMvNormal(μ = zeros(2), Σ = [1 0.7; 0.7 1], m = 1000, θ₀ = zeros(2));
julia> cov(gibbsDraws)
2x2 Matrix{Float64}:
 1.00295  0.694255
 0.694255  0.998105
```

Box 9.3: Gibbs sampling for a multivariate normal target distribution in Julia, including an example call of the function at the end. The `Not` function from the `InvertedIndices.jl` package selects all indices except the one in the argument, and \cdot is the usual (dot) vector product.

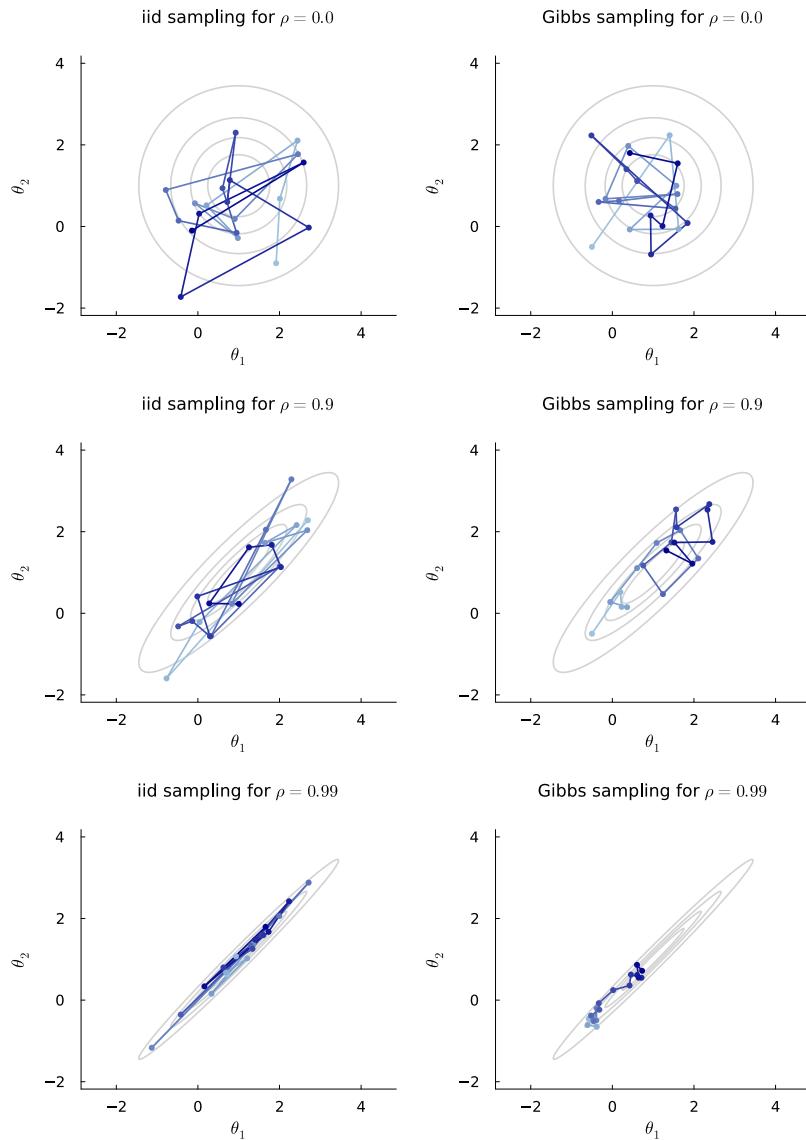


Figure 9.1: Comparing sampling paths of iid sampling (left column) vs Gibbs sampling (right column) for a bivariate normal target distribution in different correlations ρ (rows). The sampling path starts in the lightest blue and ends in the darkest blue.

leads to less efficient estimates of, for example, the posterior mean and standard deviation, or more generally the whole posterior distribution. One way to quantify this loss of efficiency compared to iid sampling is to compare the variance of the sample mean $\bar{\theta}_{1:m}$ of the simulated draws, as a frequentist estimator of the posterior mean $\mathbb{E}(\theta|y)$. For iid sampling we immediately obtain the usual variance formula for sample mean:

$$\mathbb{V}_{\text{iid}}(\bar{\theta}_{1:m}) = \frac{\mathbb{V}(\theta|y)}{m}, \quad (9.2)$$

When the draws are autocorrelated the variance can for large m be approximated by:

$$\mathbb{V}_{\text{mcmc}}(\bar{\theta}_{1:m}) = \frac{\mathbb{V}(\theta|y)}{m} \left(1 + 2 \sum_{k=1}^{\infty} \rho_k\right), \quad (9.3)$$

where $\rho_k = \text{Corr}(\theta^{(i)}, \theta^{(i-k)})$, i.e. the autocorrelation coefficient at lag k . The approximation is motivated by the fact that for a stationary process one can show that (Lindgren, 2012)

$$m \cdot \mathbb{V}_{\text{mcmc}}(\bar{\theta}_{1:m}) \xrightarrow{p} \mathbb{V}(\theta|y) \left(1 + 2 \sum_{k=1}^{\infty} \rho_k\right) \text{ as } m \rightarrow \infty. \quad (9.4)$$

The loss of efficiency from having to use dependent sampling instead of iid sampling can now be quantified by the **inefficiency factor** (IF)

$$\text{IF} = \frac{\mathbb{V}_{\text{mcmc}}(\bar{\theta}_{1:m})}{\mathbb{V}_{\text{iid}}(\bar{\theta}_{1:m})} = 1 + 2 \sum_{k=1}^{\infty} \rho_k. \quad (9.5)$$

The inefficiency factor measures how many times more draws are needed to achieve the same level of accuracy as iid sampling. For example, if the inefficiency factor is $\text{IF} = 10$, then we need ten times as many draws to achieve the same level of accuracy (sampling variance) as iid sampling. This allows us to define the **effective sample size** m_{eff} of a sampling algorithm that produces dependent draws as

$$m_{\text{eff}} = \frac{m}{\text{IF}}. \quad (9.6)$$

The effective sample size m_{eff} therefore represents the number of iid draws that would have the same sampling variance as the m dependent draws. A nominal sample size of $m = 10000$ draws from a sampling algorithm with $\text{IF} = 10$ is therefore equivalent to a sample with $m_{\text{eff}} = 1000$ iid draws.

Figure 9.2 illustrates the difference between iid sampling (first row) and Gibbs sampling (second row) for a bivariate normal target distribution with mean $\mu = (1, 1)$, unit variances and correlation $\rho = 0.9$. The left column displays the draws of θ_1 across the 1000 iterations, the middle column shows the cumulative mean estimate $\bar{\theta}_1$ as

inefficiency factor

effective sample size

more draws are included in the estimate, and the rightmost column shows the autocorrelation function of the draws. As expected, the dependent Gibbs draws moves more sluggishly over the iterations (left column) and therefore takes longer to converge to the true value (middle column). The dependence in the Gibbs draws is most clearly borne out in the estimated autocorrelation function (right column), which has high autocorrelations at shorter lags and decays slowly over the lags.

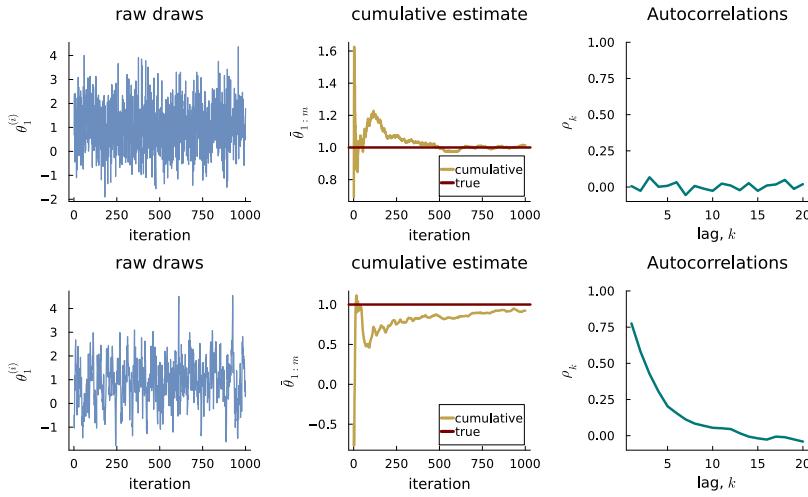


Figure 9.2: Independent sampling (first row) vs Gibbs sampling (second row) from a bivariate normal distribution $p(\theta_1, \theta_2 | \mathbf{x})$ with correlation $\rho = 0.9$.

Left column: draws of θ_1 across the 1000 iterations.

Middle column: cumulative mean estimate $\bar{\theta}_1$ as more draws are included in the estimate.

Right column: autocorrelation function of the draws.

Figure 9.3 uses simulation to explore the inefficiency factor (IF) for Gibbs sampling for a multivariate normal target density as the dimension of the multivariate normal increases. The covariance matrix of this target density is set to an equicorrelation matrix with correlation ρ and unit variance for all variables:

$$\Sigma = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{pmatrix}. \quad (9.7)$$

As expected the inefficiency factor is sharply increasing with the correlation coefficient for the larger ρ , and the inefficiency of Gibbs sampling is particularly severe in higher dimensions. For a 10-dimensional multivariate normal target density with $\rho = 0.8$ the inefficiency factor is around $IF = 30$ in the figure; hence we would need as much as 30 times as many draws to achieve the same level of accuracy as iid sampling.

Note that if the draws are negatively autocorrelated, i.e. $\rho_k < 0$, then we can have $m_{\text{eff}} > m$, meaning that the dependent draws are

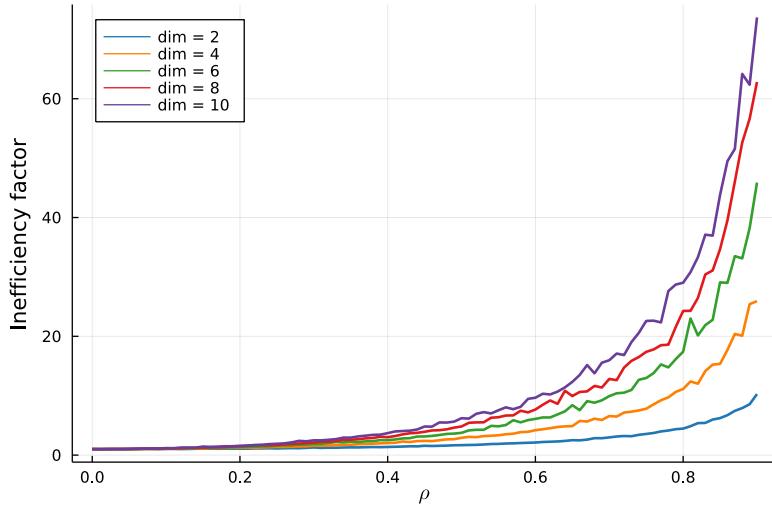


Figure 9.3: Inefficiency of Gibbs sampling for a multivariate normal target distribution in different dimensions with an equicorrelation matrix with correlation ρ .

more efficient than iid sampling. This makes sense if one considers that negative autocorrelation means by definition that one draw above the mean tends to be followed by a draw below the mean. The average of draws that tends to alternate between above and below the mean will therefore be a very good estimator of the mean. This is referred to as antithetic sampling in the Monte Carlo integration literature. However, with Gibbs sampling and MCMC we typically get positive autocorrelation, and the effective sample size from Gibbs sampling and MCMC is therefore nearly always smaller than the nominal sample size.

We have now seen that correlation between parameters in the posterior can be devastating for the efficiency of Gibbs sampling. When possible one can consider **reparametrizing** the model parameters to reduce the posterior correlation. In the multivariate normal distribution one can rotate the coordinate system to the principal axes (see Appendix A.2) to achieve parameters that exactly uncorrelated. However, in more serious examples it can be difficult to find the appropriate reparametrization.

A more common technique to deal with the inefficiency of Gibbs sampling is to group correlated parameters in a so called **block Gibbs sampler**, and to sample the group/block jointly from its multivariate full conditional posterior. For example, consider a posterior with three parameters $p(\theta_1, \theta_2, \theta_3 | \mathbf{y})$, where θ_1 and θ_2 are correlated and θ_3 is uncorrelated with the other two. We can then sample from the full conditional posterior by the following two-block Gibbs sam-

block Gibbs sampler

pler:

$$\text{Block 1: } (\theta_1, \theta_2) \sim p(\theta_1, \theta_2 | \theta_3, \mathbf{y})$$

$$\text{Block 2: } \theta_3 \sim p(\theta_3 | \theta_1, \theta_2, \mathbf{y})$$

We are here sampling the two correlated parameters jointly and the algorithm will be efficient since the two correlated dimensions we are traversing the cigar quickly in an iid fashion. This blocking technique can be applied in the same way with more than two blocks, and also with more than two parameters in a given block. However, we need to be able to sample from the full conditional posteriors of all blocks of parameters. While it is often that all univariate full conditional posteriors $p(\theta_1 | \theta_2, \theta_3, \mathbf{y})$, $p(\theta_2 | \theta_1, \theta_3, \mathbf{y})$ and $p(\theta_3 | \theta_1, \theta_2, \mathbf{y})$ belong to easily sampled distributional families, the bivariate $p(\theta_1, \theta_2 | \theta_3, \mathbf{y})$ may not be a recognizable, easily sampled, distribution. So, the recipe for success is to group together any parameters that are highly correlated and for which we can sample from the full conditional posterior. We will see an example of this in the next section.

9.2 Gibbs sampling for probit regression

Section 8.2 showed how we could approximate the posterior distribution for logistic regression using a normal distribution obtained with numerical optimization. Section 9.3 introduces a Gibbs sampler for the logistic regression. This section considers an alternative binary regression model, the **Probit regression**

$$\Pr(y_i = 1 | \mathbf{x}_i, \boldsymbol{\beta}) = \Phi(\mathbf{x}_i^\top \boldsymbol{\beta}), \quad (9.8)$$

where $\Phi(z)$ is the cumulative distribution function (cdf) of the standard normal distribution, see Figure 9.4. Similar to the reasoning behind the logistic function, the properties of a cdf guarantees that $\Pr(y = 1 | \mathbf{x}, \boldsymbol{\beta})$ is a number between zero and one for any values of the linear combination $\mathbf{x}^\top \boldsymbol{\beta}$. Although any cdf can be used to make this construction, the standard normal cdf makes it possible to use data augmentation to develop a Gibbs sampling algorithm for probit regression.

The probit model may be rewritten by augmenting the model with latent utility variables

$$u_i \stackrel{\text{indep}}{\sim} N(\mathbf{x}_i^\top \boldsymbol{\beta}, 1) \quad (9.9)$$

$$y_i = \begin{cases} 1 & \text{if } u_i > 0 \\ 0 & \text{if } u_i \leq 0 \end{cases} \quad (9.10)$$

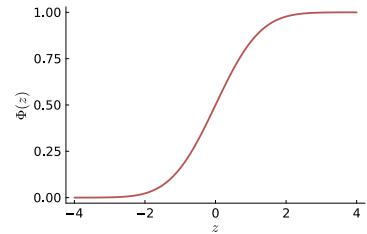


Figure 9.4: The standard normal cumulative distribution function $\Phi(z)$.

Probit regression

It is a simple calculation to show that the latent utility formulation in (9.9) and (9.10) is equivalent to the probit regression model in (9.8):

$$\begin{aligned}\Pr(y_i = 1 | \mathbf{x}_i) &= \Pr(u_i > 0) = 1 - \Pr(u_i \leq 0) \\ &= 1 - \Pr(u_i - \mathbf{x}_i^\top \boldsymbol{\beta} < -\mathbf{x}_i^\top \boldsymbol{\beta}) \\ &= 1 - \Phi(-\mathbf{x}_i^\top \boldsymbol{\beta}) = \Phi(\mathbf{x}_i^\top \boldsymbol{\beta})\end{aligned}$$

The latent utility variables u_i are not observed, and can be seen as purely artificial variables that are introduced to make Gibbs sampling possible, as we will see below. However, the latent utility variables can also be interpreted as an underlying continuous score that triggers the binary response $y_i = 1$ as soon as $u_i > 0$. As an example, consider the choice between taking public transportation to work ($y_i = 1$) or taking your own car ($y_i = 0$). We can model a person's preference for public transportation by a continuous variable u_i such that large values of u_i means that the i th person has a strong preference for public transportation. This latent utility u_i then determines if a person actually decides to go by public transport ($u_i > 0$ giving $y_i = 1$) or takes her own car ($u_i \leq 0$ giving $y_i = 0$). The probit regression models the utility as Normal random variable with a mean $\mathbf{x}_i^\top \boldsymbol{\beta}$ determined by the persons covariates, perhaps age, income, distance to work etc. The variance of the utility is by convention set to one, but could have been set to any number since the scale of the utilities are arbitrary. See Figure 9.5 for an illustration of the utility distribution and their choices for two persons with different covariate vectors \mathbf{x} .

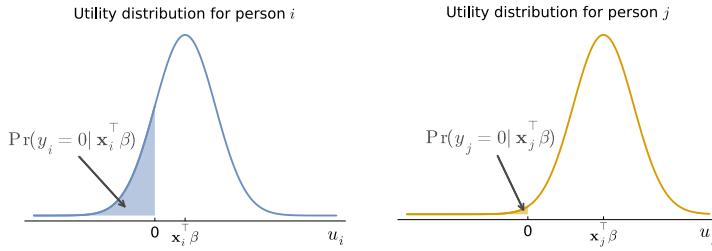


Figure 9.5: Latent utility distribution $u \sim N(\mathbf{x}^\top \boldsymbol{\beta}, 1)$ for two persons with different covariate vectors \mathbf{x} . The probability of $u \leq 0$, i.e. observing $y = 0$ is the area under the curve to the left of zero.

The beauty of introducing the latent utilities is that the model in (9.9) *conditional on* the latent utilities $\mathbf{u} = (u_1, \dots, u_n)^\top$ is just a linear regression model in $\boldsymbol{\beta}$ with the response vector being the latent utilities, \mathbf{u} . The full conditional posterior for $\boldsymbol{\beta}$ is therefore the same as for the linear Gaussian regression model in Chapter [Linear Regression](#), but now with the response vector \mathbf{u} and a *known* error variance $\sigma^2 = 1$. Let the prior distribution be $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. The full conditional posterior for $\boldsymbol{\beta}$ is then

$$\boldsymbol{\beta} | \mathbf{u}, \mathbf{y}, \mathbf{X} \sim N(\boldsymbol{\beta}_n, \boldsymbol{\Sigma}_n), \quad (9.11)$$

where $\Sigma_n = (\mathbf{X}^\top \mathbf{X} + \Sigma_0^{-1})^{-1}$ and $\beta_n = \Sigma_n(\mathbf{X}^\top \mathbf{u} + \Sigma_0^{-1}\mu_0)$. This forms the update step for β in the Gibbs sampler.

We now need to update the utilities $\mathbf{u} = (u_1, \dots, u_n)^\top$ in a separate Gibbs step. It should be clear from (9.9) and (9.10) that the individual u_i are independent conditional on β , i.e. $p(\mathbf{u}|\beta, \mathbf{y}, \mathbf{X}) = \prod_{i=1}^n p(u_i|\beta, \mathbf{y}, \mathbf{X})$. The full conditional posterior of each utility u_i needs some care, however. Using Bayes' theorem we have

$$p(u_i|\beta, \mathbf{y}, \mathbf{x}_i) \propto p(y_i|u_i)p(u_i|\beta, \mathbf{x}_i),$$

where the factors $p(y_j|u_j)$ for all other observations $j \neq i$ have been absorbed into the proportionality constant since they do not depend on u_i . The factor $p(y_i|u_i)$ is a little special however since from (9.10) we see that u_i *deterministically* determines y_i :

$$\begin{aligned} u_i > 0 &\Rightarrow p(y_i = 1|u_i) = 1 \quad \text{and} \quad p(y_i = 0|u_i) = 0 \\ u_i \leq 0 &\Rightarrow p(y_i = 1|u_i) = 0 \quad \text{and} \quad p(y_i = 0|u_i) = 1 \end{aligned}$$

We can express this using indicator functions

$$\mathbb{1}_{(a,b)}(x) = \begin{cases} 1 & \text{if } x \in (a, b) \\ 0 & \text{otherwise} \end{cases}$$

so that we can write

$$p(y_i|u_i) = \left(\mathbb{1}_{(-\infty, 0]}(u_i)\right)^{(1-y_i)} \left(\mathbb{1}_{(0, \infty)}(u_i)\right)^{y_i}$$

This expression is a little messy, but put in $y_i = 0$ and see that $p(y_i = 0|u_i) = \mathbb{1}_{(-\infty, 0]}(u_i)$. Figures 9.6 and 9.7 illustrates these two indicator functions.

Armed with the indicator functions, we can express the full conditional posterior of u_i as

$$\begin{aligned} p(u_i|\beta, \mathbf{y}, \mathbf{x}_i) &\propto p(y_i|u_i)p(u_i|\beta, \mathbf{x}_i) \\ &= \left(\mathbb{1}_{(-\infty, 0]}(u_i)\right)^{(1-y_i)} \left(\mathbb{1}_{(0, \infty)}(u_i)\right)^{y_i} N(u_i|\mathbf{x}_i^\top \beta, 1), \end{aligned}$$

which shows that the full conditional posterior for u_i is a normal distribution truncated at zero *from above*, $u_i \in (-\infty, 0]$, if $y_i = 0$ or truncated at zero *from below*, $u_i \in (0, \infty)$, if $y_i = 1$. See Box 9.4 and Figure 9.8 for the **truncated normal distribution**. In summary,

$$u_i|\beta, \mathbf{y}, \mathbf{x} \sim \begin{cases} N(\mathbf{x}_i^\top \beta, 1) \text{ truncated to } u_i \in (-\infty, 0] & \text{if } y_i = 0 \\ N(\mathbf{x}_i^\top \beta, 1) \text{ truncated to } u_i \in (0, \infty) & \text{if } y_i = 1 \end{cases} \quad (9.12)$$

The full conditional posterior for u_i is illustrated in Figure 9.9, showing both the case with $y_i = 0$ and $y_i = 1$. Note that a truncated

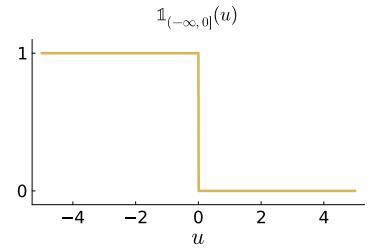


Figure 9.6: Indicator function for $y_i = 0$.

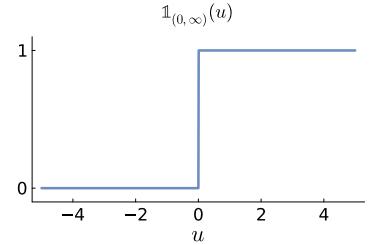


Figure 9.7: Indicator function for $y_i = 1$.

Truncated normal distribution

$X \sim N(\mu, \sigma^2, a, b)$ for $X \in [a, b]$.

$$p(x) = \frac{\phi(x|\mu, \sigma^2)}{\Phi(b|\mu, \sigma^2) - \Phi(a|\mu, \sigma^2)}$$

$$\mathbb{E}(X) = \mu + \sigma \frac{\phi(a) - \phi(b)}{\Phi(b) - \Phi(a)}$$

$\phi(x)$ is pdf of $N(0, 1)$

$\phi(x|\mu, \sigma^2)$ is pdf of $N(\mu, \sigma^2)$

$\Phi(x)$ is cdf of $N(0, 1)$

$\Phi(x|\mu, \sigma^2)$ is cdf of $N(\mu, \sigma^2)$

Box 9.4: The truncated normal distribution.

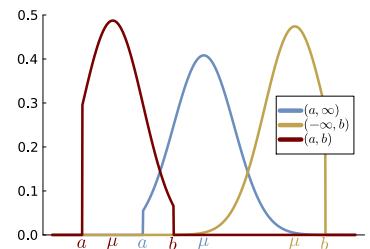


Figure 9.8: Normal distributions truncated at a from below (blue), at b from above (yellow) and from both below and above $[a, b]$ (red).

truncated normal distribution

distribution needs to be normalized so that its probability mass is 1, as for any density function (see the denominator of $f(x)$ in Box 9.4); this is why the yellow density to the left of zero is much higher than the blue density to the right of zero. The Gibbs sampling algorithm for probit regression using latent utility augmentation is summarized in Box 9.5.

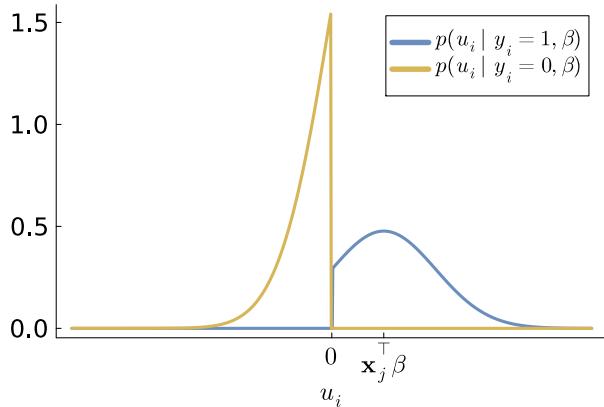


Figure 9.9: The posterior distributions for the latent utility u_i conditional on β is a normal distribution truncated at zero from above (when $y_i = 0$) or truncated at zero from below (when $y_i = 1$).

APPLIED PROBIT REGRESSION - WHO SURVIVED THE TITANIC?

Noninformative prior, $\mu_0 = \mathbf{0}$ and $\Sigma_0 = 10^2 \cdot I$. Gibbs sampling 100000 draws took 15 seconds and gave ESS of 30000, so ESS/sec = 2000. HMC with diagonal mass matrix 100000 draws after 1000 for determining adaptation took 316 sec and gave a ESS of 60000, so ESS/sec = 189.

9.3 Gibbs sampling for logistic regression

We revisit the logistic regression model here to show how a clever data augmentation trick makes it possible to develop a Gibbs sampling algorithm to sample from the posterior of the regression coefficients $p(\beta|\mathbf{y}, \mathbf{X})$. We will not give the full derivation of the Gibbs sampler here, just some of the key steps to give the reader an idea of where the sampler comes from.

Gibbs sampling for probit regression using latent utility augmentation

Input: response vector $\mathbf{y} = (y_1, \dots, y_n)^\top$
matrix ($n \times p$) with covariates \mathbf{X}
prior mean $\boldsymbol{\mu}_0$
prior covariance matrix $\boldsymbol{\Sigma}_0$
initial value $\boldsymbol{\beta}^{(0)}$
number of posterior draws m .

for k in $1:m$ **do**

- // Update latent utilities
- for** i in $1:n$ **do**

 - if** $y_i = 0$ **then**

 - $| u_i^{(k)} | \boldsymbol{\beta}^{(k-1)}, \mathbf{y}, \mathbf{x}_i \sim N(\mathbf{x}_i^\top \boldsymbol{\beta}^{(k-1)}, 1, -\infty, 0)$

 - else**

 - $| u_i^{(k)} | \boldsymbol{\beta}^{(k-1)}, \mathbf{y}, \mathbf{x}_i \sim N(\mathbf{x}_i^\top \boldsymbol{\beta}^{(k-1)}, 1, 0, \infty)$

 - end**

- end**
- // Update $\boldsymbol{\beta}$
- $\boldsymbol{\Sigma}_n \leftarrow (\mathbf{X}^\top \mathbf{X} + \boldsymbol{\Sigma}_0^{-1})^{-1}$
- $\boldsymbol{\mu}_n \leftarrow \boldsymbol{\Sigma}_n (\mathbf{X}^\top \mathbf{u} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0)$
- Draw $\boldsymbol{\beta}^{(k)} | \mathbf{u}^{(k)}$ from $N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$

end

Output: m draws $\boldsymbol{\beta}^{(1)}, \dots, \boldsymbol{\beta}^{(m)}$ from the posterior distribution $p(\boldsymbol{\beta} | \mathbf{y}, \mathbf{X})$.

Box 9.5: Gibbs sampling algorithm for the probit regression model using latent utility augmentation. The **for** loop that draws the latent utilities u_i for all observations should be replaced by fast vectorized operations in a non-compiled language.

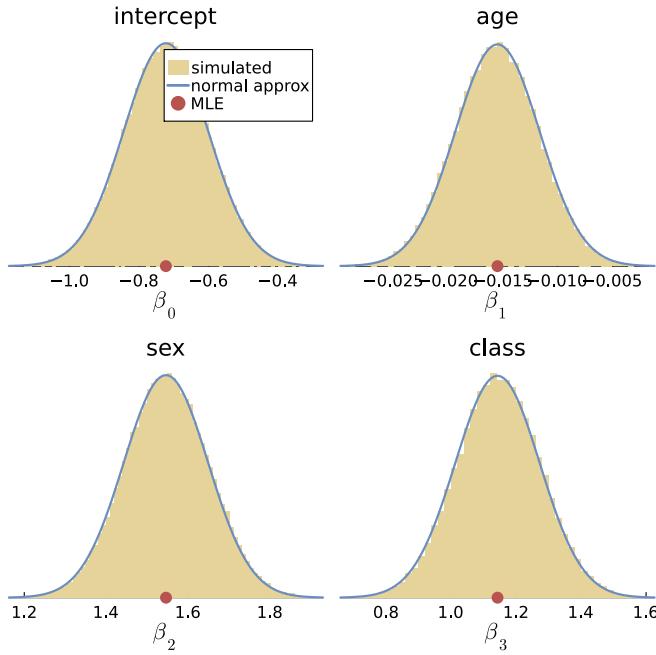


Figure 9.10: Marginal posterior distributions of the β coefficients in a probit regression fitted to the Titanic data. The solid lines are the marginal posterior based on the normal approximate posterior for β and the histograms are based on 100,000 Gibbs sampling draws. The maximum likelihood estimate for each β is shown as a red dot.

Recall that the likelihood for the logistic regression model is

$$\Pr(y_1, \dots, y_n | \mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^n \Pr(y_i = y_i | \mathbf{x}_i, \boldsymbol{\beta}), \quad (9.13)$$

where

$$\Pr(y_i = y_i | \mathbf{x}_i, \boldsymbol{\beta}) = \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta})^{y_i}}{1 + \exp(\mathbf{x}_i^\top \boldsymbol{\beta})} \quad \text{for } y_i \in \{0, 1\} \quad (9.14)$$

The data augmentation trick is to expand each $\Pr(y_i = y_i | \mathbf{x}_i, \boldsymbol{\beta})$ factor using the integral identity

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = 2^{-b} e^{\kappa\psi} \int_0^\infty e^{\omega\psi^2/2} p(\omega) d\omega,$$

where $\kappa = a - b/2$ and $p(\omega)$ is the density of a **Pólya-Gamma** random variable, $\omega \sim \text{PG}(b, 0)$. The Pólya-Gamma distribution is defined as a weighted sum of Gamma distributed variables, see Box 9.6 for details, and was specifically developed for the purpose of data augmentation in logistic regression. Now, setting $a = y_i$, $b = 1$ and $\psi = \mathbf{x}_i^\top \boldsymbol{\beta}$ in (9.3) we can write each likelihood factor in (9.14) as

$$\frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta})^{y_i}}{1 + \exp(\mathbf{x}_i^\top \boldsymbol{\beta})} = \frac{1}{2} e^{\kappa_i \mathbf{x}_i^\top \boldsymbol{\beta}} \int_0^\infty e^{\omega_i (\mathbf{x}_i^\top \boldsymbol{\beta})^2/2} p(\omega_i) d\omega_i,$$

where $\kappa_i = y_i - 1/2$ and $p(\omega_i)$ is the density of a $\text{PG}(1, 0)$ distribution for the i th augmentation variable ω_i .

Pólya-Gamma distribution

$X \sim \text{PG}(b, c)$ for $X > 0$.

A Pólya-Gamma is defined as a infinite weighted sum (convolution) of iid Gamma distributed variables

$$X \stackrel{d}{=} \sum_{k=1}^{\infty} v_k Y_k$$

where $\stackrel{d}{=}$ mean equality in distribution, the weights are

$$v_k = \frac{1}{2(k-1/2)^2 \pi^2 + c^2/2}$$

and $Y_k \stackrel{\text{iid}}{\sim} \text{Gamma}(b, 1)$.

Box 9.6: The Pólya-gamma distribution.

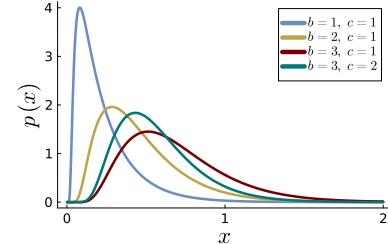


Figure 9.11: Some Pólya-Gamma distributions.

Pólya-Gamma

The updating step for the Gibbs sampler will draw β conditional on the augmentation variables $\omega = (\omega_1, \dots, \omega_n)$ from the full conditional posterior $p(\beta|\omega, \mathbf{y}, \mathbf{X})$. The likelihood function in (9.13) conditional on ω is

$$\prod_{i=1}^n \frac{\exp(\mathbf{x}_i^\top \beta)^{y_i}}{1 + \exp(\mathbf{x}_i^\top \beta)} \propto \prod_{i=1}^n e^{\kappa_i \mathbf{x}_i^\top \beta} e^{\omega_i (\mathbf{x}_i^\top \beta)^2 / 2} = \exp \left(\sum_{i=1}^n \kappa_i \mathbf{x}_i^\top \beta + \frac{\omega_i (\mathbf{x}_i^\top \beta)^2}{2} \right).$$

This expression is the exponential of a quadratic form in β , which by the usual completing of squares can be shown to be proportional to a multivariate normal density. Assuming a multivariate normal prior $\beta \sim N(\mu_0, \Sigma_0)$, it is then easy to show that the full conditional posterior of β is a multivariate normal distribution.

The augmentation variables $\omega_1, \dots, \omega_n$ are updated in a separate Gibbs sampling step, where they are drawn independently from Pólya-Gamma distributions. This derivation is not shown here, but the interested reader can find the details in Polson et al. (2013).

The Gibbs sampling using Pólya-Gamma data augmentation iterates between the following two blocks:

$$\omega_i | \beta, \mathbf{y}, \mathbf{X} \sim \text{PG}(1, \mathbf{x}_i^\top \beta), \quad i = 1, \dots, n \quad (9.15)$$

$$\beta | \omega, \mathbf{y}, \mathbf{X} \sim N(\mu_n, \Sigma_n), \quad (9.16)$$

where $\text{PG}(b, c)$ is the Pólya-Gamma distribution in Box 9.6,

$$\begin{aligned} \Sigma_n &= (\mathbf{X}^\top \Omega \mathbf{X} + \Sigma_0^{-1})^{-1} \\ \mu_n &= \Sigma_n (\mathbf{X}^\top \kappa + \Sigma_0^{-1} \mu_0), \end{aligned}$$

$\kappa = (y_1 - 1/2, \dots, y_n - 1/2)^\top$ and $\Omega = \text{Diag}(\omega_1, \dots, \omega_n)$. The complete Gibbs sampler is summarized in Box 9.7.

Both the Gibbs sampler with data-augmentation for Probit regression in the previous section and the Pólya-Gamma data augmentation for logistic regression can be easily extended to **binomial regression**. This model is suitable when each response observation is a count $y_i | \mathbf{x}_i \sim \text{Bin}(n_i, p_i)$ from a Bernoulli trial with success probability for the i th observation modeled by $p_i = \Phi(\mathbf{x}_i^\top \beta)$ in probit regression or

$$p_i = \frac{\exp(\mathbf{x}_i^\top \beta)}{1 + \exp(\mathbf{x}_i^\top \beta)}.$$

in logistic regression. This would for example be a useful model if y_i is the number of successfully growing plants in an area where n_i seeds have been planted, with the covariates \mathbf{x}_i being conditions, such as water, fertiliser and type of soil, for the i th planting area.

binomial regression

Gibbs sampling for logistic regression using Pólya-Gamma augmentation

Input: response vector $\mathbf{y} = (y_1, \dots, y_n)^\top$
 matrix ($n \times p$) with covariates \mathbf{X}
 initial value $\boldsymbol{\beta}^{(0)}$
 number of posterior draws m .

$$\boldsymbol{\kappa} \leftarrow (y_1 - 1/2, \dots, y_n - 1/2)^\top$$

for k in $1:m$ **do**

- // Update Pólya-Gamma variables
- for** i in $1:n$ **do**
- $\omega_i^{(k)} | \boldsymbol{\beta}^{(k-1)}, \mathbf{y}, \mathbf{x}_i \sim \text{PG}(1, \mathbf{x}_i^\top \boldsymbol{\beta}^{(k-1)})$
- end**
- $\Omega^{(k)} \leftarrow \text{Diag}(\omega_1^{(k)}, \dots, \omega_n^{(k)})$
- // Update $\boldsymbol{\beta}$
- $\Sigma_n \leftarrow (\mathbf{X}^\top \Omega^{(k)} \mathbf{X} + \Sigma_0^{-1})^{-1}$
- $\mu_n \leftarrow \Sigma_n (\mathbf{X}^\top \boldsymbol{\kappa} + \Sigma_0^{-1} \mu_0)$
- Draw $\boldsymbol{\beta}^{(k)} | \omega$ from $N(\mu_n, \Sigma_n)$

end

Output: m draws $\boldsymbol{\beta}^{(1)}, \dots, \boldsymbol{\beta}^{(m)}$ from the posterior distribution $p(\boldsymbol{\beta} | \mathbf{y}, \mathbf{X})$.

Box 9.7: Gibbs sampling algorithm for the logistic regression model using augmentation with Pólya-Gamma variables, $\omega_1, \dots, \omega_n$. The vector $\mathbf{X}^\top \boldsymbol{\kappa} + \Sigma_0^{-1} \mu_0$ can be pre-computed before the Gibbs sampler iterations.

9.4 Autoregressive processes

We will here develop a Gibbs sampling algorithm for the posterior distribution $p(\mu, \phi_1, \dots, \phi_p, \sigma^2 | y_{1:T})$ of the parameters in the **autoregressive model** of order p

$$y_t = \mu + \phi_1(y_{t-1} - \mu) + \dots + \phi_p(y_{t-p} - \mu) + \varepsilon_t, \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \sigma^2), \quad (9.17)$$

where y_{t-k} is the k th **lagged value** of time series. We will assume that p pre-sample values y_{-p+1}, \dots, y_0 are available, so that the lags y_{t-1}, \dots, y_{t-p} in (9.17) are available for all $t = 1, \dots, T$. The usual procedure is to use the first p values of the time series as pre-sample values.

Recall that the prior proposed in Chapter [Priors](#) for the autoregressive process is of the form

$$\begin{aligned} \mu &\sim N(m_0, \tau_\mu^2), \\ \phi | \sigma^2 &\sim N(\mu_0, \sigma^2 \Omega_0^{-1}) \\ \sigma^2 &\sim \text{Inv-}\chi^2(v_0, \sigma_0^2), \end{aligned} \quad (9.18)$$

where $\Omega_0^{-1} = \text{Diag}(\lambda^2, \lambda^2/2^2, \dots, \lambda^2/p^2)$ is a diagonal matrix with diagonal elements that decay with the lag length to encourage more shrinkage toward zero on the ϕ_k for longer lags, and $\lambda > 0$ is the prior standard deviation of the coefficient on the first lag, ϕ_1 . Note that the prior on ϕ and σ is assumed to be independent of μ , and that the joint prior $p(\phi, \sigma)$ is exactly the conjugate prior for the linear Gaussian regression model in Chapter [Linear Regression](#).

The main complication with deriving the joint posterior distribution $p(\mu, \phi_1, \dots, \phi_p, \sigma^2 | y_{1:T})$ in closed form is that the likelihood involves products of parameter pairs $\phi_k \mu$ for $k = 1, \dots, p$, and products of random variables are usually complicated. However, here is where Gibbs sampling comes to the rescue. Recall that Gibbs sampling only needs tractable posterior distributions for each parameter *conditional* on the other parameters. For example, once we condition on μ , the posterior for each ϕ_k and also for σ would be well known distributions, and it turns out that also the posterior for μ is a well known distribution, once we condition on all other model parameters. Let us now derive the full conditional posterior distributions needed for a Gibbs sampling algorithm.

Conditional on μ we can rewrite the model in (9.17) as a homoscedastic Gaussian linear regression without intercept

$$\tilde{y}_t = \tilde{\mathbf{x}}_t^\top \boldsymbol{\phi} + \varepsilon_t, \quad \varepsilon_t \stackrel{iid}{\sim} N(0, \sigma^2), \quad (9.19)$$

where $\tilde{y}_t = y_t - \mu$ and $\tilde{\mathbf{x}}_t = (y_{t-1} - \mu, \dots, y_{t-p} - \mu)^\top$ and $\boldsymbol{\phi} = (\phi_1, \dots, \phi_p)^\top$. This shows that the autoregressive model can be expressed as the linear regression in (9.19) *conditional on μ* . Recall that

autoregressive model

lagged value

we have assumed that p pre-sample values y_{-p+1}, \dots, y_0 are available, so the sample size in the regression in (9.19) is T . Since the prior for ϕ and σ^2 in (9.18) is exactly of the conjugate form for linear regression, it is immediately clear that we can sample from the joint posterior of ϕ and σ^2 conditional on μ by using the result in Figure ?? in Chapter [Linear Regression](#). The vector of autoregressive parameters ϕ now plays the role of the vector of regression coefficients β and the regression data \mathbf{y} and \mathbf{X} is constructed using model (9.19), i.e. the data with the tilde sign \sim over it; note that this data depends on μ and therefore needs to be recomputed in the algorithm every time a new draw of μ is obtained.

The remaining question is then what the full conditional posterior $p(\mu|\phi, \sigma^2, \mathbf{y})$ looks like. To derive that, note that the model in (9.17) can be rewritten by moving all the $\phi_k y_{t-k}$ terms to the left hand side to obtain

$$y_t - \phi_1 y_{t-1} - \dots - \phi_p y_{t-p} = \mu(1 - \phi_1 - \dots - \phi_p) + \varepsilon_t. \quad (9.20)$$

So by dividing both sides by $1 - \phi_1 - \dots - \phi_p$ we obtain

$$\check{y}_t = \mu + \check{\varepsilon}_t, \quad \check{\varepsilon}_t \stackrel{iid}{\sim} N(0, \check{\sigma}^2), \quad (9.21)$$

where

$$\check{y}_t = \frac{y_t - \phi_1 y_{t-1} - \dots - \phi_p y_{t-p}}{1 - \phi_1 - \dots - \phi_p}, \quad (9.22)$$

and $\check{\sigma}^2 = \sigma^2 / (1 - \phi_1 - \dots - \phi_p)^2$. The model in (9.21) says that, *conditional on ϕ and σ^2* , the transformed data $\check{\mathbf{y}}_{1:T}$ follow an iid Gaussian model with mean μ

$$\check{y}_t \stackrel{iid}{\sim} N(\mu, \check{\sigma}^2). \quad (9.23)$$

Since we condition on σ^2 and ϕ , and therefore also on $\check{\sigma}^2$, the conditional posterior of μ is then just the posterior for a mean in an iid Gaussian model with known variance, something we obtained already back in Chapter [Single-parameter models](#)

$$\mu | \mathbf{y}, \phi, \sigma^2 \sim N(\mu_T, \tau_T^2),$$

where $\tau_T^{-2} = \frac{T}{\check{\sigma}^2} + \tau_\mu^{-2}$, $\mu_T = w\bar{\check{y}} + (1-w)m_0$, $w = \frac{T}{\check{\sigma}^2} / (\frac{T}{\check{\sigma}^2} + \tau_\mu^{-2})$ and the little messy symbol $\bar{\check{y}}$ is the sample mean of the transformed response data $\check{\mathbf{y}}_{1:T}$. Note that the full conditional posterior distribution for μ is conditional on ϕ and σ^2 via the transformed data $\check{\mathbf{y}}_{1:T}$, so the transformed data need to be re-computed every time a new draw of ϕ and σ^2 is obtained. The same applies to the transformed data $\check{\mathbf{y}}$ and $\tilde{\mathbf{X}}$ in connection to (9.19) which needs to be recomputed after each update of μ .

Gibbs sampling for AR processes

```

Input: initial value  $\mu^{(0)}$   

        number of posterior draws  $m$ .  

for  $i$  in  $1:m$  do  

    // Draw from  $p(\phi, \sigma^2 | \mu, \tilde{\mathbf{y}}, \tilde{\mathbf{X}})$   

    for  $t$  in  $1:T$  do  

         $\tilde{y}_t = y_t - \mu^{(i-1)}$  and  

         $\tilde{\mathbf{x}}_t = (y_{t-1} - \mu^{(i-1)}, \dots, y_{t-p} - \mu^{(i-1)})^\top$   

    end  

    Set up  $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_T)^\top$  and  $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_T)^\top$   

    Sample  $(\sigma^2)^{(i)} | \mu^{(i-1)}, \tilde{\mathbf{y}}, \tilde{\mathbf{X}} \sim \text{Inv-}\chi^2(\nu_T, \sigma_T^2)$   

    Sample  $\phi^{(i)} | (\sigma^2)^{(i)}, \mu^{(i-1)}, \tilde{\mathbf{y}}, \tilde{\mathbf{X}} \sim N(\boldsymbol{\mu}_\phi, \sigma^2 \boldsymbol{\Omega}_\phi^{-1})$   

    // Draw from  $p(\mu | \phi, \sigma^2, \tilde{\mathbf{y}}, \tilde{\mathbf{X}})$   

    for  $t$  in  $1:T$  do  

         $\check{y}_t = \frac{y_t - \phi_1^{(i)} y_{t-1} - \dots - \phi_p^{(i)} y_{t-p}}{1 - \phi_1^{(i)} - \dots - \phi_p^{(i)}}$   

    end  

    Sample  $\mu^{(i)} | \phi^{(i)}, (\sigma^2)^{(i)}, \tilde{\mathbf{y}} \sim N(\mu_T, \tau_T^2)$   

end
Output:  $m$  draws from the joint posterior  $p(\phi, \sigma, \mu | \mathbf{y}_{1:T})$ .

```

Box 9.8: Pseudo code for Gibbs sampling from the joint posterior distribution $p(\phi, \sigma, \mu | \mathbf{y}_{1:T})$ in an autoregressive model. The two loops over the data points should be replaced by fast vectorized operations in a real implementation in a non-compiled language.

APPLICATION TO SWEDISH INFLATION DATA. We will here analyze Swedish inflation data (KPIF) during January, 1995 - December, 2020. The data are in 12-month changes, which is a common measure of yearly inflation. The time series is plotted in Figure 9.12. We fit an AR(4) model for illustration.

We use two different prior for illustration. Both priors have an informative $\mu \sim N(2, 0.25^2)$ prior for the mean inflation, tightly centered over the central bank's inflation target of 2%. The prior of the error variance is taken to be rather noninformative $\sigma^2 \sim \text{Inv-}\chi^2(3, 0.25^2)$. Both priors also use a prior mean for the AR coefficients equal to $\mu_0 = (0.9, 0, 0, 0)^\top$, i.e. the prior is centered over an AR(1) model. The difference in the two priors is in the uncertainty around this prior mean: the informative prior uses $\lambda = 0.2$ while the noninformative uses $\lambda = 1000$. Note that the latter implies a large prior standard deviation even for the fourth lag. The posterior distribution for all model parameters is obtained from the Gibbs sampler in Box 9.8. The marginal posterior densities, obtained from a kernel density estimator, are plotted for both priors in Figure 9.13. Note how the informative priors shrinks AR coefficients for lag 2-4 rather tight around zero. Figure 9.14 plots the predictive distribution for the future 48 months using posterior draws based on the informative prior.

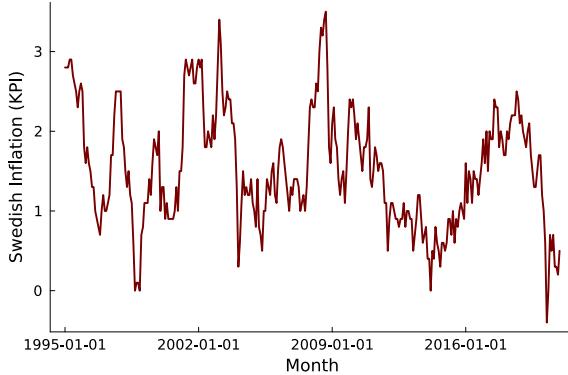


Figure 9.12: Swedish inflation data (KPIF) during January, 1995 - December, 2020. 12-month changes.

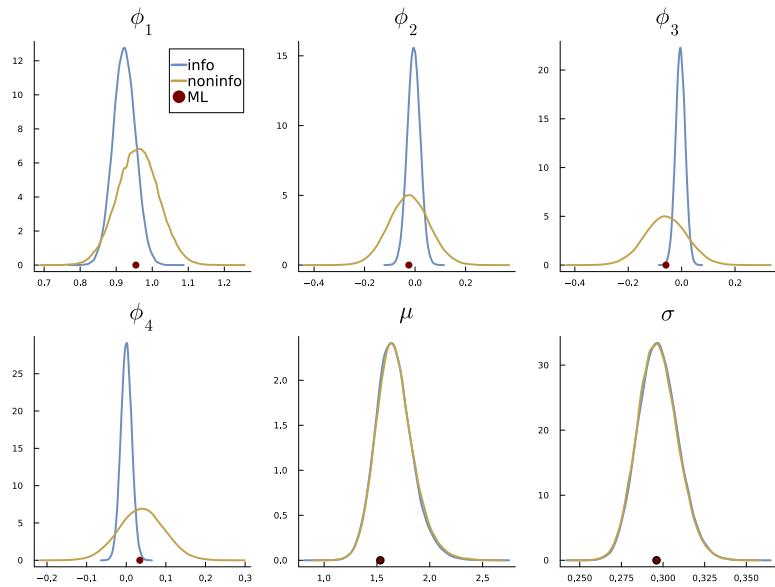


Figure 9.13: Marginal posterior distributions for the AR(4) model parameter fitted to the Swedish inflation data. The marginal densities were estimated by a kernel density estimator from the Gibbs sampling draws. The non-informative prior uses $\lambda = 1000$ whereas the informative prior has $\lambda = 0.2$.

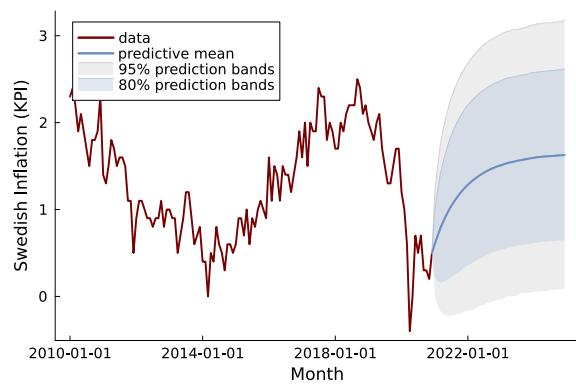


Figure 9.14: Predictive distribution 48-months-ahead for the Swedish inflation data.

10 Markov Chain Monte Carlo simulation

The main challenge in Bayesian inference is to calculate the joint posterior distribution $p(\theta_1, \dots, \theta_p | \mathbf{x})$ and posterior expectations of functions of the parameters $\mathbb{E}_{\theta|\mathbf{x}}(g(\theta))$. We often also want to compute marginal posterior distributions $p(\theta_j | \mathbf{x})$ for each parameter θ_j , and predictive distributions for future observations $p(\tilde{\mathbf{x}}|\mathbf{x})$.

In most models of practical interest, the posterior distribution is not available in closed form and we need to resort to numerical methods; such methods generally fall into three broad classes: i) deterministic approximations, for example, the multivariate normal approximation presented in Chapter 7 or variational approximations in Chapter 11, ii) deterministic numerical integration methods, for example the trapezoidal rule, or iii) simulation-based Monte Carlo methods. Numerical integration can be quite effective for problems with low-dimensional parameters, while Monte Carlo or variational methods are preferred for models with moderate to high-dimensional parameter spaces.

The previous chapter introduced Gibbs sampling as an effective method for simulating from joint posterior distributions. Gibbs sampling requires however that we can sample from the full conditional distributions of each parameter, or blocks of parameters, which is not always possible. In this chapter we will learn about more general simulation methods that do not require sampling from the full conditional distributions.

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms that generate samples from the posterior distribution using a Markov chain, and applies much more generally than Gibbs sampling. The MCMC samples can be used to approximate posterior expectations, marginal posteriors and to sample from predictive distributions and make decisions that maximize posterior expected utility.

Before we discuss MCMC methods in detail, we first introduce two simpler Monte Carlo methods: importance sampling and rejection sampling. Both methods rely on drawing independent samples from a so called proposal distribution and then either weigh (importance

sampling) or accept/reject the samples (rejection sampling). Both methods are useful in their own right, and also help in understanding the MCMC methods presented later in the chapter.

10.1 Monte Carlo

The basic Monte Carlo method was introduced already in Chapter 2 where m independent draws $\theta^{(1)}, \dots, \theta^{(m)}$ from the posterior distribution $p(\theta|y)$ was used to approximate the posterior expectation $E_{\theta|y} f(\theta) = \int f(\theta) p(\theta|y) d\theta$ by a sample mean of function evaluations

$$\hat{E}_{\theta|y} f(\theta) := \frac{1}{m} \sum_{i=1}^m f(\theta^{(i)}). \quad (10.1)$$

In Bayesian learning we are mainly interested in using Monte Carlo for approximating posterior distributions and posterior expectations. However, to keep the presentation here more general, and to simplify notation, we will consider the more generic formulation of approximating the expectation of a function $f(x)$ where x is a random vector with density $p(x)$; the case with discrete x is handled analogously with sums instead of integrals. We will write $\mu_f := E_p(f(x))$ to explicitly show that the expectation is with respect to the distribution $p(x)$. In the Bayesian context, x corresponds to the model parameters θ , and $p(x)$ to the posterior distribution $p(\theta|y)$. We assume for simplicity that the output of $f(x)$, and therefore μ_f , is one-dimensional; the extension to multi-dimensional outputs is immediate and mentioned when needed.

The **Monte Carlo estimator** draws x_1, \dots, x_M iid from $p(x)$ and estimates $\mu_f = E_p(f(x))$ with a sample mean of the function evaluations

$$\hat{\mu}_f^{\text{MC}} := \frac{1}{m} \sum_{i=1}^m f(x^{(i)}). \quad (10.2)$$

The Monte Carlo estimator is unbiased and consistent for μ_f . Consistency means that we can get arbitrarily close to the true expectation μ_f by sampling enough draws. Formally, we say that

$$\hat{\mu}_f^{\text{MC}} \xrightarrow{p} \mu_f \quad \text{as } m \rightarrow \infty,$$

where \xrightarrow{p} denotes convergence in probability. This result follows from the (weak) Law of Large Numbers (LLN), assuming that the posterior expectation μ_f is finite. There is also a stronger version of the convergence result, which states that the sample mean converges *almost surely* to μ_f .

We can quantify the typical estimation error of $\hat{\mu}_f^{\text{MC}}$ using the Central Limit Theorem (CLT). The CLT in Figure ?? says the standardized

Monte Carlo estimator

sample mean converges in distribution to the standard normal distribution as the sample size approaches infinity. This suggests the following approximate distribution for large m :

$$\hat{\mu}_f^{\text{MC}} \stackrel{\text{approx}}{\sim} N\left(\mu_f, \frac{\mathbb{V}_p(f(\mathbf{x}))}{m}\right), \quad (10.3)$$

if the variance $\mathbb{V}_p(f(\mathbf{x})) = \int (f(\mathbf{x}) - \mu_f)^2 p(\mathbf{x}) d\mathbf{x}$ is finite. The variance in the CLT can be estimated from the sample variance of the function evaluations

$$\hat{\mathbb{V}}_p(f(\mathbf{x})) := \frac{1}{m-1} \sum_{i=1}^m (f(\mathbf{x}^{(i)}) - \hat{\mu}_f^{\text{MC}})^2.$$

The normal approximation in (10.3) can be used to construct approximate confidence intervals for the posterior expectation. For example, an approximate 95% confidence interval is given by

$$\hat{\mu}_f^{\text{MC}} \pm 1.96 \sqrt{\frac{\hat{\mathbb{V}}_p(f(\mathbf{x}))}{m}}. \quad (10.4)$$

The extension to a multi-output function $\mathbf{f}(\mathbf{x})$ with a d -dimensional output is straightforward. Such functions are common, for example $\mathbf{f}(\mathbf{x}) = \mathbf{x}$ to approximate the mean vector $\mathbb{E}_p(\mathbf{x})$. The Monte Carlo estimator is

$$\hat{\mu}_f^{\text{MC}} := \frac{1}{m} \sum_{i=1}^m \mathbf{f}(\mathbf{x}^{(i)}),$$

and the CLT approximation is

$$\hat{\mu}_f^{\text{MC}} \stackrel{\text{approx}}{\sim} N\left(\mu_f, \frac{1}{m} \mathbb{V}(\mathbf{f}(\mathbf{x}))\right),$$

where $\mathbb{V}(\mathbf{f}(\mathbf{x})) = \mathbb{E}_p((\mathbf{f}(\mathbf{x}) - \mu_f)(\mathbf{f}(\mathbf{x}) - \mu_f)^\top)$ is the covariance matrix of $\mathbf{f}(\mathbf{x})$, which can be estimated by the sample covariance matrix of the function evaluations

$$\widehat{\mathbb{V}(\mathbf{f}(\mathbf{x}))} := \frac{1}{m-1} \sum_{i=1}^m (\mathbf{f}(\mathbf{x}^{(i)}) - \hat{\mu}_f^{\text{MC}})(\mathbf{f}(\mathbf{x}^{(i)}) - \hat{\mu}_f^{\text{MC}})^\top.$$

Note that the sampling covariance of the MC estimator approaches zero with the same $1/m$ rate as in the univariate case, which is an important property of Monte Carlo when working in high-dimensional spaces.

Note that the above analysis with standard errors of the MC estimator and confidence intervals are frequentist in nature. It can therefore be a bit confusing to use frequentist methods to estimate, for example, a Bayesian posterior variance $\mathbb{V}_p f(\mathbf{x})$. This is the typical approach however, implicitly motivated by the fact that the Monte Carlo sample size m is typically very large. The reader is strongly advised to clearly separate these two sources of uncertainty:

1. the **intrinsic inference problem** uses a dataset $\mathbf{y} = (y_1, \dots, y_n)$ with a *fixed data sample size* n we quantify the uncertainty about an unknown function of the parameters $f(\theta)$ with a posterior distribution $p(f(\theta)|y_1, \dots, y_n)$.
2. the **numerical problem** uses Monte Carlo with a *large user-controlled Monte Carlo sample size* m to estimate the posterior distribution and posterior expectations, with the numerical uncertainty quantified by frequentist standard errors and confidence intervals.

It is important for mental clarity to separate these two issues, and to first think about the posterior distribution without worrying about the numerical method to approximate it.

There is a separate literature on Bayesian numerical methods, often called Bayesian numerics or Probabilistic numerics, which is not discussed here; the interested readers is referred to the book [Hennig et al. \(2022\)](#). Such methods can be very useful when each evaluation of the function $f(\mathbf{x})$ is very expensive, for example when $f(\mathbf{x})$ itself involves solving a complex optimization problem or a system of differential equations, since we are then forced to use a small Monte Carlo sample size m .

EXAMPLE: Let $X \sim N(0, 1)$ and we want to compute $\mathbb{E}(\exp(X))$ using Monte Carlo. The exact value $\mathbb{E}(\exp(X)) = \exp(1/2) \approx 1.6487$ is known in this case from the mean of the log normal distribution, $\exp(X) \sim LN(0, 1)$ in Figure ???. We can estimate the expectation using the basic Monte Carlo estimator by drawing m independent draws from the $N(0, 1)$ distribution and computing the sample mean of the $\exp(X)$ evaluations

$$\hat{\mu}_f^{\text{MC}} = \frac{1}{m} \sum_{i=1}^m \exp(X^{(i)}), \quad X^{(i)} \stackrel{\text{iid}}{\sim} N(0, 1),$$

where $f(X) = \exp(X)$. We simulate $m = 1000$ draws from $N(0, 1)$ and obtain the estimate $\hat{\mu}_f^{\text{MC}} \approx 1.755$. The sample variance of the function evaluations is 5.705 and the standard error of the Monte Carlo estimate is therefore $\sqrt{5.705/1000} \approx 0.076$. Using (10.4), an approximate 95% confidence interval for $\mathbb{E}(\exp(X))$ is

$$1.755 \pm 1.96 \cdot 0.076 = (1.606, 1.904).$$

The confidence interval contains the exact value of $\exp(1/2) \approx 1.6487$. Figure 10.1 shows how the Monte Carlo estimate converges toward the exact mean as the number of draws m increases, for three independent replicates of the simulation. The figure also shows that the Monte Carlo estimate can be quite variable for small m , but that the variability decreases as m increases.

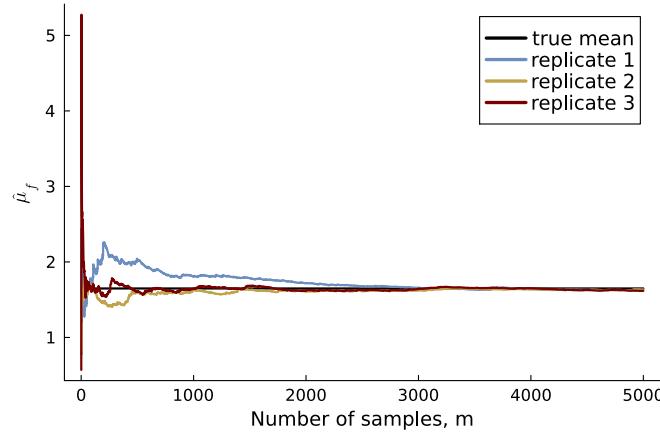


Figure 10.1: Monte Carlo estimates of the mean

$f(X) = \exp(X)$ where $X \sim N(0, 1)$ as a function of the number of draws m . The colored lines show the cumulative estimates from three independent different runs. The black horizontal line shows the exact mean.

EXAMPLE: Consider the eBay dataset from Chapter 2 with data on the number of bidders in $n = 1000$ coin auctions. Let $Y_i | \theta \stackrel{\text{iid}}{\sim} \text{Poisson}(\theta)$ be the number of bidders in the i th auction. We use the conjugate prior $\theta \sim \text{Gamma}(\alpha, \beta)$ with $\alpha = 2$ and $\beta = 1/2$. The posterior distribution from a sample $\mathbf{y} = (y_1, \dots, y_n)^\top$ is then $\theta | y_1, \dots, y_n \sim \text{Gamma}(\alpha + \sum_{i=1}^n y_i, \beta + n)$ and the posterior expectation is $\mathbb{E}(\theta | \mathbf{y}) = (\alpha + \sum_{i=1}^n y_i) / (\beta + n) \approx 3.635$. Let us pretend that we cannot compute the posterior mean analytically, and instead estimate it by Monte Carlo. Using $m = 100$ posterior draws we obtain the estimate $\bar{\theta}_m \approx 3.640$ with standard error 0.006, which gives the approximate 95% confidence interval $(3.628, 3.652)$. The Monte Carlo estimate is quite close to the exact posterior mean of 3.635, and the confidence interval contains the exact posterior mean.

10.2 Importance sampling

The Monte Carlo estimator in (10.2) requires that we can draw independent samples from the distribution $p(\mathbf{x})$, which can be expensive or even impossible. This is often the case in Bayesian inference where $p(\mathbf{x})$ is an intractable posterior distribution. Even when sampling from $p(\mathbf{x})$ is feasible, the Monte Carlo estimator can also have a large variance in certain cases.

Importance sampling (IS) is a Monte Carlo method that instead draws from a **proposal distribution** $q(\mathbf{x})$ and estimates the mean $\int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$ of a function $f(\mathbf{x})$ by a *weighted* average of the function evaluations. With a carefully chosen proposal distribution, the IS estimator can often have much smaller variance than the basic Monte Carlo estimator. The importance sampling estimator is based on the

proposal distribution

identity

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})w(\mathbf{x})q(\mathbf{x})d\mathbf{x}, \quad (10.5)$$

with the weight function $w(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$, i.e. the ratio of the **target distribution** $p(\mathbf{x})$ to the proposal distribution, $q(\mathbf{x})$. The proposal distribution needs to satisfy $q(\mathbf{x}) > 0$ for all \mathbf{x} where $f(\mathbf{x})p(\mathbf{x}) \neq 0$. The **importance sampling** method generates m independent draws from the proposal distribution $q(\mathbf{x})$ and estimates the posterior expectation as a weighted average of the function evaluations

$$\hat{\mu}_f^{\text{IS}} := \frac{1}{m} \sum_{i=1}^m f(\mathbf{x}^{(i)})w(\mathbf{x}^{(i)}) \quad (10.6)$$

with weights for each draw $w(\mathbf{x}^{(i)}) = p(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$. The IS estimator is easily shown to be an unbiased estimator for the expectation of $f(\mathbf{x})$ for any finite m , and it is consistent

$$\hat{\mu}_f^{\text{IS}} \xrightarrow{P} \mu_f \quad \text{as } m \rightarrow \infty.$$

Since the IS estimator draws from a distribution that is different from the target density $p(\mathbf{x})$, we need to assign weights to the draws based on how much more probable the draws are under the target distribution $p(\mathbf{x})$ compared to the proposal distribution $q(\mathbf{x})$. For example, if a draw has high target density but low proposal density, it should be upweighted since such draws occur too seldom. Similarly, if a draw has high proposal density but low target density, it should be given a low weight to adjust of the fact that the proposal generates too many draws in that region.

If $f(\mathbf{x}) \geq 0$, then it is easy to see that the ideal importance sampling proposal distribution is $q(\mathbf{x}) = f(\mathbf{x})p(\mathbf{x})/\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$, i.e. the proposal should be proportional to the integrand $f(\mathbf{x})p(\mathbf{x})$. This proposal gives an IS estimator with zero variance. The optimal proposal is however not useful in practice since it depends on the unknown expectation $\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$ that we want to estimate, but it shows the ideal that we want to come close to. When the function can also return negative values, the optimal proposal is $q(\mathbf{x}) = |f(\mathbf{x})|p(\mathbf{x})/\int |f(\mathbf{x})|p(\mathbf{x})d\mathbf{x}$.

EXAMPLE: Let us again compute $\mathbb{E}(\exp(X))$ where $X \sim N(0, 1)$, this time using importance sampling with proposal distribution $q(x) = N(x|\tilde{\mu}, 1)$ for some $\tilde{\mu} \neq 0$. To see what constitutes good values for $\tilde{\mu}$, note that

$$\mathbb{E}(\exp(X)) = \int_{-\infty}^{\infty} \exp(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) dx, \quad (10.7)$$

target distribution

importance sampling

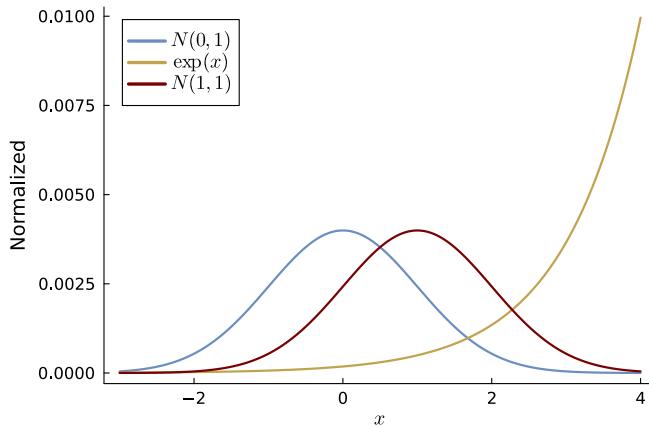


Figure 10.2: Illustrating the difference between the integrands $N(x|0, 1)$ and $\exp(x)N(x|0, 1) \propto N(x|1, 1)$, where $N(x|\mu, \sigma^2)$ is the pdf of the normal distribution with mean μ and variance σ^2 . All three curves have been normalized to integrate to one, for visualization purposes.

and that the integrand $f(\mathbf{x})p(\mathbf{x})$ can be rewritten as

$$\begin{aligned} \exp(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x-1)^2 + \frac{1}{2}\right) \\ &= \frac{e^{1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x-1)^2\right), \end{aligned}$$

which is proportional to the density for a $N(1, 1)$ distribution. This suggests that a good choice of proposal distribution is $q(x) = N(x|\tilde{\mu}, 1)$, with $\tilde{\mu} = 1$. Figure 10.2 illustrates how the multiplication of the $N(0, 1)$ target density with the function $f(x) = \exp(x)$ shifts the integrand $f(x)p(x)$ to the right, so that it is proportional to a $N(1, 1)$ density. Figure 10.3 plots kernel density estimates of the sampling distribution of the basic Monte Carlo estimator and the importance sampling estimator for different values of $\tilde{\mu}$ based on $n_{\text{Rep}} = 10000$ replicates with $m = 1000$ draws in each replicate. The figure shows that the variance of the IS estimator decreases as we approach the optimal $\tilde{\mu} = 1$.

The previous example shows that we can improve the efficiency of the estimator by using a proposal distribution that is closer to the integrand $f(x)p(x)$, which is true in general. This ideal is typically never attainable in real problems. It should also be noted that this ideal proposal distribution is tailored to a particular function f , so if we wanted to estimate the mean of another function optimally, we would need to choose a different proposal distribution. This is impractical and most often we want to use a single proposal distribution to estimate the mean of many different functions. A practical compromise is to use a proposal that is close to the target distribution $p(x)$, for all functions $f(x)$ of interest.

The tails of the proposal distribution are very important for the efficiency the IS estimator. A proposal with lighter tails than the target distribution can inflate the variance of the IS estimator, or even

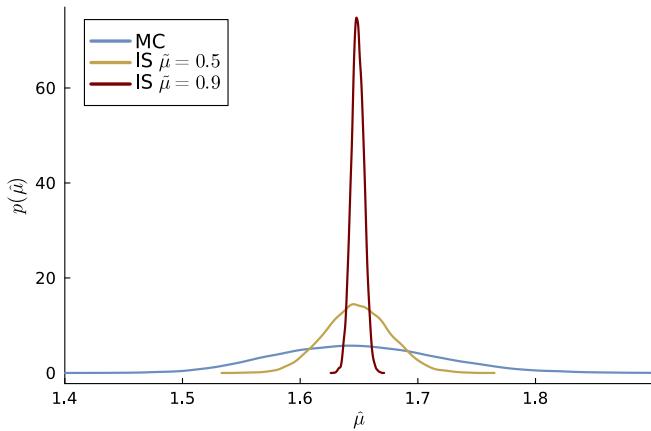
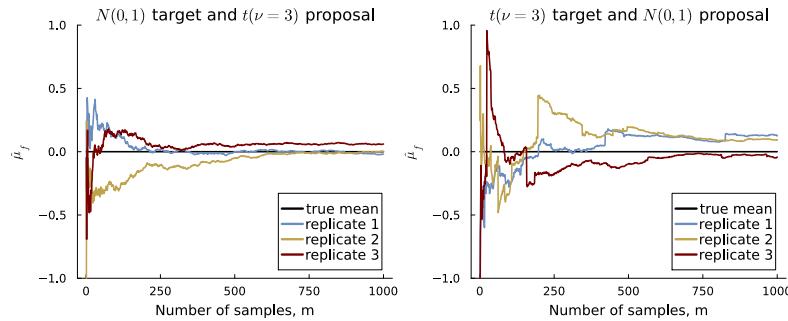


Figure 10.3: Sampling distribution of Monte Carlo (MC) and importance sampling (IS) estimators for $E(\exp(X))$ where $X \sim N(0, 1)$. The proposal distribution for the IS estimators are $N(\tilde{\mu}, 1)$.

produce an infinite variance. It is therefore common to use a heavy-tailed proposal distribution, for example a multivariate t -distribution with low degrees of freedom. The following example illustrates the importance of using a proposal distribution with sufficiently heavy tails.



EXAMPLE: Assume that we want to estimate the mean of a standard normal $p(x) = N(x|0, 1)$ target using IS with a t -distribution with $\nu = 3$ degrees of freedom as proposal; the true mean is of course zero, but we use this example to explore the converge of the IS estimate to this known mean. The left panel of Figure 10.4 shows the convergence of the IS estimator for the mean $E(X)$ in three independent replicate runs with $m = 1000$ draws from the proposal distribution. Since the $t(0, 1, \nu = 3)$ distribution has heavier tails than the $N(0, 1)$ distribution, the IS estimator converges nicely to the true mean of zero as m increases. The right panel of Figure 10.4 shows the opposite situation where we want to estimate the mean of a $t(0, 1, \nu = 3)$ target using a $N(0, 1)$ proposal distribution. The $N(0, 1)$ distribution has lighter tails than the $t(0, 1, \nu = 3)$ distribution, and the variance of the IS estimator is very large; even for

Figure 10.4: Illustrating the importance of the tails of the proposal distribution for the converge of the IS estimator. The left panel shows the IS estimator for the mean of $N(0, 1)$ target density with a $t(0, 1, \nu = 3)$ proposal distribution; here the tails of the proposal are thicker than those of the target and convergence is fine. The right panel shows the opposite situation with a $t(0, 1, \nu = 3)$ target and a thin-tailed $N(0, 1)$ proposal distribution.

$m = 1000$ draws from the proposal distribution, the IS estimate can take very large jumps caused by single draws in the tails of the proposal that get a very large weight. The figure illustrates the importance of using a proposal distribution with sufficiently heavy tails for good performance of the IS estimator.

The basic importance sampling estimator in (10.6) requires that we can evaluate the target density $p(\mathbf{x})$ *including its normalizing constant*. This is often not possible in Bayesian inference where the posterior distribution is typically only known up to a normalizing constant

$$p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta),$$

and the normalization constant $\int p(\mathbf{y}|\theta)p(\theta)d\theta$ is tractable.

Assuming now that $p(\mathbf{x}) = \tilde{p}(\mathbf{x}) / \int \tilde{p}(\mathbf{x})d\mathbf{x}$, where $\tilde{p}(\mathbf{x})$ is an unnormalized target density and the normalization constant $\int \tilde{p}(\mathbf{x})d\mathbf{x}$ is intractable. We can estimate the missing normalizing constant using the same draws from the proposal distribution by

$$\int \tilde{p}(\mathbf{x})d\mathbf{x} = \int \frac{\tilde{p}(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})d\mathbf{x} \approx \frac{1}{m} \sum_{i=1}^m \frac{\tilde{p}(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})} = \frac{1}{m} \sum_{i=1}^m \tilde{w}(\mathbf{x}^{(i)}),$$

where $\tilde{w}(\mathbf{x}^{(i)}) = \tilde{p}(\mathbf{x}^{(i)}) / q(\mathbf{x}^{(i)})$. Inserting this estimate into the IS estimator in (10.6) gives the **self-normalized importance sampling** (SNIS) estimator

$$\hat{\mu}_f^{\text{SNIS}} := \frac{\sum_{i=1}^m f(\mathbf{x}^{(i)}) \tilde{w}(\mathbf{x}^{(i)})}{\tilde{w}(\mathbf{x}^{(i)})} \quad \text{with } \mathbf{x}^{(i)} \stackrel{\text{iid}}{\sim} q(\mathbf{x}). \quad (10.8)$$

The SNIS estimator is a ratio of random variables, so it is not unbiased for finite N , but it is consistent

$$\hat{\mu}_f^{\text{SNIS}} \xrightarrow{p} \mu_f \quad \text{as } m \rightarrow \infty.$$

The variance of the SNIS estimator can be consistently estimated by

$$\hat{\mathbb{V}}(\hat{\mu}_f^{\text{SNIS}}) := \sum_{i=1}^m w^2(\mathbf{x}^{(i)}) (f(\mathbf{x}^{(i)}) - \hat{\mu}_f^{\text{SNIS}})^2,$$

where $w^{(i)}(\mathbf{x}) := \tilde{w}^{(i)}(\mathbf{x}) / \sum_{j=1}^m \tilde{w}^{(j)}$ are the normalized weights. The variance estimator can be used to construct approximate confidence intervals for the posterior expectation, just as for the basic Monte Carlo estimator.

A commonly used diagnostic for the quality of the IS estimator is the **effective sample size** (ESS), which is defined as

self-normalized importance sampling

effective sample size

$$\text{ESS} = \frac{1}{\sum_{i=1}^m w^2(\mathbf{x}^{(i)})}, \quad (10.9)$$

where $w(\mathbf{x}^{(i)})$ are the normalized weights. The ESS can be interpreted as the number of independent draws from the target distribution $p(\mathbf{x})$ that would give the same variance of the IS estimator for the mean $\mathbb{E}_p(x)$ as the current m draws from the proposal distribution $q(\mathbf{x})$. The ESS is always between 1 and m , where m is the total number of draws from the proposal distribution. As an example, if m^* of the draws have weight $w_i = 1/m^*$ and the remaining draws have zero weight, then $\text{ESS} = m^*$. Note that the ESS is for estimating the mean $\mathbb{E}_p(x)$, so the effective sample size for some other function $f(\mathbf{x})$ can be different.

EXAMPLE: Chapter 8 used a normal approximation of the posterior $p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$ in a Poisson regression for the number of bidder in the eBay data. The normal approximation followed the recipe based on numerical optimization in Chapter 7 and was given by

$$\boldsymbol{\beta}|\mathbf{y} \stackrel{\text{approx}}{\sim} N(\tilde{\boldsymbol{\beta}}, \mathbf{J}^{-1}(\tilde{\boldsymbol{\beta}})), \quad (10.10)$$

where $\tilde{\boldsymbol{\beta}}$ is the posterior mode and $\mathbf{J}(\tilde{\boldsymbol{\beta}})$ is the observed information matrix evaluated at the posterior mode. The transformed parameters $\exp(\beta_j)$ were argued to be more interpretable. We will here compute the posterior mean of $\exp(\beta_j)$ using self-normalized importance sampling with the above normal approximation as proposal distribution. This will give us the exact posterior mean of $\exp(\beta_j)$ if we simulate enough draws from the proposal distribution. Table 10.1 shows the SNIS estimates of $\exp(\beta_j)$ for $j = 1, \dots, 8$ based on $m = 1000$ draws from the normal proposal distribution. The table also shows numerical standard errors for the estimates, which are all quite small. The SNIS estimates are very close to the estimates obtained from the normal approximation in Chapter 8 in the column named 'approx', which indicates that the normal approximation is quite accurate in this case. The effective sample size for the SNIS estimates is $\text{ESS} \approx 985$.

To show the importance of a good proposal we also use the SNIS estimator with normal proposal with a mean equal to the maximum likelihood estimate and a diagonal covariance matrix with the sampling variances of each $\hat{\beta}_j$ estimate on the diagonal. This proposal is quite far from the posterior distribution, and the SNIS estimates are then quite different from the normal approximation estimates in Chapter 8, with an effective sample size of only $\text{ESS} \approx 2$. This illustrates the importance of a good proposal distribution for accurate IS estimates.

Importance sampling can also be used to estimate marginal (posterior) distributions by choosing the function $f(\mathbf{x})$ to be an indicator function for an interval. For example, we can use importance sam-

variable	approx	SNIS	std.err
intercept	2.9233	2.9204	0.0029
logbook	0.8864	0.8869	0.0008
startprice	0.1507	0.1505	0.0003
minblemish	0.9515	0.9515	0.0018
majblemish	0.8051	0.8022	0.0023
negfeedback	1.0758	1.0742	0.0019
powerseller	0.9801	0.9809	0.0012
verified	0.6771	0.6718	0.0021
sealed	1.5609	1.5597	0.0024

pling to estimate the marginal probability $\Pr(a < x_j < b)$ by choosing $f(\mathbf{x}) = \mathbb{I}(a < x_j < b)$, where

$$\mathbb{I}(a < x_j < b) = \begin{cases} 1 & \text{if } a < x_j < b, \\ 0 & \text{otherwise,} \end{cases}$$

is the indicator function for the interval (a, b) . By computing the SNIS estimate for many different intervals (a, b) we can estimate the entire marginal distribution of x_j . In the next and the following sections we will learn about other Monte Carlo methods that can be used to simulate from the target distribution $p(\mathbf{x})$, and the samples can then be used to approximate marginal distributions and expectations of functions.

10.3 Rejection sampling

Rejection sampling is a method for simulating from a target distribution $p(\mathbf{x})$ by drawing from a proposal distribution $q(\mathbf{x})$ and accepting the draws with probability proportional to the ratio of the target and proposal densities. With acceptance sampling we obtain a *sample* from the target distribution $p(\mathbf{x})$ rather than an estimate of the posterior expectation of some function as in importance sampling.

Rejection sampling needs a proposal distribution $q(\mathbf{x})$ and a *majorization constant* $M > 0$ such that

$$p(\mathbf{x}) \leq M \cdot q(\mathbf{x}) \quad \text{for all } \mathbf{x}.$$

The proposal distribution should be easy to sample from, and it should be close to the target distribution to obtain a high acceptance rate. A single draw from the target distribution is generated as follows:

1. Draw a candidate \mathbf{x}^* from the proposal distribution $q(\mathbf{x})$.
2. Draw $u \sim \text{Uniform}(0, 1)$.
3. Accept the candidate \mathbf{x}^* if

$$u < \frac{p(\mathbf{x}^*)}{M \cdot q(\mathbf{x}^*)}.$$

Table 10.1: SNIS estimates of $\mathbb{E}_{p(\beta|\mathbf{y})} \exp(\beta_j)$ with numerical standard errors for a Poisson regression fitted to the ebay data. The estimates are based on $m = 1000$ draws from the normal approximation in (10.10) to the posterior as proposal distribution. The column names 'approx' shows the estimates obtained from the normal approximation in Chapter 8.

Rejection sampling

Otherwise, reject the candidate and return to step 1.

It is not hard to show that the number of sampled candidates until one is accepted is a geometric random variable (see Box 2.12) with success probability $1/M$. The expected number of candidates that need to be drawn to obtain one accepted draw from the posterior distribution is therefore M . Hence, the proposal distribution should be as close to the posterior distribution as possible, otherwise M needs to be large.

A good majorization constant M can sometimes be found analytically, but in most cases we need to find M numerically by maximizing the ratio $p(\mathbf{x})/q(\mathbf{x})$ with respect to \mathbf{x} . This is typically done on the log scale to avoid numerical problems. The final M is often padded by multiplying with a small constant, for example 1.05, to be on the safe side. The acceptance probabilities $p(\mathbf{x})/M \cdot q(\mathbf{x})$ should be checked to be less than one for all sampled candidates \mathbf{x} , otherwise M is too small and needs to be increased and the sampling restarted.

EXAMPLE: We use rejection sampling to draw from the Logistic – Beta(α, β) distribution with $\alpha = 1$ and $\beta = 2$. The density of the Logistic – Beta(α, β) distribution is given by

$$p(x) = \frac{\sigma(x)^\alpha \sigma(-x)^\beta}{B(\alpha, \beta)}, \quad x \in \mathbb{R},$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function and $B(\alpha, \beta)$ is the beta function. The Logistic-Beta with $\alpha = 1$ and $\beta = 2$ is slightly skewed to the left. Figure 10.5 illustrates rejection sampling from this distribution using three different proposal distributions. In all three cases do we find a good majorization constant M by numerically maximizing the ratio $p(x)/q(x)$ on the log scale; and padding the final M by multiplying with 1.05. The top panel uses a uniform proposal, which is clearly a poor choice since it does not resemble the target density (top left), and the acceptance rate is therefore only 0.16; nevertheless, the histogram with 10000 accepted draws is very close to the target density (top right). The middle panel uses a student- t proposal with zero mean, unit scale and $\nu = 3$ degrees of freedom, which is better but the proposal is centered at zero while the mode of the target distribution is at $x_{\text{mode}} = \log(\alpha/\beta) \approx -0.693$. The bottom panel uses also a student- t proposal with unit scale and $\nu = 3$ degrees of freedom, but here the location parameter is set equal to the mode of the target distribution; the acceptance rate is then 0.468, which is quite good.

Rejection sampling can be used on unnormalized densities. This is clearly important for Bayesian applications where we often use the proportional form of Bayes' theorem $p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, since the

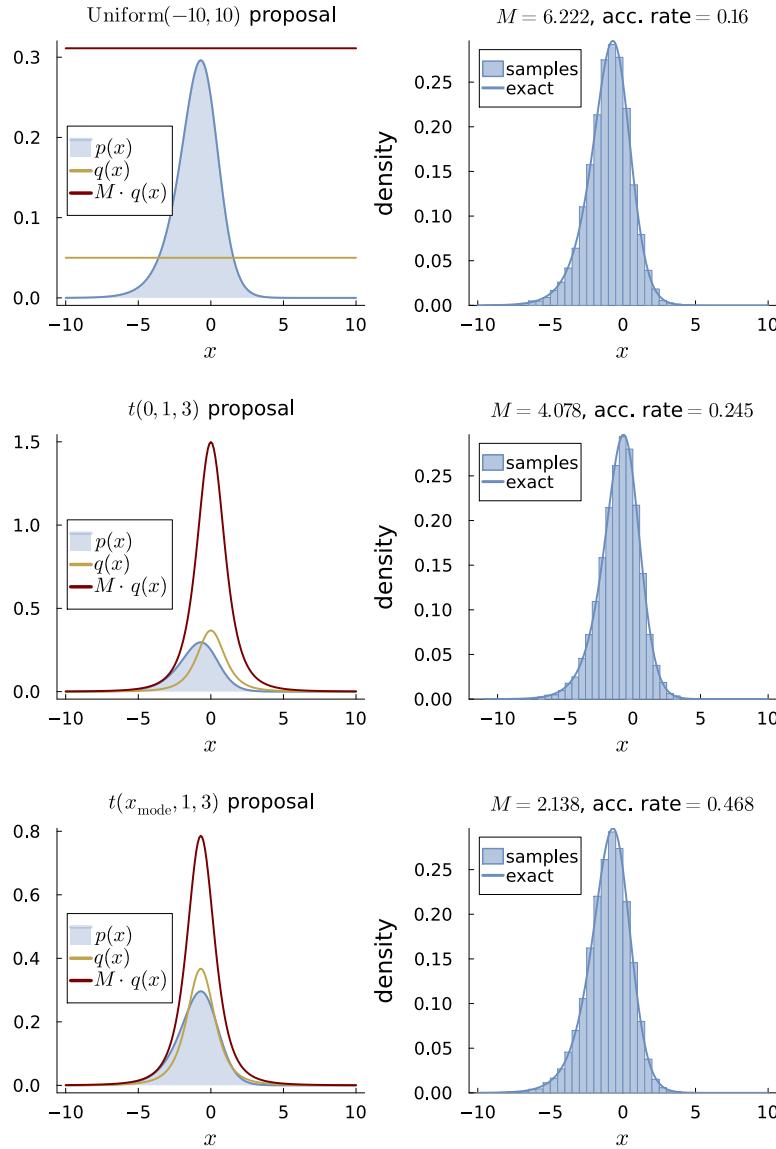


Figure 10.5: Illustrating rejection sampling from an unnormalized Logistic - Beta(α, β) with $\alpha = 1$ and $\beta = 2$. Each row of the graph uses a different proposal distributions. The top panel uses a uniform proposal. The middle panel uses a student- t proposal with zero mean, unit scale and $\nu = 3$ degrees of freedom. The bottom panel uses also a student- t proposal with unit scale and $\nu = 3$ degrees of freedom, but the location parameter is set equal to the mode of the target distribution, which is $x_{\text{mode}} = \log(\alpha/\beta) \approx -0.693$.

normalization constant is typically intractable. We need a proposal distribution $q(\theta)$ and a majorization constant $M > 0$ such that

$$p(\mathbf{y}|\theta)p(\theta) \leq M \cdot q(\theta) \quad \text{for all } \theta.$$

10.4 Markov Chain Monte Carlo

Random Walk Metropolis algorithm

```

Input: data  $\mathbf{y} = (y_1, \dots, y_n)$ 
        number of posterior draws  $m$ 
        number of burn-in draws  $b$ 
        proposal covariance matrix  $\Sigma$ 
        proposal scaling factor  $c > 0$ 
Initialize  $\theta^{(0)}$  and  $p(\mathbf{y}|\theta^{(0)})$ 
for  $i$  in  $1:(m+b)$  do
    Draw proposal  $\theta^* \sim N(\theta^*|\theta^{(i-1)}, c \cdot \Sigma)$ 
     $\alpha \leftarrow \min \left( 1, \frac{p(\mathbf{y}|\theta^*) p(\theta^*)}{p(\mathbf{y}|\theta^{(i-1)}) p(\theta^{(i-1)})} \right)$ 
    Draw  $u \sim \text{Uniform}(0, 1)$ 
    if  $u < \alpha$  then
        |  $\theta^{(i)} = \theta^*$ 
    else
        |  $\theta^{(i)} = \theta^{(i-1)}$ 
    end
end
Output: draws  $\theta^{(b+1)}, \dots, \theta^{(b+m)}$  from  $p(\theta|\mathbf{y})$ .
```

Box 10.1: The Random Walk Metropolis-Hastings algorithm for simulating from the posterior of the model parameters $p(\theta|\mathbf{y})$ with symmetric Gaussian proposal density $N(\theta^*|\theta^{(i-1)}, c \cdot \Sigma)$.

10.5 Hamiltonian Monte Carlo

10.6 Probabilistic programming frameworks

Metropolis-Hastings algorithm

```

Input: data  $\mathbf{y} = (y_1, \dots, y_n)$   

    number of posterior draws  $m$   

    number of burn-in draws  $b$   

    proposal distribution  $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})$   

Initialize  $\boldsymbol{\theta}^{(0)}$  and  $p(\mathbf{y} | \boldsymbol{\theta}^{(0)})$   

for  $i$  in  $1:(m+b)$  do  

    Draw proposal  $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})$   

     $\alpha \leftarrow \min \left( 1, \frac{p(\mathbf{y} | \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{p(\mathbf{y} | \boldsymbol{\theta}^{(i-1)}) p(\boldsymbol{\theta}^{(i-1)})} \frac{q(\boldsymbol{\theta}^{(i-1)} | \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})} \right)$   

    Draw  $u \sim \text{Uniform}(0, 1)$   

    if  $u < \alpha$  then  

         $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$   

    else  

         $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$   

    end  

end  

Output: draws  $\boldsymbol{\theta}^{(b+1)}, \dots, \boldsymbol{\theta}^{(b+m)}$  from  $p(\boldsymbol{\theta} | \mathbf{y})$ .

```

Box 10.2: The Metropolis-Hastings algorithm for simulating from the posterior of the model parameters $p(\boldsymbol{\theta} | \mathbf{y})$ with proposal density $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})$.

The leap-frog algorithm for HMC

Input: data $\mathbf{y} = (y_1, \dots, y_n)$
 number of leap-frog steps, L
 leap-frog step size, ϵ
 initial position/parameter $\boldsymbol{\theta}^{(0)}$
 initial momentum $\boldsymbol{\phi}^{(0)}$

for l in $1:L$ **do**

$$\begin{aligned}\tilde{\boldsymbol{\phi}}^{(l)} &= \boldsymbol{\phi}^{(l-1)} + \frac{\epsilon}{2} \frac{\partial \log p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}^{(l-1)}} \\ \boldsymbol{\theta}^{(l)} &= \boldsymbol{\theta}^{(l-1)} + \epsilon \cdot \mathbf{M}^{-1} \tilde{\boldsymbol{\phi}}^{(l)} \\ \boldsymbol{\phi}^{(l)} &= \tilde{\boldsymbol{\phi}}^{(l)} + \frac{\epsilon}{2} \frac{\partial \log p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\partial \theta_i} \Big|_{\boldsymbol{\theta}^{(l)}}\end{aligned}$$

end

Output: final position and momentum $\boldsymbol{\theta}^L, \boldsymbol{\phi}^{(L)}$.

Box 10.3: The leap-frog algorithm to approximate the continuous-time Hamiltonian dynamics.

The Hamiltonian Monte Carlo algorithm

```

Input: data  $\mathbf{y} = (y_1, \dots, y_n)$ 
    number of posterior draws  $m$ 
    number of burn-in draws  $b$ 
    number of leap-frog steps,  $L$ 
    leap-frog step size,  $\epsilon$ 
    mass matrix for momentum,  $\mathbf{M}$ 
    initial value  $\boldsymbol{\theta}^{(0)}$ 

for  $i$  in  $1:(m+b)$  do
    Draw initial momentum  $\boldsymbol{\phi}_s \sim N(\mathbf{0}, \mathbf{M})$ 
    Compute proposal with leap-frog algorithm
         $\boldsymbol{\theta}^*, \boldsymbol{\phi}^* = \text{LeapFrog}(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi}_s, L, \epsilon, \mathbf{M})$ 
    Accept/reject proposal
        Compute the acceptance probability
         $\alpha \leftarrow \min \left( 1, \frac{p(\mathbf{y}|\boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{p(\mathbf{y}|\boldsymbol{\theta}^{(i-1)}) p(\boldsymbol{\theta}^{(i-1)})} \frac{p(\boldsymbol{\phi}^*)}{p(\boldsymbol{\phi}_s)} \right)$ 
        Draw  $u \sim \text{Uniform}(0, 1)$ 
        if  $u < \alpha$  then
             $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$ 
        else
             $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$ 
        end
    end
Output: draws  $\boldsymbol{\theta}^{(b+1)}, \dots, \boldsymbol{\theta}^{(b+m)}$  from  $p(\boldsymbol{\theta}|\mathbf{y})Function LEAPFROG( $\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi}_s, L, \epsilon, \mathbf{M}$ )
    for  $l$  in  $1:L$  do
         $\tilde{\boldsymbol{\phi}}^{(l)} = \boldsymbol{\phi}^{(l-1)} + \frac{\epsilon}{2} \frac{\partial \log p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}^{(l-1)}}$ 
         $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{(l-1)} + \epsilon \cdot \mathbf{M}^{-1} \tilde{\boldsymbol{\phi}}^{(l)}$ 
         $\boldsymbol{\phi}^{(l)} = \tilde{\boldsymbol{\phi}}^{(l)} + \frac{\epsilon}{2} \frac{\partial \log p(\mathbf{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{\partial \theta_i}|_{\boldsymbol{\theta}^{(l)}}$ 
    end
Output: final position and momentum  $\boldsymbol{\theta}^L, \boldsymbol{\phi}^{(L)}$ .$ 
```

Box 10.4: The Hamiltonian Monte Carlo (HMC) algorithm for simulating from the posterior of the model parameters $p(\boldsymbol{\theta}|\mathbf{y})$ using the leap-frog algorithm to approximate the Hamiltonian dynamics.

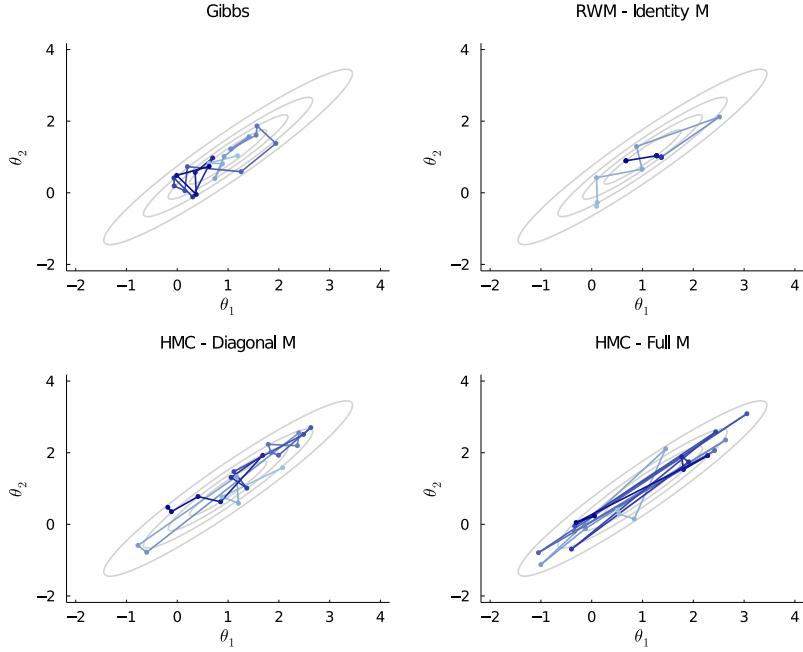


Figure 10.6: Comparing simulation paths of four algorithms for sampling from a bivariate normal target with $\mu = (1, 1)^\top$, unit variances and correlation $\rho = 0.95$. The four compared algorithms are: i) Gibbs sampling, ii) random walk Metropolis with identity scaling, iii) HMC-NUTS with diagonal mass matrix and iv) HMC-NUTS with full mass matrix.

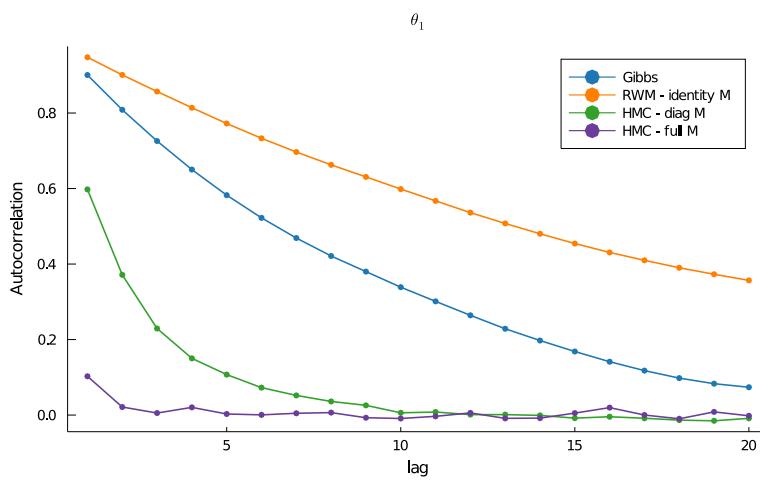


Figure 10.7: Comparing autocorrelation functions from four posterior sampling algorithms for sampling from a bivariate normal target with $\mu = (1, 1)^\top$, unit variances and correlation $\rho = 0.95$. The four compared algorithms are: i) Gibbs sampling, ii) random walk Metropolis with identity scaling, iii) HMC-NUTS with diagonal mass matrix and iv) HMC-NUTS with full mass matrix.

```

using Turing, StatsPlots, Random

# Declare the Turing model:
@model function iidbern(y, α, β)
    θ ~ Beta(α,β) # prior
    N = length(y) # number of observations
    for n in 1:N
        y[n] ~ Bernoulli(θ) # model
    end
end

# Set up the observed data
data = [0,1,1,0,0,1,1,0,1,1]

# Settings for the Hamiltonian Monte Carlo (HMC) sampler.
niter = 10000
nburn = 1000
ε = 0.1
τ = 10

# Sample the posterior using HMC
postdraws = sample(iidbern(data, 1, 2), HMC(ε, τ), niter,
    discard_initial = nburn)
plot(postdraws)

# Print and plot results
display(postdraws)
plot(postdraws)

```

Box 10.5: Turing.jl code for the iid Bernoulli model with a Beta prior.

```

using Turing, StatsPlots, Random
ScaledInverseChiSq(ν, τ²) = InverseGamma(ν/2, ν*τ²/2) # Inv-χ² distribution

# Setting up the Turing model:
@model function iidnormal(x, μ₀, κ₀, ν₀, σ²₀)
    σ² ~ ScaledInverseChiSq(ν₀, σ²₀)
    θ ~ Normal(μ₀, σ²/κ₀) # prior
    n = length(x) # number of observations
    for i in 1:n
        x[i] ~ Normal(θ, √σ²) # model
    end
end

# Set up the observed data
x = [15.77, 20.5, 8.26, 14.37, 21.09]

# Set up the prior
μ₀ = 20; κ₀ = 1; ν₀ = 5; σ²₀ = 5^2

# Settings for the Hamiltonian Monte Carlo (HMC) sampler.
niter = 10000
nburn = 1000
α = 0.65 # target acceptance probability in No U-Turn sampler

# Sample the posterior using HMC
postdraws = sample(iidnormal(x, μ₀, κ₀, ν₀, σ²₀), NUTS(α), niter,
    discard_initial = nburn)

# Print and plot results
display(postdraws)
plot(postdraws)

```

Box 10.6: Turing.jl code for the iid normal model with a conjugate prior.

```

library(rstan)

# Define the Stan model as a string
stanModelNormal = '
// The input data is a vector y of length N.
data {
    // data
    int<lower=0> N;
    vector[N] y;
    // prior
    real mu0;
    real<lower=0> kappa0;
    real<lower=0> nu0;
    real<lower=0> sigma20;
}

// The parameters in the model
parameters {
    real theta;
    real<lower=0> sigma2;
}

model {
    sigma2 ~ scaled_inv_chi_square(nu0, sqrt(sigma20));
    theta ~ normal(mu0,sqrt(sigma2/kappa0));
    y ~ normal(theta, sqrt(sigma2));
}

# Set up the observed data
data <- list(N = 5, y = c(15.77, 20.5, 8.26, 14.37, 21.09))

# Set up the prior
prior <- list(mu0 = 20, kappa0 = 1, nu0 = 5, sigma20 = 5^2)

# Sample from posterior using HMC
fit <- stan(model_code = stanModelNormal, data = c(data,prior), iter = 10000 )

# print and plot results
print(fit, pars = c("theta","sigma2"), probs=c(.1,.5,.9))
pairs(fit)
traceplot(fit, pars = c("theta", "sigma2"), nrow = 2)

```

Box 10.7: Rstan code for the iid normal model with a conjugate prior.

11 Variational inference

12 Regularization

12.1 Model complexity and overfitting

Choosing an appropriate model is one of the most important and difficult tasks in statistical modeling. The model should be complex enough to capture the underlying structure in the data, but not too complex to overfit the data. The choice of a model and its complexity depends on the intended use of the model. On one side of the spectrum is a simple and highly interpretable model that can be used for effective communication between humans; on the other side is a complex model serving mainly as an accurate prediction machine without the need to understand the underlying model mechanisms. Most cases lie somewhere in between these extremes, and model choice is often a trade-off between interpretability and prediction accuracy.

Models with a small number of parameters, for example linear regression and classification models, are simple to interpret and less likely to overfit the data. Model **overfitting** occurs when a fitted model is more complex than the underlying data generating process. An overfitted model loses track of the general tendencies in the data and tries too hard to capture individual observations. An overfitted model will therefore generalize poorly to new data points that were not included in the estimation. On the other hand, using a too simple model runs the risk of **underfitting** the data. An underfitted model fails to appropriately capture the underlying structure in the data, and will therefore also generalize poorly to new data. The aim is to find the sweet spot where the model has just the right balance between complexity and simplicity for the problem at hand.

To illustrate under- and overfitting in a simple nonlinear regression model, let us fit a Gaussian polynomial regression model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

to the `mtcars` data, which is a built-in dataset in R with data on 32 cars of different brands and specifications. We will use miles per gallon, `mpg`, as the response and the horsepower of the car, `hp`, as a

overfitting

underfitting

single explanatory variable. To have a numerically stable solution the variable hp is scaled by dividing by 100 so that hp measures hundreds of horsepowers. We use maximum likelihood to fit the model, which in this setting is the same as a least squares fit. Figure 12.1 plots the data and the fit from a linear regression model. The linear model in the top left graph is too simple to capture the non-linear relationship between mpg and hp ; the linear model is underfitting the data. The fit improves a lot when a second degree polynomial is used, and the similar fits are also produced with $p = 3$ and $p = 4$ as polynomial orders. However, with $p = 5$ there are signs of overfitting with the curve bending down beyond $hp=3$, and for $p = 8$ the polynomial model fit is erratic and wildly overfitting the data. The sharp downturn around $hp=3.2$ is an overfitting artifact that is due to the global fitting nature of polynomials, and will be discussed later in this chapter when we introduce local polynomial regression using so called splines. In the case with polynomial regression with a single covariate it is rather easy to detect overfitting by looking at the fitted curve, but in more complex models with many covariates we need other tools such as measuring the predictive on a test dataset that was not used in the model fitting.

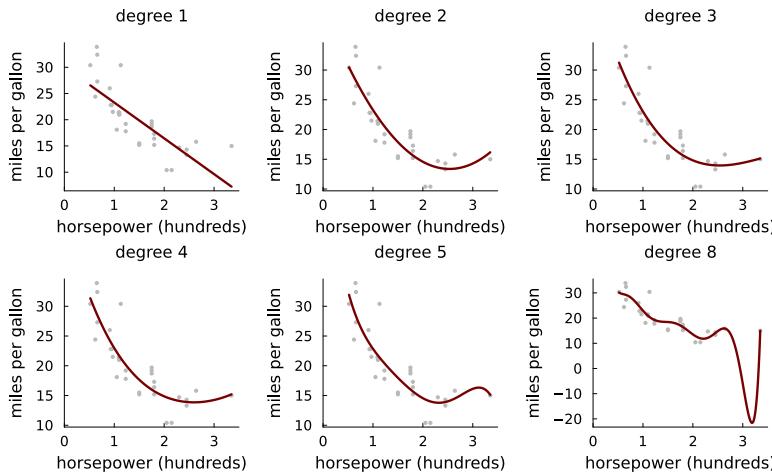


Figure 12.1: Fitting a polynomial regression to the `mtcars` data.

In this chapter we will learn how Bayesian priors can be used to control the complexity of a model and avoid overfitting. This will allow us to use flexible model with the ability to capture a wide range of different shapes, but using a prior that *encourages* simpler fits, without enforcing them as hard constraints. The widely used ridge and lasso regularization techniques will be shown to correspond to specific priors, and we will see that viewing them through our Bayesian glasses gives useful insights that can ultimately lead to extensions of these regularization techniques with better properties. We

will concentrate on two regression situations where overfitting can be a concern: i) when the number of covariates p is large compared to the number of observations n , and ii) when a flexible non-linear function is used in a nonlinear regression model. The same regularization priors can be used in classification models and other nonlinear models with covariates, but the computational cost for obtaining the posterior distribution may be higher.

12.2 *L₂-regularization and ridge regression*

The overfitting of a polynomial regression with high order has its roots in that the regression coefficients β_j are allowed take on any values, including values that may be very large (in absolute value). As an example, the fitted 8th degree polynomial in Figure 12.1 has coefficients (rounded to the nearest integer):

$$\hat{\beta} = (320, -2058, 6010, -9363, 8485, -4607, 1474, -255, 19).$$

This causes the fitted polynomial function to be very wiggly, and have a large variance from sample to sample. A simple solution to this overfitting problem is of course to use a polynomial of lower order, or even a linear model. This corresponds to setting the β_j for higher order terms *exactly* to zero; a drawback with this approach is that we may easily underfit when the data generating process actually requires a more flexible model.

A way out of this dilemma is to use a flexible model e.g. a high order polynomial, but **penalizing** large values of the regression coefficients in the fitting procedure; to impose a cost of having too large coefficients. This encourages the fitting method to produce estimates that imply a smoother model fit. One of the most commonly used regularization method is **L₂ regularization**, also known as **ridge regression**. Ridge regression modifies the least squares estimate by adding a quadratic penalty $\beta^\top \beta$ to the usual residual sum of squares $(\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta)$ fitting function; the name L₂ regularization is used since $\|\beta\|_2 = (\beta^\top \beta)^{1/2}$ is the L_2 norm, i.e the usual Euclidean length of the vector of regression coefficient β . The L₂-regularized estimate therefore minimizes the loss function

$$Q_\lambda(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta^\top \beta, \quad (12.1)$$

where $\lambda > 0$ is a regularization parameter that determines the degree of penalization. The hyperparameter λ can be set by the user, but is most commonly estimated by cross-validation in a non-Bayesian setting. The first order condition for a minimum gives p equations to

penalizing

L₂ regularization

ridge regression

solve for the p unknown β_j in β :

$$\frac{\partial Q_\lambda(\beta)}{\partial \beta} = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta = \mathbf{0}, \quad (12.2)$$

which has the following **ridge regression** estimator as solution

$$\hat{\beta}_{L_2} = (\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top \mathbf{y}. \quad (12.3)$$

The L2-penalty on large elements in β in the ridge estimator has the effect of **shrinking** the least squares estimate $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ toward the zero vector. To see this, we can first rewrite the ridge estimator in terms of the least squares estimate

$$\hat{\beta}_{L_2} = (\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top \mathbf{X} \hat{\beta}. \quad (12.4)$$

The shrinkage effect is most easily seen in the special case with an orthogonal design matrix $\mathbf{X}^\top \mathbf{X} = I_p$ where the ridge estimator becomes

$$\hat{\beta}_{L_2} = \frac{1}{1 + \lambda} \hat{\beta} = (1 - \phi) \hat{\beta}, \quad (12.5)$$

where $\phi = \lambda / (1 + \lambda)$ is the **shrinkage factor**. With $\lambda = 0$ the shrinkage factor is zero and the ridge estimate reduces to the least squares estimate; as $\lambda \rightarrow \infty$ we have $\phi \rightarrow 1$ and the ridge estimator shrinks the least squares estimate all the way to zero.

ridge regression

shrinking

shrinkage factor

An attractive feature of ridge regression is that its shrinkage has the side effect of mitigating the multicollinearity problem, and is more numerically stable due to the added λ on the diagonal of $\mathbf{X}^\top \mathbf{X}$ before taking the matrix inverse in (12.3). It even becomes possible to use more covariates than the number of observations! This case is often called the $p > n$ case (or $p \gg n$ when the number of covariates is much larger than n) since p is typically used to denote the number of covariates and n the number of observations. In such cases, $\mathbf{X}^\top \mathbf{X}$ is non-singular and there exist an infinite number of solutions to the least squares problem. Adding λ to the diagonal fixes this as $\mathbf{X}^\top \mathbf{X} + \lambda I_p$ is invertible and we can compute $\hat{\beta}_{L_2}$ uniquely.

The shrinkage in ridge regression in (12.5) is the same for all elements of β when $\mathbf{X}^\top \mathbf{X} = I_p$. In the more general case where the covariates are not orthogonal, the shrinkage of the ridge estimator is a little more elaborate, but the end result is that $\hat{\beta}_{L_2}$ applies more shrinkage along the dimensions of $\hat{\beta}$ given by the eigenvectors of $\mathbf{X}^\top \mathbf{X}$ with the smallest eigenvalues; see Appendix 17.8 for a definition of eigenvectors and eigenvalues and [Hastie et al. \(2009\)](#) for a proof. Nevertheless, ridge regression has only one λ to control the shrinkage of all regression coefficient; its shrinkage is said to be *global*. The one-dimensional global shrinkage of the ridge estimator can be restrictive in some applications; later in this chapter we will present a

global-local shrinkage method that allows for different shrinkage of different regression coefficients in a more flexible way.

Let us examine the ridge estimator in a simulated dataset where we know the ground truth. A single dataset with $n = 100$ observations are simulated from a linear regression model with unit intercept, $p = 60$ uncorrelated covariates with unit variances and a standard deviation of the error of noise $\sigma = 1$. The regression coefficient in the data generating process are chosen to be of three types: strong signal (large β), weak signal (moderately large β) and no signal (zero β). Theoretical t -values are used to quantify the strength of the signal in relation to the noise. A theoretical t -value is the data generating counterpart to the usual empirical t -values used to determine significance of covariates in frequentist analysis of regression:

$$t = \frac{\beta}{\text{SE}(\hat{\beta})},$$

where $\text{SE}(\hat{\beta}) = \sigma/\sqrt{n}$ is the frequentist standard error (standard deviation in the sampling distribution) of the least squares estimate when covariates are uncorrelated. The dataset is simulated from a linear regression with the following regression coefficients:

- five covariates with $\beta = 1$, implying a large significant effect with theoretical t -value of 10 (strong signal)
- five covariates with $\beta = 0.3$, implying a moderate effect with theoretical t -value of 3 (weak signal)
- 50 noise covariates with zero regression coefficients (no signal).

This type of model configuration where only a few of the covariates have non-zero effects is called a **sparse model**. The ridge estimates from a single data set from this data generating process are shown in Figure 12.2 for a range of λ values; the horizontal axis shows λ divided by the sample size n , which is a common scaling in many software implementations. The optimal λ value from leave-one-out cross-validation is indicated by the light gray vertical dashed line. The shrinkage effect of ridge regression for large λ values is clear, but relatively modest. We would have liked to see the β estimates for the noise covariates to be closer to zero, but the ridge estimator is unable to deliver that solution. Note also that the ridge estimates for the strong signal covariates are also shrunken away from their true values of $\beta = 1$, which is not what we would like to see.

sparse model

Consider now a Bayesian approach using the prior

$$\beta_j | \sigma^2 \stackrel{\text{iid}}{\sim} N(0, \sigma^2 / \lambda), \quad (12.10)$$

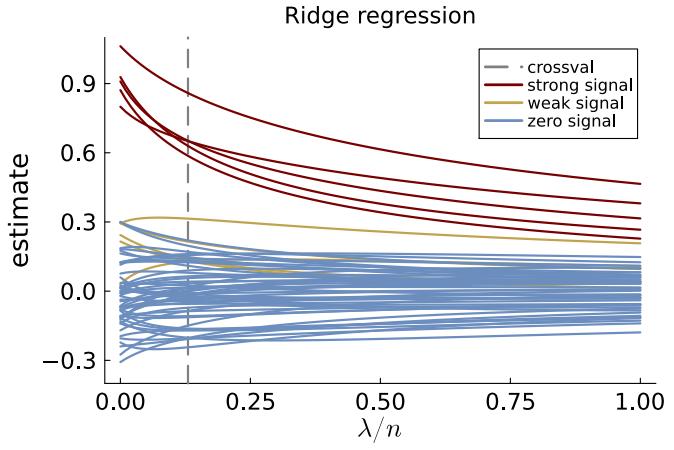


Figure 12.2: Ridge regression estimates as a function of the regularization parameter λ divided by the sample size n . The data are $n = 100$ observations simulated from a data generating process with unit intercept, $\sigma_\epsilon = 1$ and uncorrelated covariates with unit variances. The first five covariates have $\beta = 1$ with a strong signal (theoretical t -value of 10), covariates 6-10 have moderate signals with $\beta = 0.3$ (t -value of 3) and the final 50 covariates have $\beta_j = 0$. The light gray vertical dashed line indicates the optimal λ value as determined by leave-one-out cross-validation

Ridge regression uses an independent Gaussian prior

The prior

$$\beta_j | \sigma^2 \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \sigma^2 / \lambda) \quad (12.6)$$

$$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \quad (12.7)$$

for the linear regression model

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \epsilon \sim N(\mathbf{0}, \sigma^2 I_n) \quad (12.8)$$

implies a posterior mean for β equal to the ridge estimator

$$\hat{\beta}_{L_2} = (\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top \mathbf{y}. \quad (12.9)$$

Box 12.1: Ridge is Bayes with a particular Gaussian prior.

where we initially assume σ^2 to be known for simplicity. Note that this prior is a special case of the $\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$ prior used in Chapter 5 on regression with $\mu_0 = \mathbf{0}$ and $\Omega_0 = \lambda I_p$, where the simple structure for Ω_0 comes from the iid assumption in (12.10). It then follows from Figure ?? that the posterior is Gaussian with a posterior mean equal to

$$\boldsymbol{\mu}_n = \Omega_n^{-1}(\mathbf{X}^\top \hat{\boldsymbol{\beta}} + \Omega_0 \boldsymbol{\mu}_0) = (\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top \mathbf{y}, \quad (12.11)$$

which is exactly the ridge estimator in (12.3). Since this is also the marginal posterior mean of β when σ^2 is unknown following a $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$ prior (see Figure ??), we have the result in Box 12.1.

The intercept is typically not subject to regularization. This is easily implemented in a Bayesian approach by setting

$$\Omega_0 = \begin{pmatrix} 0 & 0 \\ 0 & \lambda I_{p-1} \end{pmatrix}, \quad (12.12)$$

which has zero precision on the intercept, i.e. a non-informative prior with infinite variance.

The Bayesian characterization of the ridge regression estimator gives an interesting interpretation of the otherwise rather arbitrarily defined L2-penalty: ridge regression comes from *prior belief* that the elements of β are Gaussian, independent and with the same prior precision.

To see the implication of the Gaussian assumption, consider fitting a regression model with a large number of covariates, but where only a handful of those covariates actually have a sizeable effect on the response variable; that is, most β_j are zero or very small (noise covariates), but a small number of the β_j are non-zero and potentially large (signal covariates). In such sparse situations we would like to have a method that shrinks the estimated β_j for noise covariates close to zero while at the same time leaving the estimates for the β_j for signal covariates unshrunk. As illustrated in Figure 12.2, this will typically not happen with the ridge estimator: to shrink all the unimportant β close to zero, it will have to apply a substantial shrinkage also to β estimates for the signal covariates. The end result will be a compromise where the effect of the noise covariates are not sufficiently shrunk toward zero and the effect of the signal covariates are shrunk more than one would like to. From a Bayesian point of view, this effect is caused by the homoscedastic Gaussian prior. Why? Well, a Gaussian distribution has thin tails, so it is extremely unlikely to generate large observations (in absolute value) far out in the tails. So by saying that you believe all regression coefficients to be $N(0, \sigma^2/\lambda)$ distributed a priori, you are in effect saying

I believe that all β are roughly of the same size. I do not believe that many of them are close to zero while a few of them are very large.

Hence, unless the data are really informative, the prior will lead to over-shrinkage of the true signals and under-shrinkage of the noise.

You may have wondered how exactly the ridge estimator solves the multicollinearity problem, particularly in the $p > n$ case with more covariates than observations; how is it possible to separate out the effects of different covariates when we do not have enough information in the data? The answer is that we are using *extra* information in addition to that coming from the data, and that extra information is the prior. Statistics is unfortunately not a magic wand, you need good information - data, prior or both - to make precise inferences.

One more comment about the $p > n$ case. We noted above that the implicit L₂-regularization prior in ridge regression is a special case of the $\beta|\sigma^2 \sim N(\mu_0, \sigma^2\Omega_0^{-1})$ conjugate prior. However, blindly applying the posterior formulas in Figure ?? in Chapter 5 will not work when $p > n$ since the matrix inverse $(\mathbf{X}^\top \mathbf{X})^{-1}$ in $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ does not exist when $p > n$. We only used $\hat{\beta}$ in Figure ?? to obtain a nice interpretation of the posterior mean as a compromise between data (represented by $\hat{\beta}$) and the prior. The posterior mean can however be expressed without using the least squares estimate

$$\mu_n = \Omega_n^{-1}(\mathbf{X}^\top \mathbf{X} \hat{\beta} + \Omega_0 \mu_0) = \Omega_n^{-1}(\mathbf{X}^\top \mathbf{y} + \Omega_0 \mu_0).$$

12.3 Bayesian learning of the L₂ regularization parameter

The regularization parameter λ is important for the model fit. This is a prior hyperparameter that should be determined subjectively by the user. When it is too demanding to set a particular value for λ , the user can just put a prior on λ and estimate it along with β and σ^2 . This is a hierarchical prior (see Chapter 4) where the joint prior for all unknown parameters can be decomposed as

$$p(\beta, \sigma^2, \lambda) = p(\beta|\sigma^2, \lambda)p(\sigma^2|\lambda)p(\lambda).$$

Since λ is a precision parameter we will instead analyze its inverse $\psi^2 = 1/\lambda$, which is a variance parameter. Our hierarchical prior can then be written

$$\begin{aligned} \beta|\sigma^2, \psi^2 &\sim N(\mathbf{0}, \sigma^2 \psi^2 I_p) \\ \sigma^2 &\sim \text{Inv-}\chi^2(v_0, \tau_0^2) \\ \psi^2 &\sim \text{Inv-}\chi^2(\omega_0, \psi_0^2). \end{aligned} \tag{12.13}$$

The reason for choosing a scaled Inv- χ^2 prior for ψ^2 is that it will be shown to be the conjugate prior for ψ^2 *conditional on* β and σ^2 .

We can therefore set up a Gibbs sampling algorithm to sample the joint posterior $p(\beta|\sigma^2, \psi^2|y, X)$, from which we can simply compute $\lambda = 1/\psi^2$ for each draw to obtain the marginal posterior of λ , if so desired.

With the hierarchical prior in (12.13) the user now needs to specify the prior location ψ_0^2 for ψ^2 and the degree of freedom ω_0 that determines the precision in the prior. It therefore seems that we have just pushed the problem of setting a value for $\psi^2 = 1/\lambda$ one step down the hierarchy: the user now needs to specify ψ_0^2 and ω_0 instead of ψ^2 . However, the posterior of β is typically far less sensitive to ψ_0^2 and ω_0 than it is to ψ^2 , or equivalently, λ ; this is demonstrated in an application later in this chapter.

To implement a Gibbs sampling algorithm to sample from the joint posterior $p(\beta, \sigma^2, \lambda|y)$ we need to derive the full conditional posteriors. Starting with the full conditional posterior of ψ^2 , we make use of Bayes' theorem to reverse the roles of ψ^2 and y , but still conditioning on β and σ^2 everywhere:

$$p(\psi^2|\beta, \sigma^2, y) \propto p(y|\beta, \sigma^2, \psi^2) p(\psi^2|\beta, \sigma^2) \quad (12.14)$$

However, conditional on β the distribution of the data y no longer depends on ψ^2 . This is because ψ^2 only enters via the prior for β , it has no *direct* connection to the data. So the first factor $p(y|\beta, \sigma^2, \psi^2)$ in (12.14) does not actually depend on ψ^2 and can be absorbed in the proportionality constant, i.e. $p(\psi^2|\beta, \sigma^2, y) \propto p(\psi^2|\beta, \sigma^2)$. Using Bayes' theorem one more time, this time to reverse the roles of ψ^2 and β , we can write

$$p(\psi^2|\beta, \sigma^2, y) \propto p(\psi^2|\beta, \sigma^2) \propto p(\beta|\sigma^2, \psi) p(\psi^2|\sigma^2). \quad (12.15)$$

By assumption, the prior for ψ does not depend on σ^2 so we can write the full conditional posterior of ψ^2 as

$$p(\psi^2|\beta, \sigma^2, y) \propto p(\beta|\sigma^2, \psi^2) p(\psi^2). \quad (12.16)$$

This looks a little strange since the likelihood part $p(\beta|\sigma^2, \psi^2)$ does not involve the data y at all, and β seems to play the role of the data in the full conditional posterior of ψ . This is entirely natural however since ψ^2 is a hyperparameter in the prior for β , and ψ^2 does not figure in the likelihood for y ; so *conditional* on β , these regression coefficients act like data for ψ^2 . In the *marginal* posterior $p(\psi^2|y)$ the data y does however inform us about ψ^2 . One way to think about this is that y informs us about β , and β then informs us about ψ^2 .

Now, inserting our specific priors $\beta_j|\sigma^2, \psi^2 \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \sigma^2\psi^2)$ and $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$ in (12.16) we get (remember to focus on ψ^2 , the other parameters, σ^2 and β , are fixed constants in the conditional posterior of ψ^2)

$$\begin{aligned}
p(\psi^2 | \beta, \sigma^2, \mathbf{y}) &\propto p(\beta | \sigma^2, \psi^2) p(\psi^2) \\
&\propto \prod_{i=1}^p \frac{1}{\sqrt{2\pi\sigma^2\psi^2}} \exp\left(-\frac{\beta_i^2}{2\sigma^2\psi^2}\right) \cdot (\psi^2)^{-(1+\omega_0/2)} \exp\left(-\frac{\omega_0\psi_0^2}{2\psi^2}\right) \\
&\propto (\psi^2)^{-p/2} \exp\left(-\frac{\sum_{i=1}^p (\beta_i/\sigma)^2}{2\psi^2}\right) \cdot (\psi^2)^{-(1+\omega_0/2)} \exp\left(-\frac{\omega_0\psi_0^2}{2\psi^2}\right) \\
&\propto (\psi^2)^{-(1+(\omega_0+p)/2)} \exp\left(-\frac{\sigma^{-2}\beta^\top\beta + \omega_0\psi_0^2}{2\psi^2}\right) \\
&\propto (\psi^2)^{-(1+(\omega_0+p)/2)} \\
&\quad \times \exp\left(-\frac{(\omega_0+p)(\sigma^{-2}\beta^\top\beta + \omega_0\psi_0^2)/(\omega_0+p)}{2\psi^2}\right),
\end{aligned}$$

which can be recognized as

$$\psi^2 | \beta, \sigma^2, \mathbf{y} \sim \text{Inv-}\chi^2\left(\omega_0 + p, \frac{\sigma^{-2}\beta^\top\beta + \omega_0\psi_0^2}{\omega_0 + p}\right). \quad (12.17)$$

This shows that the scaled Inv- χ^2 distribution is indeed the conditionally conjugate prior for ψ^2 since the full conditional posterior of ψ^2 belongs to the same Inv- χ^2 family as the prior.

The term $\sigma^{-2}\beta^\top\beta$ in the conditional posterior for ψ^2 suggest that inference for ψ^2 is determined by the (squared) length of the normalized regression coefficients β_i/σ for $i = 1, \dots, p$. If the data suggest that many of the normalized β_j are far from zero, then the posterior for ψ^2 will concentrate on large values, and hence the posterior for $\lambda = 1/\psi^2$ will concentrate on small values, i.e. only mild shrinkage of the β_j toward zero. Since ψ^2 is, at least conditionally, learned from the β_j coefficients as data, inference for ψ^2 will be imprecise when p is small, everything else equal.

Conditional on ψ^2 , the joint posterior $p(\beta, \sigma^2 | \psi^2, \mathbf{y})$ for β and σ^2 is directly given by Figure ?? from Chapter 5 with $\Omega_0 = \lambda I_p = \psi^{-2} I_p$. We can therefore set up a two-block Gibbs sampler with (β, σ^2) in one block and ψ^2 in the other. This is summarized in Box 12.2.

12.4 L1-regularization and the Lasso estimator

As discussed above, the ridge estimator tends to overshrink the signals (the sizeable non-zero β_j) and under-shrinking the noise (the β_j that are actually zero). We have argued that this effect can be understood from a Bayesian perspective as coming from the thin tails of the normal prior. An obvious remedy is therefore to replace the normal distribution with a more heavy-tailed prior. It turns out that

Gibbs sampling linear regression - L₂ regularization prior

The posterior for the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \varepsilon \sim N(\mathbf{0}, \sigma^2 I_n), \quad (12.18)$$

with hierarchical L₂ regularization prior

$$\begin{aligned} \boldsymbol{\beta} | \sigma^2, \psi^2 &\sim N(\mathbf{0}, \sigma^2 \psi^2 I_p) \\ \sigma^2 &\sim \text{Inv-}\chi^2(\nu_0, \tau_0^2) \\ \psi^2 &\sim \text{Inv-}\chi^2(\omega_0, \psi_0^2). \end{aligned}$$

can be sampled by a two-block Gibbs sampler:

$$\begin{aligned} \text{Block1 : } \boldsymbol{\beta} | \sigma^2, \psi^2, \mathbf{y} &\sim N(\hat{\boldsymbol{\beta}}_{L_2}, \sigma^2 (\mathbf{X}^\top \mathbf{X} + \psi^{-2} I_p)^{-1}) \\ \sigma^2 | \psi^2, \mathbf{y} &\sim \text{Inv-}\chi^2(\tau_n^2, \nu_n) \end{aligned}$$

$$\text{Block2 : } \psi^2 | \boldsymbol{\beta}, \sigma^2, \mathbf{y} \sim \text{Inv-}\chi^2(\omega_n, \psi_n^2),$$

where $\hat{\boldsymbol{\beta}}_{L_2}$ is the ridge estimator

$$\hat{\boldsymbol{\beta}}_{L_2} = (\mathbf{X}^\top \mathbf{X} + \psi^{-2} I_p)^{-1} \mathbf{X}^\top \mathbf{y} = (\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top \mathbf{y}.$$

The hyperparameters ν_n and τ_n^2 are given in Figure ?? and $\omega_n = \omega_0 + p$ and $\psi_n^2 = (\sum_{i=1}^p (\beta_i / \sigma)^2 + \omega_0 \psi_0^2) / \omega_n$.

Box 12.2: Gibbs sampling for the linear regression model with a L₂ regularization prior.

using the **Laplace distribution** in Box 12.3 and 12.3 as prior for each β_j gives rise to the well-known Lasso, or L1-regularized, estimator.

The **Lasso estimator** minimizes the usual residual sum of squares plus a penalty term that is the sum of the absolute values of the regression coefficients; that is, Lasso minimizes

$$R_\lambda(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^p |\beta_j|, \quad (12.19)$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^p |\beta_j|$ is the L_1 norm and $\lambda > 0$ is again a regularization parameter that determines the degree of penalization and can be determined by cross-validation. It is worth noticing that many software packages use a slightly different definition where the residual sum of squares in the Lasso objective function is divided by n . The regularization parameter returned from such packages should therefore be multiplied by n to be comparable with λ here. Some packages also standardize the covariates before fitting the Lasso, which we have not done in the simulation below since the covariates in that experiment have unit variance.

The Lasso estimator is not available in closed form, but the minimizer of (12.19) for a given λ can be found by an extremely efficient optimization algorithm. In fact, it is possible to compute the Lasso estimates for a whole path of λ values for the same computational cost as Ridge regression (Hastie et al., 2009). Similar to the Ridge regression estimator, the Lasso estimator shrinks the least squares estimates toward zero and the degree of shrinkage increases with λ . However, due to the kink at zero in the absolute value function in the Lasso penalty, Lasso can also shrink some estimated β_j *exactly* to zero even when $\lambda < \infty$. This means that the Lasso estimator can be used for variable selection; Lasso is short for Least Absolute Shrinkage and Selection Operator so the dual shrinkage and variable selection property of the Lasso is even part of its name. Figure 12.4 shows the Lasso estimates as a function of λ for the same simulated data as in Figure 12.2. Note how more and more β_j estimates are shrunk exactly to zero as λ increases. Figure 12.5 shows that ridge and Lasso give similar shrinkage of the high and low signal covariates, but the Lasso estimator is able to more aggressively shrink the noise covariates to zero.

Before showing the connection between the Lasso estimator and Bayesian posterior using the Laplace prior, let us first briefly explore the Laplace distribution. Laplace distribution is a symmetric distribution with a sharp peak at the mean μ with heavier tails than the normal distribution. This is illustrated in Figure 12.6 which compares some symmetric distributions, including the normal and Laplace. To clearly see the different tail behavior, the graph on the right in Figure

Laplace distribution

Lasso estimator

Laplace distribution

$X \sim \text{Laplace}(\mu, \beta)$ for $X \in \mathbb{R}$.

$$p(x) = \frac{1}{2\beta} \exp\left(-\frac{|x-\mu|}{\beta}\right)$$

$$\mathbb{E}(X) = \mu$$

$$\mathbb{V}(X) = 2\beta^2$$

Box 12.3: Laplace distribution.

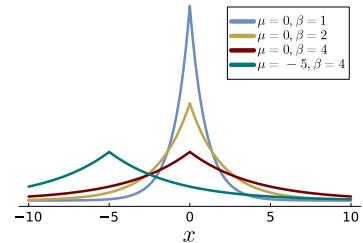


Figure 12.3: Some Laplace distributions.

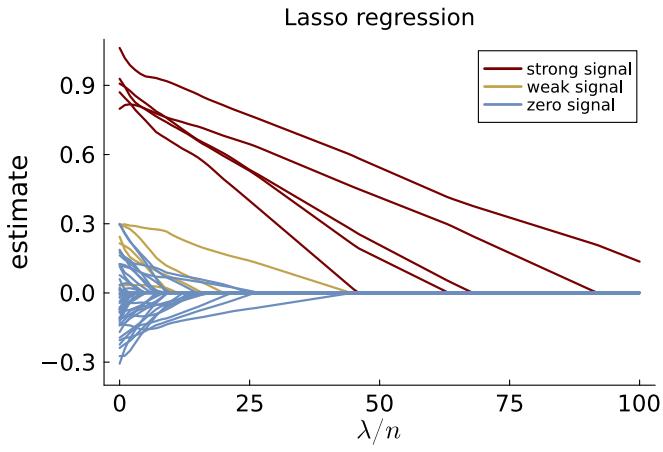


Figure 12.4: Lasso regression estimates as a function of the regularization parameter λ . The data are $n = 100$ observations simulated from a data generating process with unit intercept, $\sigma_\varepsilon = 1$ and uncorrelated covariates with unit variances. The first five covariates have $\beta = 1$ with a strong signal (theoretical t -value of 10), covariates 6–10 have moderate signals with $\beta = 0.3$ (t -value of 3) and the final 50 covariates have $\beta_j = 0$.

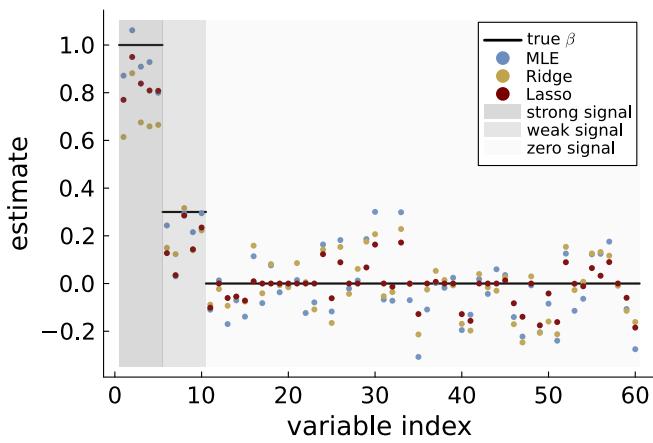
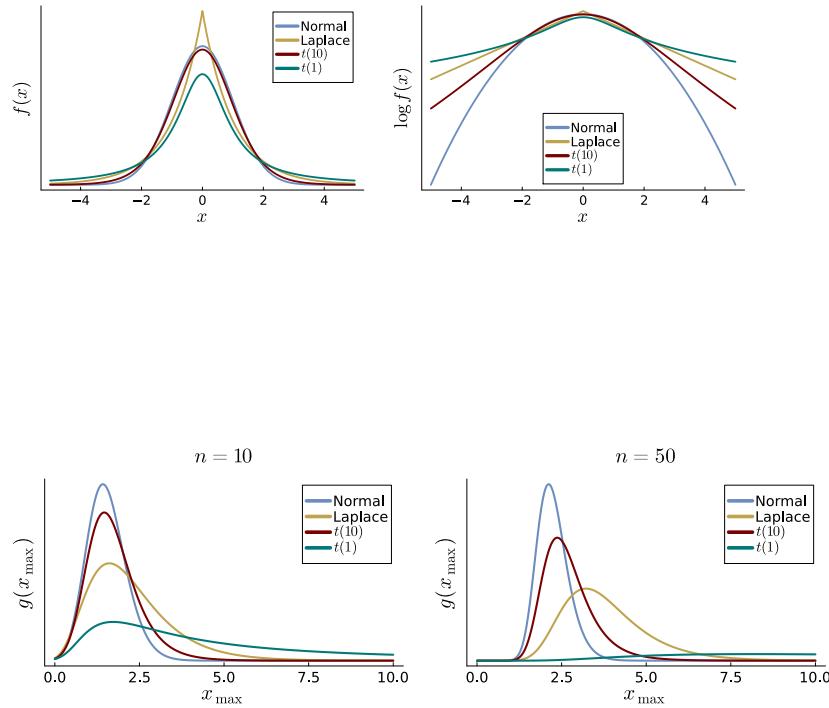


Figure 12.5: Least squares, ridge and lasso estimates of the regression coefficients for the 60 covariates in the simulated regression data. The regularization parameter in ridge and lasso is determined by cross-validation. The first five covariates have β with a strong signal (theoretical t -value of 10), covariates 6–10 have lower signal (t -value of 3) and the final 50 covariates have $\beta_j = 0$.

12.6 plots the logarithm of the pdf. Note how the tails of the normal distribution decay quadratically on the log-scale while the tails of the Laplace decays in a slower linear fashion. The student- t distribution with a low degrees of freedom have even heavier tails than the Laplace.

A distribution with heavier tails is more likely to generate extreme outcomes. Figure 12.7 illustrates this by plotting the distribution of the maximum in a sample of size n , $X_{\max} = \max(X_1, \dots, X_n)$; see Box 12.4 for the theory. It is clear from Figure 12.7 that the Laplace distribution is more likely to generate outliers than the normal distribution, in particular for the larger sample size $n = 50$ in the graph to the right. A student- t distribution with low degrees of freedom is even more extreme with a substantial probability mass on large X_{\max} already when $n = 10$ (left graph). This [observable widget](#) that compares the tails of some distributions, including the normal and Laplace.



Assume independent Laplace priors $\beta_j | \sigma^2 \stackrel{\text{iid}}{\sim} \text{Laplace}(0, 2\sigma^2/\lambda)$ for the regression coefficients in the linear regression model. The

Figure 12.6: Comparing the pdf (left) and log pdf (right) of the normal, Laplace and student- t distributions for different degrees of freedom. The log pdf of a normal distribution decays quadratically, whereas the log pdf of the Laplace decays linearly and the student- t with low degrees of freedom decays even slower than linear.

Figure 12.7: Comparing the tails of the normal, Laplace and student- t distributions by their implied distribution for the maximum in a sample of size $n = 10$ (left) and $n = 50$ (right).

Distribution of the maximum

Let X_1, \dots, X_n be iid continuous random variables with density $f(x)$ and distribution function $P(X \leq x) = F(x)$. Then the distribution of the maximum of sample of size n , $X_{\max} = \max(X_1, \dots, X_n)$ is

$$g(x_{\max}) = f(x_{\max})F(x_{\max})^{n-1}$$

Box 12.4: Distribution of the maximum of sample, also called the largest order statistic.

posterior for β conditional on σ^2 is

$$\begin{aligned} p(\beta|\sigma^2, \mathbf{X}, \mathbf{y}) &\propto p(\mathbf{y}|\sigma^2, \beta, \mathbf{X})p(\beta|\sigma^2) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)\right) \prod_{j=1}^p \exp\left(-\frac{|\beta_j|}{2\sigma^2/\lambda}\right) \\ &= \exp\left[-\frac{1}{2\sigma^2}\left((\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta) + \lambda \sum_{j=1}^p |\beta_j|\right)\right] \end{aligned}$$

where all terms not involving β have been absorbed in the proportionality constant. Maximizing $p(\beta|\sigma^2, \mathbf{X}, \mathbf{y})$ is clearly the same as minimizing

$$(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta) + \lambda \sum_{j=1}^p |\beta_j|,$$

which shows the posterior mode from independent Laplace priors for the β_j is the Lasso estimator. This is summarized in Box 12.5.

Lasso regression is the posterior mode with a Laplace prior

The prior $\beta_j|\sigma^2 \stackrel{\text{iid}}{\sim} \text{Laplace}(0, 2\sigma^2/\lambda)$ for the linear regression

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \varepsilon \sim N(\mathbf{0}, \sigma^2 I_n) \quad (12.20)$$

implies a posterior mode for β equal to the Lasso estimator.

Box 12.5: Lasso is Bayes with particular Laplace prior.

The equivalence of the Lasso estimator and a Bayesian solution with independent Laplace priors is only true when the posterior mode is used as the Bayesian point estimate. The posterior distribution for β is typically not symmetric, so the posterior mean will in general differ from the posterior mode and the Lasso estimator, and will therefore *not* shrink estimates exactly to zero. The posterior mean estimate with independent Laplace priors has been termed the **Bayesian Lasso**. This is a bit of a misnomer since the second s in Lasso stands for selection, which is something that the Bayesian Lasso does not do in general. Lasso-type variable selection happens only when using independent Laplace priors *combined with the posterior mode* as the Bayesian point estimate.

Bayesian Lasso

We have learned that the Lasso estimator, corresponding to the posterior mode estimator under a Laplace prior, also performs variable selection by setting some β exactly to zero. Chapter [Model comparison and variable selection](#) presents a direct approach to Bayesian variable selection using a so called spike-and-slab prior.

Since variable selection is fundamentally a model comparison problem where models with different subsets of variables are compared, Bayesian variable selection is most naturally discussed in connection to Bayesian model comparison methods.

12.5 Global-local regularization and the horseshoe prior

Both the L1 (Laplace) and L2 (Gaussian) regularization priors are *global* regularizers that penalize all coefficients equally using a single hyperparameter λ that acts on all elements of β . The **horseshoe prior** is instead a **global-local shrinkage prior** containing a global shrinkage parameter τ that acts on all p regression coefficients β , but also *local shrinkage* parameters λ_j for $j = 1, \dots, p$ that can modulate the global shrinkage for each β_j .

The horseshoe prior is a hierarchical prior of the form

$$\beta_j | \lambda_j^2, \tau^2, \sigma^2 \stackrel{\text{indep}}{\sim} N(0, \sigma^2 \tau^2 \lambda_j^2) \quad (12.21)$$

$$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2) \quad (12.22)$$

$$\lambda_j \sim C^+(0, 1) \quad (12.23)$$

$$\tau \sim C^+(0, 1) \quad (12.24)$$

where $C^+(0, 1)$ is the half-Cauchy distribution with location parameter 0 and scale parameter 1. The Cauchy distribution is the very heavy-tailed special case of a student- t distribution with one degree of freedom; see Box 12.6 and 12.8. The half-Cauchy distribution is the Cauchy truncated to have strictly positive support. As before the intercept β_0 is assigned a separate prior, usually non-informative, to leave the intercept more or less unaffected by the shrinkage.

Note how the global (τ^2) and local (λ_j^2) variances enter the overall variance on β_j as a product, $\sigma^2 \tau^2 \lambda_j^2$. This allows the horseshoe prior to have aggressive overall shrinkage on all β_j coefficients via a small τ^2 that encourages a sparse solution where many β_j are essentially zero, but at the same time letting a few of the β_j escape the shrinkage by inflating their local variances λ_j^2 .

To explain the origin of the name horseshoe, note first that conditional on τ and $\lambda_1, \dots, \lambda_p$, the horseshoe prior for β can be written in the conjugate form in Figure ?? with $\mu_0 = \mathbf{0}$ and

$$\Omega_0 = \text{Diag}(\tau^{-2} \lambda_1^{-2}, \dots, \tau^{-2} \lambda_p^{-2}).$$

In the case with orthogonal covariates satisfying $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_p$, then from

horseshoe prior

global-local shrinkage prior

Cauchy distribution

$X \sim C(\mu, \tau^2)$ for $X \in (-\infty, \infty)$

$$p(x) = \frac{1}{\pi \tau \left(1 + \left(\frac{x-\mu}{\tau}\right)^2\right)}$$

$$\text{median}(X) = \mu$$

$$\text{MAD}(X) = \tau,$$

where $\text{MAD}(X)$ is the median absolute deviation.

The mean and variance do not exist.

Box 12.6: The Cauchy distribution.

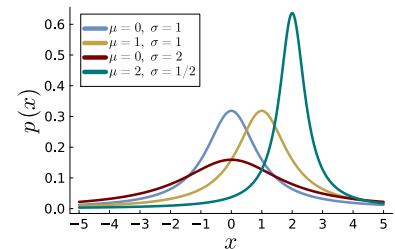


Figure 12.8: Some Cauchy distributions.

Figure ?? the posterior mean for β is

$$\begin{aligned}\boldsymbol{\mu}_n &= (\mathbf{X}^\top \mathbf{X} + \Omega_0)^{-1} (\mathbf{X}^\top \hat{\boldsymbol{\beta}} + \Omega_0 \boldsymbol{\mu}_0) \\ &= (I_p + \text{Diag}(\tau^{-2} \lambda_1^{-2}, \dots, \tau^{-2} \lambda_p^{-2}))^{-1} \hat{\boldsymbol{\beta}} \\ &= \text{Diag}(1/(1 + \tau^2 \lambda_1^2), \dots, 1/(1 + \tau^2 \lambda_p^2)) \hat{\boldsymbol{\beta}}\end{aligned}$$

so the posterior mean estimate of β_j is

$$\mu_{n,j} = (1 - \phi_j) \hat{\beta}_j,$$

where

$$\phi_j = \frac{1}{1 + \tau^2 \lambda_j^2}$$

is the *local shrinkage factor* for β_j .

The horseshoe name comes from the fact that if $\tau = 1$ and $\lambda_j \stackrel{\text{iid}}{\sim} C^+(0, 1)$ then the implied prior on each local shrinkage factor is $\phi_j \sim \text{Beta}(1/2, 1/2)$ for all j , which has a horseshoe shape; see the left graph in Figure 12.9. Hence, the horseshoe prior places most of its mass on either no shrinkage ($\phi_j = 0$) or full shrinkage ($\phi_j = 1$) for each β_j , giving a sparse solution a priori.

The right hand graph in Figure 12.9 shows the implied prior on the shrinkage factor in the L2-regularization prior when $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$; several different combinations of prior hyperparameters ω_0 and ψ_0^2 are explored to show that the prior on the shrinkage factor can vary quite a lot depending on the choice of hyperparameters. Figure 12.10 shows that the choice $\omega_0 = 0.03$ and $\psi_0^2 = 0.01$ implies a horseshoe shape, but in order to generate the two asymptotes at no shrinkage ($\phi_j = 0$) and full shrinkage ($\phi_j = 1$), the $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$ prior needs to assign very low prior density to all other value for the shrinkage factor; the horseshoe prior still has some mass on intermediate values of the shrinkage factor.

It should be remembered that regardless of the implied shrinkage factor, the horseshoe prior has a *separate* shrinkage factor for each parameter, whereas the L2-regularization prior is restricted to a single global shrinkage factor that acts on all parameters. When confronted with the data, the horseshoe prior will therefore allow for a more flexible shrinkage pattern than the L2-regularization prior.

A Gibbs sampler can be obtained by writing the half Cauchy distribution as a continuous mixture (Makalic and Schmidt, 2015)

$$X \sim C^+(0, 1) \iff X^2 | Y \sim \text{Inv-}\chi^2(1, 2/Y) \text{ and } Y \sim \text{Inv-}\chi^2(1, 2),$$

meaning that the density function of $X \sim C^+(0, 1)$ can be expressed as

$$p(x) = \int_0^\infty p(x|y)p(y)dy.$$

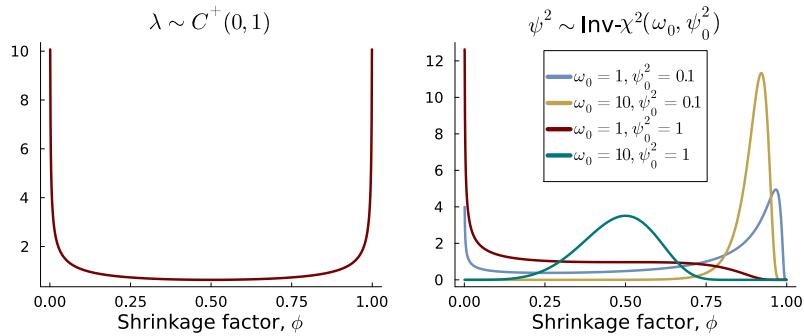


Figure 12.9: Comparing the implied shrinkage factor ϕ for the horseshoe prior (left) to the $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$ prior used in the L₂-regularization prior (right) for different values of the prior hyperparameters.

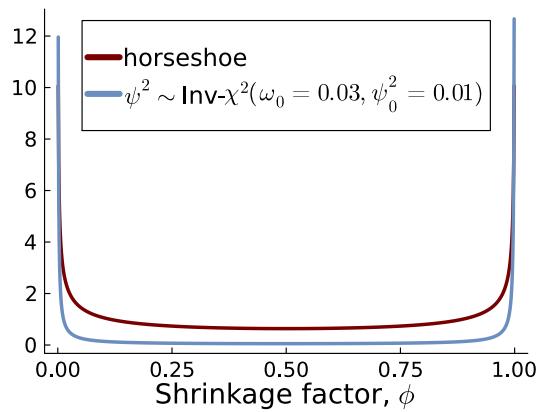


Figure 12.10: Illustration of how the L₂-regularization with the $\psi^2 \sim \text{Inv-}\chi^2(\omega_0 = 0.03, \psi_0^2 = 0.01)$ prior gives a horseshoe-like prior for the shrinkage factor, but at the cost of very low prior density for all intermediate values of the shrinkage factor.

The horseshoe prior can then be written as

$$\begin{aligned}\beta|\lambda_1, \dots, \lambda_p, \tau^2, \sigma^2 &\sim N(0, \sigma^2 \tau^2 \Lambda), \text{ where } \Lambda = \text{Diag}(\lambda_1^2, \dots, \lambda_p^2) \\ \sigma^2 &\sim \text{Inv-}\chi^2(\nu_0, \tau_0^2) \\ \lambda_j^2|\nu_j &\sim \text{Inv-}\chi^2(1, 2/\nu_j) \\ \tau^2|\xi &\sim \text{Inv-}\chi^2(1, 2/\xi) \\ \nu_1, \dots, \nu_p, \xi &\stackrel{\text{iid}}{\sim} \text{Inv-}\chi^2(1, 2)\end{aligned}$$

To obtain the full conditional posteriors for Gibbs sampling, note first that $p(\beta, \sigma^2 | \lambda_1, \dots, \lambda_p, \tau, \mathbf{y}, \mathbf{X})$ is exactly like the posterior for the linear regression in Chapter [Linear Regression](#) with $\Omega_0^{-1} = \tau^2 \Lambda$. Recall the L2-regularization case where the posterior for the hyperparameter λ given β and σ^2 was only determined from the prior $p(\beta, \sigma^2 | \lambda)$ without the likelihood for the data entering all. The same thing happens here and one can show that the full conditional posteriors for the hyperparameters in the Horseshoe prior are ([Makalic and Schmidt, 2015](#)):

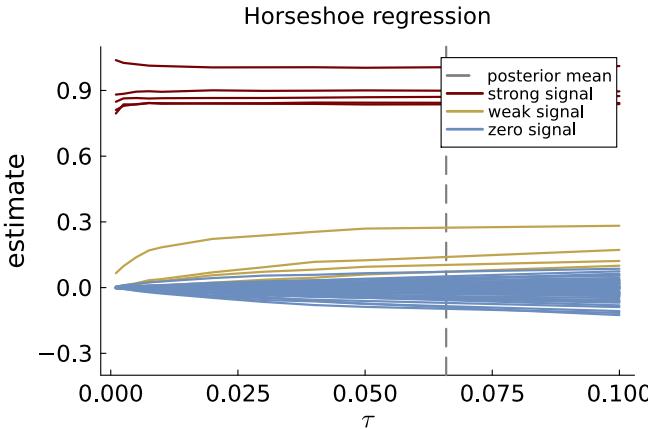
$$\lambda_j^2|\nu_j, \tau, \beta, \sigma \stackrel{\text{ind}}{\sim} \text{Inv-}\chi^2\left(2, \frac{1}{\nu_j} + \frac{1}{2} \left(\frac{\beta_j}{\sigma \tau}\right)^2\right) \quad (12.25)$$

$$\tau^2|\xi, \lambda_1, \dots, \lambda_p, \beta, \sigma \sim \text{Inv-}\chi^2\left(p+1, \frac{\frac{2}{\xi} + \sum_{j=1}^p \left(\frac{\beta_j}{\sigma \lambda_j}\right)^2}{p+1}\right) \quad (12.26)$$

$$\nu_j|\lambda_j \stackrel{\text{iid}}{\sim} \text{Inv-}\chi^2(2, 1 + 1/\lambda_j^2) \quad (12.27)$$

$$\xi|\tau \stackrel{\text{iid}}{\sim} \text{Inv-}\chi^2(2, 1 + 1/\tau^2), \quad (12.28)$$

where we have only written out explicitly the conditioning on the parameters that appear in each full conditional distribution.



TODO: Add an application with many features.

Figure 12.11: Horseshoe regression estimates (posterior means) as a function of the global standard deviation τ . The data are $n = 100$ observations simulated from a data generating process with unit intercept, $\sigma_\epsilon = 1$ and uncorrelated covariates with unit variances. The first five covariates have $\beta_j = 1$ with a strong signal (theoretical t -value of 10), covariates 6-10 have moderate signals with $\beta_j = 0.3$ (t -value of 3) and the final 50 covariates have $\beta_j = 0$.

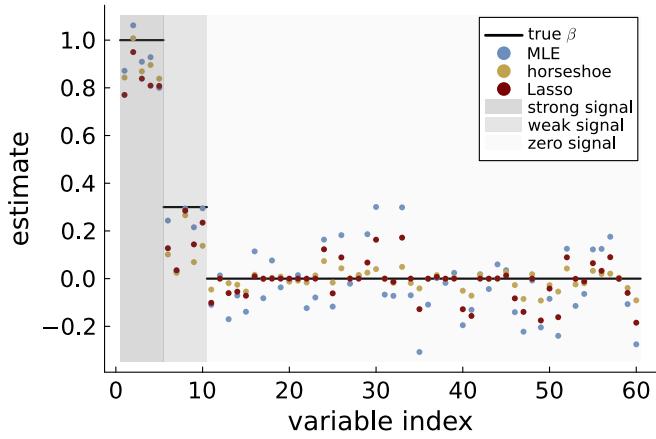


Figure 12.12: Least squares, lasso and horseshoe (posterior mean) estimates of the regression coefficients for the 60 covariates in the simulated regression data. The regularization parameter in lasso is determined by cross-validation. The first five covariates have β with a strong signal (theoretical t -value of 10), covariates 6-10 have lower signal (t -value of 3) and the final 50 covariates have $\beta_j = 0$.

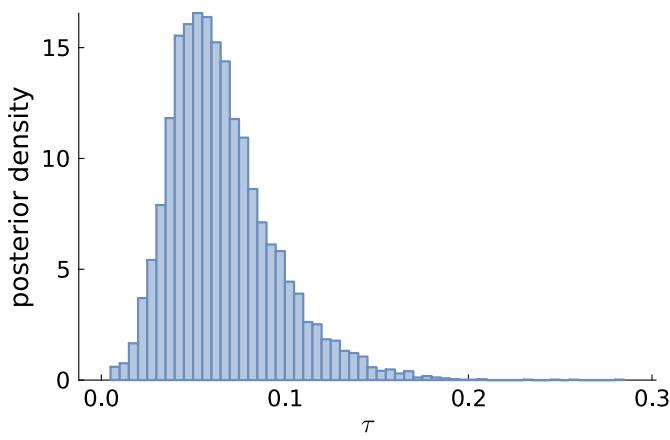


Figure 12.13: Marginal posterior of the global standard deviation τ in the horseshoe prior on the simulated data with uncorrelated covariates with unit variances.

12.6 Regularized nonlinear regression

In this section we learn how regularization priors can be used in nonlinear regression models $y = f(x) + \varepsilon$, where $f(x)$ is nonlinear function. The regularization priors encountered earlier in this chapter - the Ridge, Lasso and Horseshoe - can be effective ways of avoiding overfitting the data, even when the function $f(x)$ is a highly overparameterized flexible function.

There are many variants of nonlinear regression models, but we will here consider the practically important class of models that are nonlinear in the covariate x , but still linear in the regression coefficients. Since these models are linear in the regression coefficients, we can estimate its regression coefficients by least squares, or perform a Bayesian analysis using exactly the same machinery as for linear regression.

One example of such a model is the polynomial regression model, presented in the beginning of this chapter, where

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_p x^p \quad (12.29)$$

is a polynomial of some order p . To see that this can be written as a linear model (in the β coefficients) in exactly the form used in Chapter Linear Regression, we can expand the single covariate x into a vector of p covariates and an intercept: $\mathbf{x} = (1, x, x^2, \dots, x^p)^\top$. The **polynomial regression** model can then be written as

$$y = \mathbf{x}^\top \boldsymbol{\beta} + \varepsilon, \quad (12.30)$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^\top$ is the vector of regression coefficients. The polynomial regression model in (12.30) is a linear regression after we have expanded the single covariate x into a vector of p covariates and an intercept: $\mathbf{x} = (1, x, x^2, \dots, x^p)$.

The powers x^j in (12.29) for $j = 1, \dots, p$ are called the **basis functions** of the polynomial regression model. Note how the basis function construction to capture nonlinear regression effects effectively turned a single covariate regression into a multiple regression with potentially many (basis function) covariates. The left hand graph in the top row of Figure 12.14 shows the power basis functions used in polynomial regression (the constant basis for the intercept is not shown). The right hand graph in the top row of Figure 12.14 shows the least squares fit of a polynomial regression model to simulated data from a noisy sine curve $y = \sin(3x) + \varepsilon$ where $\varepsilon \sim N(0, 0.2^2)$.

Polynomial regression is a simple way to capture nonlinear effects in regression, but they are not without problems. One issue is that the polynomial basis function are *global* basis functions over

polynomial regression

basis functions

the whole range of the covariate x . This can have the unwanted effect that perturbation of the response variable data in one region of the covariate space can affect the fit in a completely different region. Polynomials can also extrapolate poorly outside the range of the data. Finally, the covariates constructed with the power basis functions used in polynomial regression are highly correlated, which can lead to numerical instability.

The **piecewise constant basis function** models $f(x)$ as a step function

$$f(x) = \beta_0 + \beta_1 b_1(x) + \dots + \beta_p b_p(x), \quad (12.31)$$

where the j th basis function is zero except over the interval $x \in [\kappa_{j-1}, \kappa_j]$, where it is equal to one:

$$b_j(x) = \begin{cases} 1 & \text{if } \kappa_{j-1} \leq x < \kappa_j \\ 0 & \text{otherwise} \end{cases}. \quad (12.32)$$

The $p+1$ breakpoints $\kappa_0, \dots, \kappa_p$ must be set by the user. The j th basis function models only the data in the interval $x \in [\kappa_{j-1} \leq x < \kappa_j]$ and is therefore an example of *local* basis. We can also add a linear term to $f(x)$ in (12.31) to make the model piecewise linear; see the second row of Figure 12.14 for an illustration. A drawback of the piecewise constant or piecewise linear basis function is that the function $f(x)$ is discontinuous with jumps, which can be a problem if the true function we are trying to model is smooth.

Spline basis functions combine the smoothness of polynomial basis functions with the local nature of piecewise constant basis functions.

The **local polynomial spline basis** of order q is of the form

$$b_j(x) = \begin{cases} 0 & \text{if } x < \kappa_j \\ (x - \kappa_j)^q & \text{if } x \geq \kappa_j \end{cases}. \quad (12.33)$$

where $\kappa_1, \dots, \kappa_p$ is a set of **knots** in covariate space chosen by the user. For $q = 1$ the basis function is a **local linear spline** and for $q = 2$ it is a **local quadratic spline**. The left hand graphs in the third and fourth rows of Figure 12.14 show the local linear and quadratic spline basis functions. Note how the j th knot defines a basis function that is zero for all x up to κ_j and then increases linearly like a hockey stick ($q = 1$) or quadratically like a bandy stick ($q = 2$) for all $x \geq \kappa_j$.

A spline regression model can be set up exactly as the polynomial regression model in (12.30), but with the polynomial basis functions replaced by spline basis functions. The expanded covariate vector is now $\mathbf{x} = (1, x, b_1(x), \dots, b_p(x))^\top$, where $b_j(x)$ is the spline covariate constructed as in (12.33) using the j th knot κ_j . The right hand graphs in the third and fourth rows of Figure 12.14 show the least squares

piecewise constant basis function

local polynomial spline basis

knots

local linear spline

local quadratic spline

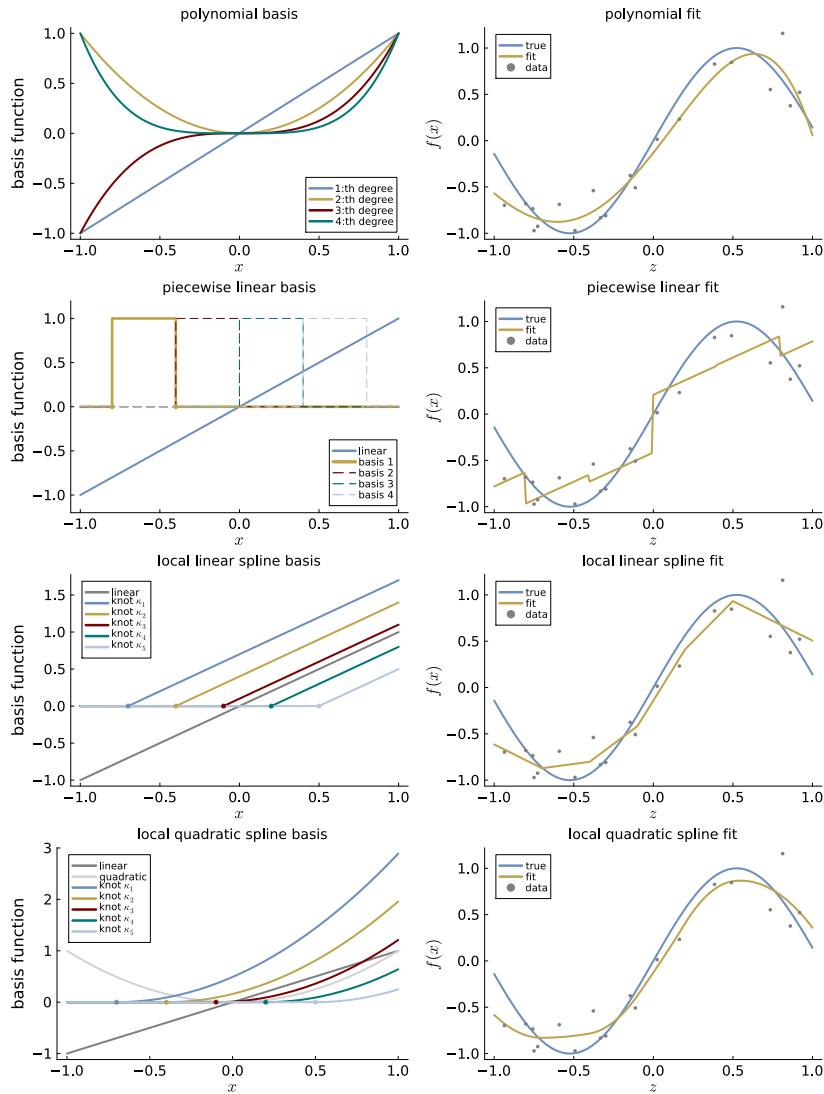


Figure 12.14: The graphs in the rows of the left column show basis functions for a i) polynomial basis, ii) piecewise basis, iii) local linear spline basis, and iv) local quadratic spline basis. The plots on the right hand side shows least squares fit of each basis function model to simulated data from a noise sine curve $y = \sin(3x) + \varepsilon$ where $\varepsilon \sim N(0, 0.2^2)$.

fit of a local linear and local quadratic spline regression model to the same simulated data as before. The local linear spline model is able to capture the general shape of the sine curve, but the local quadratic spline model is able to capture the curvature of the sine curve as well. The local linear spline model is a bit too rigid and underfits the data, whereas the local quadratic spline model is a bit too flexible and overfits the data.

There are many other types of spline basis functions, such as the *B-spline* basis functions, which are even more local than the local polynomials. It is straightforward to extend the spline regression model to have more than one covariate by adding separate spline terms for each covariate; interactions between spline covariates for different variables can be included too. There are also multi-dimensional splines that model spline surfaces in two or more dimensions.

Splines are a very flexible way to model nonlinear effects in regression, but the main difficulty is the choice of the number of knots and their locations. In principle one would like to use a few knots as possible and place them where they are needed, i.e. where the function $f(x)$ changes rapidly. This is not feasible however when the user has little prior knowledge about the function $f(x)$. In practice one often has no choice but to use a large number of knots and place them evenly over the covariate space, or perhaps at the quantiles of the empirical distribution of the covariate. However, this gives a model with many basis function covariates, i.e. a large p in $\mathbf{x} = (1, x, b_1(x), \dots, b_p(x))^\top$, which can easily lead to overfitting. It is here that a regularization prior can be very useful, allowing us to use a large number of knots, but avoiding overfitting by shrinking the estimated β coefficients of the redundant knots toward zero.

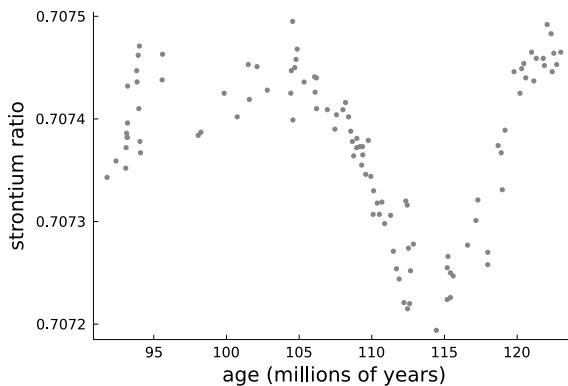


Figure 12.15: The fossil data where the strontium ratio of fossils are plotted against time.

We illustrate the use of regularization priors in spline regression for the fossil data in Figure 12.15. The data reflect global climate millions of years ago, through ratios of strontium isotopes found in

fossil shells. There is interest in whether or not the ups and downs in the data over the years reflect actual changes in the mean or are just noise (Chaudhuri and Marron, 1999). We standardize both covariate and response data to have zero mean and unit variance for numerical stability in the following.

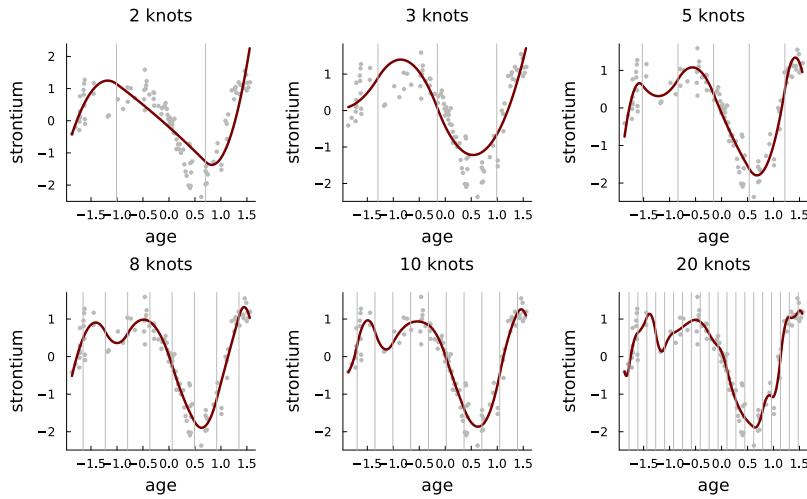


Figure 12.16: The fossil data. Fitting a quadratic spline regression with maximum likelihood for different number of knots. The location of the knots are indicated by vertical dashed lines.

Figure 12.16 shows the least squares fit of a quadratic spline regression to the fossil data for different numbers knots spread evenly over the covariate space. The fit with the model with two and three knots clearly underfits the data and does not capture the clear downturn around $x = 0.5$ well. The models with eight and ten knots do better in capturing the downturn, while the model with 20 knots seems to overfit the data with a wiggly fit.

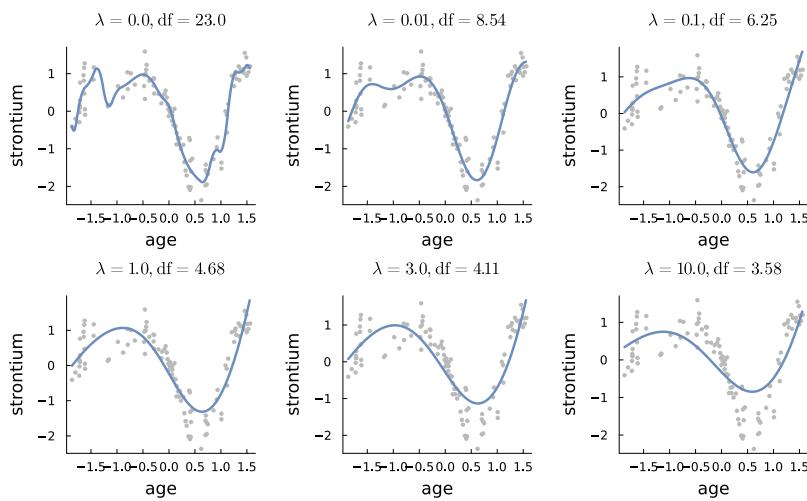


Figure 12.17: The fossil data. Fitting a quadratic spline regression with 20 knots using L₂ regularization for different values of the regularization parameter λ .

Let us now analyze the fossil data using the quadratic spline regression with 20 knots and an L₂ regularization prior. The prior $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$ is determined by setting σ_0^2 to the residual variance from a third degree polynomial regression fitted with least squares, and we set $\nu_0 = 2$, so that the prior is rather non-informative. Figure 12.17 shows the posterior mean fit for different values of the L₂-regularization parameter λ . The title in each graph in Figure 12.17 shows both the value of λ used and the so called degree of freedoms (df) of the fit (Hastie et al., 2009). The df is a measure of the *effective* number of parameters in the model, *after* the shrinkage effect from regularization, see Box 12.7 for a definition. The nominal number of parameters in the model is $p = 23$ (one β for each of the 20 knots plus three more parameters for the intercept, linear and quadratic terms), but after regularization with, for example, $\lambda = 0.1$ the effective number of parameters is reduced to just over six.

The choice of λ clearly matters for the fit, and we may be unwilling to specify an exact value for it; if we are uncertain about λ we should treat it an unknown, put a prior on it, and learn λ as a posterior distribution conditional on the data. As before, we parameterize $\psi^2 = 1/\lambda$ and put an $\text{Inv-}\chi^2(\omega_0, \psi_0^2)$ prior on ψ^2 . We initially use a rather non-informative prior with $\omega_0 = 1$ and $\psi_0^2 = 1/0.01$, and return to this choice below. We use the Gibbs sampling algorithm in Box 12.2 to simulate from the joint posterior $p(\beta, \sigma^2, \psi^2 | \mathbf{y}, \mathbf{x})$ of all model parameters; draws for $\lambda = 1/\psi^2$ are then computed and plotted as a histogram in Figure 12.18. The posterior mean and 95% credible intervals for $f(x)$ in the fossil data application are shown in Figure 12.19; the 95% predictive intervals for y are also shown as dashed lines. The posterior mean fit is quite smooth and captures the downturn around $x = 0.5$ well. With the smoothness inferred by the posterior for λ , the additional smaller downturn around $x = -1.0$ is still clearly visible. Note that the uncertainty about the regularization parameter λ , as represented by the posterior distribution in Figure 12.18, is appropriately included in the posterior and predictive intervals in Figure 12.19, in the sense that the intervals are based on posteriors that are marginalized over all parameter, including λ ; the results does not condition on an estimate of λ obtained from cross-validation or some other method.

The left hand graph of Figure 12.20 shows the sensitivity of the posterior distribution for λ to changes in the location ψ_0^2 in the prior $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$ for a given $\omega_0 = 1$. Since $\omega_0 = 1$ gives a non-informative prior, the exact location in the prior has a very small effect on the posterior for $\lambda = 1/\psi^2$. The right hand graph shows the same sensitivity for $\omega_0 = 10$. With more precise prior beliefs, the prior location ψ_0^2 starts to matter, and the posterior distribution for

Measuring flexibility

A method has a **linear fit** if its fitted values are of the form

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y},$$

where $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_n)^\top$ are the fitted values, $\mathbf{y} = (y_1, \dots, y_n)^\top$, and \mathbf{H} is the $n \times n$ **hat matrix** for the fit.

The **degrees of freedom** (df) of the fit is a measure of flexibility, and is defined as (Hastie et al., 2009)

$$\text{df} = \text{tr}(\mathbf{H})$$

where the trace (tr) is the sum of the diagonal elements.

Linear regression estimated with least squares is a linear fit with $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$, so $\text{df} = \text{tr}(\mathbf{H}) = p$, the nominal number of parameters.

With L₂-regularization the posterior mean fit is still linear, but with hat matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^\top.$$

Here $\text{df} \rightarrow 0$ as $\lambda \rightarrow \infty$.

Box 12.7: Degrees of freedom for linear fits.

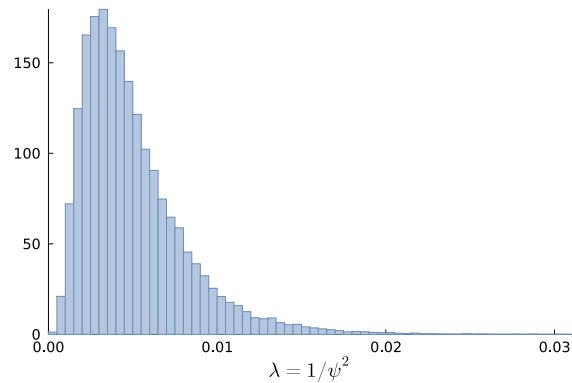


Figure 12.18: The fossil data. Posterior distribution of the regularization parameter $\lambda = 1/\psi^2$ in a quadratic spline regression with 20 knots. A vague $\psi^2 \sim \text{Inv-}\chi^2(1, 1/100)$ prior is used.

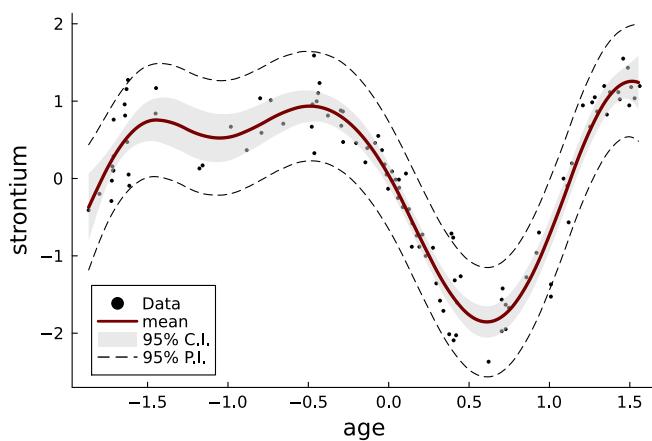


Figure 12.19: The fossil data. Posterior and predictive distribution in a quadratic spline regression with 20 knots. A vague $\psi^2 \sim \text{Inv-}\chi^2(1, 1/100)$ prior is used.

λ shifts to the right as ψ_0^2 decreases. However, these changes in the prior for the regularization parameter λ have only a small effect on the posterior mean fit, as shown in Figure 12.21.

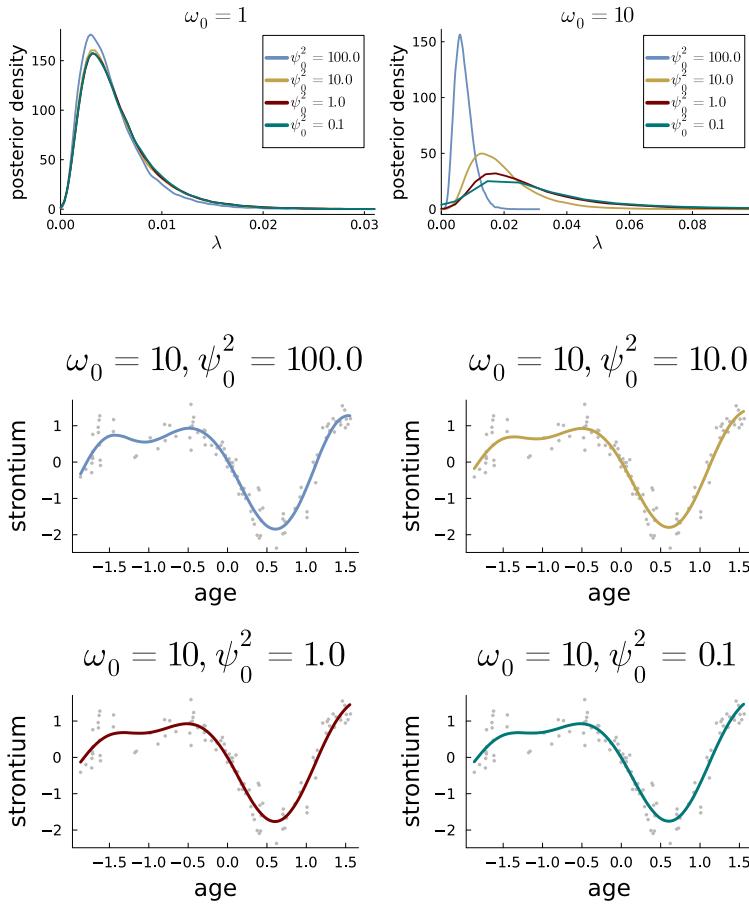


Figure 12.20: The fossil data. Sensitivity in the marginal posterior distribution for $\lambda = 1/\psi^2$ to changes in the prior for $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$.

Figure 12.21: The fossil data. Sensitivity in the posterior mean fit to changes in the prior for $\psi^2 \sim \text{Inv-}\chi^2(\omega_0, \psi_0^2)$ for $\omega_0 = 10$.

13 Mixture models and Bayesian nonparametrics

Mixture distributions appeared in the chapters [Gibbs sampling](#) and [Regularization](#) as a rather technical tool for data augmentation, to enable sampling from posterior distributions by Gibbs sampling. In this chapter we will take a closer look at mixture models as flexible distributions to model complex data. We will give particular emphasis to **discrete finite mixtures** which are mixtures based on a typically small number of component distributions. We will also introduce the concept of Bayesian nonparametrics, as a way to construct flexible models by allowing the number of parameters, or components in the case of mixtures, to grow with the data. The prominent example here will be the Dirichlet process mixture model.

discrete finite mixtures

13.1 Mixture of normals as flexible data models

The normal distribution is by far the most commonly used distribution for modeling continuous data. It has many attractive properties, for example that linear combinations $\sum_{j=1}^J c_j X_j$ of normal variables X_1, \dots, X_J are normal, even if the variables are dependent. The normal distribution has also a nice extension to the multivariate case, and the distribution plays a prominent role in most of statistics as the limiting distribution in the central limit theorem. Nevertheless, the normal model can be very restrictive: it is symmetric, unimodal and has very thin tails that cannot accommodate outliers. A mixture of normals is a flexible way to relax these assumptions.

A **mixture of normals** model is a weighted sum of normal K distributions, each with its own mean and variance:

$$p(x) = \sum_{k=1}^K \omega_k \cdot N(x|\mu_k, \sigma_k^2), \quad (13.1)$$

where the **mixture weights** sum to one $\sum_{k=1}^K \omega_k = 1$ and $N(x|\mu, \sigma^2)$ denotes the density of a normal distribution with mean μ and variance σ^2 evaluated at x . The distributions $N(x|\mu_1, \sigma_1^2), \dots, N(x|\mu_K, \sigma_K^2)$ are called the **mixture components**. Three things are important to

mixture of normals

mixture weights

mixture components

note at this stage. First, a mixture distribution is modular, it something potentially complex built up from simple parts, in this case simple normal densities. Second, the mixture in (13.1) is a **discrete mixture**, it is a sum, rather than an integral, of a finite set of normal component distributions. Third, we are *not* mixing together normal random *variables*, but normal *densities*. A weighted average of normal variables would again be normal, so no flexibility is gained from such a construction.

A weighted mixture of densities can give rise to very flexible distributional shapes. Figure 13.1 shows that a mixture of two normals can generate bimodal (top left), a mixture of three normals gives a trimodal distributions (bottom left), heavy-tails by mixing two normals with the same means but different variances (top right) and skewness (bottom right). It has in fact been shown that a mixture of normals is a *universal approximator* that can approximate *any* continuous distribution arbitrary well, given enough components. Mixture of normals are therefore often used to approximate distributions; the bottom right graph in Figure 13.1 is one such example, showing that a mixture of five normals gives a good approximation to the distribution of the log of a χ^2_1 variable, something that is used for Bayesian learning in stochastic volatility models for financial data, see Chapter [Dynamic models and sequential inference](#). This [observable widget](#) is an interactive exploration of a mixture of normals distribution.

discrete mixture

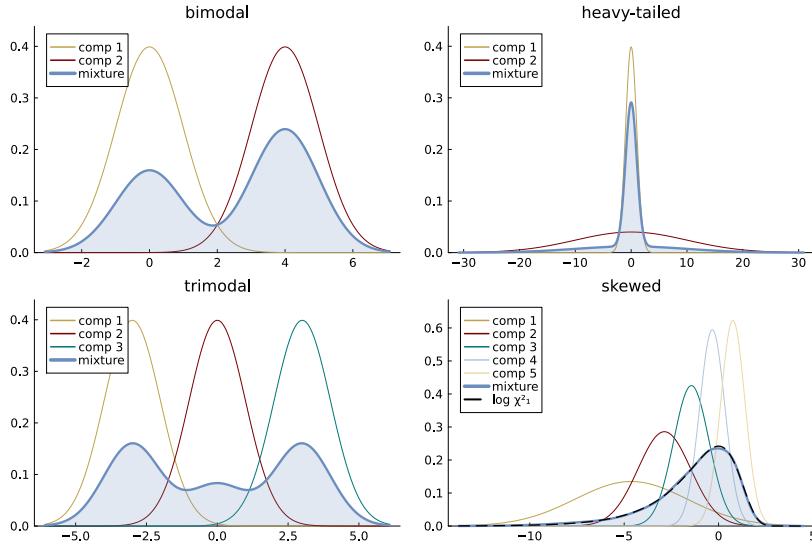


Figure 13.1: Mixture of normals.

Bimodal:

$$\mu = (0, 4)$$

$$\sigma = (1, 1)$$

$$\omega = (0.4, 0.6)$$

Heavy-tailed:

$$\mu = (0, 0)$$

$$\sigma = (1, 10)$$

$$\omega = (0.7, 0.3)$$

Trimodal:

$$\mu = (-3, 0, 3)$$

$$\sigma = (1, 1, 1)$$

$$\omega = (0.4, 0.2, 0.4).$$

Skewed:

$$\mu = (-4.63, -2.87, -1.44, -0.33, 0.76)$$

$$\sigma^2 = (8.75, 1.95, 0.88, 0.45, 0.41)$$

$$\omega = (0.13, 0.16, 0.24, 0.22, 0.25).$$

MODELING THE LENGTH OF PIKE FISH The components of a mixture can often be given a interpretation as a form of clustering of the

observations in groups or subpopulations. This is the case in Figure 13.2 where the length distribution of a sample of pike fish is rather complex (left), because of the presence of different age groups (right). Knowing the age groups makes it straightforward to model this data. We can for example assume that the length of the pikes in each age cohort follows a normal model $N(x|\mu_k, \sigma_k^2)$ for $k = 1, \dots, 5$ with separate mean and variance in each group. We can then simply learn μ_k and σ_k^2 using only data from the pikes in the year k cohort. Figure 13.3 plots the fit of this model using maximum likelihood estimates of μ_k and σ_k^2 for each cohort. The k th density in the figure is weighted by the proportion of pikes belong to the k th age cohort so that all five densities integrate to unity as a collective. The histogram in Figure 13.3 is also normalized to have unit area, i.e. it represents a density. In this example, the age groups are known since the fish were grown and tagged, but in many other applications the components are not known and the mixture model is used to learn the hidden component structure. In other situations there may not be any natural interpretation of the components, but the mixture model is used as a flexible approximation to the data distribution.

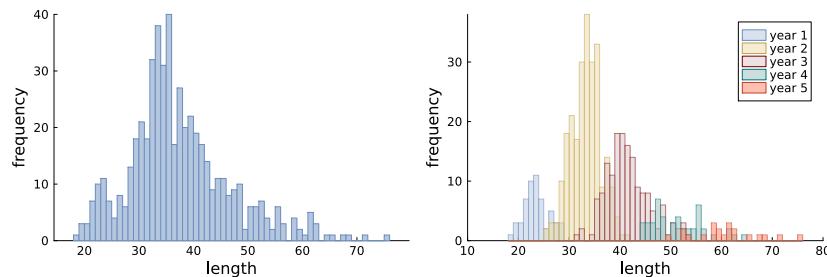


Figure 13.2: Distribution of the length (in centimeters) of 523 pike fish (left) colored coded by the year of the measured fish (right). The data is from the R package `mixdist`.

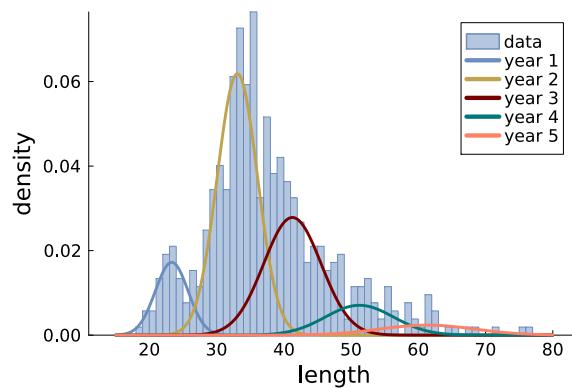


Figure 13.3: Fitting normal distributions $N(x \mid \mu_k, \sigma_k^2)$ for $k = 1, \dots, 5$ to each of the five age groups separately. Each fitted normal density have been normalized in the figure to have area equal to the proportion of fish in the corresponding age group. The histogram is also normalized to have unit area.

13.2 Simulating data from the mixture of normals model

Before explaining how to perform Bayesian inference for the mixture of normals model, let us see how we can simulate data from this model. This will give us a better understanding of the model itself, but also pave the way for simulating from the posterior distribution of the parameters in a mixture of normals.

Let us first consider a mixture of normals with only two components, for example the bimodal mixture in the top left graph of Figure 13.1. The density for a two-component mixture of normals is

$$p(x) = \omega_1 \cdot N(x|\mu_1, \sigma_1^2) + \omega_2 \cdot N(x|\mu_2, \sigma_2^2) \quad (13.2)$$

Imagine now that we want to simulate a single observation x from the mixture model in (13.2). We can simulate from this distribution by first sampling a **component allocation variable** z from a categorical distribution with support $x \in \{1, 2\}$ and probabilities $\Pr(z = 1) = \omega_1$ and $\Pr(z = 2) = \omega_2 = 1 - \omega_1$. If we obtain the outcome $z = 1$, we sample the observation x from the $N(\mu_1, \sigma_1^2)$ distribution. If we instead obtain $z = 2$, we sample the observation x from $N(\mu_2, \sigma_2^2)$. We can express this as:

$$\begin{aligned} z &\sim \text{Bern}(\omega_1) \\ X | (z = 1) &\sim N(\mu_1, \sigma_1^2) \\ X | (z = 2) &\sim N(\mu_2, \sigma_2^2), \end{aligned} \quad (13.3)$$

component allocation variable

where the Bernoulli variable z is now coded to take values in $\{1, 2\}$ rather than usual coding in $\{0, 1\}$, with probabilities $\Pr(z = 1) = \omega_1$ and $\Pr(z = 2) = \omega_2 = 1 - \omega_1$. This helps to make the notation consistent to the general case with $K > 2$ components; we can alternatively write the Bernoulli as a categorical distribution with two outcomes, $z \sim \text{Cat}(\omega_1, \omega_2)$. We can simulate a dataset with n observations from the two-component mixture of normals model by repeating this process n times. We use the letter z for the allocation variable since it is general a latent variable that is not actually observed. The fish length data is an unusual case where the mixture allocation (age group) was actually observed, but that is rarely the case.

In the same way, we can express the general mixture of normals distribution with K components

$$p(x) = \sum_{k=1}^K \omega_k \cdot N(x|\mu_k, \sigma_k^2),$$

using an allocation variable $z \in \{1, \dots, K\}$ as

$$\begin{aligned} z &\sim \text{Cat}(\omega_1, \dots, \omega_K) \\ X | z &\sim N(\mu_z, \sigma_z^2), \end{aligned} \quad (13.4)$$

where μ_z and σ_z^2 are the mean and variance of component z . It is now easy to see how we can generalize to simulate a single observation from a mixture of K normal distributions:

- simulate an allocation variable from a categorical distribution $z \sim \text{Cat}(\omega_1, \dots, \omega_K)$, where the probabilities are the mixture weights.
- simulate the observation x from the selected normal mixture component with mean μ_z and variance σ_z^2 .

The complete algorithm for simulating a dataset with n observations from a mixture of normals model with K components is given in Box 13.1.

Simulating data from a mixture of normals

Input: the number of simulated data observations n
 mixture weights $\omega = (\omega_1, \dots, \omega_K)$
 mixture component means $\mu_{1:K} = (\mu_1, \dots, \mu_K)$
 mixture component variances $\sigma_{1:K}^2 = (\sigma_1^2, \dots, \sigma_K^2)$

for i in $1:n$ **do**
 // Simulate component allocation variable
 Draw $z_i \sim \text{Cat}(\omega_1, \dots, \omega_K)$
 // Simulate from selected mixture component
 Draw $x_i|z_i \sim N(\mu_{z_i}, \sigma_{z_i}^2)$

end
Output: n iid observations $\mathbf{x} = (x_1, \dots, x_n)$ from the mixture of normals model
 $p(x) = \sum_{k=1}^K \omega_k \cdot N(x|\mu_k, \sigma_k^2).$

Box 13.1: Simulating a dataset with n observations from a mixture of normals model with K components by first drawing the mixture component each observation comes from (z_i for $i = 1, \dots, n$) and then simulating the observation from the selected component (x_i for $i = 1, \dots, n$).

13.3 Inference for the mixture of normals model

We will now consider computing the joint posterior distribution for all the unknown parameters in the mixture of normals model

$$p(x) = \sum_{k=1}^K \omega_k \cdot N(x|\mu_k, \sigma_k^2). \quad (13.5)$$

The parameters in this model are the component means $\boldsymbol{\mu}_{1:K} = (\mu_1, \dots, \mu_K)$, the component variances $\sigma_{1:K}^2 = (\sigma_1^2, \dots, \sigma_K^2)$ and the mixture weights $\boldsymbol{\omega}_{1:K} = (\omega_1, \dots, \omega_K)$.

We need the joint prior distribution for all parameters. Assume first that the parameters in different components, μ_k, σ_k^2 , are independent a priori, and independent of the mixture weights $\omega_{1:K}$. The priors for μ_k and σ_k^2 in each component are the same as for the normal model in Chapter [Multi-parameter models](#), a normal prior for μ_k conditional on σ_k^2 and a ScaledInv- χ^2 prior for σ_k^2 . Since the mixture weights are probabilities restricted to the unit simplex $\sum_{k=1}^K \omega_k = 1$, a natural prior for ω is the Dirichlet distribution. The full prior is then

$$\begin{aligned} \boldsymbol{\omega} &\sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K) \\ \sigma_k^2 &\sim \text{ScaledInv-}\chi^2(\nu_{0,k}, \sigma_{0,k}^2) \text{ and } \mu_k | \sigma_k^2 \sim N(\mu_{0,k}, \sigma_k^2 \tau_{0,k}^2), \end{aligned} \quad (13.6)$$

where the extra subscript k on $\nu_{0,k}, \sigma_{0,k}^2, \mu_{0,k}$ and $\tau_{0,k}^2$ shows that we allow for different prior hyperparameters in each mixture component. It is quite common in applications to use the same prior hyperparameter for all components, i.e. to set $\nu_{0,k} = \nu_0, \sigma_{0,k}^2 = \sigma_0^2, \mu_{0,k} = \mu_0$ and $\tau_{0,k}^2 = \tau_0^2$ for $k = 1, \dots, K$.

If we try to follow the usual route of obtaining the posterior by multiplying the likelihood with the prior we obtain a complicated expression which cannot be recognized as any known distribution. The main problem is the likelihood of the mixture of normals model

$$p(x_1, \dots, x_n | \boldsymbol{\mu}_{1:K}, \sigma_{1:K}^2, \boldsymbol{\omega}_{1:K}) = \prod_{i=1}^n \sum_{k=1}^K \omega_k \cdot N(x_i | \mu_k, \sigma_k^2),$$

which is nasty since it is a product of a (weighted) sum. It is not even possible to obtain the maximum likelihood estimator in closed form.

There are two additional problems with the likelihood for mixture models. First, a mixture model is not fully identifiable, meaning that the likelihood is the same for different parameter values, like we saw for the multinomial logistic model in Chapter [Classification and Generalized regression](#). The identification problem here is that we can swap the indices of the mixture components and obtain the same likelihood. Consider for example the two-component mixture with parameter vector $\boldsymbol{\theta} = (\mu_1, \sigma_1^2, \omega_1, \mu_2, \sigma_2^2, \omega_2)$. Now consider the

Gibbs sampling for mixture of normals

Input: data $\mathbf{x} = (x_1, \dots, x_n)$ and prior hyperparameters
initial mixture allocation $\mathbf{z}^{(0)} = (z_1^{(0)}, \dots, z_n^{(0)})$
number of posterior draws m .

```

for  $j$  in  $1:m$  do
    // Update component parameters
    for  $k$  in  $1:K$  do
        Set  $\mathbf{x}_k = \{x_i \text{ such that } z_i^{(j-1)} = k\}$ 
        Draw  $(\sigma_k^2)^{(j)} | \mathbf{x}_k \sim \text{ScaledInv-}\chi^2(\nu_{n,k}, \sigma_{n,k}^2)$ 
        Draw  $\mu_k^{(j)} | (\sigma_k^2)^{(j)}, \mathbf{x}_k \sim N(\mu_{n,k}, \tau_{n,k}^2)$ 
    end

    // Update component weights
    Set  $n_k = |\mathbf{x}_k|$ , number of obs in component  $k$ 
    Draw  $\boldsymbol{\omega}^{(j)} \sim \text{Dirichlet}(\alpha_1 + n_1, \dots, \alpha_K + n_K)$ 

    // Update mixture allocations
    for  $i$  in  $1:n$  do
        for  $k$  in  $1:K$  do
             $\tilde{\omega}_k \propto \omega_k^{(j)} \cdot N(x_i | \mu_k^{(j)}, \sigma_k^{(j)})$ 
        end
        normalize  $\tilde{\omega}_1, \dots, \tilde{\omega}_K$  to sum to one
        simulate allocation  $z_i^{(j)} \sim \text{Cat}(\tilde{\omega}_1, \dots, \tilde{\omega}_K)$ 
    end
end
```

Output: m autocorrelated draws from the joint posterior
 $p(\boldsymbol{\mu}_{1:K}, \sigma_{1:K}^2, \boldsymbol{\omega}_{1:K} | \mathbf{x})$.

Box 13.2: Gibbs sampling algorithm for sampling from a joint posterior distribution $p(\boldsymbol{\mu}_{1:K}, \sigma_{1:K}^2, \boldsymbol{\omega}_{1:K} | \mathbf{x})$ of the parameters in a mixture of normals model with K components. The unknown mixture allocation for each observation $\mathbf{z} = (z_1, \dots, z_n)$ is drawn in a separate updating step. The updating steps for each μ_k and σ_k^2 follow the same conjugate update for normal distributions in Chapter [Multi-parameter models](#) but using only the subset of data currently allocated to component k according to \mathbf{z} , hence the subscript k on the posterior hyperparameters $\mu_{n,k}, \tau_{n,k}^2, \nu_{n,k}$ and $\sigma_{n,k}^2$.

same mixture, but with the order of the two components swapped (permuted): $\tilde{\theta} = (\mu_2, \sigma_2^2, \omega_2, \mu_1, \sigma_1^2, \omega_1)$. Clearly the likelihood value for these two parameter vector values is the same: $p(x_1, \dots, x_n | \theta) = p(x_1, \dots, x_n | \tilde{\theta})$. This may seem like a trivial problem, but can cause problems for posterior sampling algorithms, as we discuss later. One solution is to impose a constraint on the parameters, for example by requiring that the mixture components are ordered by means: $\mu_1 < \mu_2 < \dots < \mu_K$ since this now rules out permutations of the component indices.

A second problem, particularly for maximum likelihood, is that the likelihood is unbounded. By setting one of the component means μ_k equal to one of the data points x_i , the likelihood can be made arbitrarily large by setting the variance σ_k^2 arbitrarily small. The maximum likelihood estimate for a mixture model is therefore one of these degenerate solutions, which is not very useful. This is not a problem for Bayesian inference if the prior on the component variances σ_k^2 decays rapidly to zero as $\sigma_k^2 \rightarrow 0$. Nevertheless, the likelihood function is complicated and analytical solutions for obtaining the posterior distribution are not available.

We have already seen in Chapter [Gibbs sampling](#) that some intractable models can be made tractable by data augmentation. This is also the case for the mixture of normals model. We can augment each data observation x_i with mixture allocation variable a_i for $i = 1, \dots, n$. As we have already seen, conditional on the allocation variables $z = (z_1, \dots, z_n)$ we can just split up the sample in K subsets and update the parameters in each mixture component k using the k th subset. The allocation variables are of course not actually known, but can be updated in a separate Gibbs sampling update conditional on draws of the component parameters. The full Gibbs sampling algorithm for the mixture of normals model is given in [Box 13.2](#).

[Figure 13.4](#) shows a mixture of normals model with $K = 1, 2, \dots, 6$ number of components fitted to the Pike length data. The red line in the graphs are the posterior mean density for the data $p(x) = \sum_{k=1}^K \omega_k N(x | \mu_k, \sigma_k^2)$ averaged over 100,000 draws from the posterior $p(\mu_{1:K}, \sigma_{1:K}^2, \omega_{1:K} | \mathbf{x})$. A model with two components captures the skewness of the distribution, but missed the mode for the youngest fish cohort. A three-component mixture uses its third component to capture this mode. Additional components beyond $K = 3$ seem to only marginally improve the fit. The number of components can be analyzed by computing the marginal likelihood or the log predictive scores for different K and comparing them, or by computing the posterior distribution over $K = 1, 2, \dots, 6$, as explained in Chapter [Model comparison and variable selection](#). Towards the end of this chapter we will also see how to use the Dirichlet process mixture

model to compute the posterior distribution over the number of components.

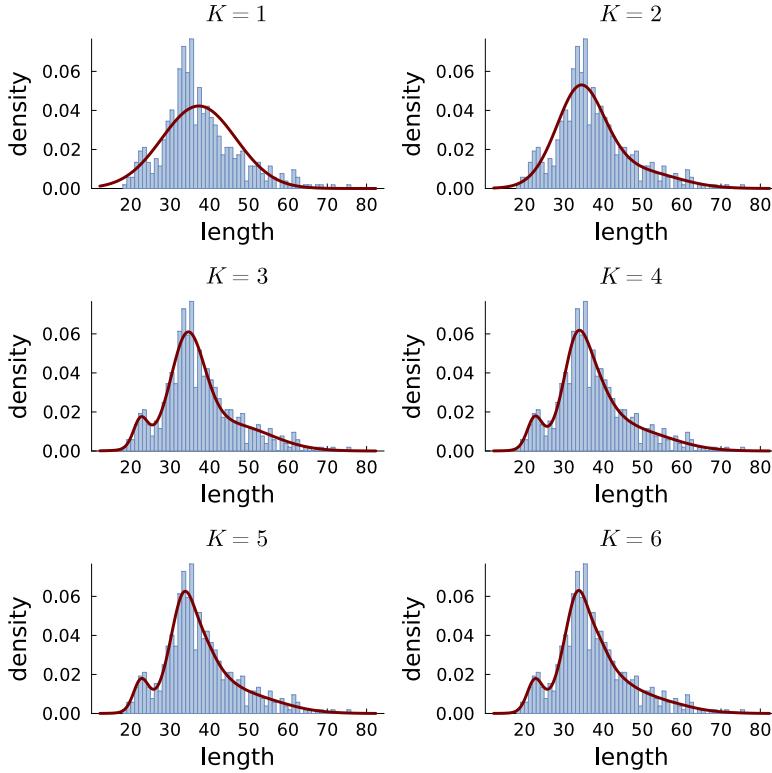


Figure 13.4: Mixture of normals model fitted to the Pike length data for different number of mixture components. The blue histogram is the observed data, the red line is the posterior mean density for the data $p(x) = \sum_{k=1}^K \omega_k N(x | \mu_k, \sigma_k^2)$ averaged over 100,000 draws from the posterior $p(\mu_{1:K}, \sigma_{1:K}^2 | \mathbf{x})$.

13.4 Mixture of Poissons for count data

As we have already discussed, a potential limitation with the Poisson distribution for modeling count data is that it is *equi-dispersed*, meaning that it is restricted to have the same mean and variance. In Section 8.5 we saw how the negative binomial model could be a useful alternative for over-dispersed data. The negative binomial model, $X \sim \text{NegBin}(x|\mu, \psi)$ is parametrized by the mean $\mu = \mathbb{E}(X)$ and the over-dispersion parameter ψ such that $\mathbb{V}(X) = \mu(1 + \mu/\psi) \geq \mathbb{E}(X) = \mu$. The negative binomial distribution can be derived as a continuous mixture of Poisson distributions with Gamma mixing weights:

$$\text{NegBin}(x|\mu, \psi) = \int_0^\infty \text{Poisson}(x|\lambda) \cdot \text{Gamma}(\lambda|\psi, \psi/\mu) d\lambda. \quad (13.7)$$

We can also replace the Gamma mixing density with some other distribution with positive support $\lambda > 0$, but the marginal density for X will no longer be a negative binomial. One particularly useful mixing distribution is a discrete distribution with probabilities $\{\omega_1, \dots, \omega_K\}$

on the support points $\{\lambda_1, \dots, \lambda_K\}$. This finite mixture of Poisson distributions is a flexible model for over-dispersed, which we now describe in more detail.

A **finite mixture of Poisson** distributions is of the form

$$p(x) = \sum_{k=1}^K \omega_k \cdot \text{Poisson}(x|\lambda_k), \quad (13.8)$$

where $\text{Poisson}(x|\lambda_k)$ denotes the Poisson pmf with mean λ_k evaluated at $X = x$. The mean is $\mu = \mathbb{E}(X) = \sum_{k=1}^K \omega_k \lambda_k$. The variance can be derived with law of total variance formula by first conditioning on the mixture component $I = k$, where $I \in \{1, \dots, K\}$ is mixture allocation variable and then undoing the conditioning:

$$\begin{aligned} \mathbb{V}(X) &= \mathbb{E}_I \mathbb{V}(X|I) + \mathbb{V}_I \mathbb{E}(X|I) \\ &= \mathbb{E}_I \lambda_I + \mathbb{V}_I \lambda_I \\ &= \sum_{k=1}^K \omega_k \lambda_k + \sum_{k=1}^K \omega_k (\lambda_k - \mu)^2 \\ &= \mu + \sum_{k=1}^K \omega_k (\lambda_k - \mu)^2, \end{aligned}$$

which is clearly overdispersed since $\sum_{k=1}^K \omega_k (\lambda_k - \mu)^2 > 0$. This overdispersion is illustrated in the top left graph of Figure 13.6 where $\mathbb{E}(X) = 3.88 < \mathbb{V}(X) \approx 12$.

It is also quite common in applications with count data to find an excess number of zeros compared to the Poisson distribution, a phenomenon referred to as **zero-inflation**. There are distributions specifically designed for zero-inflated counts, for example the **zero-inflated Poisson** (ZIPoisson) distribution. The top right graph of the figure shows that a mixture of two Poissons can also generate zero-inflation; the ZIPoisson is actually a limiting special case of the two-component mixture of Poisson distribution with one of the mixture components as a degenerate point mass at zero counts $x = 0$. A mixture of Poissons is of course more flexible than a ZIPoisson distribution and can generate other bimodal, trimodal patterns and more, see Figure 13.6. An interactive exploration of a mixture of Poissons distribution is provided in this [observable widget](#).

A MIXTURE OF POISSONS FOR THE NUMBER OF EBAY BIDDERS. Let us return to the eBay data from Section 2.4 and fit a mixture of Poissons model to the number of bidders. We will for simplicity use a non-informative $\text{Gamma}(0.01, 0.01)$ prior for λ in all mixture components and a uniform prior on the weights, i.e. $\omega \sim \text{Dirichlet}(1, \dots, 1)$. Figure 13.7 shows the fit of the mixture of Poissons model with vary-

Zero-inflated Poisson distribution

$X \sim \text{ZIPoisson}(\pi, \lambda)$ for $X = 0, 1, 2, \dots$

$$p(x) = \begin{cases} \pi + (1-\pi)e^{-\lambda} & \text{if } x = 0 \\ \frac{(1-\pi)\lambda^x e^{-\lambda}}{x!} & \text{if } x = 1, \dots \end{cases}$$

$$\mathbb{E}(X) = (1-\pi)\lambda$$

$$\mathbb{V}(X) = \lambda(1-\pi)(1+\pi\lambda)$$

Box 13.3: The zero-inflated Poisson distribution.

mixture of Poisson

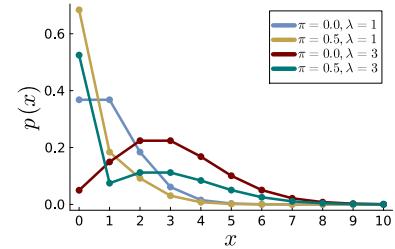


Figure 13.5: Some zero-inflated Poisson distributions.

zero-inflation

zero-inflated Poisson

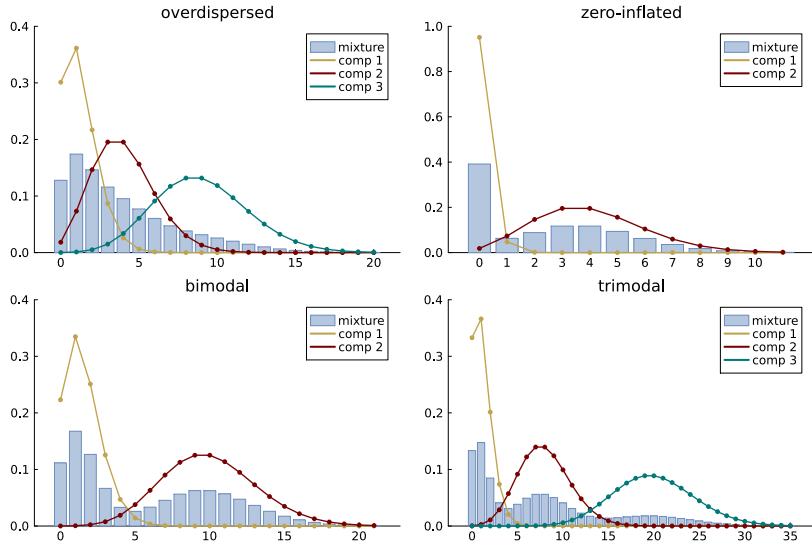


Figure 13.6: Mixture of Poissons.

Overdispersed:

$$\lambda = (1.2, 4, 9)$$

$$\omega = (0.4, 0.4, 0.2)$$

Zero-inflated:

$$\lambda = (1.5, 10)$$

$$\omega = (0.5, 0.5)$$

Bimodal:

$$\lambda = (0, 0)$$

$$\omega = (0.7, 0.3)$$

Trimodal:

$$\lambda = (1.1, 8, 20)$$

$$\omega = (0.4, 0.4, 0.2).$$

ing number of mixture components. The blue histogram is the observed data, the red line is the posterior mean density for the data $p(x) = \sum_{k=1}^K \omega_k \text{Pois}(x \mid \lambda_k)$ averaged over 100,000 draws from the posterior $p(\lambda_{1:K}, \omega_{1:K} \mid \mathbf{x})$. The largest difference is between the one-component and the two-component model, where the two-component model fit the data much better; the additional improvement in fit from adding a third component is marginal.

The posterior distribution for the parameters λ_1 and λ_2 in the two-component Poisson mixture is shown in Figure 13.8. There was no label switching in the Gibbs sampling iterations (not shown) so the two components are interpretable. The posterior for the smaller λ_1 resembles the posterior for the number of bidders in the auctions with high reservation prices and the posterior for larger λ_2 looks a lot like the posterior for the number of bidders in the auctions with low reservation prices, see Section 2.4; so it seems like the two-component mixture picks up this hidden dimension in the data. We can use the sampled allocation variable z_i for each auction to compute the probability that auction i belongs to the first mixture component with the smaller λ_1 , and then classify each observation to the first component if this probability is smaller than 0.5. With this classification, Table 13.1 shows the number of auctions with low and high reservation prices versus large and small λ in the two-component Poisson mixture. The mixture component with the smaller λ_1 does indeed consist almost exclusively of auctions with high reservation prices. In this dataset we actually had access to the reservation prices, but in many other applications the components are not known and the mixture model is used to learn the hidden component structure.

Gibbs sampling for mixture of Poissons

Input: data $\mathbf{x} = (x_1, \dots, x_n)$ and prior hyperparameters
initial mixture allocation $\mathbf{z}^{(0)} = (z_1^{(0)}, \dots, z_n^{(0)})$
number of posterior draws m .

for j in $1:m$ **do**

// Update component parameters
for k in $1:K$ **do**
| Set $\mathbf{x}_k = \{x_i \text{ such that } z_i^{(j-1)} = k\}$
| Draw $\lambda_k^{(j)} | \mathbf{x}_k \sim \text{Gamma}(\alpha_k^{(j)}, \beta_k^{(j)})$
end

// Update component weights

Set $n_k = |\mathbf{x}_k|$, number of obs in component k
Draw $\boldsymbol{\omega}^{(j)} \sim \text{Dirichlet}(\alpha_1 + n_1, \dots, \alpha_K + n_K)$

// Update mixture allocations

for i in $1:n$ **do**
| **for** k in $1:K$ **do**
| | $\tilde{\omega}_k \propto \omega_k^{(j)} \cdot \text{Pois}(x_i | \lambda_k^{(j)})$
| **end**
| normalize $\tilde{\omega}_1, \dots, \tilde{\omega}_K$ to sum to one
| $z_i^{(j)} \sim \text{Cat}(\tilde{\omega}_1, \dots, \tilde{\omega}_K)$
end

end

Output: m autocorrelated draws from the joint posterior

$p(\lambda_{1:K}, \boldsymbol{\omega}_{1:K} | \mathbf{x})$.

Box 13.4: Gibbs sampling algorithm for sampling from a joint posterior distribution $p(\lambda_{1:K}, \boldsymbol{\omega}_{1:K} | \mathbf{x})$ of the parameters in a mixture of Poissons model with K components. The unknown mixture allocation for all observations $\mathbf{z} = (z_1, \dots, z_n)$ is drawn in a separate updating step. The updating steps for each λ_k follow the same conjugate update for Poisson distributions in Chapter Single-parameter models but using only the subset of data \mathbf{x}_k currently allocated to component k according to \mathbf{z} , hence the subscript k on the posterior hyperparameters α_k and β_k . The only updating steps that differ from the Gibbs sampling algorithm for a mixture of normals model in Box 13.2 are written in orange font.

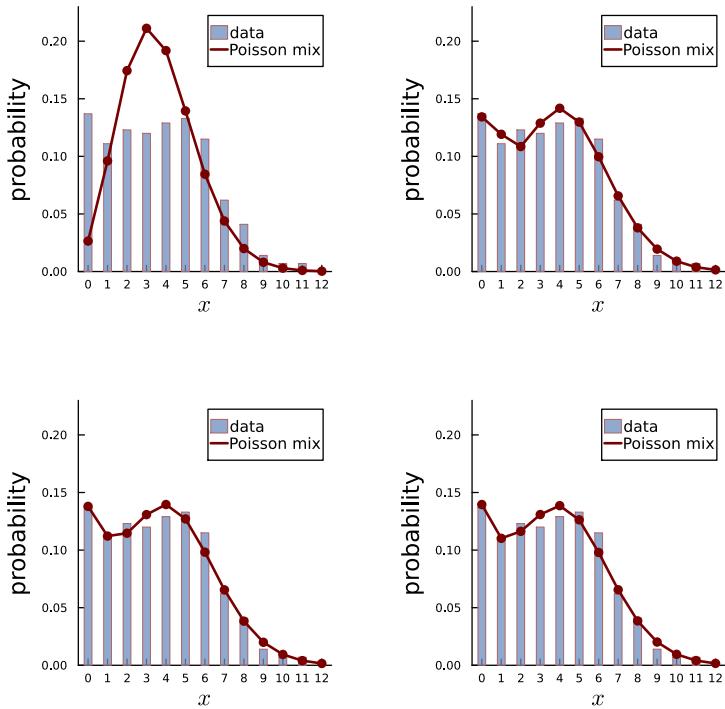


Figure 13.7: Mixture of Poissons model fitted to the ebay bids data for different number of mixture components. The blue histogram is the observed data, the red line is the posterior mean density for the data $p(x) = \sum_{k=1}^K \omega_k \text{Pois}(x | \lambda_k)$ averaged over 100,000 draws from the posterior $p(\lambda_{1:K}, \omega_{1:K} | \mathbf{x})$.

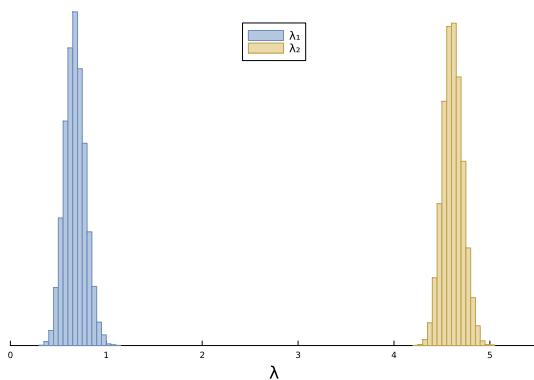


Figure 13.8: Posterior distribution for λ_1 and λ_2 in the two-component Poisson mixture fitted to the number of bidders in the eBay data. There was no label switching so the two components are interpretable.

	low reservation price	high reservation price
large λ	543	209
small λ	7	241

Table 13.1: Cross tabulation of the number of auctions with low and high reservation prices versus large and small λ in the two-component Poisson mixture fitted to the number of bidders in the eBay data. The mixture component with the small λ consist almost exclusively of auctions with high

13.5 Exponential family mixtures and multivariate mixtures

We can similarly obtain a flexible mixture model by taking density components from any distributional family $f(x|\theta)$ with parameter vector θ and mix them together in a discrete mixture

$$f(x) = \sum_{k=1}^K \omega_k \cdot f(x|\theta_k). \quad (13.9)$$

There is also nothing stopping us from taking the density components from different distributional families. For example, a mixture of a normal and a Laplace component for continuous data

$$f(x) = \omega_1 \cdot N(x|\mu_1, \sigma_1^2) + \omega_2 \cdot \text{Laplace}(x|\mu_2, \sigma_2^2), \quad (13.10)$$

or a mixture of a Poisson and a negative binomial component

$$f(x) = \omega_1 \cdot \text{Poisson}(x|\lambda_1) + \omega_2 \cdot \text{NegBin}(x|\mu_2, \phi_2), \quad (13.11)$$

for count data. See Figure 13.9 for illustrations of these mixtures.

We can similarly construct flexible multivariate distributions by mixing together multivariate distributions. For example, a mixture of K multivariate normals

$$f(\mathbf{x}) = \sum_{k=1}^K \omega_k \cdot N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (13.12)$$

where $N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the density of a multivariate normal distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$ evaluated at \mathbf{x} .

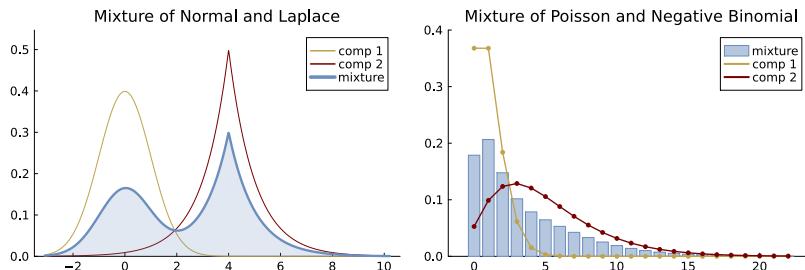


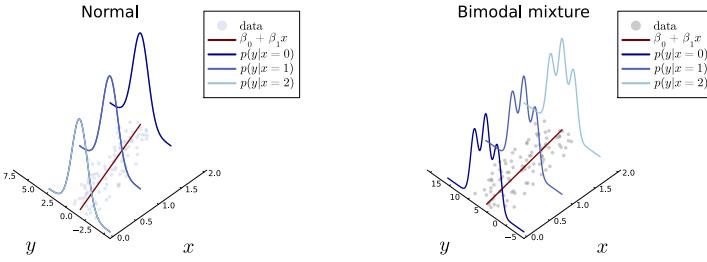
Figure 13.9: Mixture of a $N(0, 1)$ and a $\text{Laplace}(4, 1)$ distribution (left) and a mixture of a $\text{Poisson}(1)$ and a $\text{NegBin}(\mu = 5, \psi = 3)$ distribution (right).

13.6 Mixture of regressions and mixture of experts

In a regression setting where a response variable y is regressed on a vector of covariates \mathbf{x} , we can use a mixture for modeling the distribution of the error term ε

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \stackrel{\text{iid}}{\sim} \text{MoN}(\boldsymbol{\mu}_{1:K}, \sigma_{1:K}^2, \omega_{1:K}) \text{ for } i = 1, \dots, n, \quad (13.13)$$

where MoN denotes a mixture of normals distribution. The model in (13.13) is no longer restricted to normal errors, and allows for a wide range of distributional shapes, e.g. heavy tailed or skewed errors. It is however restricted to have the *same* mixture of normals distribution for all observations, i.e. for all values of \mathbf{x} . See Figure 13.10 for an illustration.



Recall that a regression model is a conditional distribution $p(y|\mathbf{x}) = N(y | \mathbf{x}^\top \boldsymbol{\beta}, \sigma^2)$ for the response y given the covariates \mathbf{x} , see the left graph of Figure 13.10 for a graphical illustration. Instead of using a mixture on the error terms in a regression, we can directly construct a flexible conditional distribution $p(y|\mathbf{x})$ by taking a mixture of such regression conditional distributions $p(y|\mathbf{x}, \boldsymbol{\beta}_k, \sigma_k^2)$, for example $N(y | \mathbf{x}^\top \boldsymbol{\beta}_k, \sigma_k^2)$, with *different* regression coefficients and error variances in each component, hence the subscript k on $\boldsymbol{\beta}_k$ and σ_k^2 in the k th component. The **mixture of linear Gaussian regressions** model is then

$$p(y_i|\mathbf{x}_i) = \sum_{k=1}^K \omega_k \cdot N(y_i | \mathbf{x}_i^\top \boldsymbol{\beta}_k, \sigma_k^2). \quad (13.14)$$

This is different from the above model which used a mixture of normals for the error term, since here we are allowing for different *regression lines* $\mathbf{x}^\top \boldsymbol{\beta}_k$ in each mixture component. A mixture of regression models can be seen as a probabilistic way of clustering regression data into K groups, where the k th group follows its own regression line $\mathbf{x}^\top \boldsymbol{\beta}_k$ and the size of the k th group is proportional to the mixture weight ω_k . The left graph in Figure 13.11 shows a mixture of two linear Gaussian regressions and some data simulated from the model. Since we simulated the data in the figure we actually know the true mixture component for each observation, but in practice the allocation to components is unknown and the mixture model is used to learn the hidden component structure, hence performing a model-based clustering of the data. Note also how the mixture model is able to capture the bimodal structure in the data, which would not be possible with a single regression line, even if we used transformations of the data, or a non-linear regression model.

Figure 13.10: Regression with mixture of normal errors.

Left: Regression with the standard assumptions with $\varepsilon \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$.

Right: Regression with a three-component mixture of normals for ε with $\mu = (-3, 0, 3)$, $\sigma = (1, 1, 1)$ and $\omega = (0.3, 0.4, 0.3)$.

mixture of linear Gaussian regressions

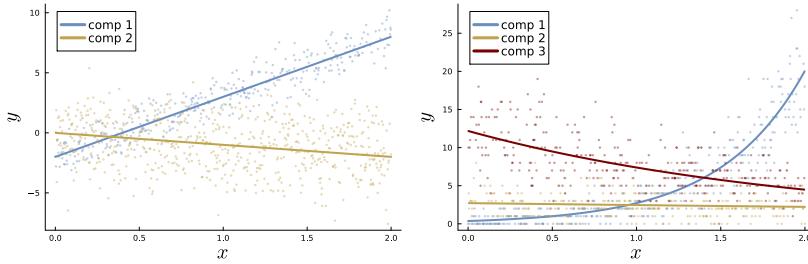


Figure 13.11: A mixture of two linear Gaussian regressions (left) and three (right) Poisson regressions (right).

We can easily extend the Gibbs sampling algorithm in Box 13.2 to fit a mixture of regressions model, by the following modifications of the algorithm:

- replace the updates of μ_k and σ_k^2 in the iid normal model to updates of the regression parameters β_k and error variance σ_k^2 in the k th component. The updates for these parameters are exactly the same conjugate updates as for the normal regression in Chapter Linear Regression, but again using only the subset of data $\{y_k, \mathbf{X}_k\}$ currently allocated to component k according to the allocation variables \mathbf{z} . Here we use the notation \mathbf{X}_k for the $n_k \times p$ matrix with covariate observations for the n_k observations in the k th subset and \mathbf{y}_k is a vector with corresponding response observations.
- in the updating step for the allocation variables z_i for $i = 1, \dots, n$, replace the normal pdfs $N(x_i | \mu_k, \sigma_k^2)$ in the iid normal model with the conditional normal pdfs $N(y_i | \mathbf{x}_i^\top \beta_k, \sigma_k^2)$ from the regression model in the k th component.

The same construction can be used for other regression models, for example a mixture of Poisson regressions for count data

$$p(y_i | \mathbf{x}_i) = \sum_{k=1}^K \omega_k \cdot \text{Pois}(y_i | \lambda_k = \exp(\mathbf{x}_i^\top \beta_k)). \quad (13.15)$$

The right graph of 13.11 shows some simulate data from a mixture of three Poisson regressions.

A **mixture of experts** takes the idea of a mixture of regressions one step further by allowing the mixture weights to be a function on the covariates \mathbf{x}_i . For example, a mixture of Gaussian regression experts is of the form

$$p(y_i | \mathbf{x}_i) = \sum_{k=1}^K \omega_k(\mathbf{x}_i) \cdot N(y_i | \mathbf{x}_i^\top \beta_k, \sigma_k^2) \quad (13.16)$$

with mixture weights now depending on the covariates \mathbf{x}_i , modeled

mixture of experts

for example as a multinomial logistic regression

$$\omega_k(\mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\gamma}_k)}{\sum_{j=1}^K \exp(\mathbf{x}_i^\top \boldsymbol{\gamma}_j)}, \quad (13.17)$$

where $\boldsymbol{\gamma}_K = \mathbf{0}$ for identification, exactly like in Chapter [Classification and Generalized regression](#). Note that the mixture weights $\omega_1, \dots, \omega_K$ are no longer explicit parameters in the model, but are instead implicitly computed from the covariates \mathbf{x}_i and the new multinomial regression parameters $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_K$.

The Gibbs sampling algorithm for a mixture of experts model is similar to the one for a mixture of regressions, but with the Dirichlet updating step for the weights $\boldsymbol{\omega}$ replaced by a step for updating the multinomial regression parameters $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_K$ for the mixture weights. The categorical response observations in the multinomial logistic regression are here the allocation variables $\mathbf{z} = (z_1, \dots, z_n)$, which therefore change in every Gibbs iterations following the update step of \mathbf{z} . The update of the $\boldsymbol{\gamma}_k$ parameters is done using the same Polya-Gamma data augmentation trick as in Chapter [Gibbs sampling](#).

13.7 Bayesian histograms

A histogram is a nonparametric estimator of a probability density $p(x)$ for a continuous variable. The estimator has a single tuning parameter, the number of bins K , or more generally a partitioning of the support into a set of bins, that may or many not have equal length. A little more formally, consider a dataset with n iid observations x_1, \dots, x_n from an underlying probability model with density $X \sim p(x)$. A **histogram estimator** is a partitioning of the support of X into K disjoint intervals B_1, \dots, B_K with an estimate of the density $p(x)$ over B_k proportional to the number of observations in B_k

$$\hat{p}(x) = \frac{n_k/n}{\Delta}, \quad \text{for } x \in B_k, \quad (13.18)$$

where n_k is the number of observations in B_k and Δ is the width of the interval B_k , which is for simplicity assumed to be the same for all k . Note that $\hat{p}(x)$ is indeed a density:

$$\int \hat{p}(x) dx = \sum_{k=1}^K \frac{n_k/n}{\Delta} \Delta = \frac{1}{n} \sum_{k=1}^K n_k = 1, \quad (13.19)$$

since $\sum_{k=1}^K n_k = n$. The histogram estimator in (13.18) can be trivially reexpressed using indicator functions for the intervals B_k as

$$\hat{p}(x) = \sum_{k=1}^K \frac{1}{n\Delta} \mathbb{1}(x \in B_k), \quad (13.20)$$

where $\mathbb{1}(x \in B_k) = 1$ if $x \in B_k$ and zero otherwise.

Rather than using a histogram as an estimator, we can view the data as coming from a **histogram probability model**

$$p(x) = \sum_{k=1}^K \frac{\omega_k}{\Delta_k} \mathbb{1}(x \in B_k), \quad (13.21)$$

where the partition B_1, \dots, B_K is fixed while the probabilities for the bins $\omega_1, \dots, \omega_K$ are unknown parameters to be learned from data.

For a fixed partition and a given dataset x_1, \dots, x_n , the likelihood of the histogram model is that of a multinomial distribution with n trials and K categories, one for each bin in the partition

$$(n_1, n_2, \dots, n_K) \sim \text{Multinomial}(\omega_1, \omega_2, \dots, \omega_K),$$

where $\omega_k = \Pr(B_k)$ and n_k is the number of observations in the k th bin. The conjugate prior for $\omega = (\omega_1, \dots, \omega_K)^\top$ is therefore the Dirichlet distribution

$$\omega \sim \text{Dirichlet}(\alpha), \quad (13.22)$$

where $\alpha = (\alpha_1, \dots, \alpha_K)^\top$. The posterior distribution for ω is then

$$\omega | \mathbf{x} \sim \text{Dirichlet}(\alpha + \mathbf{n}), \quad (13.23)$$

where $\mathbf{n} = (n_1, \dots, n_K)$.

Recall that the vector of hyperparameters $\alpha = (\alpha_1, \dots, \alpha_K)^\top$ in the Dirichlet distribution is only proportional to the mean vector:

$$\mathbb{E}(\omega) = \frac{1}{\alpha_0} \alpha, \quad (13.24)$$

where $\alpha_0 = \sum_{j=1}^K \alpha_j$ is a measure of the precision in the distribution; the smaller α_0 , the larger the variance of the elements in ω . The subscript 0 is here used in the usual sense of zero observation in a prior, and should not be interpreted as the zero:th bin. Denoting the mean vector $\bar{\alpha} = \mathbb{E}(\omega)$ we therefore have the relation $\alpha = \alpha_0 \bar{\alpha}$, and it will be convenient to write the $\text{Dirichlet}(\alpha)$ distribution as $\text{Dirichlet}(\alpha_0 \bar{\alpha})$ to explicitly separate the precision parameter α_0 from the mean vector $\bar{\alpha}$. The posterior distribution in this notation is then

$$\omega | \mathbf{x} \sim \text{Dirichlet}\left((\alpha_0 + \mathbf{n}) \frac{\alpha + \mathbf{n}}{\alpha_0 + \mathbf{n}}\right), \quad (13.25)$$

where we immediately see that the posterior mean for the histogram probability $\omega_k = \Pr(B_k)$ in bin k is

$$\mathbb{E}(\omega_k | \mathbf{x}) = \frac{\alpha_k + n_k}{\alpha_0 + n} \quad (13.26)$$

i.e. the relative proportion of *total* counts in the k th bin, i.e. the number of data observations in bin n_k plus the number of imaginary prior observations α_k in bin k .

The prior mean vector $\bar{\alpha}$ can be specified as follows:

histogram probability model

- specify a prior guess $p_0(x)$ for the underlying density, e.g. a $N(\mu, \tau^2)$ density for some given values of μ and τ .
- compute the implied probabilities for each bin

$$\Pr_0(B_k) = \int_{x \in B_k} p_0(x) dx \text{ for } k = 1, \dots, K.$$

- set the prior mean vector to match the computed bin probabilities:

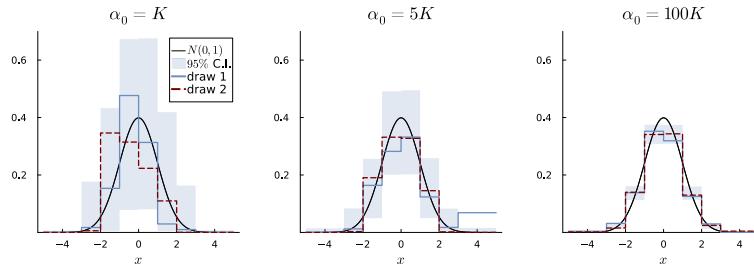
$$\bar{\alpha} = (\Pr_0(B_1), \dots, \Pr_0(B_K))^\top.$$

- set the precision parameter α_0 to a value that reflects the uncertainty in the prior guess $p_0(x)$.

As an example consider a prior guess $p_0(x) = N(0, 1)$ and a partition of the support into $K = 8$ bins using thresholds that are $\pm 1, \pm 2$ and ± 3 standard deviations from the mean, i.e. the bins are

$$B_1 = (-\infty, -3), B_2 = [-3, -2), \dots, B_7 = [2, 3), B_8 = [3, \infty),$$

with probabilities $\Pr_0(B_k) = \bar{\alpha}_k$ computed from the standard normal pdf. Figure 13.12 illustrates the prior distribution for the histogram model with the Dirichlet prior $\omega \sim \text{Dirichlet}(\alpha_0 \bar{\alpha})$ with $\bar{\alpha}$ determined from the prior guess $p_0(x) = N(0, 1)$ for different values of the precision parameter α_0 . The shaded regions in the figure are 95% pointwise prior credible intervals for the histogram. Two draws from the prior are also shown. Note how each realization is a histogram and how the prior distribution is more concentrated around the prior mean $\bar{\alpha}$ for larger values of the precision parameter α_0 .



The bin width was $\Delta = 1$ in Figure 13.12, giving a rather coarse histogram. We can also use a finer partitioning of the support by using smaller bin widths, e.g. $\Delta = 0.5$ or $\Delta = 0.1$. Figure 13.13 shows the prior distribution for the Bayesian histogram model with a Dirichlet prior $\omega \sim \text{Dirichlet}(\alpha_0 \bar{\alpha})$ for different bin sizes (rows) and different prior precisions α_0 (columns).

For a given dataset $\mathbf{x} = (x_1, \dots, x_n)^\top$, the binning results in multinomial counts $\mathbf{n} = (n_1, \dots, n_K)^\top$ over the K bins. With a Dirichlet

Figure 13.12: The prior distribution for the Bayesian histogram model with a Dirichlet prior $\omega \sim \text{Dirichlet}(\alpha_0 \bar{\alpha})$ with bin size $\Delta = 1$ for different prior precisions α_0 . The prior mean $\bar{\alpha}$ is set from the prior guess $p_0(x) = N(0, 1)$ (black line) for different values of the precision parameter α_0 . The shaded area is the 95% prior credible interval for the histogram. Two draws from the prior are also shown.

prior, the posterior distribution for the bin probabilities ω is therefore also Dirichlet

$$\omega | \mathbf{x} \sim \text{Dirichlet}\left((\alpha_0 + n)\left(\frac{\alpha_0}{\alpha_0 + n}\bar{\alpha} + \frac{n}{\alpha_0 + n}\bar{\mathbf{n}}\right)\right), \quad (13.27)$$

where $\bar{\alpha}$ is the prior mean vector for ω and $\bar{\mathbf{n}} = (n_1/n, \dots, n_K/n)^\top$ is the empirical distribution of the data over the bins.

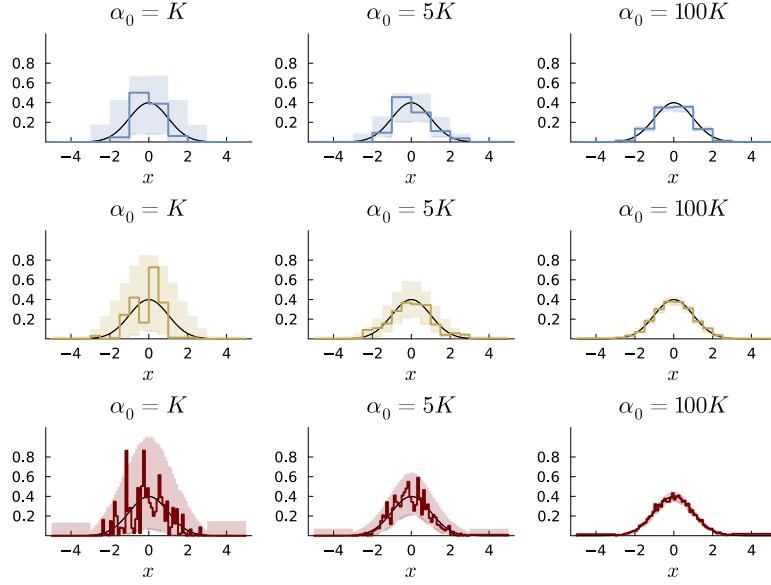


Figure 13.13: The prior distribution for the Bayesian histogram model with a Dirichlet prior $\omega \sim \text{Dirichlet}(\alpha_0, \alpha)$ for different bin sizes (rows) and prior precisions α_0 (columns). The first row uses a bin size of $\Delta = 1$, the second row $\Delta = 0.5$ and the last row $\Delta = 0.1$. The prior mean $\bar{\alpha}$ is set from the prior guess $p_0(x) = N(0, 1)$ (black line) for different values of the precision parameter α_0 .

Using the notation $\tilde{\omega}_k$ for the posterior mean of the bin probabilities ω_k in (13.27) we can define a **Bayesian histogram** estimate (under squared loss) as

$$\hat{p}(x) = \sum_{k=1}^K \frac{\tilde{\omega}_k}{\Delta_k} \mathbb{1}(x \in B_k), \quad (13.28)$$

13.8 Dirichlet process priors

The Dirichlet distribution is a distribution over distributions:

$$(P(B_1), \dots, P(B_k)) \sim \text{Dirichlet}(\alpha P_0(B_1), \dots, \alpha P_0(B_k))$$

where P_0 is a fixed probability measure, e.g. $N(0, 1)$.

Given a Dirichlet distribution over $K = 3$ bins, $(n_1, n_2, n_3) \sim \text{Dirichlet}(\alpha_1, \alpha_2, \alpha_3)$, what would happen if we merged the first two bins into a single bin? The resulting distribution can be shown to also be a Dirichlet distribution over the two remaining bins

$$(n_1 + n_2, n_3) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3).$$

Bayesian histogram

This merging property holds generally for a Dirichlet distribution over K bins, and similarly for the splitting of bins:

- merge two bins to one and the distribution over the $K - 1$ remaining bins is still Dirichlet, and the parameters of the merged bin is the sum of the parameters of the two merged bins.
- split a bin into two and the distribution over the $K + 1$ bins is still Dirichlet with the parameters of the splitted bin divided between the two new bins.

We say that the *Dirichlet distribution* is *closed under merging and splitting of bins*. This property paves the way for the following definition of the Dirichlet process.

Definition. A random probability measure P follows a **Dirichlet process** $P \sim DP(\alpha_0 P_0)$ with base measure P_0 and precision $\alpha_0 > 0$ if and only if

Dirichlet process

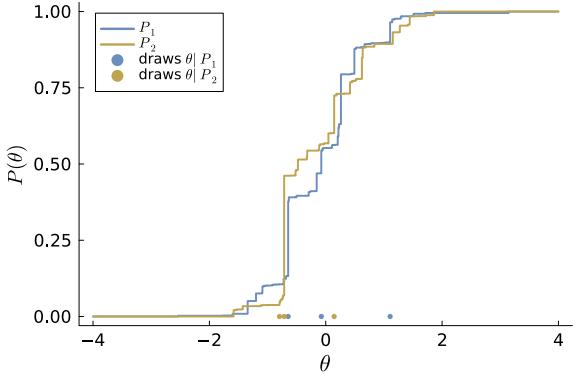
$$(P(B_1), \dots, P(B_K)) \sim \text{Dirichlet}(\alpha_0 P_0(B_1), \dots, \alpha_0 P_0(B_K))$$

for any finite measurable partition B_1, \dots, B_K .

We are here using terminology from measure theory, the mathematical foundation of probability theory. If you are not familiar with measure theory you can ignore the word measurable in the above definition and take the simplified approach of equating the phrase *probability measure* with a probability distribution.

Note that the phrase *random probability measure* P in Definition 13.8 is not the usual setting where P is a fixed distribution from which *random variables* $X \sim P$ can be generated. We actually mean that the whole probability distribution is a random object: a random function that satisfies the usual properties of a distribution function (CDF), for example being a non-decreasing function $0 \leq P(x) \leq 1$ with $\lim_{x \rightarrow -\infty} P(x) = 0$ and $\lim_{x \rightarrow \infty} P(x) = 1$. We can simulate a draw \tilde{P} of the probability measure P (see below for more on how this is done) and then simulate a random variable $X \sim \tilde{P}$ from that realized distribution \tilde{P} . Hence, a Dirichlet process is a probability distribution over probability distributions; every draw from a Dirichlet process is a probability distribution. Let that sink in.

Figure 13.14 shows two realized P (CDF curves) from a Dirichlet Process with base measure $P_0 = N(0, 1)$ and $\alpha_0 = 10$. Three realized $X \sim P$ draws from each of the two realized distributions are plotted as dots in the same color as the P that generated them. You may note that both realizations of P are discrete probability distributions, even though the base measure P_0 is a continuous distribution. This is a property of the Dirichlet process: it is a distribution over distributions that are discrete with probability 1.



To make things more concrete, let us focus on a single bin B . For a realized probability distribution P from the Dirichlet process we can compute the probability of that bin, $P(B)$. We know from Definition 13.8 that the distribution of *any* partition is a Dirichlet distribution, so $(P(B), P(B^c))$ where B^c is the complement, follows a Dirichlet distribution with only $K = 2$ categories. But this then means that $P(B)$ follows a Beta distribution. Hence, if $P \sim DP(\alpha P_0)$ then

$$P(B) \sim \text{Beta} [\alpha P_0(B), \alpha (1 - P_0(B))] \quad (13.29)$$

$$E[P(B)] = P_0(B) \quad (13.30)$$

$$\text{Var}[P(B)] = P_0(B) [1 - P_0(B)] / (1 + \alpha), \quad (13.31)$$

where the mean and variance follows from the usual formulas for the Beta distribution $X \sim \text{Beta}(a, b)$ with $E(X) = a/(a + b)$ and $\text{Var}(X) = ab/((a + b)^2(a + b + 1))$.

Stick-breaking representation of the Dirichlet process $P \sim DP(\alpha P_0)$

$$P(\cdot) = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_i} \quad (13.32)$$

$$\pi_h = V_h \prod_{\ell < h} (1 - V_\ell) \quad (13.33)$$

$$V_h \stackrel{\text{iid}}{\sim} \text{Beta}(1, \alpha) \quad (13.34)$$

$$\theta_h \stackrel{\text{iid}}{\sim} P_0 \quad (13.35)$$

Consider the model where the observations are iid from some unknown probability distribution P

$$y_i | P \stackrel{\text{iid}}{\sim} P, \text{ for } i = 1, \dots, n \quad (13.36)$$

with a Dirichlet process prior $P \sim DP(\alpha P_0)$. Consider now a partition of the outcome space B_1, \dots, B_K as in the histogram model. Let $P(B_k)$ be the probability for the k th bin B_k according to the probability measure P . By the definition of the Dirichlet process, the joint prior distribution for the partition $P(B_1), \dots, P(B_K)$ is $\text{Dirichlet}(\alpha P_0(B_1), \dots, \alpha P_0(B_K))$.

Figure 13.14: Illustration that a Dirichlet process is a distribution over distributions. Each line represent a realization from the Dirichlet process $P \sim DP(\alpha_0 P_0)$, with $\alpha_0 = 10$ and P_0 is the standard normal distribution. Note how each realization is a discrete probability distribution even though P_0 is a continuous distribution. The blue dots are three draws from the realized distribution in blue and the yellow dots are the same for the yellow distribution realization.

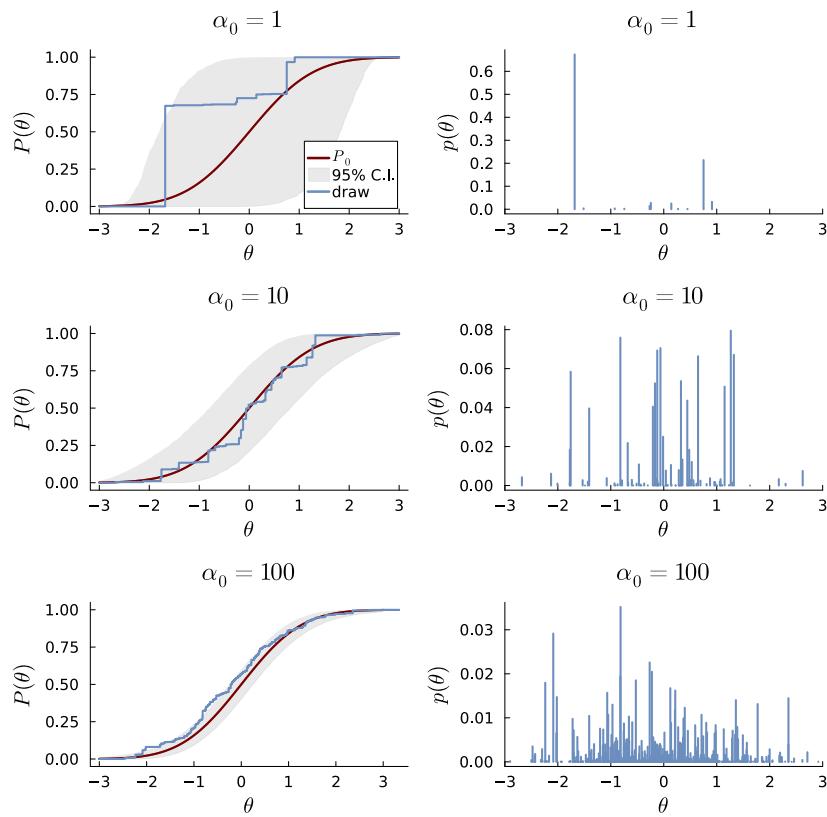


Figure 13.15: Simulating prior draws a DP

The posterior distribution for the partition given the data y_1, \dots, y_n is then also a Dirichlet distribution:

$$\text{Dirichlet} \left(\alpha P_0(B_1) + \sum_{i=1}^n 1_{y_i \in B_1}, \dots, \alpha P_0(B_k) + \sum_{i=1}^n 1_{y_i \in B_k} \right) \quad (13.37)$$

where $\sum_{i=1}^n 1_{y_i \in B_k}$ is just a fancy way of counting the number of observations falling into bin B_k .

Now, the posterior distribution in (13.37) is for a specific choice of bins B_1, \dots, B_K . Since the same property would hold for any choice of bins, we express this more generally by saying the posterior distribution of P is a Dirichlet process:

$$P|y_1, \dots, y_n \sim \text{DP} \left(\alpha P_0 + \sum_{i=1}^n \delta_{y_i} \right). \quad (13.38)$$

Note that

13.9 Dirichlet process mixtures

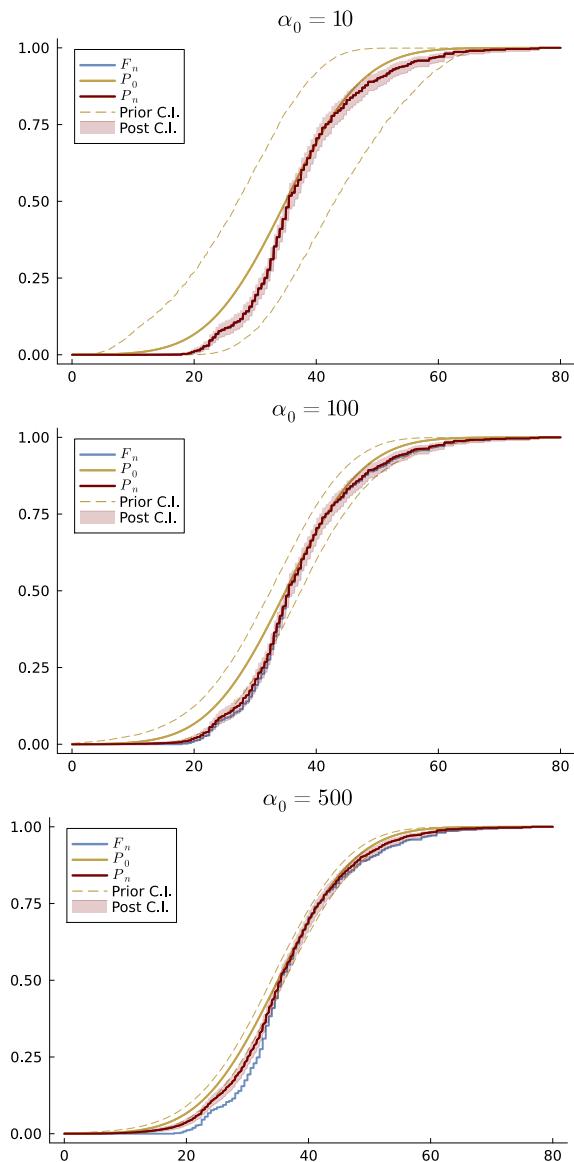


Figure 13.16: Analyzing the Fish length data with the prior $P \sim DP(\alpha_0 P_0)$ with base distribution $P_0 = N(35, 10^2)$ for three different prior precisions α_0 . The empirical distribution function $F_n(x)$ is shown in blue, the prior mean P_0 and 95% credible bands in beige and posterior mean P_n and 95% credible bands in red.

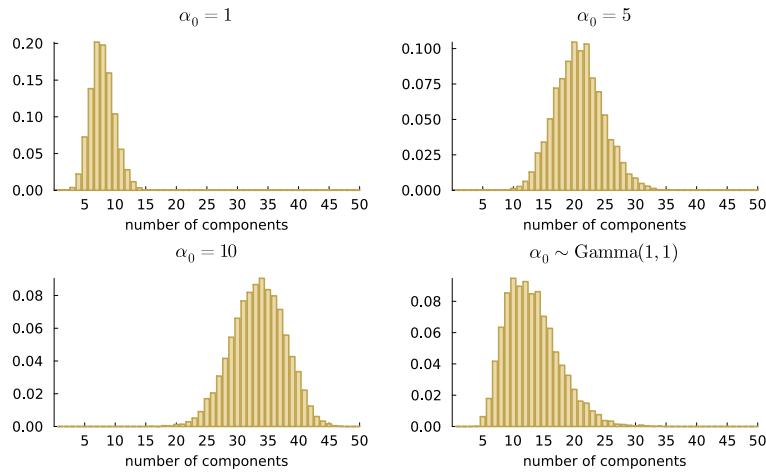


Figure 13.17: Fish length data. Posterior mean of the Dirichlet process mixture model for the fish length data for different prior precisions α_0 . The graph in the bottom right shows the fitted density when α_0 is estimated with a $\alpha_0 \sim \text{Gamma}(1, 1)$ prior.

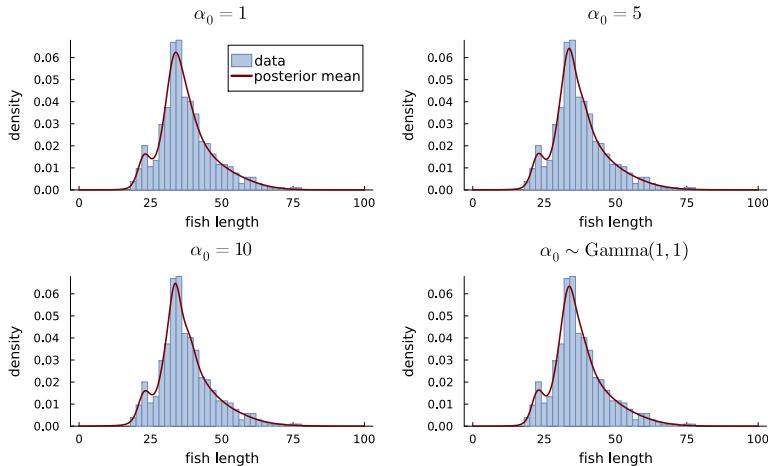


Figure 13.18: Fish length data. Posterior distribution for the number of mixture components in a Dirichlet process mixture model for different values of the prior precision α_0 . The graph in the bottom right shows the posterior distribution when α_0 is estimated with a $\alpha_0 \sim \text{Gamma}(1, 1)$ prior.

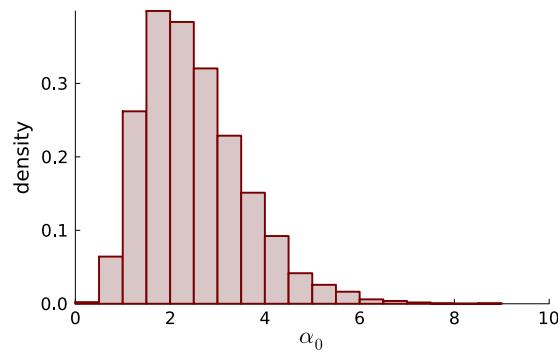


Figure 13.19: Fish length data. Posterior distribution for the prior precision α_0 when $\alpha_0 \sim \text{Gamma}(1, 1)$ prior.

14 Model comparison and variable selection

14.1 Posterior model probabilities and the marginal likelihood

In most applications we have more than one potential model for the data. For example, count data can be modelled with a Poisson, geometric or negative binomial distribution. Income data can be modelled by a log-normal or a Gamma distribution. In regression analysis we usually have a multitude of models formed from different combinations of the covariates. This variable selection problem will be discussed in detail in Section [Bayesian variable selection](#).

Let $\mathcal{M} = \{M_1, \dots, M_K\}$ denote the set of potential models for a dataset \mathbf{x} . Each model has its own set of parameters, θ_k for model M_k . Consider first the rather unrealistic **\mathcal{M} -closed** case where one of these models is believed to be the **data generating process** (DGP). The Bayesian solution to the model comparison problem is then clear: compute the posterior distribution for the unknown true model $M \in \mathcal{M}$:

$$\Pr(M = M_k | \mathbf{x}) \propto p(\mathbf{x} | M_k) \cdot \Pr(M_k), \quad (14.1)$$

where $\Pr(M = M_k)$ is the prior distribution over \mathcal{M} and $p(\mathbf{x} | M_k)$ is the probability of the observed data \mathbf{x} in model M_k . Table 14.1 is an example where a uniform prior distribution over four models $\mathcal{M} = \{M_1, \dots, M_4\}$ is updated to posterior distribution; after observing the data, model M_2 is the most probable model.

	M_1	M_2	M_3	M_4
$\Pr(M_k)$	0.25	0.25	0.25	0.25
$\Pr(M_k \mathbf{y})$	0.05	0.81	0.10	0.04

\mathcal{M} -closed
data generating process

Table 14.1: Example of prior-to-posterior updating of model probabilities.

The likelihood contribution to (14.1), $p(\mathbf{x} | M_k)$, does not condition on the parameters θ_k in model M_k ; the parameters have been marginalized out and

$$p(\mathbf{x} | M_k) = \int p(\mathbf{x} | \theta_k, M_k) p(\theta_k | M_k) d\theta_k, \quad (14.2)$$

is therefore usually called the **marginal likelihood**. The alternative

marginal likelihood

name **evidence** is often used in machine learning. It is important to note that the parameters are integrated out by the *prior* and that the marginal likelihood is the prior expected likelihood function:

$$p(\mathbf{x}|M_k) = \mathbb{E}_{\theta_k}(p(\mathbf{x}|\theta_k, M_k)). \quad (14.3)$$

The marginal likelihood is therefore the **prior predictive distribution** for the training data $p(\mathbf{x}|M_k)$ when the parameters are drawn from the prior distribution. The marginal likelihood $p(\mathbf{x}|M_k)$ is therefore typically much more sensitive to the prior $p(\theta_k|M_k)$ than the posterior $p(\theta_k|\mathbf{x}, M_k)$ for the model parameters. We will explore this prior sensitivity in this chapter, and also present some alternative model comparison measures that are less sensitive to the prior.

The **Bayes factor** comparing model M_1 to model M_2 is defined as

$$B_{12}(\mathbf{x}) = \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})}. \quad (14.4)$$

The (modified) Jeffreys' scale of evidence (Kass and Raftery, 1995) is often used to interpret the strength of evidence of a Bayes factor:

- Barely worth mentioning: 1–3
- Positive: 3–20
- Strong: 20–150
- Very strong: > 150.

This scale is rather arbitrary, but can potentially be useful as a rough guide.

BERNOULLI MODEL

Let $x_1, \dots, x_n \stackrel{\text{iid}}{\sim} \text{Bern}(\theta)$ and assume the prior $\theta \sim \text{Beta}(\alpha, \beta)$. The marginal likelihood is then

$$\begin{aligned} p(x_1, \dots, x_n) &= \int p(x_1, \dots, x_n | \theta) p(\theta) d\theta \\ &= \int \theta^s (1-\theta)^f \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} d\theta \\ &= \frac{1}{B(\alpha, \beta)} \int \theta^{\alpha+s-1} (1-\theta)^{\beta+f-1} d\theta \\ &= \frac{B(\alpha+s, \beta+f)}{B(\alpha, \beta)}, \end{aligned}$$

where the last equality follows since the integral is with respect to the kernel of the $\text{Beta}(\alpha+s, \beta+f)$ density. Note that we need to retain the normalizing constant $1/B(\alpha, \beta)$ in the prior when computing a marginal likelihood; we are not allowed to use the proportional form of Bayes' theorem here.

evidence

prior predictive distribution

Bayes factor

14.2 Normal model

Consider first the iid Normal model $x_1, \dots, x_n \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$ with known σ^2 . We will compare two versions of this model: a null model M_0 where $\theta = \mu_0$ exactly, and a model M_1 with unrestricted θ following $\theta \sim N(\mu_0, \sigma^2/\kappa_0)$ a priori. This can be seen as the Bayesian equivalent of testing a sharp null hypothesis $H_0 : \theta = \mu_0$ vs $H_1 : \theta \neq \mu_0$. Note that the prior in the unrestricted model M_1 is centered on the null hypothesis, which is sensible given the hypothesis testing setup.

The marginal likelihood for model M_1 is obtained by integrating the likelihood with respect to the prior for the unknown θ :

$$p(\mathbf{x}|M_1) = \int \prod_{i=1}^n N(x_i|\theta, \sigma^2) N(\theta|\mu_0, \sigma^2/\kappa_0) d\theta. \quad (14.5)$$

This integral can be calculated by completing the squares in the exponentials of the two Gaussian densities and integrating out θ using properties of the normal density. We will take a different route here that highlights the role of the sample mean \bar{x} in the Bayes factor comparing M_0 to M_1 .

Using the same algebra as when deriving the posterior for θ in the normal model in Chapter [Single-parameter models](#) we can express the likelihood as

$$\begin{aligned} p(\mathbf{x}|\theta, \sigma^2) &= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2\sigma^2}n(\bar{x}-\theta)^2\right) \\ &= c(\sigma^2, s^2) N(\bar{x}|\theta, \sigma^2/n), \end{aligned}$$

where $s^2 = (n-1)^{-1} \sum_{i=1}^n (x_i - \bar{x})^2$, $c(\sigma^2, s^2) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{ns^2}{2\sigma^2}\right) (2\pi\sigma^2/n)^{1/2}$ and $N(\bar{x}|\theta, \sigma^2/n)$ denotes the density function of the sample mean: $\bar{x}|\theta, \sigma^2 \sim N(\theta, \sigma^2/n)$. The constant $c(\sigma^2, s^2)$ will be shown to appear in both $p(\mathbf{x}|M_0)$ and $p(\mathbf{x}|M_1)$, and will therefore cancel out in the Bayes factor.

The marginal likelihood under M_0 is trivial since this model does not contain any unknown parameters, so we just insert $\theta = \mu_0$ in the likelihood:

$$p(\mathbf{x}|M_0, \sigma^2) = c(\sigma^2, s^2) N(\bar{x}|\mu_0, \sigma^2/n).$$

The marginal likelihood for model M_1 is

$$\begin{aligned} p(\mathbf{x}|M_1, \sigma^2) &= \int p(\mathbf{x}|\theta) p(\theta) d\theta \\ &= c(\sigma^2, s^2) \int N(\bar{x}|\theta, \sigma^2/n) N(\theta|\mu_0, \sigma^2/\kappa_0) d\theta. \end{aligned}$$

We have seen a similar integral when deriving the predictive distribution for the iid Gaussian model, $p(\tilde{x}|\mathbf{x}) = \int N(\tilde{x}|\theta, \sigma^2) N(\theta|\mu_n, \tau_n^2) d\theta$

as $N(\bar{x}|\mu_n, \sigma^2 + \tau_n^2)$. Analogous arguments shows that

$$p(\mathbf{x}|M_1, \sigma^2) = c(\sigma^2, s^2)N(\bar{x}|\mu_0, \sigma^2(1/n + 1/\kappa_0)), \quad (14.6)$$

and the Bayes factor for a given σ^2 is

$$\text{BF}_{01}(\mathbf{x}, \sigma^2) = \frac{p(\mathbf{x}|M_0, \sigma^2)}{p(\mathbf{x}|M_1, \sigma^2)} = \frac{N(\bar{x}|\mu_0, \sigma^2/n)}{N(\bar{x}|\mu_0, \sigma^2(1/n + 1/\kappa_0))}. \quad (14.7)$$

The expression in (14.7) shows that the Bayes factor compares prior predictive densities for the two models with respect to the data compressed into the sufficient statistic \bar{x} . We can also clearly see the limiting behavior of BF_{01} with respect to the prior sample size κ_0 :

- $B_{01} \rightarrow 1$ as $\kappa_0 \rightarrow \infty$. The prior under M_1 tends to a point mass at $\theta = \mu_0$ when $\kappa_0 \rightarrow \infty$, and M_0 and M_1 are therefore identical models in the limit.
- $B_{01} \rightarrow \infty$ as $\kappa_0 \rightarrow 0$, regardless of how close \bar{x} is to μ_0 . This is the case since the $\mathbb{V}(\bar{x}|M_1) = \sigma^2(1/n + 1/\kappa_0) \rightarrow \infty$ as $\kappa_0 \rightarrow 0$; model M_1 therefore assigns lower and lower predictive density to the observed \bar{x} when $\kappa_0 \rightarrow 0$. A marginal likelihood evaluates the combination of a likelihood and a prior; if you make your prior "stupid" enough, the simpler null model M_0 will eventually win, even when \bar{x} is not very likely to come from M_0 .

The Bayes factor when the variance is assumed unknown is obtained by integrating $p(\mathbf{x}|M_0, \sigma^2)$ and $p(\mathbf{x}|M_1, \sigma^2)$ with respect to the $\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$ prior. The end result is a ratio of two student- t distributions for \bar{x} and is not given here.

INTERNET SPEED DATA. Figure 14.1 plots the Bayes Factor comparing $M_0: N(20, 5^2)$ to $M_1: N(\theta, 5^2)$ for the internet speed data as a function of the prior sample size κ_0 . The shaded region marks out the κ_0 where $\text{BF}_{01} > 1$, i.e. where the evidence supports M_0 . The region for "barely worth mentioning" in the Jeffreys scale of evidence for is marked out by horizontal orange dashed lines. Unless the prior is very spread out, there is no evidence in favor of either model.

Figure 14.2 illustrates how the prior predictive density assigns increasingly lower density to the observed $\bar{x} = 15.99$ when κ_0 decreases.

Figure 14.3 illustrates the Bayes factor for the internet speed data with \bar{x} artificially changed from 15.99 to $\bar{x} = 12$; the figure plots both the Bayes factor and the Jeffreys scale of evidence in logs for visibility. With \bar{x} so far from the null value $\mu_0 = 20$, there is now positive or even close to strong evidence in favor of M_1 for all $\kappa_0 \in (0.01, 3)$. This is also clear from Figure 14.2 if we move the purple data point to $\bar{x} = 12$.

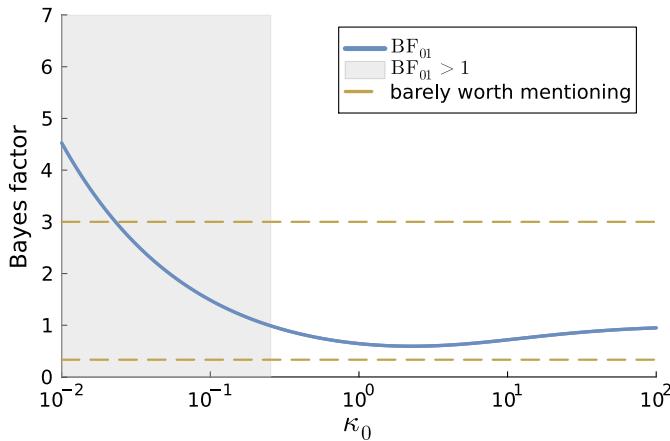


Figure 14.1: Bayes factor for the internet speed data with known variance $\sigma^2 = 5^2$. The graph plots the Bayes factor BF_{01} as a function of the prior sample size κ_0 in log-scale. The shaded region shows the values for κ_0 where $BF_{01} > 1$, i.e. where there is support in favor of the null model. The limits for "barely worth mentioning" in the Jeffreys scale of evidence are marked out as horizontal orange dashed lines.

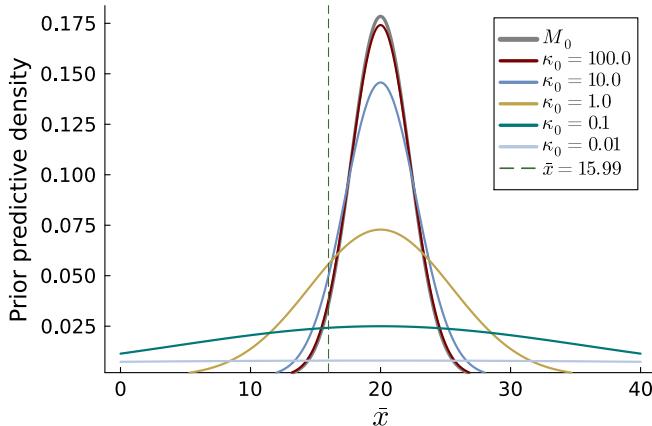


Figure 14.2: Internet speed data with known variance $\sigma^2 = 5^2$. Prior predictive densities for \bar{x} in the models M_0 and M_1 for different values of the prior hyperparameter κ_0 . The realized data of $\bar{x} = 15.99$ is shown as a purple dot with dashed line.

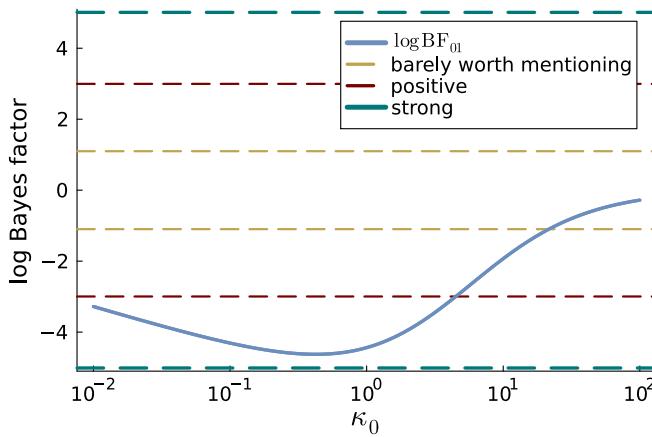


Figure 14.3: Log Bayes factor for the internet speed data with \bar{x} artificially set to $\bar{x} = 12$ instead of the actually observed $\bar{x} = 15.99$. The graph plots the log Bayes factor BF_{01} as function of the prior sample size κ_0 in log-scale. The limits for Jeffreys' scale of evidence (in logs) are marked out as horizontal dashed lines.

Properties of posterior model probabilities

EXAMPLE: GEOMETRIC vs POISSON

Consider count data and the comparison of the two models:

- $M_1: x_1, \dots, x_n | \theta_1 \stackrel{\text{iid}}{\sim} \text{Geo}(\theta_1)$ with prior $\theta_1 \sim \text{Beta}(\alpha_1, \beta_1)$
- $M_2: x_1, \dots, x_n | \theta_2 \stackrel{\text{iid}}{\sim} \text{Poisson}(\theta_2)$ with prior $\theta_2 \sim \text{Gamma}(\alpha_2, \beta_2)$.

See Box 2.12 for the definition of the geometric distribution.

The marginal likelihoods are (see Exercise X)

$$\begin{aligned} p(x_1, \dots, x_n | M_1) &= \int p(x_1, \dots, x_n | \theta_1, M_1) p(\theta_1 | M_1) d\theta_1 \\ &= \frac{\Gamma(\alpha_1 + \beta_1)}{\Gamma(\alpha_1) \Gamma(\beta_1)} \frac{\Gamma(n + \alpha_1) \Gamma(n\bar{y} + \beta_1)}{\Gamma(n + n\bar{y} + \alpha_1 + \beta_1)}. \end{aligned}$$

and

$$\begin{aligned} p(x_1, \dots, x_n | M_2) &= \int p(x_1, \dots, x_n | \theta_2, M_2) p(\theta_2 | M_2) d\theta_2 \\ &= \frac{\Gamma(n\bar{y} + \alpha_2)}{\Gamma(\alpha_2)(n + \beta_2)^{n\bar{y} + \alpha_2}} \frac{1}{\prod_{i=1}^n y_i!}. \end{aligned}$$

For consistency, we set $\alpha_1/\beta_1 = \beta_2/\alpha_2$ so that both models have the same prior predictive mean, $\mathbb{E}(\tilde{x}|M_1) = E(\tilde{x}|M_2)$ (Bernardo and Smith, 2009). We will specifically use $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 10$ in the illustrations, and equal prior model probabilities $\Pr(M_1) = \Pr(M_2) = 1/2$.

To investigate how the posterior model probabilities $\Pr(M_1|x)$ and $\Pr(M_2|x)$ behave as the sample size grows large, I simulate a data set with $n = 500$ from the $\text{Pois}(\theta_2 = 1)$ model, so the M_2 is the true data generating process. We then compute $\Pr(M_2|x)$ sequentially using a larger and larger sample size until all $n = 500$ observations have been used up. Figure 14.4 shows the results from this experiment repeated four times to also see the sampling variation. The graph to the left in Figure 14.4 zooms in on the first $n = 100$ observations; there is quite some sampling variability in the model probabilities, but there is a clear tendency for the posterior probability on the Poisson model to tend to 1. The right hand graph shows the results for the full sample of $n = 500$ observations; the probability $\Pr(M_2|x)$ clearly tends to 1 for all four replications.

The asymptotic behavior in Figure 14.4 is what one would expect, and one can indeed prove that Bayesian posterior model probabilities are consistent in the \mathcal{M} -closed setting where the data generating process is among the compared models:

$$\Pr(M_k^*|x) \xrightarrow{p} 1 \text{ as } n \rightarrow \infty, \quad (14.8)$$

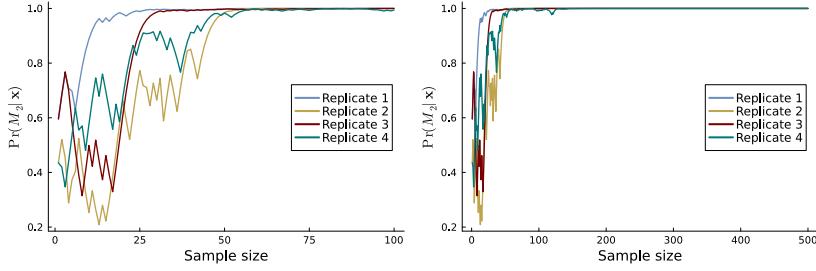


Figure 14.4: Asymptotic behavior of posterior model probabilities in \mathcal{M} -closed when comparing the models:

$M_1: \text{Geo}(\theta_1), \theta_1 \sim \text{Beta}(10, 10)$

$M_2: \text{Pois}(\theta_2), \theta_2 \sim \text{Gamma}(10, 10)$

~

The graphs show the evolution of the posterior probability for the Poisson model as the sample size increases. Each line corresponds to a replication of the experiment. The data are generated from the iid $\text{Pois}(1)$ model. The left graph shows the subset of the first 100 data points and the right graph shows all 500 data points.

where M_k^* is the data generating process.

What happens asymptotically when the data generating process is not among the compared models? This **\mathcal{M} -open** setting is more realistic since models are typically just approximations to reality. To explore this let us change previous experiment and generate data from a negative binomial distribution; see Figure ?? for the definition of the negative binomial distribution.

Figure 14.5 (left) shows the asymptotic behaviour of the posterior model probabilities for the Poisson and Geometric models when both models are wrong and data actually comes from the $\text{NegBin}(2, 0.5)$ distribution; the posterior probabilities seem to converge to a solution where the Geometric model gets a probability of one as n grows.

The right hand graph of the figure explains why this is happening by plotting the $\text{NegBin}(2, 0.5)$ data generating distribution as a bar chart with the optimal fit of each compared model overlayed. The optimal fit is defined as the fit that minimizes the Kullback-Leibler divergence of the model from the data generating process. Specifically, let $g_\theta(x)$ be the data density of a model and let $f(x)$ denote the data generating process. The optimal fit for the model $g_\theta(x)$ is then obtained by minimizing the Kullback-Leibler divergence

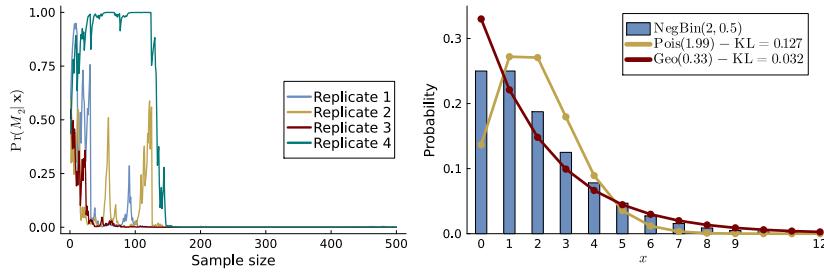
$$d(f, g) = \int \log \left(\frac{f(x)}{g_\theta(x)} \right) f(x) dx$$

with respect to the model parameters θ .

The legend of Figure 14.5 (right) shows that the Geometric model is closer to the data generating process (smaller KL divergence) than the Poisson model which explains why the Geometric model wins asymptotically. The asymptotic tendency seen in Figure 14.5 can be proved to hold quite generally in that

$$\Pr(M_k^*|x) \xrightarrow{P} 1 \text{ as } n \rightarrow \infty, \quad (14.9)$$

where M_k^* is the model in \mathcal{M} with the smallest Kullback-Leibler divergence from the data generating process.



Marginal likelihood in linear regression

The marginal likelihood for the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \stackrel{iid}{\sim} N(0, \sigma^2 I_n), \quad (14.10)$$

is given by

$$p(\mathbf{y}|\mathbf{X}) = \int \int p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{X}) p(\boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2. \quad (14.11)$$

The marginal likelihood is a special case of the posterior predictive distribution in Figure ?? when the posterior is based on $n = 0$ data points, i.e. when the parameters are integrated with respect to the prior, and the object of prediction is the training data \mathbf{y} ; for this reason, the marginal likelihood is sometimes called the **prior predictive distribution**. Note that the marginal likelihood is not measuring in-sample training error since the prediction for the training data \mathbf{y} is only using prior information for the model parameters $\boldsymbol{\beta}$ and σ^2 . Hence setting $n = 0$ and $\tilde{\mathbf{y}} = \mathbf{y}$ we immediately have the marginal likelihood for the linear regression model

$$\mathbf{y}|\mathbf{X} \sim t_{\nu_0} \left(\mathbf{X}\boldsymbol{\mu}_0, \sigma_0^2 (\mathbf{I}_n + \mathbf{X}\Omega_0^{-1}\mathbf{X}^\top) \right). \quad (14.12)$$

TODO! add comparison of models with different predictors for the salaries data. Then point forward to variable selection section and take it up there again.

14.3 The Laplace approximation of the marginal likelihood

There are many methods for approximating the marginal likelihood when it cannot be derived analytically. An obvious approach comes from the marginal likelihood being the prior expected likelihood

$$p(\mathbf{x}) = \int p(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \mathbb{E}_{p(\boldsymbol{\theta})} p(\mathbf{x}|\boldsymbol{\theta}),$$

and can therefore be computed by simple Monte Carlo simulation

$$\widehat{p}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m p(\mathbf{x}|\boldsymbol{\theta}^{(i)}), \quad (14.13)$$

Figure 14.5: Asymptotic behavior of posterior model probabilities in \mathcal{M} -open when comparing the models:
 $M_1: \text{Geo}(\theta_1), \theta_1 \sim \text{Beta}(10, 10)$
 $M_2: \text{Pois}(\theta_2), \theta_2 \sim \text{Gamma}(10, 10).$

The left graph shows the evolution of the posterior probability for the Poisson model as the sample size increases. Each line corresponds to a replication of the experiment. The data are generated from the iid NegBin(2, 0.5) model. The right graph shows the fit of the models with KL-optimal parameters.

prior predictive distribution

where $\theta^{(i)} \stackrel{\text{iid}}{\sim} p(\theta)$ are m draws from the prior.

Unfortunately, the simple Monte Carlo estimator in (14.13) usually has disastrously large variance and is rarely used in practice. The problem with the estimator in (14.13) is that the likelihood is often much more concentrated than the prior and the estimate will then be dominated by the few prior draws that happen to end up where the likelihood is concentrated. Importance sampling can be used to reduce the variance, see for example the modified harmonic estimator in [Geweke \(1999\)](#). There are also many methods based on MCMC, in particular Chib's methods for Gibbs sampling ([Chib, 1995](#)) and its extension to Metropolis-Hastings ([Chib and Jeliazkov, 2001](#)). We will here present a simple but often quite accurate method for approximating the marginal likelihood, the Laplace approximation.

The Laplace approximation of the log marginal likelihood for a model with p parameters is

$$\ln \hat{p}(x) = \ln p(x|\hat{\theta}) + \ln p(\hat{\theta}) + (1/2) \ln |J_{x,\hat{\theta}}^{-1}| + (p/2) \ln(2\pi), \quad (14.14)$$

where $\hat{\theta}$ is the posterior mode and $|J_{x,\hat{\theta}}|$ is the determinant of the observed information matrix as in Chapter [Multi-parameter models](#), but here defined for the posterior instead of the likelihood:

$$J_{\hat{\theta},x} = -\frac{\partial^2 \ln p(x|\theta)p(\theta)}{\partial \theta \partial \theta^\top} \Big|_{\theta=\hat{\theta}}, \quad (14.15)$$

where $\hat{\theta}$ is the posterior mode.

BERNOULLI MODEL. We have already computed the marginal likelihood for the Bernoulli model in closed form earlier in this chapter, so there is really no need to approximate it. However, it gives us a chance to practice deriving the marginal likelihood and we can also assess how accurate the approximation is since we know the true answer here. We have:

$$\begin{aligned} \ln p(x|\theta)p(\theta) &= (\alpha + s - 1) \ln \theta + (\beta + f - 1) \ln(1 - \theta) \\ \frac{\partial \ln p(x|\theta)p(\theta)}{\partial \theta} &= \frac{\alpha + s - 1}{\theta} - \frac{\beta + f - 1}{1 - \theta} \\ \frac{\partial^2 \ln p(x|\theta)p(\theta)}{\partial \theta^2} &= -\frac{\alpha + s - 1}{\theta^2} - \frac{\beta + f - 1}{(1 - \theta)^2} \end{aligned}$$

Solving $\partial \ln p(x|\theta)p(\theta)/\partial \theta = 0$ for θ gives the posterior mode

$$\hat{\theta} = \frac{\alpha + s - 1}{\alpha + \beta + n - 2},$$

and therefore

$$J_{x,\hat{\theta}}^{-1} = -\left[\frac{\partial^2 \ln p(\theta|x)}{\partial \theta^2} \Big|_{\theta=\hat{\theta}} \right]^{-1} = \frac{(\alpha + s - 1)(\beta + f - 1)}{(\alpha + \beta + n - 2)^3}.$$

To examine the accuracy of this approximation, let us consider a dataset with $s = 6$ successes in $n = 10$ trials and the uniform prior with $\alpha = \beta = 1$. Here, $\hat{\theta} = s/n = 0.6$ and $J_{x,\hat{\theta}}^{-1} = sf/n^3 = 0.024$. The Laplace approximation of the log marginal likelihood in (14.14) is therefore

$$\ln \hat{p}(x) = 6 \ln(0.6) + 4 \ln(0.4) + (1/2) \ln(0.024) + (1/2) \ln(2\pi) \approx -7.676,$$

which is quite close to the true log marginal likelihood $\ln p(x) = -7.745$. Consider for example using this marginal likelihood for comparing a model against a null model where $\theta = 0.5$. The true Bayes factor is then $0.5^{10}/\exp(-7.745) \approx 2.559$ and the Bayes factor from the Laplace approximation is $0.5^{10}/\exp(-7.676) \approx 2.105$; the approximate Bayes factor and the exact Bayes factor both lead to the conclusion that the evidence in favor of the null model is "barely worth mentioning" according to the Jeffreys scale of evidence.

14.4 Log predictive score

The marginal likelihood is by construction usually sensitive to the exact specification of the prior. A precise prior elicitation is sometimes hard, or at least time-consuming, particularly in models with many parameters where the prior dependence can be especially hard to get right. Several alternative measures for Bayesian model comparison that are less sensitive to the prior have therefore been developed. The log predictive score measure in this section sacrifices some data to make the marginal likelihood more robust to variations in the prior.

The marginal likelihood is the joint prior predictive distribution for all observations and can therefore be decomposed as sequences of conditional densities:

$$p(x_1, \dots, x_n) = p(x_1)p(x_2|x_1) \cdots p(x_n|x_1, x_2, \dots, x_{n-1}) \quad (14.16)$$

The i th factor in this decomposition is the intermediate predictive density

$$p(x_i|x_1, \dots, x_{i-1}) = \int p(x_i|x_1, \dots, x_{i-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|x_1, \dots, x_{i-1}) d\boldsymbol{\theta},$$

where $p(\boldsymbol{\theta}|x_1, \dots, x_{i-1})$ is the intermediate posterior for $\boldsymbol{\theta}$ conditional on the data subset x_1, \dots, x_{i-1} . For iid data we have the usual simplification $p(x_i|x_1, \dots, x_{i-1}, \boldsymbol{\theta}) = p(x_i|\boldsymbol{\theta})$.

In a time series context where the observations have a natural ordering in time, the factor $p(x_i|x_1, \dots, x_{i-1})$ in the decomposition in (14.16) is the one-step-ahead predictive distribution for the observation at time i given data up to time $i - 1$. When the data are not specifically ordered, for example iid data, the decomposition in

(14.16) can be done in many different ways by ordering the observations differently; we return this interpretation later in this section.

The decomposition in (14.16) is interesting for at least three reasons. First, it can be used to diagnose why a model has a low marginal likelihood by inspecting each of the terms in the decomposition to see which observations are poorly predicted. Second, it gives a clear connection between the marginal likelihood and sequential out-of-sample predictive performance of a model, particularly for time series data. Third, the decomposition in (14.16) can be used to highlight the effect of the prior on the marginal likelihood and suggest a way to reduce the influence of the prior in Bayesian model comparisons using the marginal likelihood.

To elaborate on this last point consider the iid Normal model with known variance: $x_1, \dots, x_n | \theta \stackrel{\text{iid}}{\sim} N(\theta, \sigma^2)$ with prior $\theta \sim N(\mu_0, \sigma^2/\kappa_0)$. We will be particularly interested in the sensitivity of the marginal likelihood with respect to κ_0 . The intermediate predictive distribution for observation x_i in decomposition (14.16) is

$$x_i | x_1, \dots, x_{i-1} \sim N\left(\mu_{i-1}, \sigma^2 \left(1 + \frac{1}{i-1+\kappa_0}\right)\right), \quad (14.17)$$

where $\mu_{i-1} = w_{i-1}\bar{x}_{i-1} + (1-w_{i-1})\mu_0$, \bar{x}_{i-1} is the sample mean of the first $i-1$ observations, and $w_{i-1} = (i-1)/(i-1+\kappa_0)$. This result is simply the predictive distribution for the Gaussian model with known variance in [Prediction and Decision making](#) with $n = i-1$ data points in the posterior.

Consider now $n = 100$ observations simulated from the $N(20, 5^2)$ distribution, to mimic the setting in the Internet speed data; the original dataset with only $n = 5$ observations is too small for the point I want to make here. The upper graph in Figure 14.6 plots the log of the marginal likelihood decomposition

$$\log p(x_1, \dots, x_n) = \log p(x_1) + \log p(x_2|x_1) + \dots + \log p(x_n|x_1, \dots, x_{n-1}), \quad (14.18)$$

for three different values of κ_0 . The log marginal likelihood for the model with $\kappa_0 = 1$ is for example the sum of the values in the orange line. A careful examination of the graph shows that the prior sensitivity of log marginal likelihoods is entirely driven by the first term in the decomposition (14.18). The lower graph in Figure 14.6 makes this more visible by zooming in on the 20 first observations. This comes as no surprise since it is clear from (14.17) that the first terms will be affected by κ_0 , but the later terms in the sequence where i is large remain essentially unaffected by κ_0 .

An obvious way to reduce the prior sensitivity while still remaining close to the marginal likelihood is therefore to discard the first terms in (14.18). This is the **log predictive score (LPS)**:

log predictive score

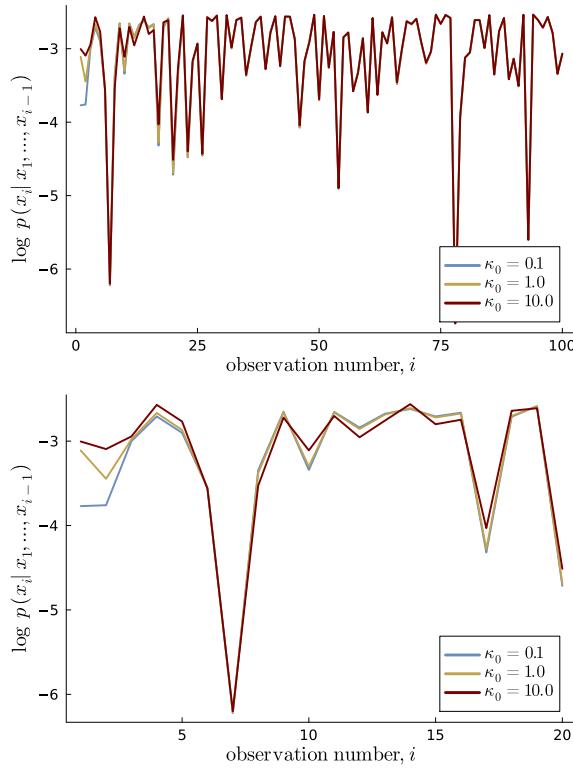


Figure 14.6: Decomposition of the log marginal likelihood for the simulated internet speed data with $n = 100$ observations for three different values for the prior sample size κ_0 . The bottom graph zooms in on the gray shaded region with the first 20 observations.

$$\text{LPS} = \sum_{i=i^*+1}^n \log p(x_i | x_1, \dots, x_{i-1}). \quad (14.19)$$

The LPS is effectively using the first i^* observations to train the prior $p(\theta)$ into an intermediate posterior $p(\theta | x_1, \dots, x_{i^*})$ which is then used as the new prior for the remaining test data x_{i^*+1}, \dots, x_n . There are also variants of LPS which scales by $1/(n - i^*)$ so that the LPS is the average log predictive observation per test observation. The form in (14.19) has the advantage that Jeffreys' scale of evidence can still be used since the number of terms in the LPS is the number of test observations; the training data have been sacrificed to reduce the sensitivity to the prior and can therefore not be used in the evidence for the model.

Figure 14.7 plots the LPS as a function of the training fraction $f = i^*/n$ for each of the three κ_0 values. The LPS in the figure is scaled by $n/(n - i^*)$ to keep the same scale on the LPS for all training fractions for presentation purposes. The LPS in Figure 14.7 with training fraction $f = 0$ is the original log marginal likelihood where the prior (κ_0) has a substantial effect on the LPS. Already with a training fraction of 15% is the LPS insensitive to $\kappa_0 \in [0.1, 10]$.

The LPS in (14.19) discards the first i^* observations. This makes sense for time series where the observations are ordered in time.

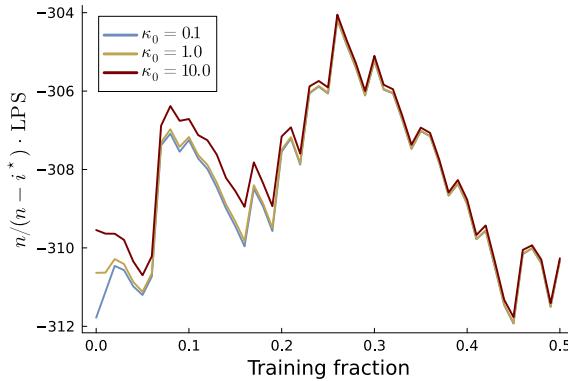


Figure 14.7: Log predictive score (scaled) as a function of the training fraction for the simulated internet speed data with $n = 100$ observations.

For cross-sectional data, e.g. iid data, there is no natural ordering and it is then common practise to use a cross-validated version of the LPS. The idea with **K-fold cross-validated LPS** is to split, or partition, the data into K folds, use one of the K folds for training and then evaluate the predictive performance on the $K - 1$ folds left out. This is repeated K times, each time with a new fold as the training fold. Table 14.2 illustrates the data partitioning. Note that this is different from the usual cross-validation used in machine learning where instead $K - 1$ folds would be used for training and the single remaining used for testing. The reason is that cross-validation in machine learning aims at estimating the generalization performance of the model on future data. The cross-validated LPS still aims for something close to the marginal likelihood, but uses cross-validation to lessen the arbitrary choice of which observations to use in the training and test when computing the LPS. Bayesian cross-validation methods that aim to estimate the generalization performance of the model are discussed in the next section.

K-fold cross-validated LPS

n data observations					
	1, 2, ..., $n - 1, n$				
Split 1:	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2:	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3:	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4:	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5:	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Table 14.2: The data partitioning for 5-fold cross-validation of the LPS. For each of the K splits, the observations in each fold in blue is used to train the prior into a posterior. The observations in the remaining folds in the same row are used to compute the LPS for the split.

14.5 Bayesian estimators of generalization performance

Leave-one-out and cross-validation. WAIC?

14.6 Bayesian variable selection

Regression and classification modeling involves deciding on which covariates to include in the model. This can often be determined from an understanding of the application area, but in many problems the number of possible covariates is large and it may not be obvious what the relevant covariates are.

We have already seen that using too many covariates can lead to overfitting and that regularization priors, like the normal and Laplace, can help to mitigate this problem. Another way to manage many potential covariates is *variable selection* where the goal is to select a subset of the covariates that best explains the response variable, without overfitting. A classical variable selection method is **forward selection**; this method starts by including the single best covariate (for example the one with largest absolute t -ratio) and then sequentially adds more covariates given the previously included covariates until some stopping criteria is met (for example that all the remaining covariates have a t -ratio smaller than some threshold).

Bayesian variable selection is another way to select the best covariate subset, where one obtains a probability distribution over all possible subsets of covariates. Since each covariate subset corresponds to a separate regression model, this is essentially a model comparison problem where we compute posterior model probabilities over models. A single best subset can then be chosen based on this posterior distribution, for example the subset with highest posterior model probability. Alternatively, we do not need to *choose* a specific subset, we can instead *average* over all subset models using the posterior model probabilities as weights. For example, if the aim is prediction, we can average the predictive distributions from each of the subset models to obtain a predictive distribution that marginalizes over the correct subset of covariates in the model; the uncertainty about the correct subset is then included in the predictive distribution.

Another view of Bayesian variable selection is that it is just another regularization prior for the regression coefficients $\beta_j, j = 1, \dots, p$, just like the normal and Laplace priors, but with a particular mixture distribution with a point mass at $\beta_j = 0$. This point mass explicitly encodes the information that β_j may be *exactly* zero, which of course implies that the j th covariate is excluded from the model.

The traditional Bayesian approach to variable selection uses a particular mixture distribution for the regression coefficients $\beta_j, j = 1, \dots, p$. The **spike-and-slab** prior is a mixture of a **point mass distribution** (see Figure 14.8 and Box 14.1) at $\beta_j = 0$ and a

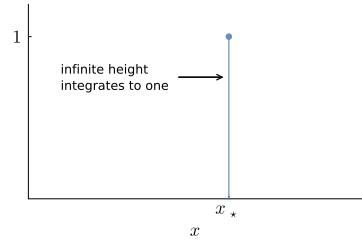


Figure 14.8: Illustration of a Dirac delta function as a point mass distribution at $x = x_*$.

forward selection

Bayesian variable selection

Point mass distribution

A point mass distribution has all its probability mass at the single point x_* and is defined using a **Dirac delta function** $\delta_{x_*}(x)$ located at x_* . Informally this can be viewed as a density function for the random variable X with infinite height at x_* that integrates to one over the support of X :

$$\int_{-\infty}^{\infty} \delta_{x_*}(x) dx = 1,$$

with the property that

$$\int_{-\infty}^{\infty} f(x) \delta_{x_*}(x) dx = f(x_*),$$

for an integrable function $f(x)$.

Box 14.1: Point mass distribution using a Dirac Delta function.

spike-and-slab

normal distribution with a large variance:

$$p(\beta_j | \sigma^2) = \omega \cdot N(\beta_j | 0, \tau^2 \sigma^2) + (1 - \omega) \cdot \delta_0(\beta_j), \quad (14.20)$$

where the second mixture component $\delta_0(\beta_j)$ with weight $1 - \omega$ is a Dirac delta function, i.e. a point mass at $\beta_j = 0$. This *spike* component therefore gives particular prior mass to the situation where $\beta_j = 0$, i.e. when the j th covariate is excluded from the model. The other mixture component, the *slab*, with weight ω is a normal distribution with mean zero and a large variance $\tau^2 \sigma^2$, which gives prior mass to the situation where $\beta_j \neq 0$, i.e. when the j th covariate does have a significant effect on the response variable. The spike-and-slab prior is illustrated in Figure 14.9.

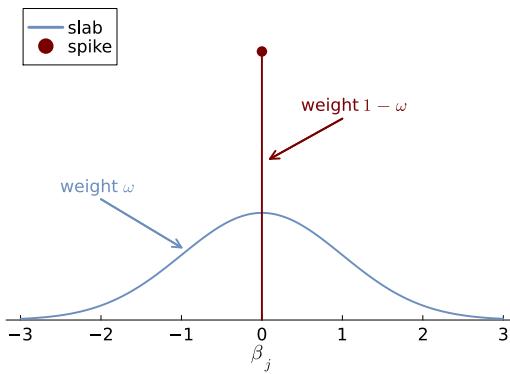


Figure 14.9: Illustration of the spike-and-slab prior. Note that the spike is a Dirac delta function, i.e. a point mass at zero, which in a way has finite height but is drawn with finite height for illustration purposes.

We can express the spike-and-slab prior using a mixture allocation variable z_j for each covariate j , where $z_j = 1$ indicates that the j th covariate is included in the model ($\beta_j \neq 0$) and $z_j = 0$ indicates that the j th covariate is excluded from the model ($\beta_j = 0$). The spike-and-slab prior is then

$$z_1, \dots, z_p \sim \text{Bernoulli}(\omega) \quad (14.21)$$

$$\beta_j | z_j \sim \begin{cases} N(0, \tau^2 \sigma^2) & \text{if } z_j = 1 \\ = 0 & \text{if } z_j = 0 \end{cases} \quad (14.22)$$

This will be useful when we later design a Gibbs sampling algorithm for a regression model with a spike-and-slab prior. Note that there is here an allocation variable z_j connected to each *covariate* in the model, not to each *observation* like we have seen when analyzing the mixture models for the data earlier in the chapter. It is therefore common to refer to the z_j as *variable selection indicators* in the context of Bayesian variable selection. Consider an example with $p = 3$ potential covariates. The outcome $\mathbf{z} = (z_1, z_2, z_3) = (1, 0, 1)$ means that the first and third covariates are included in the model ($\beta_1 \neq 0$ and $\beta_3 \neq 0$) whereas the second covariate is not ($\beta_2 = 0$).

The weight ω is called the **prior inclusion probability** and a small ω therefore encodes the prior information that few covariates are expected to have non-zero regression coefficients, i.e. it encodes *spar-sity*, a form of regularization that prevents overfitting. Hence we see again that regularization comes from the prior distribution. The spike-and-slab prior is also used in Bayesian variable selection in high-dimensional regression models, where the number of covariates p is much larger than the number of observations n . This is possible since the effective model size, i.e. the number of covariate with non-zero β_j , is typically much smaller than p due to the spike-and-slab prior.

Since the spike-and-slab prior is a mixture it is tempting to set up a Gibbs sampler where the regression coefficients β and the error variance σ^2 are drawn conditional on the variable selection indicators $\mathbf{z} = (z_1, \dots, z_p)$ and then update the selection indicators conditional on β and σ^2 . The fact that one of the mixture components is a Dirac delta spike leads to a problem with this approach. Since $z_j = 0$ dictates that $\beta_j = 0$, then as soon as the Gibbs sampler assigns $z_j = 0$ to a covariate j , the regression coefficient β_j will be stuck at zero for the rest of the Gibbs iterations. The solution to this problem is to instead draw from the *joint* posterior using the following marginal-conditional decomposition

$$p(\beta, \sigma^2, \mathbf{z} | \mathbf{y}, \mathbf{X}) = p(\beta, \sigma^2 | \mathbf{z}, \mathbf{y}, \mathbf{X}) p(\mathbf{z} | \mathbf{y}, \mathbf{X}). \quad (14.23)$$

Simulating from the first factor $p(\beta, \sigma^2 | \mathbf{z}, \mathbf{y}, \mathbf{X})$ is straightforward since this distribution is conditional on the variable selection indicators \mathbf{z} . We can therefore just select the subset of variables determined by \mathbf{z} and simulate from the posterior for a Gaussian linear regression as in Chapter [Linear Regression](#) using the matrix of covariates \mathbf{X}_z , where the subscript denotes that this matrix contains only the covariates selected by \mathbf{z} . In the example above with $\mathbf{z} = (1, 0, 1)$ we have that $\mathbf{X}_z = (\mathbf{x}_1, \mathbf{x}_3)$ is an $n \times 2$ matrix.

The marginal posterior for \mathbf{z} in (14.23) can be written using Bayes' theorem as

$$p(\mathbf{z} | \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} | \mathbf{z}, \mathbf{X}) p(\mathbf{z}), \quad (14.24)$$

where $p(\mathbf{z})$ is the prior $z_1, \dots, z_p \sim \text{Bernoulli}(\omega)$ with probability mass function

$$p(\mathbf{z}) = \omega^{\sum_{j=1}^p z_j} (1 - \omega)^{p - \sum_{j=1}^p z_j}. \quad (14.25)$$

It remains to derive the factor $p(\mathbf{y} | \mathbf{z}, \mathbf{X})$. This is the distribution for the response vector \mathbf{y} given the covariates selected by \mathbf{z} . Using our previous notation we can therefore write this as $p(\mathbf{y} | \mathbf{X}_z)$. But this is just the *marginal likelihood* for the Gaussian linear regression model with the covariates selected by \mathbf{z} in (14.12). The marginal likelihood is

prior inclusion probability

therefore

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}_z) &= \iint p(\mathbf{y}|\boldsymbol{\beta}_z, \sigma^2, \mathbf{X}_z) p(\boldsymbol{\beta}_z, \sigma^2 | \mathbf{X}_z) d\boldsymbol{\beta}_z d\sigma^2 \\ &= t_{v_0, z} \left(\mathbf{y} | \mathbf{0}, \sigma_{0,z}^2 (\mathbf{I}_n + \mathbf{X}_z \Omega_{0,z}^{-1} \mathbf{X}_z^\top) \right) \end{aligned} \quad (14.26)$$

where $\boldsymbol{\beta}_z$ is the vector of regression coefficients for the covariates selected by \mathbf{z} . The zero mean in the t -distribution is due to the prior mean μ_0 being zero in the slab component of the prior. We subscript v_0 , σ_0^2 and Ω_0 with \mathbf{z} since these quantities are here defined from the subset of covariates selected by \mathbf{z} .

We can now use Gibbs sampling to simulate from the joint posterior $p(\mathbf{z}|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{z}, \mathbf{X})p(\mathbf{z})$ by simulating each z_j conditional on the rest of the allocation variables $\mathbf{z}_{-j} = (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_p)$. Since each z_j is binary we only need to compute $p(\mathbf{z}|\mathbf{y}, \mathbf{X})$ for two different values of z_j , i.e. $z_j = 0$ and $z_j = 1$. For example assume that we are currently at $\mathbf{z} = (1, 0, 1)$ and want to simulate z_1 . We then compute first for the model that excludes the first covariate

$$\begin{aligned} p(\mathbf{z} = (0, 0, 1) | \mathbf{y}, \mathbf{X}) &\propto p(\mathbf{y}|\mathbf{z} = (0, 0, 1), \mathbf{X})p(\mathbf{z} = (0, 0, 1)) \\ &\propto p(\mathbf{y}|\mathbf{X}_{(0,0,1)}) \cdot (1 - \omega) \end{aligned} \quad (14.27)$$

and similarly for the model including the first covariate

$$p(\mathbf{z} = (1, 0, 1) | \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}_{(1,0,1)}) \cdot \omega \quad (14.28)$$

We then normalize these two probabilities to sum to one and draw z_1 from a Bernoulli distribution with these probabilities. We then repeat this for each z_j and iterate through all p covariates until we have sampled a \mathbf{z} from the joint posterior $p(\mathbf{z}|\mathbf{y}, \mathbf{X})$.

Given the sampled \mathbf{z} we can then simulate a draw from the joint posterior $p(\boldsymbol{\beta}, \sigma^2 | \mathbf{z}, \mathbf{y}, \mathbf{X})$ as described above. This concludes a single Gibbs sampling iteration sampling from the joint posterior $p(\boldsymbol{\beta}, \sigma^2, \mathbf{z} | \mathbf{y}, \mathbf{X})$. Repeating this process for many iterations gives us a sample from the joint posterior. Box 14.2 summarizes the Gibbs sampling algorithm. The algorithm in Box 14.2 computes the marginal likelihood $p(\mathbf{y}|\mathbf{X}_z)$ from scratch in every iteration even though we are actually only changing the inclusion/exclusion of a single covariate in each update. This is wasteful and a more numerically sound approach reuses and updates a previously computed marginal likelihood, see Smith and Kohn (1996).

Gibbs sampling for Bayesian variable selection in regression

Input: $n \times p$ matrix with p covariates as columns \mathbf{X}
vector \mathbf{y} with response observations
slab variance τ^2
prior inclusion probability ω
initial variable indicators $\mathbf{z}^{(0)} = (z_1^{(0)}, \dots, z_p^{(0)})$
number of posterior draws m .

```

for  $j$  in  $1:m$  do
    // Update regression parameters
    Draw  $(\sigma^2)^{(j)} | \mathbf{y}, \mathbf{X}_{\mathbf{z}^{(j-1)}} \sim \text{ScaledInv-}\chi^2(\nu_n, \sigma_n^2)$ 
    Draw  $\boldsymbol{\beta}_{\mathbf{z}^{(j-1)}}^{(j)} | (\sigma^2)^{(j)}, \mathbf{y}, \mathbf{X}_{\mathbf{z}^{(j-1)}} \sim N(\boldsymbol{\mu}_n, (\sigma^2)^{(j)} \Omega_n^{-1})$ 
    Set  $\boldsymbol{\beta}^{(j)}[\mathbf{z}^{(j-1)}] = \boldsymbol{\beta}_{\mathbf{z}^{(j-1)}}^{(j)}$  and  $\boldsymbol{\beta}^{(j)}[\text{Not}(\mathbf{z}^{(j-1)})] = 0$ 

    // Update mixture allocations
    Set  $\tilde{\mathbf{z}} = \mathbf{z}^{(j-1)}$ 
    for  $k$  in  $1:p$  do
        Set  $\tilde{\mathbf{z}}_0$  to  $\tilde{\mathbf{z}}$  but with  $k$ th element equal to 0
        Set  $\tilde{\mathbf{z}}_1$  to  $\tilde{\mathbf{z}}$  but with  $k$ th element equal to 1
        Compute  $\tilde{\omega}_{k,0} \propto (1 - \omega) \cdot p(\mathbf{y} | \mathbf{X}_{\tilde{\mathbf{z}}_0})$ 
        Compute  $\tilde{\omega}_{k,1} \propto \omega \cdot p(\mathbf{y} | \mathbf{X}_{\tilde{\mathbf{z}}_1})$ 
        Normalize  $\tilde{\omega}_{k,0}$  and  $\tilde{\omega}_{k,1}$  to sum to one
        Simulate allocation  $z_k^{(j)} \sim \text{Bernoulli}(\tilde{\omega}_{k,1})$ 
        Update  $\tilde{\mathbf{z}}$  with the new allocation  $z_k^{(j)}$ 
    end
    Set  $\mathbf{z}^{(j)} = \tilde{\mathbf{z}}$ 
end

```

Output: m autocorrelated draws from the joint posterior $p(\boldsymbol{\beta}, \sigma^2, \mathbf{z} | \mathbf{y}, \mathbf{X})$.

Box 14.2: Gibbs sampling algorithm for sampling Bayesian variable selection using the spike-and-slab prior.

15 Gaussian processes

Earlier chapters have presented several types of regression and classification models, starting from the basic linear regression for a continuous response variable with a normal distribution for the error term. The linear model was then extended to a binary response variable (logistic and probit regression), to a count data response (Poisson and negative binomial regression) or more generally a generalized linear model in the exponential family. All those models are still fundamentally linear models, driven by a linear combination of the covariates $\mathbf{x}^\top \boldsymbol{\beta}$, with the effect of having rather restrictive linear decision boundaries for classification. We have shown how the models can easily be extended with nonlinear effects by replacing the linear combination $\mathbf{x}^\top \boldsymbol{\beta}$ with a non-linear function of the covariates using polynomial and spline bases. The models remained linear in the parameters however, which is convenient for inference and prediction, but can be limiting in terms of flexibility.

In this chapter we will enter the realm of **nonparametric regression** where the relationship between the covariates and response variable will be given by unknown function whose functional form is left completely unspecified and learned from data. We are of course still doing Bayesian inference and will start to put priors over whole functions using the concept of a *Gaussian process* from the stochastic process literature. Such priors typically incorporate the notation of smoothness of the underlying regression functions and therefore have a regularizing effect.

nonparametric regression

15.1 Gaussian processes priors

Gaussian process regression is a general nonparametric approach where the regression function is not specified by a fixed set of parameters, but instead viewed as a general function $f(\mathbf{x})$ of the covariates \mathbf{x} . The function $f(\mathbf{x})$ is not restricted to be linear or even continuous, but we will use a Gaussian process prior to express the beliefs that $f(\mathbf{x})$ should have a certain degree of smoothness or not be excessively wiggly. Such prior beliefs will be seen to have a regularizing effect

Gaussian process regression

that can prevent overfitting.

Consider the following non-linear regression model, initially with a single covariate x

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2), \quad (15.1)$$

where the conditional mean $\mathbb{E}(y|x) = f(x)$ is a general function of x , not necessarily linear or polynomial. To see the non-parametric aspect of this model we can view it as having a separate parameter at every data point

$$y_i = f_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_\varepsilon^2), \quad (15.2)$$

where the conditional mean $f_i = f(x_i)$ is a *separate* parameter for every data point, see Figure 15.1 for an illustration. Note that even though the model actually has an abundance of parameters, one for each data point, we use the term *nonparametric* here to describe the absence of a restrictive parametric function for the conditional mean $f(x)$. The model in (15.1) is partly parametric, however, since the error term is additive and Gaussian.

The model in (15.2) is clearly a ridiculously flexible model that would lead to dramatic overfitting with a fitted function $f(x)$ passing through every data point to give a perfect fit. The real problem is that every data point has its own mean f_i , completely *independent of the f_j for other data points x_j* . One would think that two covariate values x_i and x_j that are very close should have rather similar conditional means for the response variable, i.e. if $x_i \approx x_j$ then $f(x_i)$ should be close to $f(x_j)$. We can solve this by imposing a parametric structure on the function $f(x)$, e.g. linear $f(x) = \beta_0 + \beta_1 x$, since the individual f_i would then be deterministically connected through the parameters β_0 and β_1 . However, the assumption of linearity or other simple parametric forms can sometimes fit the data poorly. Gaussian process regression takes an intermediate approach to this problem by allowing $f_i = f(x_i)$ to be a separate parameter for every data point but places a prior distribution on the function values $\mathbf{f}_{1:n} = (f_1, \dots, f_n)^\top$ with the prior information that $f(x)$ is likely to be similar for x values that are close to each other, i.e. it encodes the belief that the function $f(x)$ should be smooth.

Rather than putting a prior on the function values in the sample, $\mathbf{f}_{1:n} = (f_1, \dots, f_n)^\top$, a Gaussian process gives a prior on the whole function $f(x)$ for all x values in the domain of interest. This will allow us to generalize to new data, for example in prediction. We say that we specify a Gaussian process prior for the unknown function $f(x)$. A Gaussian process is a generalization of a multivariate normal distribution to an infinite number of variables, i.e. a distribution over functions instead of vectors. With a finite set of parameters θ , a draw

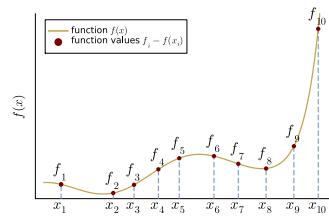
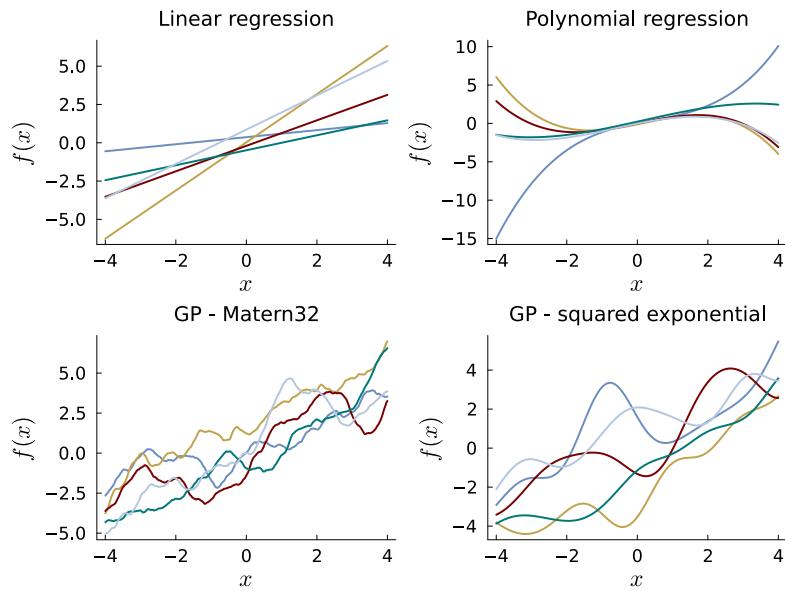


Figure 15.1: In nonparametric regression every data point has its own parameter $f_i = f(x_i)$.

from the prior or posterior distribution is a point in a p -dimensional space. With a Gaussian process, every draw is a *function*, i.e. a curve when x is a scalar, or a surface when x is a vector.

A Bayesian linear regression can also be viewed as prior on functions $f(x)$, but where every realization is restricted to be a linear function $f(x) = \mathbf{x}^\top \boldsymbol{\beta}$. This is because every prior draw of the vector with regression coefficients $\boldsymbol{\beta}^{(i)}$ implies a draw of the regression function $f^{(i)}(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}^{(i)}$; see the top left graph in Figure 15.2. Similarly, a polynomial regression is also a prior over functions, but where every realization has to be a polynomial function of a certain order (top right graph in Figure 15.2). A Gaussian process takes this a step further and allows the function to arbitrary, but instead imposes restrictions more implicitly, typically in some form of smoothness of the function realizations. This is illustrated in the bottom part of Figure 15.2, where five realizations from two different Gaussian process priors are plotted; the two Gaussian processes prior are defined below. Note how the realizations from the Gaussian process to the left in the figure are less smooth than the realization from the Gaussian process to the right.



Going back to the case with a separate parameter $f_i = f(x_i)$ for each data point (x_i, y_i) we can try to use a multivariate normal distribution as a joint prior for the vector of function values:

$$\mathbf{f}_{1:n} = (f_1, \dots, f_n)^\top \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Omega}_0), \quad (15.3)$$

where the prior covariance matrix $\boldsymbol{\Omega}_0$ can be specified to have large correlation between any pair of function values $f(x_i)$ and $f(x_j)$

Figure 15.2: Illustrating the concept of a prior over functions by plotting five realizations from different priors.

A linear regression with prior $(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1) \sim N((1, 1), 0.5 \cdot I_2)$, a polynomial regression of degree 3 with prior $\boldsymbol{\beta} \sim N((1, 1, 0, 0), 0.1 \cdot I_4)$, a Gaussian process prior with a Matern32 kernel and a Gaussian process prior with a squared exponential kernel. The mean function in both Gaussian process priors is the linear identity function $\mu_0(x) = x$.

whenever the covariate values x_i and x_j are close. It would clearly be tedious to manually specify the covariance between all pairs of elements in $f_{1:n}$. To aid in this we use a **covariance function** $k_0(x, x')$ that specifies the covariance between $f(x)$ and $f(x')$ for any pair of covariate values x and x' ; note that x' is just another x -value, and not the vector transpose. As usual we use the subscript 0 on $k_0(x, x')$ to denote that the covariance function it is part of the prior, based on zero observations. The covariance function $k_0(x, x')$ is often called the **covariance kernel**. The prior covariance matrix Ω_0 in (15.3) is then constructed by evaluating the covariance function $k_0(x, x')$ for all pairs of covariate values x and x' in the data set

$$\Omega_0 = \begin{pmatrix} k_0(x_1, x_1) & \cdots & k_0(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k_0(x_n, x_1) & \cdots & k_0(x_n, x_n) \end{pmatrix} \quad (15.4)$$

Similarly, we can specify the mean vector μ_0 in (15.3) by evaluating a mean function $\mu_0(x)$ at all covariate values x_1, \dots, x_n .

Perhaps the most widely used covariance function is the **Gaussian kernel**

$$\mathbb{C}(f(x), f(x')) = k_0(x, x') = \sigma_f^2 \cdot \exp\left(-\frac{(x - x')^2}{2\ell^2}\right), \quad (15.5)$$

where $\mathbb{C}(\cdot, \cdot)$ is the covariance, $\ell > 0$ is a **length scale** parameter that controls the wigglyness of the function $f(x)$. This kernel is usually referred to as the **squared exponential kernel** in the machine learning literature. Note how the squared exponential kernel gives a high covariance between $f(x)$ and $f(x')$ if the covariate values x and x' are close to each other; see Figure 15.3. The scale parameter $\sigma_f > 0$ controls the prior standard deviation of the function values $f(x)$, i.e. how much the function is allowed to vary around its mean function $\mu_0(x)$. Figure 15.4 plots realizations from a Gaussian process with the squared exponential kernel for four different settings of the kernel hyperparameters ℓ and σ_f . Note how ℓ affects the wigglyness of the realizations and σ_f the variation around the mean function $\mu_0(x) = x$.

The above construction of a multivariate normal prior for the values of an unknown $f(x)$ at a finite set of covariate points x_1, \dots, x_n makes the way for the following formal definition of a Gaussian process.

Definition. A **Gaussian process** (GP) is a collection of random variables where the joint distribution of any finite number of the random variables is multivariate Gaussian.

The above definition of a Gaussian process can be shown to give a properly defined probability distribution over the somewhat abstract

covariance function

covariance kernel

Gaussian kernel

length scale

squared exponential kernel

Gaussian process

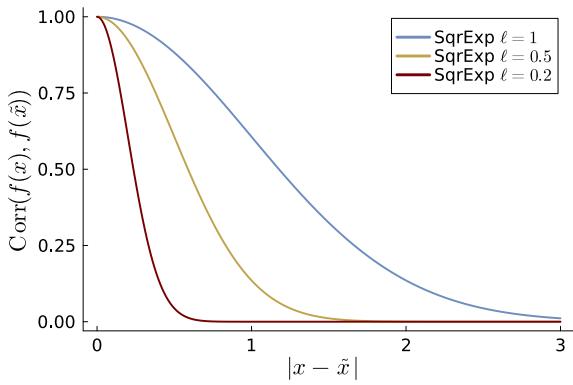


Figure 15.3: Illustrating how the correlation between function values $f(x)$ and $f(x')$ decays with the distance $|x - x'|$ for different length scales ℓ in the squared exponential kernel.

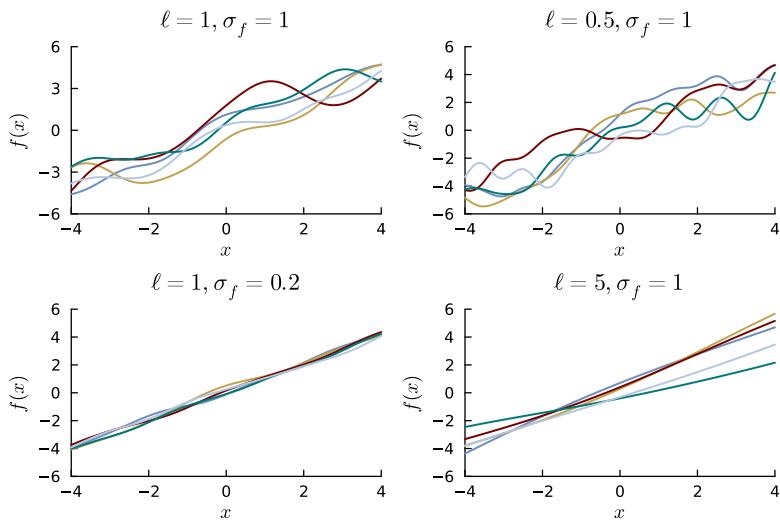


Figure 15.4: Realizations from a Gaussian process with the squared exponential kernel for four different settings of the kernel hyperparameters ℓ and σ_f . The mean function is $\mu_0(x) = x$ in all four plots.

space of *functions*, in such a way that every finite selection, or sampling, of function values $f(x_1), \dots, f(x_n)$ has a multivariate normal distribution. This is very convenient: we have an abstract definition of a probability distribution over functions, but when we sample a finite set of function values from $f(\mathbf{x})$, those values follow a multivariate normal distribution.

We can now see how to generate a random function $f(x)$ from a Gaussian process prior:

1. choose a fine grid of x -values, $\mathbf{x}_{\text{grid}} = (x_1, \dots, x_n)^\top$.
2. evaluate the mean function $\mu_0(x)$ at all $x \in \mathbf{x}_{\text{grid}}$ to get the n -dimensional mean vector $\boldsymbol{\mu}_0$.
3. compute the covariance kernel $k_0(x, x')$ at all pairs of $x \in \mathbf{x}_{\text{grid}}$ and construct the $n \times n$ covariance matrix $\boldsymbol{\Omega}_0$. Add a small number to the diagonal of $\boldsymbol{\Omega}_0$ to ensure that it is positive definite.
4. draw a vector of n function values $f(x)$ for all $x \in \mathbf{x}_{\text{grid}}$ from the multivariate normal distribution $(f_1, \dots, f_n)^\top \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Omega}_0)$.
5. plot a connected line through the function values $(f_1, \dots, f_n)^\top$ at the grid points.

The addition of a small number, sometimes called a **nugget**, in Step 3 is particularly important when a very fine grid is used; any two neighboring points on the grid will then be so close to each other that the correlation between their function values will nearly one, making $\boldsymbol{\Omega}_0$ near singular or singular. The above five-step procedure was used to generate the five draws in the two bottom graphs in Figure 15.2.

nugget

The same procedure can be used to generate a sample of a function $f(\mathbf{x})$ with multi-dimensional input \mathbf{x} from a Gaussian process prior. Here the realizations will be surfaces or hypersurfaces over the multi-dimensional space for the vector of covariates \mathbf{x} . The *function values* $f(\mathbf{x})$ are still scalars even though the input \mathbf{x} is multi-dimensional, so $\boldsymbol{\mu}_0$ is still a n -dimensional vector and $\boldsymbol{\Omega}_0$ is still an $n \times n$ covariance matrix. We need a multi-dimensional kernel function $k_0(\mathbf{x}, \mathbf{x}')$ that takes two p -dimensional vectors \mathbf{x} and \mathbf{x}' and returns the covariance between their respective function values $f(\mathbf{x})$ and $f(\mathbf{x}')$. The squared exponential kernel in (15.5) immediately generalizes to the case with multiple covariates $f(\mathbf{x})$ by replacing the scalar distance $x - x'$ with the Euclidean distance $\|\mathbf{x} - \mathbf{x}'\| = (\sum_{k=1}^p (x_k - x'_k)^2)^{1/2}$ between the vectors \mathbf{x} and \mathbf{x}' :

$$\mathbb{C}(f(\mathbf{x}), f(\mathbf{x}')) = k_0(\mathbf{x}, \mathbf{x}') = \sigma_f \cdot \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right). \quad (15.6)$$

Just as we write $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote that the random vector \mathbf{x} follows a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, we write

$$f(\mathbf{x}) \sim \text{GP}(\mu_0(\mathbf{x}), k_0(\mathbf{x}, \mathbf{x}')) \quad (15.7)$$

to denote that the random function $f(\mathbf{x})$ follows a Gaussian process with mean function $\mu_0(\mathbf{x})$ and covariance kernel $k_0(\mathbf{x}, \mathbf{x}')$:

$$\begin{aligned} \mu_0(\mathbf{x}) &= \mathbb{E}(f(\mathbf{x})) \\ k_0(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left((f(\mathbf{x}) - \mu_0(\mathbf{x}))(f(\mathbf{x}') - \mu_0(\mathbf{x}'))\right)^\top \end{aligned} \quad (15.8)$$

To make it clear that a Gaussian process is a probability distribution for a function we often write $f(\cdot)$ instead of $f(\mathbf{x})$ since the input \mathbf{x} is just a placeholder for any covariate vector. Hence a Gaussian process is often written as

$$f(\cdot) \sim \text{GP}(\mu_0(\cdot), k_0(\cdot, \cdot)), \quad (15.9)$$

since it is completely determined by the mean function and covariance kernel.

There are many other kernel functions than the squared exponential, see [Williams and Rasmussen \(2006\)](#) for a collection. Any function $k_0(\mathbf{x}, \mathbf{x}')$ can be used as covariance kernel provided that the covariance matrix generated from it is positive definite (see [Appendix A.2](#)). One of the most popular covariance kernel is the Matérn kernel

$$k_0(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{(2\nu)r}}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{(2\nu)r}}{\ell} \right), \quad (15.10)$$

where $K_\nu(\cdot)$ is the so called modified Bessel function of the second kind; see [Appendix A.1](#) for the definition. This function is available in all major numerical programming languages. Compared to the squared exponential kernel, the Matérn kernel has an additional hyperparameter, the so called degrees of freedom or smoothness parameter $\nu > 0$. The correlation function of some different Matérn kernels are plotted in [Figure 15.5](#).

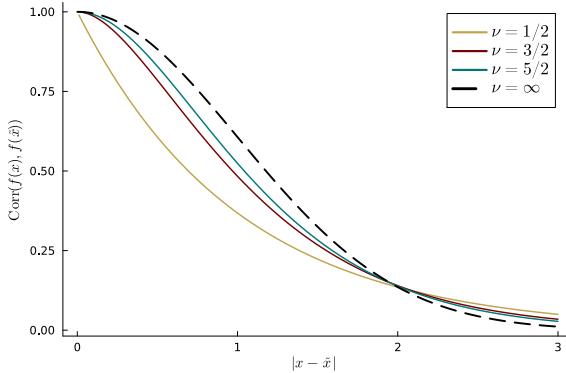


Figure 15.5: Illustrating how the correlation between function values $f(x)$ and $f(x')$ decays with the distance $|x - x'|$ for different Matérn kernels with length scale $\ell = 1$. The Matérn kernel with $\nu = \infty$ is the squared exponential kernel.

The degrees of freedom parameter is related to the differentiability of the Gaussian process. Since we are dealing with stochastic processes here rather than deterministic functions, we need an different definition of differentiability than the usual one from calculus. It is common to define differentiability in terms of **mean square convergence** of random variables (see Box 15.1-15.3). A Gaussian process with Matérn covariance function is k -times mean square (MS) differentiable if and only if $\nu > k$; hence lower values of ν gives realizations that are less smooth in the sense of having fewer MS continuous derivatives. The values $\nu = 1/2$, $\nu = 3/2$ and $\nu = 5/2$ are particularly used in practice, and it is common to refer to them as Matern₁₂, Matern₃₂ and Matern₅₂, respectively. Note that Matern₁₂ is MS continuous but has no MS derivatives, Matern₃₂ has one MS derivative and Matern₅₂ has two MS derivatives. These different smoothnesses are clearly visible in Figure 15.6 which shows three realizations from Matern₁₂, Matern₃₂ and Matern₅₂ and the squared exponential ($\nu = \infty$). It should be noted however that the differentiability of the *sample functions* requires some additional conditions in addition to those for MS differentiability, see Lindgren (2012) for details.

15.2 Gaussian process regression

Assume now that we have a regression dataset $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$ that we want to model with the following nonparametric regression model with Gaussian errors

$$y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2). \quad (15.11)$$

We use a Gaussian process prior for the unknown conditional mean function $f(\mathbf{x})$ with a certain prior mean function $\mu(\mathbf{x})$ and prior covariance kernel $k_0(\mathbf{x}, \mathbf{x}')$. Our aim is now to derive the posterior distribution for the function $f(\mathbf{x})$ conditional on the observed data. We

Mean-square convergence

A sequence of random variables X_1, \dots, X_n converges in mean-square to a random variable X if

$$\mathbb{E}(|X_n - X|^2) \rightarrow 0 \text{ as } n \rightarrow \infty.$$

We write $X_n \xrightarrow{ms} X$.

Box 15.1: Mean-square convergence of random variables.

Mean-square continuous stochastic process

A stochastic process $f(x)$ is mean-square continuous at x if

$$f(x+h) \xrightarrow{ms} f(x) \text{ as } h \rightarrow 0.$$

Box 15.2: Mean-square continuous stochastic process.

Mean-square differentiable stochastic process

A stochastic process $f(x)$ is mean-square differentiable at x with mean-square derivative $f'(x)$ if

$$\frac{f(x+h) - f(x)}{h} \xrightarrow{ms} f'(x),$$

as $h \rightarrow 0$.

Box 15.3: Mean-square differentiable stochastic process.

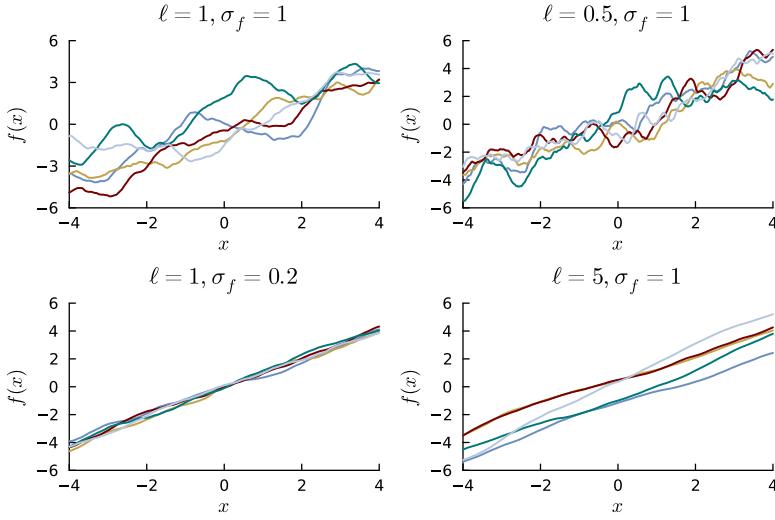


Figure 15.6: Realizations from a Gaussian process with the Matern₃₂ ($\nu = 3/2$) kernel for four different settings of the kernel hyperparameters ℓ and σ_f . The mean function is $\mu_0(x) = x$ in all four plots.

condition on noise standard deviation σ_ϵ and treat the case with unknown σ_ϵ and potentially unknown kernel hyperparameters in Section 15.3.

We are interested in the posterior for the function $f(\mathbf{x})$ over a set of \tilde{n} covariate vectors $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{\tilde{n}}$. That is, our parameter of interest is the vector $\tilde{\mathbf{f}} = (f(\tilde{\mathbf{x}}_1), \dots, f(\tilde{\mathbf{x}}_{\tilde{n}}))^\top$. The evaluation points $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{\tilde{n}}$ can be a so called *test set* for which we want predictions, or they can be a grid of values over which we want to plot the posterior distribution of $f(\mathbf{x})$. For notational convenience, we place the \tilde{n} covariate test observation vectors as rows in the $\tilde{n} \times p$ matrix $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{\tilde{n}})^\top$ and the test responses are denoted by $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_{\tilde{n}})^\top$.

Our aim is to obtain the joint distribution of $\tilde{\mathbf{f}}$ conditional on the observed *training data* $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$, or simply $\mathbf{y} = (y_1, \dots, y_n)^\top$, if we follow the usual convention in regression and consider the covariate data as fixed. The usual route to derive the posterior distribution using Bayes' theorem

$$p(\tilde{\mathbf{f}}|\mathbf{y}) \propto p(\mathbf{y}|\tilde{\mathbf{f}})p(\tilde{\mathbf{f}}),$$

is less straightforward for Gaussian process regression. The reason is that the likelihood factor $p(\mathbf{y}|\tilde{\mathbf{f}})$ is not easy to write down since the test set $\tilde{\mathbf{X}}$ is usually different from the training set \mathbf{X} . That is, while $p(\mathbf{y}|\mathbf{f}) = N(\mathbf{y}|\mathbf{f}, \sigma_\epsilon^2 \mathbf{I}_n)$ is simple, $p(\mathbf{y}|\tilde{\mathbf{f}}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\tilde{\mathbf{f}})d\mathbf{f}$ is a little more involved. We will instead follow the presentation in Williams and Rasmussen (2006) and use the conditioning properties of the multivariate normal distribution to derive the posterior distribution for the function $f(\mathbf{x})$.

To derive the conditional distribution $\tilde{\mathbf{f}}|\mathbf{y}$ we first express the joint distribution $p(\tilde{\mathbf{f}}, \mathbf{y})$, and then obtain the sought posterior distribution

$p(\tilde{\mathbf{f}}|\mathbf{y})$ by conditioning on \mathbf{y} , using the convenient conditioning properties of a multivariate normal distribution. The joint distribution is of the form

$$\begin{pmatrix} \mathbf{y} \\ \tilde{\mathbf{f}} \end{pmatrix} = N\left(\begin{pmatrix} \boldsymbol{\mu}_0 \\ \tilde{\boldsymbol{\mu}}_0 \end{pmatrix}, \begin{pmatrix} \Omega_{ff} + \sigma_\epsilon^2 \mathbf{I}_n & \Omega_{ff}\tilde{\mathbf{f}} \\ \Omega_{\tilde{f}f} & \Omega_{\tilde{f}\tilde{f}} \end{pmatrix} \right), \quad (15.12)$$

where $\Omega_{ff} = \mathbb{V}(\mathbf{f})$ is the $n \times n$ covariance matrix for the function values in the training data, $\Omega_{\tilde{f}\tilde{f}} = \mathbb{V}(\tilde{\mathbf{f}})$ is the $\tilde{n} \times \tilde{n}$ covariance matrix for function values in the test data and $\Omega_{ff} = \mathbb{C}(\mathbf{f}, \tilde{\mathbf{f}})$ is the $n \times \tilde{n}$ matrix with covariances between pairs of elements in the training set $f(\mathbf{x})$ and the test set $f(\tilde{\mathbf{x}})$. The subvectors and submatrices in (15.12) can be derived by considering the model in vector form

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}_n) \quad (15.13)$$

from which is clear that

$$\mathbb{E}(\mathbf{y}) = \mathbb{E}(\mathbf{f}) + \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{\mu}_0 \quad (15.14)$$

and, because of the independence between $\boldsymbol{\epsilon}$ and \mathbf{f} ,

$$\mathbb{V}(\mathbf{y}) = \mathbb{V}(\mathbf{f}) + \mathbb{V}(\boldsymbol{\epsilon}) = \Omega_{ff} + \sigma_\epsilon^2 \mathbf{I}_n$$

Similarly we have

$$\mathbb{C}(\mathbf{y}, \tilde{\mathbf{f}}) = \mathbb{E}(\mathbf{y} - \boldsymbol{\mu}_0)(\tilde{\mathbf{f}} - \tilde{\boldsymbol{\mu}}_0)^\top = \mathbb{E}(\mathbf{f} + \boldsymbol{\epsilon} - \boldsymbol{\mu}_0)(\tilde{\mathbf{f}} - \tilde{\boldsymbol{\mu}}_0)^\top = \Omega_{f\tilde{f}},$$

since $\boldsymbol{\epsilon}$ is independent of $\tilde{\mathbf{f}}$.

Now, recall from Figure ?? that for a joint normal distribution $\mathbf{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with the partitioning into subvectors,

$$\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix},$$

we know that

$$\mathbf{z}_2 | \mathbf{z}_1 \sim N(\tilde{\boldsymbol{\mu}}_2, \tilde{\boldsymbol{\Sigma}}_2)$$

where

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_2 &= \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{z}_1 - \boldsymbol{\mu}_1) \\ \tilde{\boldsymbol{\Sigma}}_2 &= \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}. \end{aligned}$$

Applying this conditioning on the joint normal in (15.12) we obtain the posterior distribution result in Box 15.4.

Recall the definition of a Gaussian process: a collection of random variables where every finite subset is multivariate normal. Hence, since the posterior for $\tilde{\mathbf{f}}|\mathbf{y}$ according to Box 15.4 is multivariate normal for *any* choice of test points $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{\tilde{n}}$, the posterior distribution

Posterior updating Gaussian process regression

Model

$$y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2),$$

Gaussian process prior

$$f \sim \text{GP}(\mu(\cdot), k_0(\cdot, \cdot))$$

Posterior at \tilde{n} test points, $\tilde{\mathbf{f}} = (f(\tilde{\mathbf{x}}_1), \dots, f(\tilde{\mathbf{x}}_{\tilde{n}}))^\top$

$$\tilde{\mathbf{f}} | \mathbf{y} \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Omega}_n)$$

where

$$\boldsymbol{\mu}_n = \tilde{\boldsymbol{\mu}}_0 + \boldsymbol{\Omega}_{\tilde{f}f} (\boldsymbol{\Omega}_{ff} + \sigma^2 I_n)^{-1} (\mathbf{y} - \boldsymbol{\mu}_0) \quad (15.15)$$

$$\boldsymbol{\Omega}_n = \boldsymbol{\Omega}_{\tilde{f}\tilde{f}} - \boldsymbol{\Omega}_{\tilde{f}f} (\boldsymbol{\Omega}_{ff} + \sigma^2 I_n)^{-1} \boldsymbol{\Omega}_{f\tilde{f}}. \quad (15.16)$$

and

$$\boldsymbol{\Omega}_{ff} = \mathbb{V}(\mathbf{f})$$

$$\boldsymbol{\Omega}_{\tilde{f}\tilde{f}} = \mathbb{V}(\tilde{\mathbf{f}})$$

$$\boldsymbol{\Omega}_{f\tilde{f}} = \mathbb{C}(\mathbf{f}, \tilde{\mathbf{f}}).$$

Box 15.4: Posterior for the Gaussian process regression.

Conjugate prior for GP regression with Gaussian noise

In the Gaussian process regression with known σ_ε^2

$$y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2),$$

a conjugate prior for the function $f(\cdot)$ is the Gaussian process

$$f \sim \text{GP}(\mu_0(\cdot), k_0(\cdot, \cdot)) \stackrel{\text{data}}{\Rightarrow} f | \mathbf{y} \sim \text{GP}(\mu_n(\cdot), k_n(\cdot, \cdot))$$

where $\mu_n(\cdot)$ is the posterior mean function corresponding to μ_n in (15.15) and $k_n(\cdot, \cdot)$ is the posterior covariance kernel corresponding to Ω_n in (15.16).

Box 15.5: The Gaussian process prior is the conjugate prior for Gaussian process regression.

for the function $f(\mathbf{x})$ is a Gaussian process. This means that the Gaussian process prior is the conjugate prior for the Gaussian process regression model: a Gaussian process prior gives a Gaussian process posterior. We summarize this observation in Box 15.5.

The result in Box 15.5 should not come as a big surprise. We have already seen in many places that combining a normal prior with a normal data likelihood with a known variance gives a normal posterior, both in the univariate (Section 2.2) and the multivariate case (Section 3.6), and in the linear Gaussian regression model in Chapter 5.

A SIMULATED EXAMPLE. Let us fit a Gaussian process regression model to a simulated dataset where we know the true mean function

$$y = f(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2), \quad (15.17)$$

where $f(x) = \sin(3x)$, $x \sim \text{Uniform}(-2, 2)$ and $\sigma_\varepsilon = 0.2$. We use a zero prior mean function $\mu_0(x) = 0$ and a squared exponential kernel with hyperparameters σ_f and ℓ . We will for simplicity analyze the data conditional on the true $\sigma_\varepsilon = 0.2$. Figure 15.7 shows fit of the Gaussian process regression model to $n = 20$ data points from the model (15.17), each graph using different kernel hyperparameters σ_f and ℓ as indicated by each graph title. Each plot shows the true mean function $f(x)$ (blue line), the posterior mean of $f(x)$ (red line) and the 95% pointwise credible interval for the function $f(x)$ (gray shaded area). Note in particular in the lower left plot how a small length scale ℓ gives a more wiggly fit and large credibility intervals for x values in regions without any observed data. This is because a small ℓ implies that the correlation between function values decreases rapidly with the distance in x -space, so points in data sparse regions have very little help from even the nearest observed data points. The posterior for $f(x)$ at a point x in a data sparse regions is therefore essentially just the prior. The lower right graph shows that a small value for σ_f gives more shrinkage toward the zero mean function.

The 95% credibility intervals in Figure 15.7 are for the conditional mean $f(x)$, and are therefore not meant to capture the individual data points. Each individual response observation \tilde{y}_i in the test set is disturbed by some noise $\tilde{\varepsilon}_i \sim N(0, \sigma_\varepsilon^2)$, and we need to compute a 95% posterior *predictive interval* to properly capture individual observations:

$$\mu_n(\tilde{x}) \pm 1.96 \sqrt{k_n(\tilde{x}, \tilde{x}) + \sigma_\varepsilon^2},$$

where $\mu_n(\tilde{x}) = \mathbb{E}(f(\tilde{x})|\mathbf{y})$ and $k_n(\tilde{x}, \tilde{x}) = \mathbb{V}(f(\tilde{x})|\mathbf{y})$ are the posterior mean and variance of $f(\tilde{x})$, respectively. Figure 15.8 shows the posterior predictive bands (dashed lines) for the simulated sine wave data.

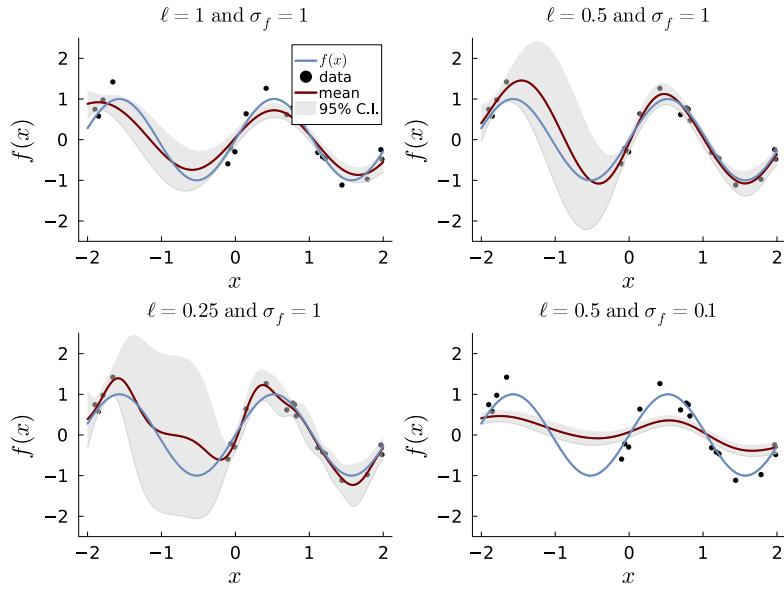


Figure 15.7: Fitting a Gaussian process regression to the simulated data from the sine wave model in (15.17). The blue line is the true mean function $f(x)$, the red line is the posterior mean of $f(x)$ and the shaded area are the 95% pointwise credible intervals for the function $f(x)$.

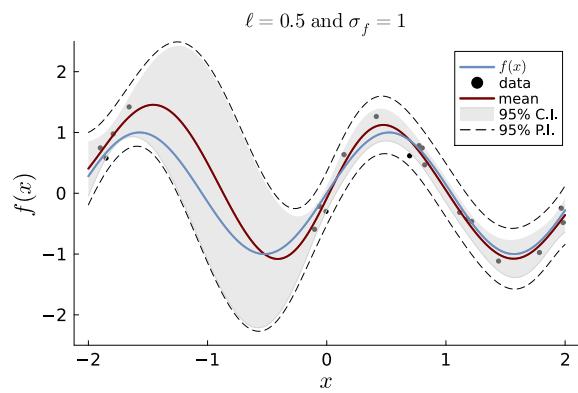


Figure 15.8: Prediction bands from Gaussian process regression fitted to the simulated data from the sine wave model in (15.17). The dashed lines area are the 95% pointwise prediction intervals for a new data point \tilde{y} .

GAUSSIAN PROCESS REGRESSION FOR LIDAR DATA.

Lidar (light detection and ranging) is a remote sensing method that uses light in the form of a pulsed laser to measure distances to objects. One of the early uses was to monitor pollution by detecting chemical compounds in the atmosphere. The **Lidar dataset** from Holst et al. (1996) plotted in Figure 15.9 contains 221 Lidar measurements with `logratio` as the logarithm of the ratio of the received light at the resonance frequency of the compound against the received light on a frequency off this target resonance frequency. The explanatory variable `distance` is the distance travelled before the light is reflected back to the sensor (normalized to the interval [0, 1] here). The relationship between `logratio` and `distance` is clearly highly nonlinear. Ruppert et al. (2003) show that the data can not be transformed to a linear relationship by the usual transformations applied in applied regression modeling.

Lidar dataset

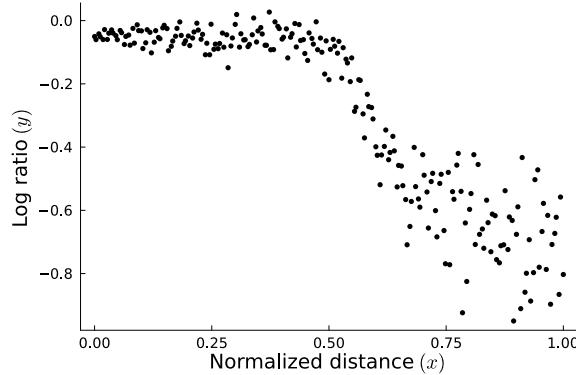


Figure 15.9: The Lidar data with 221 observation on the log ratio contrast of received light at the resonance frequency of the target compound plotted against the distance travelled before the light is reflected back to the sensor (normalized to the interval [0, 1]).

We will initially ignore the clear heteroscedasticity in the data and model it with a homoscedastic Gaussian process regression model

$$\text{logratio} = f(\text{distance}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2),$$

with a zero mean GP prior

$$f \sim \text{GP}(0, k_0(\cdot, \cdot)).$$

We will experiment with both the squared exponential and Matern32 kernels, and assess the effect of the two kernel hyperparameters σ_f and ℓ . The error variance is fixed to $\sigma_\varepsilon = 0.05$ throughout the analysis, and we return to this choice in Section 15.3 where all hyperparameters are inferred.

Figure 15.10 shows the posterior distribution for $f(x)$ for the Lidar data using the squared exponential kernel with different hyperparameters. The posterior for $f(x)$ is represented by the posterior mean function ($\mu_n(x)$ plotted as the red line) and the 95% pointwise credible intervals (light gray shaded regions). The 95% predictive intervals

is given as dashed lines. The conditional mean in the data is modeled quite nicely with the two smaller length scales in the top row of Figure 15.10. A length scale of $\ell = 1$ used in the lower left plot shows clear signs of underfitting with a too smooth function that is unable to fully capture the nonlinearity in the data. As expected, predictive intervals from the homoscedastic Gaussian process model is not able to capture the heteroscedasticity in the data.

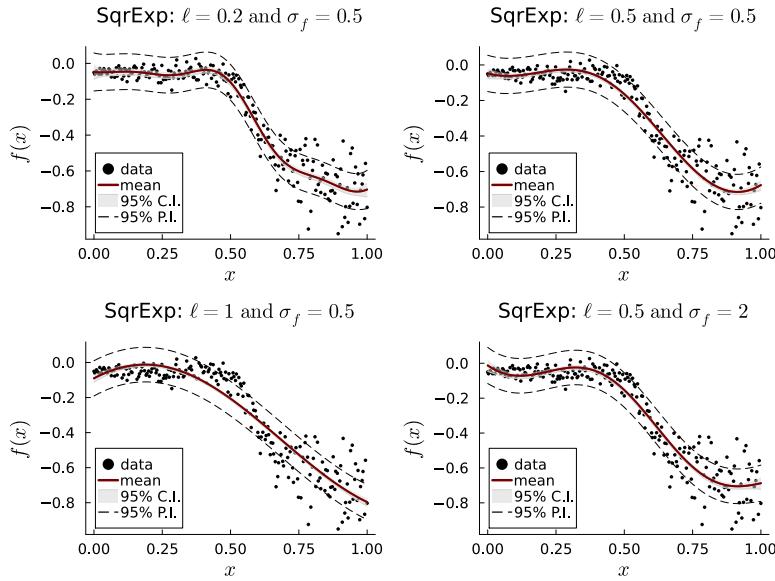


Figure 15.10: Posterior distribution for $f(x)$ (gray shaded regions) and predictive intervals (dashed lines) for Lidar data using the squared exponential kernel with different hyperparameters.

Figure 15.11 shows the posterior distribution for $f(x)$ in the Lidar data using the Matern32 ($\nu = 3/2$) kernel with the same choice for the kernel hyperparameters as for the squared exponential in Figure 15.10. The Matern32 kernel gives in general a less smooth fit than the squared exponential kernel, but the fit with the longer length scale $\ell = 1$ in the lower left graph seems to strike a good balance.

Box 15.5 assumes that the error variance σ_ε^2 is known. The extension to unknown σ_ε^2 can be handled exactly like in the linear Gaussian regression model by using a $\sigma_\varepsilon^2 \sim \text{Inv-}\chi^2$ prior. The posterior would then be given by the conditional-marginal decomposition

$$p(\tilde{\mathbf{f}}, \sigma_\varepsilon^2 | \mathbf{y}) = p(\tilde{\mathbf{f}} | \sigma_\varepsilon^2, \mathbf{y}) p(\sigma_\varepsilon^2 | \mathbf{y}), \quad (15.18)$$

where $\tilde{\mathbf{f}} | \sigma_\varepsilon^2, \mathbf{y}$ is given by Box 15.4 and $\sigma_\varepsilon^2 | \mathbf{y}$ can be shown to be a $\text{Inv-}\chi^2$ distribution. However, it is more common, particularly in the machine learning literature, to analyze the noise variance σ_ε^2 as part of the other prior hyperparameters in the mean function and covariance kernel, for example the scale parameter σ_f^2 and the length scale parameter ℓ in the squared exponential kernel; see the next section.

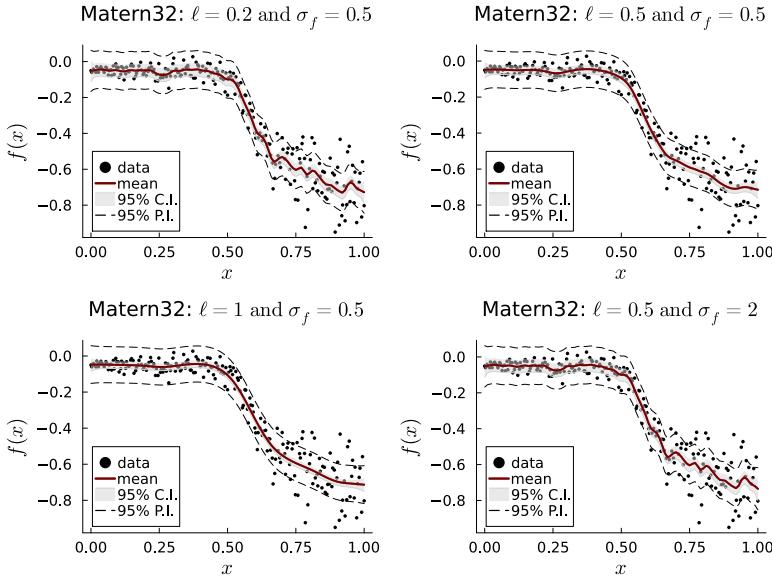


Figure 15.11: Posterior distribution for $f(x)$ (gray shaded regions) and predictive intervals (dashed lines) for Lidar data using the Matern32 kernel with different hyperparameters.

15.3 Learning the kernel hyperparameters

The choice of kernel hyperparameters $\theta = (\sigma_f, \ell)^\top$ is clearly important for the fit of the Gaussian process regression model. A too small length scale ℓ gives a very wiggly fit that overfits the data, while a large length scale gives a very smooth fit that runs the risk of underfitting. The choice of hyperparameters is a subjective choice based on the user's prior beliefs about the underlying function $f(\mathbf{x})$. If the user feels unable to specify the prior hyperparameters θ she can use a hierarchical prior

$$\begin{aligned} f|\theta &\sim \text{GP}(0, k_{0,\theta}(\cdot, \cdot)) \\ \theta &\sim p(\theta), \end{aligned}$$

where we use the subscript θ on the covariance kernel $k_{0,\theta}(\cdot, \cdot)$ to explicitly show that the kernel depends on the hyperparameters θ . The prior mean function can different from zero and its parameters can be estimated along with the kernel hyperparameters.

The joint posterior distribution for the function $f(\mathbf{x})$ at the test points $\tilde{\mathbf{f}}$ and the hyperparameters θ is given by

$$p(\tilde{\mathbf{f}}, \theta | \mathbf{y}) \propto p(\tilde{\mathbf{f}} | \theta, \mathbf{y}) p(\theta | \mathbf{y}) \quad (15.19)$$

where $p(\tilde{\mathbf{f}} | \theta, \mathbf{y})$ is the Gaussian process posterior given in Box 15.4 and $p(\theta | \mathbf{y})$ is the marginal posterior distribution for the hyperparameters. By Bayes' theorem we have

$$p(\theta | \mathbf{y}) \propto p(\mathbf{y} | \theta) p(\theta), \quad (15.20)$$

where $p(\mathbf{y}|\boldsymbol{\theta})$ is the marginal likelihood of the data given the hyperparameters:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})d\mathbf{f}. \quad (15.21)$$

Since the Gaussian process regression model is $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$ where $\mathbf{f} \sim N(\boldsymbol{\mu}_0, \Omega_{ff})$ and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma_\varepsilon^2 \mathbf{I}_n)$ are independent, we immediately have that the marginal likelihood is

$$\mathbf{y}|\boldsymbol{\theta} \sim N(\boldsymbol{\mu}_0, \Omega_{ff} + \sigma_\varepsilon^2 \mathbf{I}_n). \quad (15.22)$$

Note that the marginal likelihood is a (complicated) function of the hyperparameters $\boldsymbol{\theta}$ through the covariance matrix Ω_{ff} . While we can easily *evaluate* the probability density in (15.22) for a given set of hyperparameters $\boldsymbol{\theta}$, the posterior distribution for $\boldsymbol{\theta}$ in (15.20) based on this marginal likelihood is usually not a known distribution for any prior $p(\boldsymbol{\theta})$; we have to resort to MCMC, normal or variational approximations to approximate the hyperparameter posterior $p(\boldsymbol{\theta}|\mathbf{y})$.

An alternative approach is to estimate the hyperparameters by numerically maximizing the marginal likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ with respect to the $\boldsymbol{\theta}$. This is sometimes referred to as **empirical Bayes** (EB). Once such a point estimate of the hyperparameters $\hat{\boldsymbol{\theta}}_{\text{EB}}$ is obtained, the posterior distribution for the function $f(\mathbf{x})$ can be computed by plugging in $\hat{\boldsymbol{\theta}}_{\text{EB}}$ in the posterior distribution in Box 15.4, $p(\tilde{\mathbf{f}}|\hat{\boldsymbol{\theta}}_{\text{EB}}, \mathbf{y})$. Note that this is not a fully Bayesian procedure since it ignores the uncertainty in the hyperparameters, but it is usually computationally much faster than MCMC or variational approximations. Figure 15.12 shows the level contours of the log marginal likelihood for the Lidar data with the Matern32 kernel conditional on the empirical Bayes estimate of the noise standard deviation $\hat{\sigma}_\varepsilon = 0.079$. The empirical Bayes estimate of $\ell = 0.661$ and $\sigma_f = 0.441$ is marked out with an orange dot. Note that the log marginal likelihood is quite flat around the maximum, indicating that the uncertainty in the hyperparameters is quite large. Figure 15.13 plots the posterior distribution for $f(x)$ and the predictive bands using the Matern32 kernel with hyperparameters estimated by maximizing the marginal likelihood.

empirical Bayes

Figure 15.14 shows the marginal posterior distribution for all three hyperparameters in the Lidar data obtained from 5000 Hamiltonian Monte Carlo draws using the NUTS sampler (histograms) and normal approximation (solid lines). The normal approximation was obtained for log-transformed hyperparameters, so the densities in the figure are log-normal distributions. The prior on the hyperparameters was for simplicity chosen to be the non-informative and improper uniform distribution on the log scale. Note how the normal approximation does a decent job in approximating the posterior, but misses a bit of the heavy right tail in the marginal posterior distribution for σ_f and ℓ .

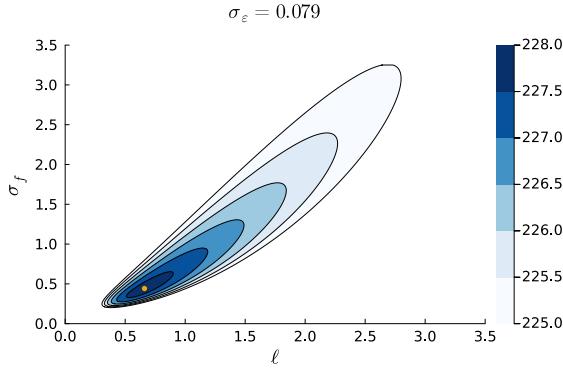


Figure 15.12: Level contours of the log marginal likelihood for the Lidar data conditional on the empirical Bayes estimate of the noise standard deviation $\hat{\sigma}_\varepsilon = 0.079$. The empirical Bayes estimate of $\ell = 0.661$ and $\sigma_f = 0.441$ is marked out with an orange dot.

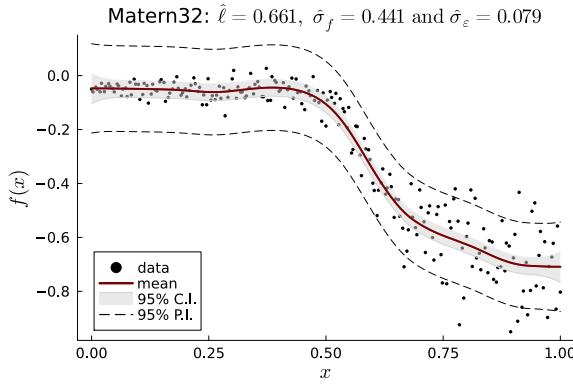


Figure 15.13: Posterior distribution for $f(x)$ (gray shaded regions) and predictive intervals (dashed lines) for Lidar data using the Matern32 kernel with hyperparameters estimated by maximizing the marginal likelihood.

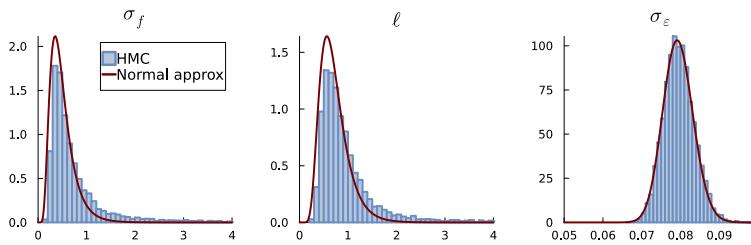


Figure 15.14: Marginal posterior distributions for the hyperparameters in the Lidar data using the Matern32 kernel. The histograms are computed from 5000 Hamiltonian Monte Carlo draws using the NUTS sampler. The normal approximation was obtained for log-transformed hyperparameters, so the densities in the figure are log-normal distributions.

15.4 Heteroscedastic Gaussian processes regression

The standard Gaussian process regression model assumes the errors $\varepsilon_1, \dots, \varepsilon_n$ to be independent, i.e. $\varepsilon_i \stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$. Generalizing to a regression model with $\varepsilon \sim N(\mathbf{0}, \Sigma_\varepsilon)$ with a known positive definite covariance matrix Σ_ε is straightforward; just replace $\sigma_\varepsilon^2 \mathbf{I}_n$ in (15.12) with Σ_ε to obtain the same formulae for the posterior as in (15.15) and (15.16) with $\sigma_\varepsilon^2 \mathbf{I}_n$ replaced by Σ_ε . In the case with unknown Σ_ε we proceed by assigning a Inverse Wishart prior to Σ_ε as in Section 3.7 and obtain a conditional-marginal decomposition of the posterior similar to the iid case.

A particularly important noise variance structure is the *heteroscedastic Gaussian process regression* where the ε terms are independent, but have different variances over the observations

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_{\varepsilon_i}^2). \quad (15.23)$$

In this heteroscedastic model, the diagonal noise covariance matrix is diagonal, $\Sigma_\varepsilon = \text{Diag}(\sigma_{\varepsilon_1}^2, \dots, \sigma_{\varepsilon_n}^2)$. The variances on the diagonal are typically modelled with a *variance function*, a common choice is the exponential function $\sigma_{\varepsilon_i}^2 = \exp(\mathbf{z}_i^\top \boldsymbol{\beta})$, for some set of covariates \mathbf{z} that are believed to determine the noise variances. Such heteroscedastic models need to be handled with MCMC or posterior approximations. We can also have nonparametric variance function with another Gaussian process prior, see Section 15.5.

We will here fit a heteroscedastic Gaussian process regression model to the Lidar data where we allow the noise variance to also depend on the distance covariate. We use the Matern32 kernel and variance function

$$\sigma_\varepsilon^2 = \exp(\beta_0 + \beta_1 \cdot \text{distance}).$$

The heteroscedasticity is easier to interpret if we look at the standard deviation and rewrite the model slightly to clearly show its multiplicative form

$$\sigma_\varepsilon = \tilde{\beta}_0 \cdot \tilde{\beta}_1^{\text{distance}},$$

where $\tilde{\beta}_0 = \exp(\beta_0/2)$ and $\tilde{\beta}_1 = \exp(\beta_1/2)$. The interpretation of $\tilde{\beta}_0$ is the standard deviation of the noise at the shortest distance (since $\text{distance}=0$ here as it normalized from 0 to 1), and $\tilde{\beta}_1$ is the multiplicative factor that increases the standard deviation from the smallest distance to the largest in the dataset (since a unit increase in distance means going from the smallest to the largest distance when the covariate is normalized to $[0, 1]$).

The regression coefficients β_0 and β_1 are hyperparameters to be estimated jointly with the kernel hyperparameters ℓ and σ_f , using for

example MCMC or HMC sampling. Alternatively, we can approximate the posterior of the four-dimensional hyperparameter vector $\theta = (\log \ell, \log \sigma_f, \beta_0, \beta_1)$ with a multivariate normal distribution obtained by numerical optimization. A multivariate normal has univariate normal marginal distributions, hence the approximate univariate normal $\beta_0 | \mathbf{y}, \mathbf{X} \sim N(m, s^2)$ implies that $\tilde{\beta}_0 = \exp(\beta_0/2)$ is log-normally distributed with parameters $\mu = m/2$ and variance $s^2/4$. Similarly for $\tilde{\beta}_1$. Figure 15.15 shows these approximate marginal posterior distributions and histograms from HCM sampling. The normal approximation to the parameters in the standard deviation function is more or less perfectly aligned with the histogram from the HMC draws, while the normal approximation is not fully capturing the heavy right tail of the posterior for the kernel hyperparameter, similar to the homoscedastic case. Figure 15.16 shows the posterior distribution for $f(x)$ and the predictive intervals for the Lidar data using the Matern32 kernel with heteroscedastic variance, using the ML estimates of all hyperparameters. Note in particular how the explicit modelled increasing variance with distance gives wider predictive intervals for larger distances and therefore fits the data much better.

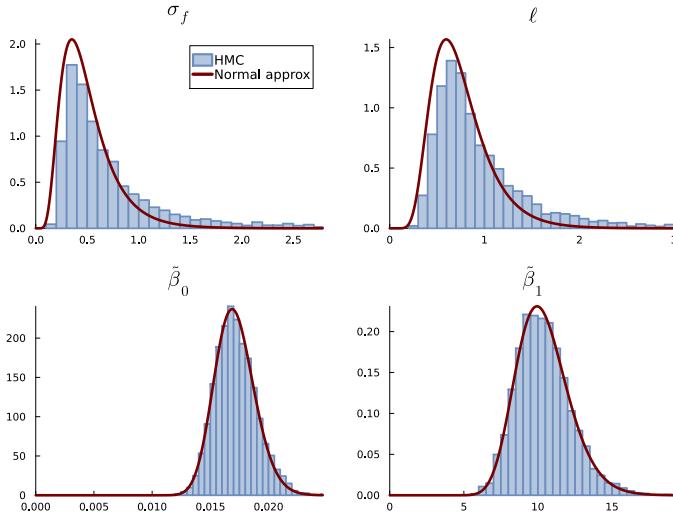


Figure 15.15: Marginal posterior distributions of the hyperparameters ℓ and σ_f in the Matern32 kernel, and the parameters in the standard deviation function $\tilde{\beta}_0 + \tilde{\beta}_1 \text{distance}$ in the heteroscedastic Gaussian process regression model fitted to the Lidar data. The histograms are from 5000 HMC draws and the solid curves are from the normal approximation of the posterior of the unrestricted parameter vector $\theta = (\log \ell, \log \sigma_f, \beta_0, \beta_1)^\top$.

15.5 Gaussian processes for classification

Gaussian processes priors can be used for nonparametric estimation of functions in a wide range of problems, not only regression models for a continuous response. In this section, we show how Gaussian processes can be used to make logistic and Poisson regression models more flexible.

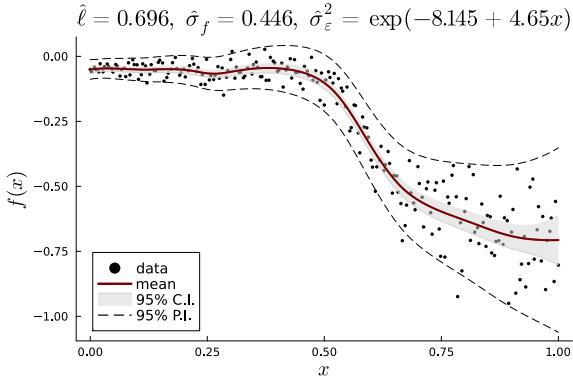


Figure 15.16: Posterior distribution for $f(x)$ (red line and gray shaded regions) and predictive intervals (dashed lines) for Lidar data using the Matern32 kernel with heteroscedastic variance.

The chapter [Classification and Generalized regression](#) introduced the logistic regression model for a binary response y conditional on a vector of covariates \mathbf{x}

$$y_i | \mathbf{x}_i, \boldsymbol{\beta} \sim \text{Bernoulli}(\theta(\mathbf{x}_i^\top \boldsymbol{\beta})), \quad (15.24)$$

where

$$\theta(z) = \frac{1}{1 + \exp(-z)}$$

is the logistic function that maps the linear predictor $\mathbf{x}_i^\top \boldsymbol{\beta}$ to the success probability $\Pr(y_i | \mathbf{x}_i) = \theta_i = \theta(\mathbf{x}_i^\top \boldsymbol{\beta}) \in [0, 1]$, thereby guaranteeing that the output is a probability for any $\mathbf{x} \in \mathbb{R}^p$ and $\boldsymbol{\beta} \in \mathbb{R}^p$. The logistic function can be said to ‘squash’ the linear predictor to the unit interval $[0, 1]$ ([Williams and Rasmussen, 2006](#)). The Gaussian cdf $\Phi(z)$ can be used as an alternative squashing function, giving rise to the probit regression model.

The logistic regression model is essentially a linear model since the log-odds

$$\log \left(\frac{\Pr(y=1|\mathbf{x})}{\Pr(y=0|\mathbf{x})} \right) = \mathbf{x}^\top \boldsymbol{\beta}. \quad (15.25)$$

is a linear function, with the implication that the two classes $y = 0$ and $y = 1$ are separated by linear decision boundaries in \mathbf{x} -space. Gaussian processes can be used to make the decision boundaries much more flexible, even nonparametric.

Gaussian process logistic regression is a nonparametric extension of the logistic regression model where the linear function of the covariates $\mathbf{x}^\top \boldsymbol{\beta}$ inside the squashing function is replaced by a general function $f(\mathbf{x})$ following a Gaussian process prior

$$y_i | \mathbf{x}_i, \boldsymbol{\beta} \stackrel{\text{indep}}{\sim} \text{Bernoulli}(\theta(f(\mathbf{x}_i))) \quad (15.26)$$

$$f(\cdot) \sim \text{GP}(\mu_0(\cdot), k_0(\cdot, \cdot)) \quad (15.27)$$

Note how this is exactly the same ideas as when the linear combination $\mathbf{x}^\top \boldsymbol{\beta}$ in Gaussian linear regression was replaced by a more

Gaussian process logistic regression

flexible function following a GP prior. Figure 15.17 illustrates how the realizations from the Gaussian process with a Matern52 kernel (left) gets squashed by the logistic function to the unit interval [0, 1] (right).

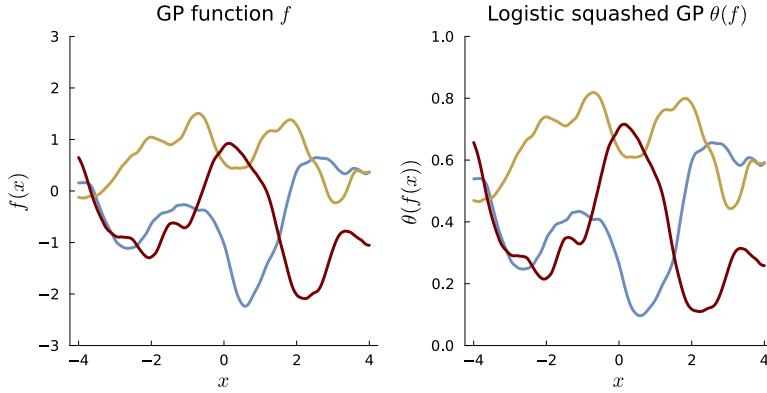


Figure 15.17: Illustrating how realizations from the Gaussian process with a Matern52 kernel (left) gets squashed by the logistic function $\theta(z) = 1/(1 + \exp(-z))$ to the unit interval [0, 1] (right).

The posterior distribution for the function $f(\cdot)$ for the logistic Gaussian process models is not tractable, even if we condition on the kernel hyperparameters. We can sample from the joint posterior using Hamiltonian Monte Carlo or a specific simulation algorithm designed for model with Gaussian process priors called elliptical slice sampling (Murray et al., 2010). The next section presents a normal approximation approach based on optimization for Gaussian process Poisson regression. The same approach is applicable to any generalized linear model with a Gaussian process prior, including the logistic regression in section. (Williams and Rasmussen, 2006) gives a detailed treatment of the normal approximation for the logistic regression model.

15.6 Gaussian processes for Poisson regression

It is by now quite easy to see how many other models can be extended to be nonparametric by replacing the linear predictor with a Gaussian process prior. This section presents approximate posterior inference and prediction for the Gaussian process Poisson regression model for count data:

$$y_i | \mathbf{x}_i, \boldsymbol{\beta} \stackrel{\text{indep}}{\sim} \text{Pois}(\theta(f(\mathbf{x}_i))) \quad (15.28)$$

$$f(\cdot) \sim \text{GP}(\mu_0(\cdot), k_0(\cdot, \cdot)), \quad (15.29)$$

where $\theta(z) : \mathbb{R} \rightarrow \mathbb{R}^+$ is any twice differentiable function, most commonly the exponential function $\theta(z) = \exp(z)$.

As in Chapter Normal posterior approximation we will here approximate the otherwise intractable posterior of \mathbf{f} conditional on

the prior hyperparameters θ by a multivariate normal distribution obtained from optimization

$$\mathbf{f}|\mathbf{y}, \theta \sim N(\tilde{\mathbf{f}}, \mathbf{J}_{\mathbf{f}, \mathbf{y}}^{-1}), \quad (15.30)$$

where $\tilde{\mathbf{f}}$ is the posterior mode and

$$\mathbf{J}_{\mathbf{f}, \mathbf{y}} = -\frac{\partial^2 \log p(\mathbf{f}|\mathbf{y}, \theta)}{\partial \mathbf{f} \partial \mathbf{f}^\top} \Big|_{\mathbf{f}=\tilde{\mathbf{f}}}$$

is the usual observed information matrix. The log density for an individual observation $y_i|f_i \sim \text{Pois}(\exp(f_i))$ is

$$\log p(y_i|f_i) = y_i f_i - \exp(f_i) - \log y_i!$$

so the log-likelihood for the training data is $\mathbf{y} = (y_1, \dots, y_n)^\top$ is

$$\log p(\mathbf{y}|\mathbf{f}, \theta) = \sum_{i=1}^n (y_i f_i - \exp(f_i) - \log y_i!)$$

Since $\mathbf{f} \sim N(\mu_0, \Omega_0)$ a priori, the log posterior is

$$\log p(\mathbf{f}|\mathbf{y}, \theta) = \sum_{i=1}^n (y_i f_i - \exp(f_i) - \log y_i!) - \frac{1}{2} (\mathbf{f} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Omega}_0^{-1} (\mathbf{f} - \boldsymbol{\mu}_0) + c, \quad (15.31)$$

where c is a constant that does not depend on \mathbf{f} and therefore cancels when taking derivatives.

Note that the each f_i only enters exactly in one of the n terms in the sum in the log likelihood, which simplifies the gradient and Hessian computations. The gradient vector and Hessian matrix are given by

$$\begin{aligned} \frac{\partial \log p(\mathbf{f}|\mathbf{y}, \theta)}{\partial \mathbf{f}} &= (y_1 - \exp(f_1), \dots, y_n - \exp(f_n))^\top - \boldsymbol{\Omega}_0^{-1}(\mathbf{f} - \boldsymbol{\mu}_0) \\ \frac{\partial^2 \log p(\mathbf{f}|\mathbf{y}, \theta)}{\partial \mathbf{f} \partial \mathbf{f}^\top} &= -\text{Diag}\left((\exp(f_1), \dots, \exp(f_n))^\top\right) - \boldsymbol{\Omega}_0^{-1}, \end{aligned}$$

where $\text{Diag}(\mathbf{v})$ is a diagonal matrix with the vector $\mathbf{v} = (v_1, \dots, v_n)^\top$ on the diagonal. We can not analytically solve for the posterior mode $\tilde{\mathbf{f}}$ from the systems of equations $\frac{\partial \log p(\mathbf{f}|\mathbf{y}, \theta)}{\partial \mathbf{f}} = \mathbf{0}$, but Newton's algorithm presented in Chapter [Normal posterior approximation](#) can be used to find the mode iteratively, starting from an initial guess $\mathbf{f}^{(0)}$, for example the prior mean $\boldsymbol{\mu}_0$.

The posterior distribution of $\tilde{\mathbf{f}} = (\tilde{f}_1, \dots, \tilde{f}_n)^\top$ at a set of test covariates $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{\tilde{n}})^\top$ is

$$p(\tilde{\mathbf{f}}|\tilde{\mathbf{X}}, \mathbf{y}, \theta) = \int p(\tilde{\mathbf{f}}|\mathbf{f}, \tilde{\mathbf{X}}, \theta) p(\mathbf{f}|\mathbf{y}, \theta) d\mathbf{f}, \quad (15.32)$$

where we have used that $\tilde{\mathbf{f}}$ is conditionally independent of \mathbf{y} given \mathbf{f} . This distribution is intractable, but we can simulate from it using

the above derived normal approximation of $p(\mathbf{f}|\mathbf{y}, \boldsymbol{\theta})$. We can also simulate from the predictive distribution of new test responses $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_{\tilde{n}})^\top$. The steps are as follows:

- draw $\mathbf{f}^{(j)}|\mathbf{y}, \boldsymbol{\theta}$ from the normal approximation $N(\tilde{\mathbf{f}}, \mathbf{J}_{\mathbf{f}, \mathbf{y}}^{-1})$
- draw $\tilde{\mathbf{f}}^{(j)}$ from $p(\tilde{\mathbf{f}}|\mathbf{f}^{(j)}, \tilde{\mathbf{X}}, \boldsymbol{\theta})$
- draw $\tilde{y}_i^{(j)} \sim \text{Pois}(\exp(\tilde{f}_i^{(j)}))$ independently for $i = 1, \dots, \tilde{n}$.

The distribution $p(\tilde{\mathbf{f}}|\mathbf{f}, \tilde{\mathbf{X}}, \boldsymbol{\theta})$ in the second step is easily seen to be multivariate normal by using the same conditioning properties of the multivariate normal distribution as before, but this time with respect to the following joint distribution

$$\begin{pmatrix} \mathbf{f} \\ \tilde{\mathbf{f}} \end{pmatrix} = N\left(\begin{pmatrix} \boldsymbol{\mu}_0 \\ \tilde{\boldsymbol{\mu}}_0 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Omega}_{ff} & \boldsymbol{\Omega}_{f\tilde{f}} \\ \boldsymbol{\Omega}_{\tilde{f}f} & \boldsymbol{\Omega}_{\tilde{f}\tilde{f}} \end{pmatrix}\right). \quad (15.33)$$

Conditioning on \mathbf{f} gives

$$\tilde{\mathbf{f}}|\mathbf{f} \sim N(\boldsymbol{\mu}_{\tilde{f}|f}, \boldsymbol{\Omega}_{\tilde{f}|f})$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\tilde{f}|f} &= \tilde{\boldsymbol{\mu}}_0 + \boldsymbol{\Omega}_{\tilde{f}f} \boldsymbol{\Omega}_{ff}^{-1} (\mathbf{f} - \boldsymbol{\mu}_0) \\ \boldsymbol{\Omega}_{\tilde{f}|f} &= \boldsymbol{\Omega}_{\tilde{f}\tilde{f}} - \boldsymbol{\Omega}_{\tilde{f}f} \boldsymbol{\Omega}_{ff}^{-1} \boldsymbol{\Omega}_{f\tilde{f}}. \end{aligned}$$

15.7 Bayesian optimization

16 Interaction models

16.1 Surface splines

16.2 Bayesian regression trees

17 Dynamic models and sequential inference

Data often arrive sequentially, for example as a time series $y_{1:T} = (y_1, \dots, y_T)$ observed at equidistant time points $t = 1, 2, \dots, T$, or at irregular time points t_1, t_2, \dots, t_T , where the time between successive observations can vary. Although sequences observed in time will be the main focus of this chapter, the algorithms introduced will often also be applicable to other sequences, for example a sequence of words in a text.

We have already seen some models for time series data, for example autoregressive models and time series regression. In this chapter we will learn about a general class of models called state-space models, which includes those previously introduced time series models, but also a wide range of other models.

The idea of online learning was introduced in Chapter [Single-parameter models](#), where it was shown how the posterior distribution of a parameter θ can be updated sequentially as new data arrives. The parameter θ in that chapter remained constant throughout time and we learned more and more about it as additional data points became available. Here we generalize this online setting to the case where the unknowns themselves change over time, for example a time-varying parameter θ_t in a model that takes a new value in every time period $t = 1, \dots, T$. Such *time-varying parameter models* are often used in economics, engineering, and other fields to capture the changing nature of many systems.

17.1 Some examples of state-space models

To motivate the study of the state-space models, we will first go through some models that later will be shown to be state-space models, and will postpone the actual definition of a state-space model until a few pages later. For the moment, we only need to know that state-space models consist of a vector of **state variables** z_t that change over time. The state variables are so called **latent variables** that are not directly observable, so in this sense they are more like parameters that change over time. We can however observe a vector

state variables
latent variables

of **measurement variables** \mathbf{y}_t that depend on the state variables. We use the sequence of measurements $\mathbf{y}_{1:t} = (\mathbf{y}_1, \dots, \mathbf{y}_t)^\top$ to learn about the underlying latent state \mathbf{z}_t and its evolution in time. We typically use greek letters for unknown parameters, but we will in this chapter use the Roman letter \mathbf{z}_t for the unknown state variable at time t . This follows the convention in the state-space literature, and also more generally for models with latent variables.

ROBOT LOCALIZATION. To navigate and clean a room efficiently, a robot vacuum cleaner needs to locate its position in a room. For the robot, its position is therefore an unknown state vector $\mathbf{z}_t = (\text{lon}_t, \text{lat}_t)^\top$ with longitude and latitude coordinates. The robot uses measurements \mathbf{y}_t from its many sensors (e.g. infrared light sensors, camera images) to infer its location. Exactly how the state evolves over time and how the sensor measurements are linked to the state position depends on the specific robot and its sensors. The robot can infer its position by computing the posterior distribution of the current position $\mathbf{z}_t = (\text{lon}_t, \text{lat}_t)^\top$ conditional on the observed sensor data up to that time point $\mathbf{y}_{1:t} = (\mathbf{y}_1, \dots, \mathbf{y}_t)^\top$. Such a posterior distribution gives the robot a complete probabilistic description of its current position which it can use to plan the next move.

FINDING THE STARS IN MACROECONOMICS. Economic policy is a balance act: when firms are operating at maximal production capacity we get low unemployment, but typically also high inflation. Economic policies may therefore aim to minimize the *natural unemployment rate* u_t^* – the employment rate that is consistent with stable inflation – while also maximizing *potential output* y_t^* , defined as the maximal output that can be produced in an economy without creating an excessive inflationary pressure. The $*$ superscript is typically used for these variables to indicate that they are not directly observable, we cannot simply go out in the world and measure them. We therefore try to infer the time series of u_t^* and y_t^* from other variables that are observable: inflation p_t (from price surveys), the unemployment rate u_t (from employment surveys) and industrial production y_t (from production surveys). In the language of state-space models, $\mathbf{z}_t = (u_t^*, y_t^*)^\top$ is the state vector, which is inferred from the observation vector, $\mathbf{y}_t = (p_t, u_t, y_t)^\top$. How the observable time series data are linked to the underlying latent state variables is determined by economic theory. With state-space models and Bayesian learning we can obtain the posterior distribution of the unknown state variables, u_t^* and y_t^* for $t = 1, \dots, T$ from observed time series of inflation, unemployment and industrial production; we can "find the stars $*$ in macroeconomics".

measurement variables

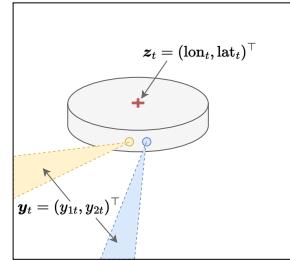


Figure 17.1: A vacuum cleaning robot learning its location $\mathbf{z}_t = (\text{lon}_t, \text{lat}_t)^\top$ (red cross) with the help of its two sensor measurements $\mathbf{y}_t = (y_{1t}, y_{2t})^\top$.

LOCAL LEVEL MODEL. A piecewise constant level model for a time series y_t , $t = 1, \dots, T$, with K different levels is of the form

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2) \quad (17.1)$$

$$\mu_t = \begin{cases} \mu_1 & \text{if } 1 \leq t \leq t_1 \\ \mu_2 & \text{if } t_1 < t \leq t_2 \\ \vdots & \vdots \\ \mu_K & \text{if } t_{K-1} < t \leq T, \end{cases} \quad (17.2)$$

so that the K different time segments $(t_{k-1}, t_k]$ each have their own constant level μ_k and the time series is just random noise ε_t around that level. A simulated realization is given in Figure 17.2 and this [observable widget](#) can be used to simulate data with more segments. To estimate this model we would have to locate all K constant segments $(t_{k-1}, t_k]$ and then estimate the level μ_k for each segment. An alternative model is the local level model, where the level μ_t is allowed to change by a random amount in each time period. The **local level model** is of the form

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2) \\ \mu_t &= \mu_{t-1} + \eta_t, & \eta_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\eta^2) \end{aligned} \quad (17.3)$$

This is a state space model with an unobservable state μ_t evolving as a random walk over time, and the measurements y_t are independent *conditional* on the state μ_t . Figure 17.3 shows a realization from this model, see also this [observable widget](#). Contrary to the previous two examples where the state was a real world object, albeit unobservable, the state μ_t in the local level model is more of an abstract model parameter that evolves over time.

DYNAMIC REGRESSION. It is not uncommon that the relationship between a response variable y_t and a set of explanatory variables \mathbf{x}_t changes over time. For example, how the temperature affects biking habits may change over time as bikes get more adapted to colder weather. Unless the model includes additional covariates to capture these structural changes, the model may not be able to accurately predict the response variable y_t in the future. A simple way to account for structural changes is to allow the regression coefficients to change over time:

$$\begin{aligned} y_t &= \mathbf{x}_t^\top \boldsymbol{\beta}_t + \varepsilon_t, & \varepsilon_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2) \\ \boldsymbol{\beta}_t &= \boldsymbol{\beta}_{t-1} + \boldsymbol{\eta}_t, & \boldsymbol{\eta}_t &\stackrel{\text{iid}}{\sim} N(0, \boldsymbol{\Sigma}_\eta), \end{aligned} \quad (17.4)$$

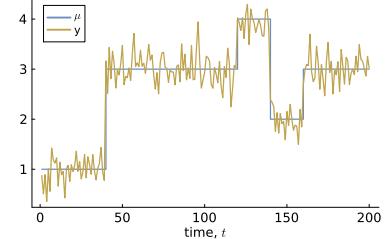


Figure 17.2: Simulated time series from a piecewise constant level model with $\sigma_\varepsilon = 0.25$.

local level model

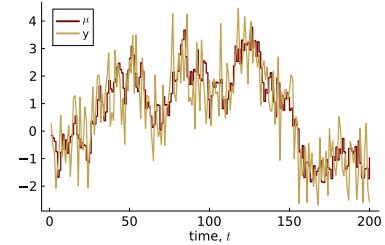
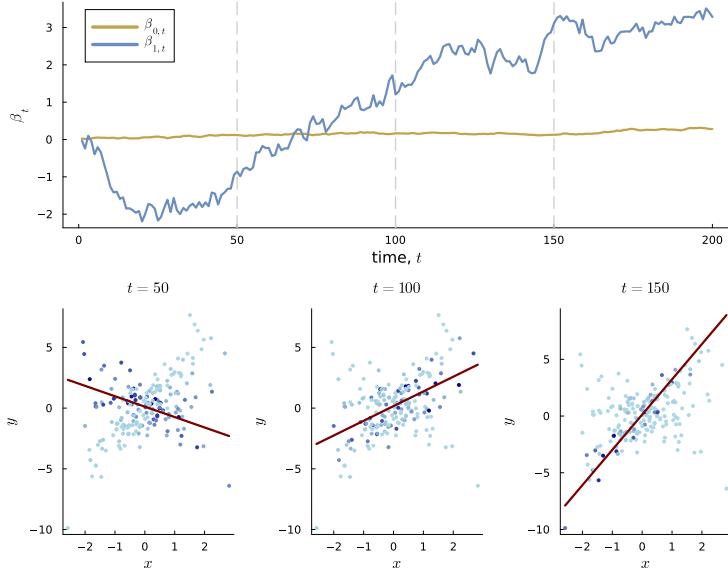


Figure 17.3: Simulated time series from a local level model with $\sigma_\eta = 0.5$ and $\sigma_\varepsilon = 1$.

where the vector of regression coefficients β_t is now indexed by time t to indicate that it changes from period to period. How fast the regression coefficients change is determined by the innovation covariance matrix Σ_η . It is typically assumed that the elements of β_t evolve independently over time, i.e. that Σ_η is a diagonal matrix. Figure 17.4 shows a simulation from the dynamic regression model in (17.4) with parameters evolving over time (top) and snapshots of the regression line at three time points (bottom).



Here is a good time to pause to consider what we are trying to accomplish with these time-varying models, or state-space models in general. Estimating a dynamic regression model seems like an impossible task since we have a new vector of unknown regression coefficients β_t at every time point; that is, it seems that we are trying to estimate β_t at time t from a single data observation (y_t, \mathbf{x}_t) , which is usually not a great idea in regression. The reason why this actually works is that we are also assuming that the β_t are evolving smoothly in time as a random walk $\beta_t = \beta_{t-1} + \eta_t$, which connects up the β_t at different time periods. From a Bayesian perspective, the random walk evolution of the β_t can be viewed as a prior $\beta_t | \beta_{1:t-1} \sim N(\beta_{t-1}, \Sigma_\eta)$. From this perspective we can say that estimation of time-varying parameter models is possible by adding information via the prior. Note how this falls in the category of smoothness priors discussed in Chapter [Priors](#) where the smoothness is now in time: we believe a priori that β_t is similar to the previous value β_{t-1} . Exactly how smoothly is determined by the innovation covariance matrix Σ_η . It is a question of semantics whether the random walk for the β_t is part

Figure 17.4: Simulated time series from a dynamic linear regression $y_t = \beta_{0,t} + \beta_{1,t}x_t + \varepsilon_t$, $\varepsilon_t \stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$ with $\beta_{0,0} = \beta_{1,0} = 0$, $\sigma_\varepsilon = 1$ and $\Sigma_\eta = \text{Diag}(0.01^2, 0.2^2)$.
Top: realized time evolution of the two regression coefficients, where the slope coefficient (blue) is changing more rapidly than the intercept (orange).
Bottom: snapshots of the regression lines at three points in time. Datapoints observed nearer to the time of the snapshot are plotted in darker color.

of the model or the prior, the generating process for the data remains the same and so does the posterior.

17.2 The linear Gaussian state-space model

With the motivating examples in the previous subsection for inspiration, we are now ready to formally define the class of state space models, beginning with important subclass of linear Gaussian models.

The **linear Gaussian state-space model** for a m -dimensional observation vector \mathbf{y}_t and s -dimensional state vector \mathbf{z}_t is of the form

$$\text{measurement model: } \mathbf{y}_t = \mathbf{C}\mathbf{z}_t + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \stackrel{\text{iid}}{\sim} N(0, \boldsymbol{\Sigma}_{\varepsilon}) \quad (17.5)$$

$$\text{state dynamics: } \mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \stackrel{\text{iid}}{\sim} N(0, \boldsymbol{\Sigma}_{\eta}), \quad (17.6)$$

where \mathbf{A} is an $s \times s$ matrix governing the transition of the state vector over time, \mathbf{C} is a $m \times s$ matrix mapping the state vector to the observation vector. The positive definite matrix $\boldsymbol{\Sigma}_{\varepsilon}$ is the covariance matrix for the **measurement errors** $\boldsymbol{\varepsilon}_t$ while $\boldsymbol{\Sigma}_{\eta}$ is the positive definite covariance of the **state innovations** $\boldsymbol{\eta}_t$. The initial state \mathbf{z}_0 is either assumed known or assigned a prior distribution, $\mathbf{z}_0 \sim N(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Sigma}_{0|0})$. We are here using a notation where the subscript $0|0$ indicates that the prior distribution is for the state at time $t = 0$ (the zero before the vertical bar) and is based on all the data up to and including time $t = 0$ (the zero after the vertical bar).

The linear Gaussian state-space model in (17.5) and (17.6) makes four assumptions:

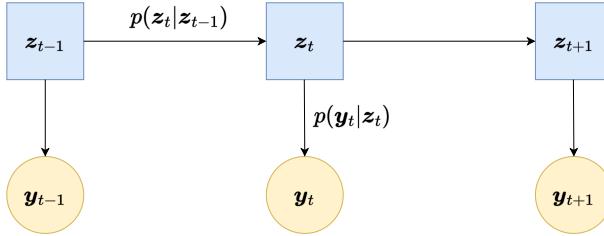
- A1 **(Markov state transitions)** The state \mathbf{z}_t is a first order Markov process, i.e. \mathbf{z}_t depends only on \mathbf{z}_{t-1} .
- A2 **(Conditional independence of measurements)** The observations \mathbf{y}_t are independent conditional on the current state vector \mathbf{z}_t .
- A3 **(Linearity)** The state dynamics and the mapping from the states to the measurements are both linear.
- A4 **(Gaussianity)** The measurement noise $\boldsymbol{\varepsilon}_t$, the state innovations $\boldsymbol{\eta}_t$ and the prior distribution for the initial state \mathbf{z}_0 are all Gaussian.

The conditional independence assumptions in A1 and A2 are visualized in the graphical model of the state-space model in Figure 17.5. A graphical model represents the conditional independence assumptions in a model, where each node in the graph represents a random variable (blue boxes for the state variables and yellow circles for the observations) and the edges/arrows between the nodes represent

linear Gaussian state-space model

measurement errors
state innovations

conditional dependencies; a missing edge between two nodes therefore means that the two variables are conditionally independent. The graphical model in Figure 17.5 shows that the measurement vector \mathbf{y}_t at time t is independent of the measurements at other time periods *conditional on* the state \mathbf{z}_t at time t . The graphical model also shows that the state is first order Markov since \mathbf{z}_t only depends on \mathbf{z}_{t-1} .



LOCAL LEVEL MODEL IN STATE-SPACE FORM. The local level model in (17.3) is a state-space model with $\mathbf{z}_t = \mu_t$, $m = 1$, $s = 1$, $\mathbf{A} = 1$, $\mathbf{C} = 1$, $\Sigma_\varepsilon = \sigma_\varepsilon^2$, $\Sigma_\eta = \sigma_\eta^2$.

DYNAMIC REGRESSION IN STATE-SPACE FORM. The time series regression in (17.4) is a state-space model with $\mathbf{z}_t = \beta_t$, $m = 1$, $s = p$, $\mathbf{C}_t = \mathbf{x}_t^\top$, $\mathbf{A} = \mathbf{I}_p$, $\Sigma_\varepsilon = \sigma_\varepsilon^2$, $\Sigma_\eta = \Sigma_\eta$. Note that the matrix $\mathbf{C}_t = \mathbf{x}_t^\top$ varies over time here, but in a completely deterministic way since we know the vector of covariates \mathbf{x}_t in each time period. The generalization to deterministically time-varying \mathbf{A} , \mathbf{C} , Σ_ε and σ_ε^2 is straightforward to handle for the algorithms described later, we just need to insert the current time-indexed \mathbf{A}_t , \mathbf{C}_t , $\Sigma_{t,\varepsilon}$ and $\sigma_{t,\varepsilon}^2$ at each time-step of the algorithm. Parameters that are constant over time are often called *static* parameters to distinguish them from the time-varying state \mathbf{z}_t .

The state-space model in (17.5) and (17.6) lacks the concept of **control variables**. In many applications we have some control over the state dynamics, for example the robot in Figure 17.1 can control its movement by using the motor and steering to choose the speed, acceleration and heading direction. The control variables are typically determined by a control person (in this case the robot herself) and are therefore not random. A control application aims to find the sequence of control signals $\mathbf{u}_1, \mathbf{u}_2, \dots$ that maximizes some desirable objective (cleaning a room in short time). In economics, the control variables constitute the economic policy that is under control of politicians and other policy makers. This is a type of sequential decision problems that goes under the heading of *Markov Decision processes* in automatic control and statistics or **reinforcement learning**

Figure 17.5: Graphical model representation of the state-space model with state \mathbf{z}_t and measurement vector \mathbf{y}_t . Note how each measurement vector \mathbf{y}_t at time t is independent of the measurements at other time periods given the state \mathbf{z}_t at time t . The graphical model also shows that the state is first order Markov since \mathbf{z}_t only depends on \mathbf{z}_{t-1} .

control variables

reinforcement learning

in the machine learning field. The general state-space model with control variables is of the form

$$\text{measurement: } \mathbf{y}_t = \mathbf{C}\mathbf{z}_t + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \stackrel{\text{iid}}{\sim} N(0, \boldsymbol{\Sigma}_{\varepsilon}) \quad (17.7)$$

$$\text{state dynamics: } \mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{B}\mathbf{u}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \stackrel{\text{iid}}{\sim} N(0, \boldsymbol{\Sigma}_{\eta}), \quad (17.8)$$

where \mathbf{u}_t is the vector with control variables, which are also called *control signals*. The state at time t is determined by the control signal \mathbf{u}_t and some random disturbance $\boldsymbol{\eta}_t$ to the state dynamics.

The state-space model in (17.5) and (17.6) can be further generalized to allow for dependence between measurement errors $\boldsymbol{\varepsilon}_t$ and state innovations $\boldsymbol{\eta}_t$, and, as mentioned in connection to time-varying regression, to also have system matrices \mathbf{A} , \mathbf{B} and \mathbf{C} that depend on time t . We will continue the rest of the chapter with the simpler model in (17.5) and (17.6) with constant system matrices and no control variables.

17.3 Bayesian filtering

The **filtering posterior** distribution $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ is the posterior distribution of the state vector \mathbf{z}_t at time t given the observations up to time t , which we denote by $\mathbf{y}_{1:t} = (y_1, \dots, y_t)^\top$. This is the *instantaneous* posterior where we look at the state at the current time point t using only the data available up that time point. The filtering posterior is the relevant distribution for online prediction, for example a robot determining its current location based on sensor data. In the next section we will discuss the smoothing, or retrospective, posterior $p(\mathbf{x}_t | \mathbf{y}_{1:T})$, which is the posterior distribution of the state vector \mathbf{x}_t at time t given *all* observations until the end of the time series, $\mathbf{y}_{1:T} = (y_1, \dots, y_T)^\top$.

filtering posterior

Note that our focus in Bayesian filtering is the posterior for the unknown state \mathbf{z}_t , assuming that the parameters \mathbf{A} , \mathbf{C} , $\boldsymbol{\Sigma}_{\eta}$ and $\boldsymbol{\Sigma}_{\varepsilon}$, in the state-space model are known. This may be true in some applications; for example, the robot in Figure 17.1 has have been calibrated in a lab before being deployed in the real world. In other applications, the parameters may be unknown and need to be estimated from the data. We will discuss this case in Section 17.6, but for now we will assume that the parameters are known.

Our goal here is to develop an algorithm for computing the filtering posterior that recursively updates the prior distribution $p(\mathbf{z}_t | \mathbf{y}_{1:t-1})$ at time t to the posterior distribution $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ at time t . Note that the term *prior* at time t refers to our beliefs based on all the data up to and including time $t - 1$, but *before* (prior to) observing the measurement \mathbf{y}_t at time t . Similarly, the term *posterior* at time t refers

to our beliefs based on all the data up to and including time t , *after* (posterior to) observing the measurement \mathbf{y}_t at time t . The posterior at time t is obtained with Bayes' rule

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{y}_{1:t}) &\propto p(\mathbf{y}_t | \mathbf{z}_t, \mathbf{y}_{1:t-1}) p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \\ &\propto p(\mathbf{y}_t | \mathbf{z}_t) p(\mathbf{z}_t | \mathbf{y}_{1:t-1}), \end{aligned} \quad (17.9)$$

where the last equality follows from Assumption A2 above: the measurement \mathbf{y}_t conditional on the current state \mathbf{z}_t is independent of all previous measurements $\mathbf{y}_{1:t-1}$, so we can cross out the dependence on $\mathbf{y}_{1:t-1}$.

The prior distribution at time t in (17.9) can be obtained from an application of the law of total probability in Figure ??

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) &= \int p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{y}_{1:t-1}) p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{z}_{t-1} \\ &= \int p(\mathbf{z}_t | \mathbf{z}_{t-1}) p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{z}_{t-1}, \end{aligned} \quad (17.10)$$

where the last equality follows from the Markov property in the state-space model in Assumption A1: the current state \mathbf{z}_t is independent of previous measurements $\mathbf{y}_{1:t-1}$ conditional on the previous state \mathbf{z}_{t-1} . Equation (17.10) thus takes us from the posterior distribution of \mathbf{z}_{t-1} at time $t-1$, $p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})$, to the prior distribution of \mathbf{z}_t at time $t-1$, $p(\mathbf{z}_t | \mathbf{y}_{1:t-1})$. We say that we are *propagating* (carrying forward) the posterior distribution of \mathbf{z}_{t-1} through the state dynamics model to obtain the prior distribution for the state in the next time period \mathbf{z}_t , i.e. prior to observing the data \mathbf{y}_t .

The updating of the prior distribution at time t , $p(\mathbf{z}_t | \mathbf{y}_{1:t-1})$, to the posterior distribution $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ at time t can hence be achieved in two steps: i) prior propagation (17.10) (solid green arrow in Figure 17.6) followed by a ii) measurement update (17.9) (dashed blue arrow in Figure 17.6). The two steps are summarized in Box 17.1 and illustrated in Figure 17.6.

17.4 Bayesian filtering in linear Gaussian models

The two steps in the Bayes filter in Box 17.1 are intractable for most models and priors. An important exception is the linear Gaussian state-space model, where analytical expressions can be derived for both prior propagation and measurement update steps, leading to the famous Kalman filtering algorithm. The Kalman filter has been hugely successful in an astonishing range of applications. The original algorithm was derived to minimize the mean squared forecast error of the state vector. As we will see, the Kalman filter is a special case of the Bayes filter, derived by the usual Bayesian prior-to-

Bayes filter update

$$p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t})$$

prior propagation

$$\underbrace{p(\mathbf{z}_t | \mathbf{y}_{1:t-1})}_{\text{prior at } t} = \int p(\mathbf{z}_t | \mathbf{z}_{t-1}) \underbrace{p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1})}_{\text{posterior at } t-1} d\mathbf{z}_{t-1}$$

measurement update

$$\underbrace{p(\mathbf{z}_t | \mathbf{y}_{1:t})}_{\text{posterior at } t} \propto p(\mathbf{y}_t | \mathbf{z}_t) \underbrace{p(\mathbf{z}_t | \mathbf{y}_{1:t-1})}_{\text{prior at } t}$$

Box 17.1: The two phases of the Bayes filter update from time $t - 1$ to time t for a general state-space model.

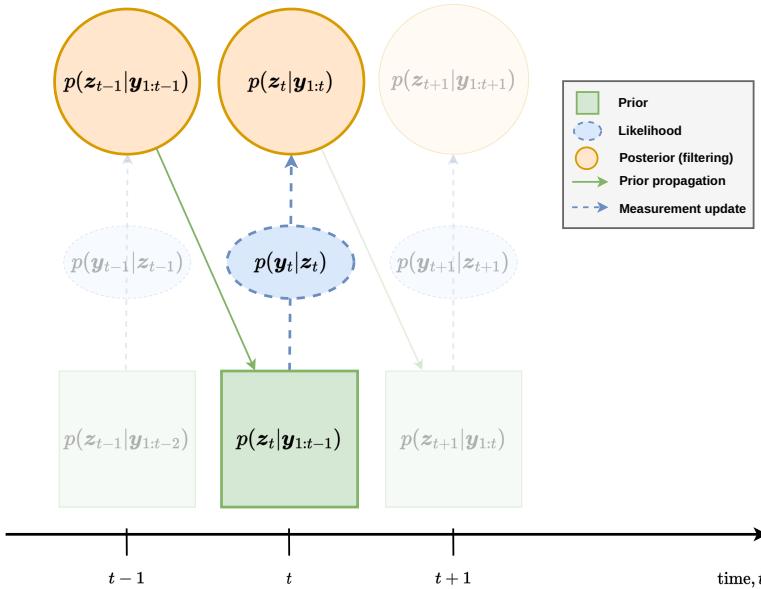


Figure 17.6: Illustration of the sequential nature of the Bayes filter where the posterior from the previous time step is first propagated forward one time step (solid green arrow) to become the current prior. That prior is then combined with the likelihood of the measurement at the current time step (dashed blue arrow) to obtain the (filtering) posterior of the current state.

posterior updating of the evolving state vector \mathbf{z}_t as new data \mathbf{y}_t arrives. When something works well in practice, Bayes is often lurking in the background.

We will now show that the filtering posterior $p(\mathbf{z}_t|\mathbf{y}_{1:t})$ in the linear Gaussian state-space model after observing $\mathbf{y}_{1:t}$ is Gaussian: $\mathbf{z}_t|\mathbf{y}_{1:t} \sim N(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Omega}_{t|t})$. The notation $\boldsymbol{\mu}_{t|s} = \mathbb{E}(\mathbf{z}_t|\mathbf{y}_{1:s})$ and $\boldsymbol{\Omega}_{t|s} = \mathbb{V}(\mathbf{z}_t|\mathbf{y}_{1:s})$ is meant to clearly show i) which state \mathbf{z}_t the posterior is for (time index before the $|$ sign) and ii) how much data is available when computing the posterior (time index after the $|$ sign). To derive the Bayes filter update from $p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})$ to $p(\mathbf{z}_t|\mathbf{y}_{1:t})$, we need to derive the prior propagation step and the measurement update step. If we assume that the prior $p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})$ is Gaussian for every t then we will show that the posterior $p(\mathbf{z}_t|\mathbf{y}_{1:t})$ is also Gaussian. Since the assumption of a Gaussian prior is true by assumption when $t = 1$, the initial prior is assumed to be Gaussian $\mathbf{z}_0 \sim N(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Omega}_{0|0})$, we immediately then see by induction that the posteriors $p(\mathbf{z}_t|\mathbf{y}_{1:t})$ are Gaussian for all $t = 1, \dots, T$. We will now derive the posterior means $\boldsymbol{\mu}_{t|t}$ and covariance matrices $\boldsymbol{\Omega}_{t|t}$ at each time step.

The prior propagation step in Box 17.1 is given by

$$p(\mathbf{z}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{z}_{t-1}.$$

Now, since $\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \boldsymbol{\eta}_t$, and both $\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}$ and $\boldsymbol{\eta}_t$ are Gaussian, the propagated prior $\mathbf{z}_t|\mathbf{y}_{1:t-1}$ is also Gaussian: $\mathbf{z}_t|\mathbf{y}_{1:t-1} \sim N(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Omega}_{t|t-1})$. The mean $\boldsymbol{\mu}_{t|t-1}$ and covariance matrix $\boldsymbol{\Omega}_{t|t-1}$ can be obtained with the iteration laws for the mean and variance in Figure ??, by first conditioning on \mathbf{z}_{t-1} and then undoing the conditioning by marginalizing with respect to $p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})$. By the law of iterated expectation, we have

$$\boldsymbol{\mu}_{t,t-1} \equiv \mathbb{E}_{\mathbf{z}_t|\mathbf{y}_{1:t-1}}(\mathbf{z}_t) = \mathbb{E}_{\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}}\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_{t-1}}(\mathbf{z}_t) = \mathbf{A}\boldsymbol{\mu}_{t-1,t-1}$$

using the conditional independence property $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{y}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{z}_{t-1})$ and that $\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_{t-1}}(\mathbf{z}_t) = \mathbf{A}\mathbf{z}_{t-1}$. The covariance of $p(\mathbf{z}_t|\mathbf{y}_{1:t-1})$ is by the law of total variance

$$\begin{aligned} \boldsymbol{\Omega}_{t,t-1} &\equiv \mathbb{V}_{\mathbf{z}_t|\mathbf{y}_{1:t-1}}(\mathbf{z}_t) = \mathbb{E}_{\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}}\mathbb{V}_{\mathbf{z}_t|\mathbf{z}_{t-1}}(\mathbf{z}_t) + \mathbb{V}_{\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}}\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_{t-1}}(\mathbf{z}_t) \\ &= \mathbb{E}_{\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}}\boldsymbol{\Sigma}_\eta + \mathbb{V}_{\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}}(\mathbf{A}\mathbf{z}_{t-1}) \\ &= \boldsymbol{\Sigma}_\eta + \mathbf{A}\boldsymbol{\Omega}_{t-1,t-1}\mathbf{A}^\top. \end{aligned}$$

With prior propagated forward to time t , we can perform the

measurement update step in (17.9) using Bayes' rule

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{y}_{1:t}) &\propto p(\mathbf{y}_t | \mathbf{z}_t) p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \\ &\propto N(\mathbf{y}_t | \mathbf{C}\mathbf{z}_t, \boldsymbol{\Sigma}_\varepsilon) N(\mathbf{z}_t | \boldsymbol{\mu}_{t,t-1}, \boldsymbol{\Omega}_{t,t-1}) \\ &\propto N(\mathbf{z}_t | \boldsymbol{\mu}_{t,t}, \boldsymbol{\Omega}_{t,t}), \end{aligned}$$

where $N(\mathbf{z} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate Gaussian density with argument \mathbf{z} . The mean and covariance matrix in the last step can be derived by completing the square in the exponent of the product of the two Gaussian densities, just like we did for β in the linear regression case in Chapter [Linear Regression](#) (with \mathbf{z}_t playing the role of β here). The posterior mean after this bit of algebra is given by

$$\begin{aligned} \boldsymbol{\mu}_{t,t} &= \boldsymbol{\mu}_{t,t-1} + \boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top (\mathbf{C}\boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon)^{-1} (\mathbf{y}_t - \mathbf{C}\boldsymbol{\mu}_{t,t-1}) \\ &= \boldsymbol{\mu}_{t,t-1} + \mathbf{K}_t (\mathbf{y}_t - \mathbf{C}\boldsymbol{\mu}_{t,t-1}), \end{aligned}$$

where $\mathbf{K}_t = \boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top (\mathbf{C}\boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon)^{-1}$ is the so called **Kalman gain** that determines how much weight to put on the most recent measurement \mathbf{y}_t relative to the prior $\boldsymbol{\mu}_{t,t-1}$; for example, with noisy measurements $\boldsymbol{\Sigma}_\varepsilon$ is large and \mathbf{K}_t close to the zero matrix, causing the Kalman filter to essentially ignore the one-step-ahead prediction error $\mathbf{y}_t - \mathbf{C}\boldsymbol{\mu}_{t,t-1}$ at time t in the update. The covariance matrix is given by

$$\begin{aligned} \boldsymbol{\Omega}_{t,t} &= \boldsymbol{\Omega}_{t,t-1} - \boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top (\mathbf{C}\boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon)^{-1} \mathbf{C}\boldsymbol{\Omega}_{t,t-1} \\ &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \boldsymbol{\Omega}_{t,t-1}. \end{aligned}$$

The Kalman filter algorithm consists of updating the above posterior mean and covariance matrix recursively in time. The algorithm is summarized in Box 17.2, which uses the slightly more general version which also has a control signal \mathbf{u}_t in the state dynamics. The control signal is conventionally assumed to be available before the measurement \mathbf{y}_t is observed, which is why it appears in the prior propagation step. Code for a single time update with the Kalman filter algorithm is given in Box ??, and the full Kalman filter algorithm for all time periods $t = 1, \dots, T$ is given in Figure 17.8.

EXAMPLE: NILE FLOW DATA. We illustrate the filtering on the Nile flow data, a dataset often used for state-space modeling in the literature. The data, available in the `datasets` package in R, is a time series with measurements of the annual flow of the river Nile at Aswan during 1871–1970. The data is plotted in Figure 17.9 with a vertical line marking out a potential changepoint noted in the literature.

We analyze the Nile flow data using a local level model

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2) \\ \mu_t &= \mu_{t-1} + \eta_t, & \eta_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\eta^2). \end{aligned}$$

Kalman filter update for linear Gaussian state-space models

$$p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t})$$

prior propagation

$$N(\boldsymbol{\mu}_{t-1,t-1}, \boldsymbol{\Omega}_{t-1,t-1}) \Rightarrow N(\boldsymbol{\mu}_{t,t-1}, \boldsymbol{\Omega}_{t,t-1})$$

$$\boldsymbol{\mu}_{t,t-1} = \mathbf{A}\boldsymbol{\mu}_{t-1,t-1} + \mathbf{B}\mathbf{u}_t$$

$$\boldsymbol{\Omega}_{t,t-1} = \mathbf{A}\boldsymbol{\Omega}_{t-1,t-1}\mathbf{A}^\top + \boldsymbol{\Sigma}_\eta$$

measurement update

$$N(\boldsymbol{\mu}_{t,t-1}, \boldsymbol{\Omega}_{t,t-1}) \Rightarrow N(\boldsymbol{\mu}_{t,t}, \boldsymbol{\Omega}_{t,t})$$

$$\boldsymbol{\mu}_{t,t} = \boldsymbol{\mu}_{t,t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}\boldsymbol{\mu}_{t,t-1})$$

$$\boldsymbol{\Omega}_{t,t} = (\mathbf{I} - \mathbf{K}_t \mathbf{C}) \boldsymbol{\Omega}_{t,t-1}$$

with Kalman gain

$$\mathbf{K}_t = \boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top (\mathbf{C} \boldsymbol{\Omega}_{t,t-1} \mathbf{C}^\top + \boldsymbol{\Sigma}_\epsilon)^{-1}$$

Box 17.2: The Kalman filter update from time $t = 1$ to t for a linear Gaussian state-space model with control signal \mathbf{u}_t before observing measurement at time t .

```
function kalmanfilter_update(μ, Ω, u, y, A, B, C, Σε, Σn)
    # Prediction step - moving state forward without new measurement
    ȳ = A*μ + B*u
    Ȣ = A*Ω*A' + Σn

    # Measurement update - updating the N(ȳ, Ȣ) prior with the new data point
    K = Ȣ*C' / (C*Ȣ*C' .+ Σε) # Kalman Gain
    μ = ȳ + K*(y .- C*ȳ)
    Ω = (I - K*C)*Ȣ
    return μ, Ω
end
```

Figure 17.7: Julia code for an update step of the Kalman filter algorithm for the state-space model with a control signal \mathbf{u} . The code uses the bar sign for hyperparameters in the prior so that the prior mean is denoted by $\bar{\mu} = \boldsymbol{\mu}_{t,t-1}$ while μ without the bar sign is the posterior mean $\mu = \boldsymbol{\mu}_{t,t}$. The prior and posterior covariance matrix are similarly denoted by $\bar{\Omega} = \boldsymbol{\Omega}_{t,t-1}$ and $\Omega = \boldsymbol{\Omega}_{t,t}$, respectively.

```

function kalmanfilter(U, Y, A, B, C, Σ_e, Σ_n, μ_o, Σ_o)

T = size(Y,1) # Number of time steps
n = length(μ_o) # Dimension of the state vector

# Storage for the mean and covariance state vector trajectory over time
μ_traj = zeros(T, n)
Σ_traj = zeros(n, n, T)

# The Kalman iterations
μ = μ_o
Σ = Σ_o
for t = 1:T
    μ, Σ = kalmanfilter_update(μ, Σ, U[t,:]', Y[t,:]', A, B, C, Σ_e, Σ_n)
    μ_traj[t,:] = μ
    Σ_traj[:, :, t] = Σ
end

return μ_traj, Σ_traj
end

```

Figure 17.8: Julia code for the Kalman filter algorithm using the Kalman filter update in Box ???. The $T \times n$ matrix \mathbf{Y} contains the T observations \mathbf{y}_t stacked on top of each other row-wise, and the $T \times m$ matrix \mathbf{U} is a the analogous matrix for the control vector \mathbf{u}_t .

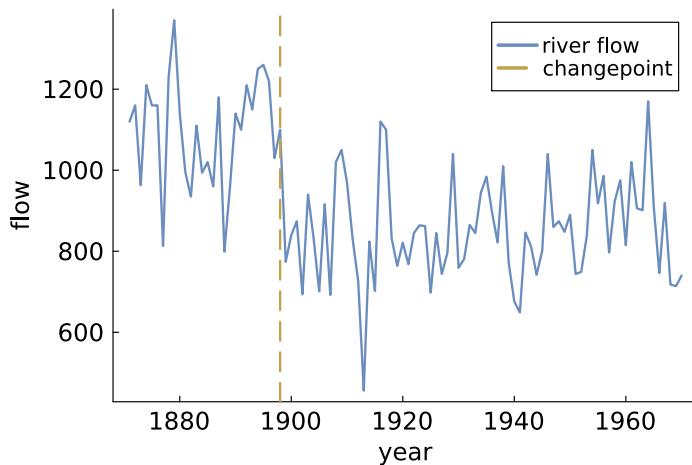


Figure 17.9: Time series of annual measurements (in $10^8 m^3$) of the river Nile at Aswan during 1871–1970. The vertical line marks a potential changepoint in the time series at year 1898.

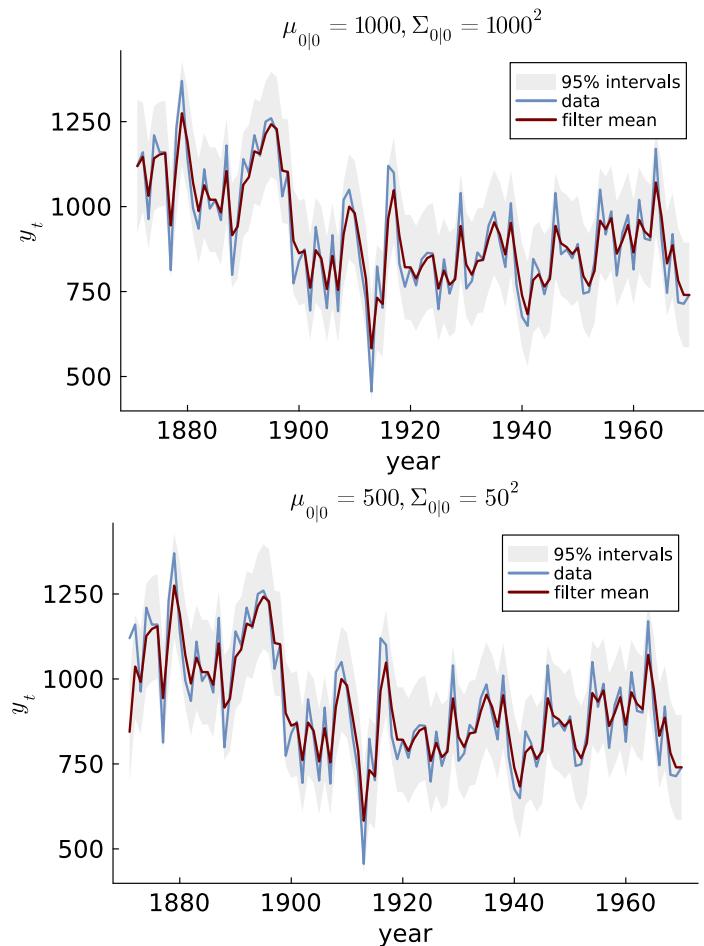


Figure 17.10: The filtering posterior mean (red line) and 95% credible intervals for the local level model fitted to the Nile flow data (blue line) with different priors on the initial state. Note how the filtering posterior is mainly affected by the prior in the first few time periods.

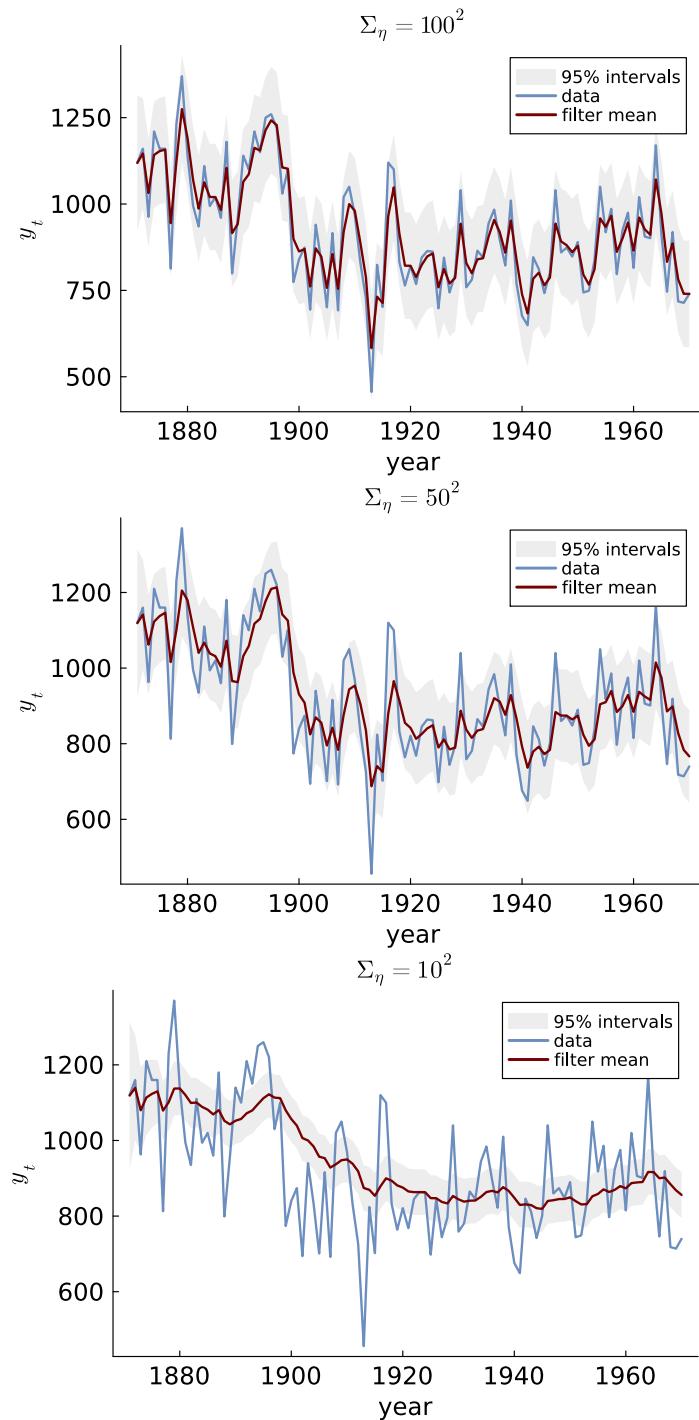


Figure 17.11: The filtering posterior mean (red line) and 95% credible intervals for the local level model fitted to the Nile flow data (blue line) with different parameter evolution variance. Smaller Σ_η makes the filtering mean smoother over time.

As explained above, this model can be cast as a state space model with the local level as a single state variable $\mathbf{z}_t = \mu_t$. For the local level model where have the state-space parameters: $\mathbf{A} = 1$, $\mathbf{C} = 1$, $\Sigma_\epsilon = \sigma_\epsilon^2$, $\Sigma_\eta = \sigma_\eta^2$. We will initially simply set $\sigma_\epsilon = \sigma_\eta = 100$. The prior for the initial state $N(\mu_{0|0}, \Omega_{0|0})$ with $\mu_{0|0} = 1000$ and $\Omega_{0|0} = 1000^2$, a rather uninformative prior. The top graph of Figure 17.10 shows the filtering posterior mean (red line) and 95% credible intervals for the local level model fitted to the Nile flow data (blue line). The bottom graph of Figure 17.10 shows the effect of changing the prior for the initial state to $\mu_{0|0} = 500$ and $\Omega_{0|0} = 50^2$, a more informative prior. The initial prior has a large effect on the filtering posterior in the first few time periods, but the effect diminishes as more data is observed.

Figure 17.11 shows how the filtering posterior depends on the innovation variance. As Σ_η is gradually decreased from $\Sigma_\eta = 100^2$ down to $\Sigma_\eta = 10^2$, the filtering posterior becomes more smooth since the state \mathbf{z}_t is allowed to change more slowly over time. In Section 17.6 we will discuss how to estimate the model parameters Σ_ϵ and Σ_η from the data. Figure 17.12 shows the filtering posterior mean when both model parameters Σ_ϵ and Σ_η are estimated by the posterior mode. Note however that this is the *filtering* posterior $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ so that the inference at every time point t in Figure 17.12 is only based on the data observed up to that time point $\mathbf{y}_{1:t}$. The effect of changing the model parameters and estimating the parameters can be explored in this [observable widget](#).

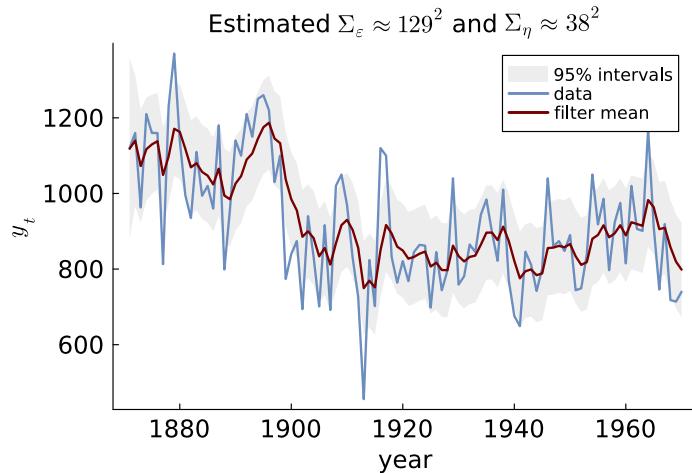


Figure 17.12: The filtering posterior mean (red line) and 95% credible intervals for the local level model fitted to the Nile flow data (blue line). The model parameters Σ_ϵ and Σ_η are estimated by the posterior mode.

17.5 Bayesian smoothing in linear Gaussian models

So far we have learned how compute the filtering posterior $p(\mathbf{z}_t|\mathbf{y}_{1:t})$, which is the online, immediate, posterior distribution at any time point t using all the data available up to that date. However, in many applications we are also interesting in the retrospective posterior $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ that uses all the data available up to the end of the time series. The posterior distribution $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ is the so called **joint smoothing distribution** looks back at time $t < T$ when standing at the final time period T . One example is inferring the potential output at some past date using all available data on inflation, employment and industrial production.

The **forward filtering backward sampling algorithm** simulates from the joint smoothing posterior $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ in linear Gaussian state space models. The algorithm is based on the fact that the joint posterior distribution of the state vector $\mathbf{z}_{1:T}$ can be factorized *backward* in time as

$$p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}) = p(\mathbf{z}_T|\mathbf{y}_{1:T}) \prod_{t=1}^{T-1} p(\mathbf{z}_t|\mathbf{z}_{t+1:T}, \mathbf{y}_{1:T}). \quad (17.11)$$

This is just another instance of the usual factorization of a joint density into a product of conditional densities, but here the first (marginal) factor is for the final time point $t = T$, and we then work backward in time. The factorization in (17.11) is convenient since we already know its first factor $p(\mathbf{z}_T|\mathbf{y}_{1:T})$ from the final iteration of the Kalman filter in Box 17.2: $\mathbf{z}_T|\mathbf{y}_{1:T} \sim N(\boldsymbol{\mu}_{T|T}, \boldsymbol{\Omega}_{T|T})$. The idea then is to simulate a draw $\mathbf{z}_T^{(1)} \sim N(\boldsymbol{\mu}_{T|T}, \boldsymbol{\Omega}_{T|T})$ and then simulate a draw $\mathbf{z}_{T-1}^{(1)}$ from the conditional distribution $p(\mathbf{z}_{T-1}|\mathbf{z}_T, \mathbf{y}_{1:T})$ given the draw $\mathbf{z}_T^{(1)}$. We continue like this backward in time until $t = 1$, at time step t drawing $\mathbf{z}_t^{(1)}$ conditional on the previously generated states at later time points, $\mathbf{z}_{t+1}^{(1)}, \dots, \mathbf{z}_T^{(1)}$. This gives us a single realized path of the state vector $\mathbf{z}_{1:T}^{(1)}$ from the smoothing posterior. We can repeat this process to obtain more sampled paths $\mathbf{z}_{1:T}^{(i)}$, for $i = 1, \dots, m$ from the smoothing posterior $p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T})$. The only missing piece in this algorithm is how to draw from the conditional distribution $p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{y}_{1:T})$ for time steps $t = T-1, T-2, \dots, 1$, which we will now derive.

First, note that one can easily show that the Markov property of the state space model also holds in reverse, so that we can write

$$p(\mathbf{z}_t|\mathbf{z}_{t+1:T}, \mathbf{y}_{1:T}) = p(\mathbf{z}_t|\mathbf{z}_{t+1}, \mathbf{y}_{1:t}). \quad (17.12)$$

This is also intuitive in that knowing \mathbf{z}_{t+1} makes the subsequent state evolution $\mathbf{z}_{t+2:T}$ irrelevant for \mathbf{z}_t . Also, if we condition on \mathbf{z}_{t+1} the

joint smoothing distribution

forward filtering backward sampling algorithm

noisy measurement \mathbf{y}_{t+1} is clearly useless for inferring \mathbf{z}_t , and the same certainly holds for subsequent measurements $\mathbf{y}_{t+2:T}$. By Bayes' theorem we then have

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{y}_{1:t}) &\propto p(\mathbf{z}_{t+1} | \mathbf{z}_t, \mathbf{y}_{1:t}) p(\mathbf{z}_t | \mathbf{y}_{1:t}) \\ &= N(\mathbf{z}_{t+1} | \mathbf{A}\mathbf{z}_t, \boldsymbol{\Sigma}_\eta) N(\mathbf{z}_t | \boldsymbol{\mu}_{t|t}, \boldsymbol{\Omega}_{t|t}), \end{aligned} \quad (17.13)$$

since the first factor is just the state transition density and the second factor is by definition the filtering posterior at time t . Since the expression in (17.13) is the product to two multivariate normal densities, and \mathbf{z}_t appears linearly in each quadratic form, we can complete the square in the usual fashion to show that $p(\mathbf{z}_t | \mathbf{z}_{t+1:T}, \mathbf{y}_{1:T})$ is a multivariate normal density. After some algebra we can express the mean and covariance matrix in this distribution by the following backward recursion formulas

$$\begin{aligned} \boldsymbol{\mu}_{t|t+1} &= \boldsymbol{\mu}_{t|t} + \boldsymbol{\Omega}_{t|t} \mathbf{A}^\top \boldsymbol{\Omega}_{t+1|t}^{-1} (\mathbf{z}_{t+1} - \mathbf{A}\boldsymbol{\mu}_{t|t}) \\ \boldsymbol{\Omega}_{t|t+1} &= \boldsymbol{\Omega}_{t|t} - \boldsymbol{\Omega}_{t|t} \mathbf{A}^\top \boldsymbol{\Omega}_{t+1|t}^{-1} \mathbf{A} \boldsymbol{\Omega}_{t|t}. \end{aligned} \quad (17.14)$$

Note how the mean of $p(\mathbf{z}_t | \mathbf{z}_{t+1:T}, \mathbf{y}_{1:T})$ explicitly depends on the future \mathbf{z}_{t+1} and remember that we will simulate \mathbf{z}_t given a previously simulated \mathbf{z}_{t+1} . The complete FFBS algorithm is given in Box 17.3.

17.6 Parameter inference in linear Gaussian state-space models

So far we have considered the model parameters \mathbf{A} , \mathbf{C} , $\boldsymbol{\Sigma}_\eta$ and $\boldsymbol{\Sigma}_\epsilon$ in the linear Gaussian state-space model as known when computing the filtering and smoothing posteriors. It is typically the case that the state-space parameters \mathbf{A} , \mathbf{C} , $\boldsymbol{\Sigma}_\eta$ and $\boldsymbol{\Sigma}_\epsilon$ are parametrized by a smaller dimensional vector of parameters $\boldsymbol{\theta}$. In many applications, the model parameters $\boldsymbol{\theta}$ are unknown and need to be estimated from data. For example, in the time varying regression model we had known $\mathbf{A} = \mathbf{I}_p$ and time-varying but known $\mathbf{C}_t = \mathbf{x}_t^\top$, but typically unknown measurement variance $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2$ and unknown parameter innovation variances $\boldsymbol{\Sigma}_\eta = \text{Diag}(\sigma_{\eta_1}^2, \dots, \sigma_{\eta_p}^2)$. Hence here we have $\boldsymbol{\theta} = (\sigma_\epsilon^2, \sigma_{\eta_1}^2, \dots, \sigma_{\eta_p}^2)$. From a Bayesian point of view, having unknown model parameters means obtaining the joint posterior distribution of the model parameters $\boldsymbol{\theta}$ and the states $\mathbf{z}_{1:T}$:

$$p(\boldsymbol{\theta}, \mathbf{z}_{1:T} | \mathbf{y}_{1:T}) = p(\mathbf{z}_{1:T} | \boldsymbol{\theta}, \mathbf{y}_{1:T}) p(\boldsymbol{\theta} | \mathbf{y}_{1:T}), \quad (17.15)$$

The marginal-conditional decomposition in (17.15) is convenient since we already know how to explore $p(\mathbf{z}_{1:T} | \boldsymbol{\theta}, \mathbf{y}_{1:T})$ by the FFBS algorithm for a given value of the model parameters.

However, we also need to simulate from the marginal posterior of the model parameters $p(\boldsymbol{\theta} | \mathbf{y}_{1:T})$ in (17.15), i.e. we need to integrate

Sampling the joint smoothing posterior in the LGSS model

```

Input: time series  $\mathbf{y}_{1:T} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$   

    model parameters  $\theta$   

    initial state prior mean  $\mu_{0|0}$   

    initial state prior covariance matrix  $\Omega_{0|0}$   

    number of samples from the posterior  $m$   

 $\mathbf{A}, \mathbf{C}, \Sigma_\eta, \Sigma_\varepsilon \leftarrow \text{SETUPSTATESPACE}(\theta)$   

 $\{\mu_{t|t}, \Omega_{t|t}, \mu_{t|t-1}, \Omega_{t|t-1}\}_{t=1}^T \leftarrow \text{KALMANFILTER}(\mathbf{y}_{1:T}, \mathbf{A},$   

 $\mathbf{C}, \Sigma_\eta, \Sigma_\varepsilon, \mu_{0|0}, \Omega_{0|0})$   

for  $i$  in  $1, \dots, m$  do  

    Simulate  $\mathbf{z}_T^{(i)} \sim N(\mu_{T|T}, \Omega_{T|T})$   

    for  $t$  in  $T-1, T-2, \dots, 1$  do  

         $\mu_{t|t+1} \leftarrow \mu_{t|t} + \Omega_{t|t} \mathbf{A}^\top \Omega_{t+1|t}^{-1} (\mathbf{z}_{t+1}^{(i)} - \mu_{t+1|t})$   

         $\Omega_{t|t+1} \leftarrow \Omega_{t|t} - \Omega_{t|t} \mathbf{A}^\top \Omega_{t+1|t}^{-1} \mathbf{A} \Omega_{t|t}$   

        Simulate  $\mathbf{z}_t^{(i)} | \mathbf{z}_{t+1}^{(i)} \sim N(\mu_{t|t+1}, \Omega_{t|t+1})$   

    end  

end  

Output: Draws  $\mathbf{z}_{1:T}^{(1)}, \dots, \mathbf{z}_{1:T}^{(m)}$  from the joint smoothing  

    posterior  $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T}, \theta)$  conditional on the  

    model parameters.
```

Box 17.3: Forward filtering backward sampling (FFBS) from the joint smoothing posterior $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T}, \theta)$. The function `SETUPSTATESPACE` sets up all the matrices in the state space model from a vector of model parameters θ and the function `KALMANFILTER` runs the Kalman filter to return the sequence of filtering means and covariances $\{\mu_{t|t}, \Omega_{t|t}\}_{t=1}^T$ as well as the corresponding quantities $\{\mu_{t|t-1}, \Omega_{t|t-1}\}_{t=1}^T$ from each prior propagation step. The simulation is performed backward in time so that the state at time t is simulated conditional on the simulated state at time $t+1$ (marked out in orange font).

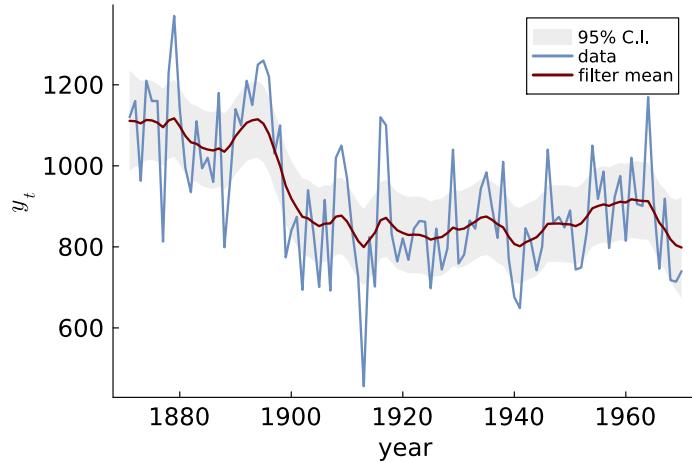


Figure 17.13: The smoothing posterior mean (red line) and 95% credible intervals for the local level model fitted to the Nile flow data (blue line) estimate from simulation from joint smoothing posterior $p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T})$ using the FFBS algorithm in Box 17.3. The model parameters Σ_ϵ and Σ_η are estimated by the posterior mode.

out the complete evolution of the state $\mathbf{z}_{1:t}$. It turns out that for the linear Gaussian state-space model the Kalman filter provides everything we need to compute $p(\theta|\mathbf{y}_{1:T})$, as we will now explain.

By Bayes' theorem, the marginal posterior of θ is

$$p(\theta|\mathbf{y}_{1:T}) \propto p(\mathbf{y}_{1:T}|\theta)p(\theta). \quad (17.16)$$

Using the usual sequential decomposition of a joint density into a product of conditional densities we can express the marginal likelihood as

$$\begin{aligned} p(\mathbf{y}_{1:T}|\theta) &= p(\mathbf{y}_1|\theta)p(\mathbf{y}_2|\mathbf{y}_1, \theta) \cdots p(\mathbf{y}_T|\mathbf{y}_{1:T-1}, \theta) \\ &= \prod_{t=1}^T p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta). \end{aligned}$$

We can obtain each term in this product by applying the continuous form of the law of total probability where we first condition on \mathbf{z}_t to enjoy the simplifications coming for the state space model's conditional independence assumptions, and then removing \mathbf{z}_t again by integrating with respect to \mathbf{z}_t . This gives

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta) &= \int p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_{1:t-1}, \theta)p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \theta)d\mathbf{z}_t \\ &= \int p(\mathbf{y}_t|\mathbf{z}_t, \theta)p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \theta)d\mathbf{z}_t \\ &= \int N(\mathbf{y}_t|\mathbf{C}\mathbf{z}_t, \Sigma_\epsilon)N(\mathbf{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Omega}_{t|t-1})d\mathbf{z}_t \\ &= N(\mathbf{y}_t|\mathbf{C}\boldsymbol{\mu}_{t|t-1}, \mathbf{C}\boldsymbol{\Omega}_{t|t-1}\mathbf{C}^\top + \Sigma_\epsilon), \end{aligned}$$

where the last equality follows from the fact that the product of two Gaussian densities is proportional to a Gaussian density with mean and covariance matrix given by the Kalman filter update equations in

Section 17.4, similar to the derivation of the prior propagation step in the Kalman filter. The marginal likelihood is hence given by

$$p(\mathbf{y}_{1:T}|\boldsymbol{\theta}) = \prod_{t=1}^T N(\mathbf{y}_t | \mathbf{C}\boldsymbol{\mu}_{t|t-1}, \mathbf{C}\boldsymbol{\Omega}_{t|t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}_\epsilon), \quad (17.17)$$

where we note that $\boldsymbol{\mu}_{t|t-1}$ and $\boldsymbol{\Omega}_{t|t-1}$ comes from the *prior propagation step* of the Kalman filter, i.e. before the measurement update. This is because a marginal likelihood is really a measure of the accuracy of the one-step-ahead *predictions*, as explained in Chapter [Model comparison and variable selection](#).

With this Kalman filter evaluation of the marginal likelihood we can evaluate the posterior $p(\boldsymbol{\theta}|\mathbf{y}_{1:T}) \propto p(\mathbf{y}_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$ vector. The posterior of the model parameters $\boldsymbol{\theta}$ in (17.16) can therefore be simulated by the Metropolis-Hastings algorithm or Hamiltonian Monte Carlo (HMC) algorithm, presented in Chapter [Markov Chain Monte Carlo simulation](#), where for every proposed $\boldsymbol{\theta}$ value we need to evaluate the marginal likelihood by running the Kalman filter in Figure 17.8 for $t = 1, \dots, T$ using the proposed $\boldsymbol{\theta}^*$ value when setting up the linear Gaussian state-space model parameters \mathbf{A} , \mathbf{C} , $\boldsymbol{\Sigma}_\eta$ and $\boldsymbol{\Sigma}_\epsilon$.

Given samples $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(m)}$ from the marginal posterior $p(\boldsymbol{\theta}|\mathbf{y}_{1:T})$, we can then complete the simulation from the joint posterior of the model parameters and the states $p(\boldsymbol{\theta}, \mathbf{z}_{1:T}|\mathbf{y}_{1:T}) = p(\boldsymbol{\theta}|\mathbf{y}_{1:T})p(\mathbf{z}_{1:T}|\boldsymbol{\theta}, \mathbf{y}_{1:T})$ by simulating the states using the FFBS algorithm in Box 17.3 for each value of $\boldsymbol{\theta}^{(i)}$. The end result will be m samples from the joint posterior of the states and model parameters $p(\mathbf{z}_{1:T}, \boldsymbol{\theta}|\mathbf{y}_{1:T})$.

An alternative strategy is to use a two-block Gibbs sampler that iterates for $i = 1, \dots, m$:

- **Block 1:** Simulate $\boldsymbol{\theta}^{(i)}|\mathbf{z}_{1:T}^{(i)} \sim p(\boldsymbol{\theta}|\mathbf{z}_{1:T}^{(i)}, \mathbf{y}_{1:T})$
- **Block 2:** Simulate $\mathbf{z}_{1:T}^{(i)}|\boldsymbol{\theta}^{(i)} \sim p(\mathbf{z}_{1:T}|\boldsymbol{\theta}^{(i)}, \mathbf{y}_{1:T})$

The advantage of this approach is that the posterior of the parameters conditional on the states in Block 1 can often be simulated from directly using conjugate priors, so no need to run the Kalman filter for computing the marginal likelihood. The disadvantage is that the Gibbs sampler can be slow to converge when the states and model parameters are correlated in the posterior, and may therefore require a large number of iterations compared to the above approach of drawing from the marginal-conditional decomposition $p(\boldsymbol{\theta}, \mathbf{z}_{1:T}|\mathbf{y}_{1:T}) = p(\boldsymbol{\theta}|\mathbf{y}_{1:T})p(\mathbf{z}_{1:T}|\boldsymbol{\theta}, \mathbf{y}_{1:T})$.

17.7 Non-linear non-Gaussian models

So far we have only considered linear Gaussian state-space models that satisfies all four assumptions listed in Subsection 17.2. Those as-

Marginal likelihood - linear Gaussian state-space model

```

Input: time series  $\mathbf{y}_{1:T} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$   

    model parameters  $\theta$   

    initial state prior mean  $\mu_{0|0}$   

    initial state prior covariance matrix  $\Omega_{0|0}$   

 $\mathbf{A}, \mathbf{C}, \Sigma_\eta, \Sigma_\epsilon \leftarrow \text{SETUPSTATESPACE}(\theta)$   

 $\{\mu_{t|t-1}, \Omega_{t|t-1}\}_{t=1}^T \leftarrow \text{KALMANFILTER}(\mathbf{y}_{1:T}, \mathbf{A}, \mathbf{C}, \Sigma_\eta,$   

 $\Sigma_\epsilon, \mu_{0|0}, \Omega_{0|0})$   

 $\text{lml} = 0$   

for  $t$  in  $1:T$  do  

    |  $\text{lml} = \text{lml} + \log N(\mathbf{y}_t | \mathbf{C}\mu_{t|t-1}, \mathbf{C}\Omega_{t|t-1}\mathbf{C}^\top + \Sigma_\epsilon)$   

end  

Output: log marginal likelihood  $p(\mathbf{y}_{1:T} | \theta)$ ,  $\text{lml}$ .

```

Box 17.4: Computing the log marginal likelihood $\log p(\mathbf{y}_{1:T} | \theta)$ in (17.17) at a given θ using the Kalman filter to return the sequence of prior means and covariances $\{\mu_{t|t-1}, \Omega_{t|t-1}\}_{t=1}^T$.

sumptions made it possible to derive the Kalman filter for efficiently computing the filter posterior $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ and to develop an efficient algorithm for simulating from the smoothing posterior $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T})$. There are however many interesting state-space models that violate one or several of the four assumptions. We will here consider models with non-linear and/or non-Gaussian aspects, but still satisfy Assumption 1 (Markov state transitions) and 2 (Conditional independence of measurements). Before stating the general model, we first introduce two commonly used non-linear non-Gaussian models.

POISSON WITH TIME-VARYING INTENSITY FOR TIME SERIES COUNTS
 Consider a time series of counts, for example the daily number of persons in intensive care during a pandemic. The Poisson model is a natural choice for count data, and the state-space methodology allows us to extend the model to the time series setting with *dependent* counts over time; if there is a large number of people in intensive care today, we also expect many there tomorrow. The Poisson model with autocorrelated time-varying intensity is

$$y_t | z_t \stackrel{\text{indep}}{\sim} \text{Pois}(\exp(z_t)) \\ z_t = z_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2). \quad (17.18)$$

Here we model the data as independent Poisson counts conditional on a time-varying and autocorrelated intensity $\lambda_t = \exp(z_t)$. Similar

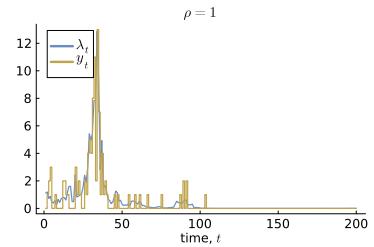


Figure 17.14: Simulated realization ($T = 200$) of the latent intensity process and the observed counts from the Poisson time series model in (17.18) with $\sigma_\eta = 0.5$ and initial value $z_0 = 0$.

to a Poisson regression, we use the exponential function to ensure that the intensity is always positive; put differently, we model the log intensity $\log \lambda_t = z_t$ as a random walk. Figure 17.7 shows a realization of the intensity $\lambda_t = \exp(z_t)$ process and the corresponding realized count time series.

STOCHASTIC VOLATILITY A commonly used model for financial returns is the stochastic volatility model, which in its most basic form is

$$\begin{aligned} y_t &= \mu + \varepsilon_t, & \varepsilon_t &\stackrel{\text{indep}}{\sim} N(0, \exp(z_t)), \\ z_t &= z_{t-1} + \eta_t, & \eta_t &\stackrel{\text{iid}}{\sim} N(0, \sigma_\eta^2). \end{aligned} \quad (17.19)$$

The stochastic volatility model has a heteroscedastic variance $\sigma_t^2 = \exp(z_t)$ that evolves over time via a latent random walk process, z_t . This gives rise to *volatility clustering*, which means periods of persistently low or high volatility. Figure 17.7 shows a realization of the standard deviation $\sigma_t = \exp(z_t/2)$ from the stochastic volatility model and the corresponding realized return time series y_t ; the volatility clustering is clearly visible.

The Poisson time series model in (17.18) and the stochastic volatility model in (17.19) are both examples of non-linear non-Gaussian state-space models. A general state-space model is given by

$$\begin{aligned} \text{measurement model: } & y_t \stackrel{\text{indep}}{\sim} h(y_t | \mathbf{z}_t) \\ \text{state dynamics: } & \mathbf{z}_t \sim g(\mathbf{z}_t | \mathbf{z}_{t-1}), \end{aligned} \quad (17.20)$$

where $g(\mathbf{z}_t | \mathbf{z}_{t-1})$ is the state transition model, i.e. a distribution for \mathbf{z}_t conditional on the past state \mathbf{z}_{t-1} , and $h(y_t | \mathbf{z}_t)$ is the measurement model, i.e. a distribution for the measurement y_t at time t conditional on the state \mathbf{z}_t at time t . The state transition model $g(\mathbf{z}_t | \mathbf{z}_{t-1})$ in (17.20) can be essentially any distribution and the current state \mathbf{x}_t can depend non-linearly on the previous state \mathbf{x}_{t-1} , but has to be first order Markov (Assumption A1). Similarly, the measurement model $h(y_t | \mathbf{z}_t)$ in (17.20) can be any distribution and the observations y_t can depend non-linearly on the current state \mathbf{z}_t , but should be conditionally independent given the current state (Assumption A2).

It is also possible to have non-linear/non-Gaussian parts mixed with linear/Gaussian parts. For example, the Poisson time series model in (17.18) has a non-Gaussian (Poisson) measurement model with non-linear dependence on the state ($\mathbb{E}(y_t | z_t) = \exp(z_t)$), but the state transition model is both linear and Gaussian. The stochastic volatility model in 17.7 is linear and Gaussian in the state transition model, has a Gaussian measurement model, but the measurements depend non-linearly on the state, $\mathbb{V}(y_t | z_t) = \exp(z_t)$.

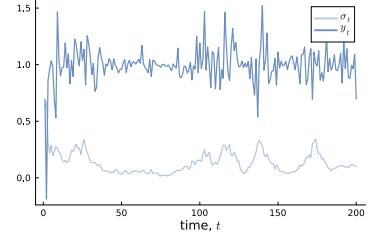


Figure 17.15: Simulated realizations ($T = 200$) of the latent volatility and the observed returns from the stochastic volatility model in (17.19) with $\mu = 1$, $\sigma_\varepsilon = 1$, $\sigma_\eta = 0.5$ and initial value $z_0 = 0$.

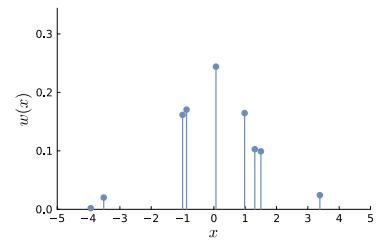


Figure 17.16: Illustration of a univariate weighted particle system.

In state-space models with non-linear and/or non-Gaussian parts, the Kalman filter and the FFBS algorithm for sampling from the joint smoothing posterior are no longer applicable. The Bayes filter and Bayes smoother can instead be implemented by so called particle methods, which we will now briefly describe.

17.8 Sequential inference in non-linear non-Gaussian models

For the general state-space model in (17.20), we will approximate the joint posterior filtering distribution $p(\mathbf{z}_{1:t}|\mathbf{y}_{1:t})$ with a discrete distribution with weighted point masses at a set of support points, also called **weighted particle system**, see Figure 17.16 for an illustration. A **point mass distribution** has all its probability mass at the single point $x = x_*$ and is defined using a **Dirac delta function** $\delta_{x_*}(\cdot)$ centered at x_* , see Figure ?? in Chapter Model comparison and variable selection. We can generalize the concept of Dirac delta functions to the multivariate setting by defining $\delta_{\mathbf{x}_*}(\mathbf{x})$ as a probability distribution with all its probability mass at the single point $\mathbf{x} = \mathbf{x}_*$ in \mathbb{R}^p .

A general multivariate probability distribution $p(\mathbf{x})$ can be approximated by a weighted sum of Dirac delta functions with weight w_j at location \mathbf{x}_j

$$p(\mathbf{x}) = \sum_{j=1}^M w_j \delta_{\mathbf{x}_j}(\mathbf{x}), \quad (17.21)$$

which is illustrated for the univariate case in Figure 17.16 and in Figure 17.17 for the bivariate case. The support points (17.21) \mathbf{x}_j are usually referred to as **particles**, the weights w_j as **particle weights** and the approximation in (17.21) as a **weighted particle system**.

Our aim now is to approximate the posterior filtering distribution with a weighted particle system

$$\hat{p}(\mathbf{z}_t|\mathbf{y}_{1:t}) = \sum_{j=1}^M w_t^{(j)} \delta_{\mathbf{z}_t^{(j)}}(\mathbf{z}_{1:t}), \quad (17.22)$$

which we can easily simulate from by drawing from the categorical distribution $j \sim \text{Cat}(w_t^{(1)}, \dots, w_t^{(M)})$ and returning $\mathbf{z}_t^{(j)}$, the j th particle. We obtain the weighted particle system in (17.8) by a recursive procedure called a **particle filter**.

We will now describe a simple but widely used particle filter, the so called *bootstrap filter*. Recall first the Bayes filter update from time $t - 1$ to time t in Box 17.1 with a prior propagation step followed by a measurement update step:

$$p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t|\mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t|\mathbf{y}_{1:t}). \quad (17.23)$$

When the model was linear and Gaussian we could derive both the prior propagation step and the measurement update step analytically.

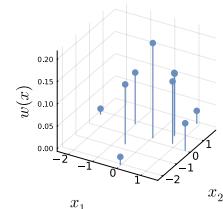


Figure 17.17: Illustration of a bivariate weighted particle system.

weighted particle system
point mass distribution
Dirac delta function

particles
particle weights
weighted particle system

particle filter

For non-linear and/or non-Gaussian models, this no longer possible and we resort to simulation.

The **bootstrap filter** at time t has three main steps:

bootstrap filter

- **Prior propagation.** Approximates the prior propagation step by propagating each of M particles $\mathbf{z}_{t-1}^{(j)}$ from $p(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1})$ forward in time by drawing particles from the state transition model $\mathbf{z}_t^{(j)} \sim g(\mathbf{z}_t|\mathbf{z}_{t-1}^{(j)})$, one for each particle $\mathbf{z}_{t-1}^{(j)}$.
- **Measurement update.** The measurement update step is then implemented by assigning weights $w_t^{(j)}$ to each propagated particle $\mathbf{z}_t^{(j)}$ based on how well the measurement \mathbf{y}_t agrees with the particle $\mathbf{z}_t^{(j)}$, i.e. with weights proportional to $p(\mathbf{y}_t|\mathbf{z}_t^{(j)})$.
- **Resampling.** The measurement update can lead to a very uneven distribution of the weights $w_t^{(j)}$ over time with most of the weight concentrated on a few particles, which is detrimental for sampling efficiency. The resampling step therefore draws M new particles $\mathbf{z}_t^{(j)}$ from the current particle system $\{\mathbf{z}_t^{(j)}, w_t^{(j)}\}_{j=1}^M$ with replacement.

The bootstrap particle filter is a form of sequential importance sampling where at each time step the particles are generated from the importance density $p(\mathbf{z}_t|\mathbf{y}_{1:t-1})$ and then assigned weights equal to the target posterior divided by the proposal density:

$$w_t^{(j)} = \frac{p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t})}{p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t-1})} \propto \frac{p(\mathbf{y}_t|\mathbf{z}_t^{(j)})p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t-1})}{p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t-1})} = p(\mathbf{y}_t|\mathbf{z}_t^{(j)}).$$

Note that the resampling step is done with replacement so that some particles may be drawn multiple times while others are not drawn at all. This creates a genealogy of the particles, where several particles at time t can come from the same *ancestor* at a previous time period.

The bootstrap filter algorithm is summarized in Box 17.5.

The bootstrap filter is easily implemented with simple weights, but can be inefficient since it is using the propagated prior $p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t-1})$ as importance density. If the current observation \mathbf{y}_t is very informative so that the target posterior $p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t})$ is very different from the importance density $p(\mathbf{z}_t^{(j)}|\mathbf{y}_{1:t-1})$, then the weights tend to concentrate on a few particles, ultimately leading to a high variance when the particle approximation is used to estimate posterior moments. This problem is known as **weight degeneracy** and can be alleviated by using more sophisticated importance densities which take the current observation \mathbf{y}_t into account, for example the so called **auxillary particle filter**. The weights will then no longer have the simple form

weight degeneracy

auxillary particle filter

Bootstrap filter update for general state-space model

$$p(\mathbf{z}_{t-1} | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t-1}) \Rightarrow p(\mathbf{z}_t | \mathbf{y}_{1:t})$$

Input: particles from previous step $\mathbf{z}_{t-1}^{(1)}, \dots, \mathbf{z}_{t-1}^{(M)}$
 measurement \mathbf{y}_t
 control signal \mathbf{u}_t

```

for  $j$  in  $1:M$  do
  Prior propagation
  draw state particle  $\mathbf{z}_t^{(j)} \sim g(\mathbf{z}_t | \mathbf{z}_{t-1}^{(j)})$ 
  Measurement update
  compute unnormalized weight  $\tilde{w}_t^{(j)} = p(\mathbf{y}_t | \mathbf{z}_t^{(j)}, \theta)$ 
end
normalize weights
 $w_t^{(j)} = \tilde{w}_t^{(j)} / \sum_{k=1}^M \tilde{w}_t^{(k)}$  for  $j = 1, \dots, M$ 
Resampling
for  $j$  in  $1:M$  do
   $k_j \sim \text{Cat}(w_t^{(1)}, \dots, w_t^{(M)})$ 
   $\mathbf{z}_t^{(j)} \leftarrow \mathbf{z}_t^{(k_j)}$ 
end
Output:  $M$  draws  $\mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(M)}$  from  $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ .
```

Box 17.5: The bootstrap particle filter update at time t . At time $t = 1$ the prior propagation step is replaced by drawing particles from the prior $p(\mathbf{z}_0)$.

$w_t^{(j)} \propto p(\mathbf{y}_t | \mathbf{z}_t^{(j)})$ as in the bootstrap filter.

A by-product of particle filters is an unbiased estimate of the marginal likelihood $\hat{p}(\mathbf{y}_{1:r} | \boldsymbol{\theta}) = \prod_{t=1}^r \hat{p}(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta})$ where each factor $p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta})$ is approximated by the average of the unnormalized weights $w_t^{(j)}$ at time t : $\hat{p}(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta}) = M^{-1} \sum_{j=1}^M w_t^{(j)}$.

Finally, let us now consider how to sample from the joint filtering posterior $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T})$ using the particle filter. As mentioned above the particle filter implicitly defines a genealogy of particles where for any draw $\mathbf{z}_t^{(j)}$ at time t we can find its ancestor $\mathbf{z}_{t-1}^{(j)}$ at time $t-1$, and in the same way we can trace back the whole ancestral path back to the beginning of time at $t=1$. This is illustrated in Figure 17.18. This means that we effectively sample M whole trajectories $\mathbf{z}_{1:T}^{(j)}$ from the joint filtering posterior $p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T})$ by tracing out the ancestral path of each of the M particles $\mathbf{z}_T^{(j)}$ drawn at the final time step $t=T$ with the bootstrap filter. The weights $w_T^{(j)}$ at the last step of the bootstrap filter therefore gives a weighted particle system approximation to the joint smoothing posterior

$$p(\mathbf{z}_{1:T} | \mathbf{y}_{1:T}) \approx \sum_{j=1}^M w_T^{(j)} \delta_{\tilde{\mathbf{z}}_{1:T}^{(j)}}(\mathbf{z}_{1:T}), \quad (17.24)$$

where $\tilde{\mathbf{z}}_{1:T}^{(j)}$ is the ancestral path of the particle $\mathbf{z}_T^{(j)}$. This is illustrated in Figure (17.18) where the red and yellow trajectories are draws from the joint filtering posterior. However, the particle system approximation to the joint smoothing posterior in (17.8) is not useful in practice since the ancestral paths tend to be traced back to one common ancestor. This lack of particle diversity makes the approximation to the joint smoothing posterior in (17.8) very poor in the early time periods. There are many improved smoothing algorithms in the literature that aim to solve this problem, but we will not go into details here.

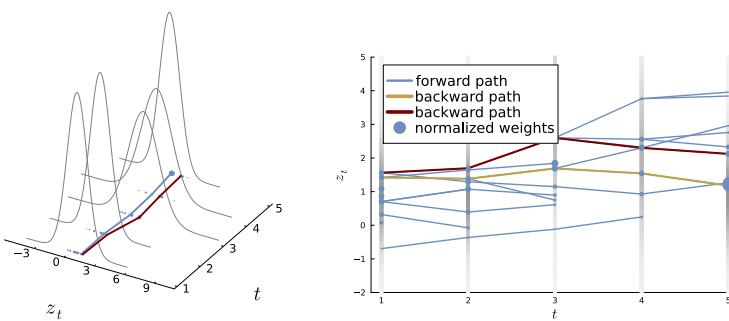


Figure 17.18: Illustrating how a sequence of filtering posteriors densities $p(\mathbf{z}_t | \mathbf{y}_{1:t})$ are approximated by particles (blue dots with size proportional to their weight at time t) and how particles are resampled and producing offsprings (connecting light blue lines). The red and yellow trajectories are samples from the joint filtering posterior $p(\mathbf{z}_t | \mathbf{y}_{1:t})$.

Bibliography

- Andersson, P. G. (2023). The wald confidence interval for a binomial p as an illuminating “bad” example. *The American Statistician*, 77(4):443–448.
- Bernardo, J. M. and Smith, A. F. (2009). *Bayesian theory*, volume 405. John Wiley & Sons.
- Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.
- Chaudhuri, P. and Marron, J. S. (1999). Sizer for exploration of structures in curves. *Journal of the American Statistical Association*, 94(447):807–823.
- Chib, S. (1995). Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321.
- Chib, S. and Jeliazkov, I. (2001). Marginal likelihood from the metropolis–hastings output. *Journal of the American Statistical Association*, 96(453):270–281.
- Dua, D. and Graff, C. (2017). UCI machine learning repository.
- Fanaee-T, H. and Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15.
- Fox, J. and Weisberg, S. (2019). *An R companion to applied regression*. Sage publications.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*, volume 3rd edition. CRC press.
- Geweke, J. (1999). Using simulation methods for bayesian econometric models: inference, development, and communication. *Econometric reviews*, 18(1):1–73.
- Harville, D. A. (1998). Matrix algebra from a statistician’s perspective.

- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Hennig, P., Osborne, M. A., and Kersting, H. P. (2022). *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press.
- Holst, U., Hössjer, O., Björklund, C., Ragnarson, P., and Edner, H. (1996). Locally weighted least squares kernel regression and statistical evaluation of lidar measurements. *Environmetrics*, 7(4):401–416.
- Irony, T. Z. and Singpurwalla, N. D. (1997). Non-informative priors do not exist - a dialogue with José M. Bernardo. *Journal of Statistical Planning and Inference*, 65(1):159–177.
- Jeffreys, H. (1998). *The theory of probability*. OUP Oxford.
- Jin, S., Thulin, M., and Larsson, R. (2017). Approximate bayesianity of frequentist confidence intervals for a binomial proportion. *The American Statistician*, 71(2):106–111.
- Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, 90(430):773–795.
- Lindgren, G. (2012). *Stationary stochastic processes: theory and applications*. CRC Press.
- Lindholm, A., Wahlström, N., Lindsten, F., and Schön, T. B. (2022). *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press.
- Makalic, E. and Schmidt, D. F. (2015). A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182.
- Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate analysis*, 1979. Academic Press Inc.
- Migon, H. S., Gamerman, D., and Louzada, F. (2014). *Statistical inference: an integrated approach*. CRC press.
- Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 541–548. JMLR Workshop and Conference Proceedings.
- O'Hagan, A. (2019). Expert knowledge elicitation: subjective but scientific. *The American Statistician*, 73(sup1):69–81.

- Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349.
- Rajan, R. G. and Zingales, L. (1995). What do we know about capital structure? some evidence from international data. *The journal of Finance*, 50(5):1421–1460.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric regression*. Cambridge university press.
- Smith, M. and Kohn, R. (1996). Nonparametric regression using bayesian variable selection. *Journal of econometrics*, 75(2):317–343.
- Sundberg, R. (2019). *Statistical modelling by exponential families*, volume 12. Cambridge University Press.
- Thrun, S., Burgard, W., and Fox, D. (2006). Probabilistic robotics.
- Villani, M. (2009). Steady-state priors for vector autoregressions. *Journal of Applied Econometrics*, 24(4):630–650.
- Wegmann, B. and Villani, M. (2011). Bayesian inference in structural second-price common value auctions. *Journal of Business & Economic Statistics*, 29(3):382–396.
- Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge MA.

Appendix: Some Mathematical results

A.1 Some calculus

Some special functions

A little combinatorics

The **factorial** $n!$ of an positive integer n is defined as

$$n! = n(n-1) \cdots 2 \cdot 1 \quad (25)$$

The **binomial coefficient**

$$\binom{n}{s} = \frac{n!}{s!(n-s)!} \quad (26)$$

is the number of ways that s elements can be placed in n positions, if the order does not matter.

The **gamma function** has definition

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \quad (27)$$

The **beta function**

$$\text{Beta}(x, y) = \int_0^\infty t^{x-1} (1-t)^{y-1} dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \quad (28)$$

The **modified Bessel function of the first kind** is defined as

$$I_\nu(x) = \sum_{m=0}^{\infty} \frac{1}{m!\Gamma(m+\nu+1)} \left(\frac{x}{2}\right)^{2m+\nu} \quad (29)$$

The **modified Bessel function of the second kind** is defined as

$$K_\nu(x) = \frac{\pi}{2} \frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\nu x)}. \quad (30)$$

Derivatives and optimization

- Derivative of a function $f'(x)$
- Rules for derivatives: sum rule, product rule, chain rule.

- Second derivative $f''(x)$
- Higher order derivatives and the Taylor approximation
- Optimization: solve $f'(x) = 0$ and check $f''(x_{mode}) < 0$ for a maximum.
- Partial derivative $\frac{\partial f}{\partial x_j}$

Limits and Integration

- The limit concept $\lim_{x \rightarrow a} f(x)$
- Riemann integral

A.2 Some linear algebra

This section summarizes some selected results from matrix algebra and multivariate analysis. The results are mostly given without proof, and the reader is referred to for example Harville (1998) for an extensive account or Appendix A in Mardia et al. (1979) for a more condensed treatment. The starred sections are not strictly required for understanding the material in this book, but are widely used results that every statistician should know about.

Vectors, matrices and their products

Let

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix}$$

be a vector with p elements. We always define vectors as *column* vectors. A vector can be turned into a row vector by the **vector transpose** $\mathbf{a}^\top = (a_1, a_2, \dots, a_p)$.

The **dot product** of two vectors \mathbf{a} and \mathbf{b} with the same number elements is defined as

$$\mathbf{a}^\top \mathbf{b} = \sum_{j=1}^p a_j b_j,$$

which is often written as $\mathbf{a} \cdot \mathbf{b}$. Two vectors \mathbf{a} and \mathbf{b} are **orthogonal** (perpendicular) to each other if and only if $\mathbf{a} \cdot \mathbf{b} = 0$; see Figure A.2.

The *Euclidean length*, or *L_2 -norm*, of a vector is defined as

$$\|\mathbf{a}\|_2 = (\mathbf{a}^\top \mathbf{a})^{1/2} = \left(\sum_{j=1}^p a_j^2 \right)^{1/2}.$$

Another common norm is the *L_1 -norm*

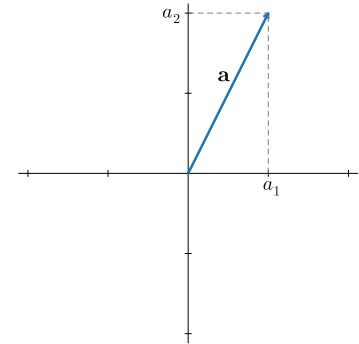


Figure A.21: Geometric illustration of the vector $\mathbf{a} = (a_1, a_2)^\top$.

vector transpose

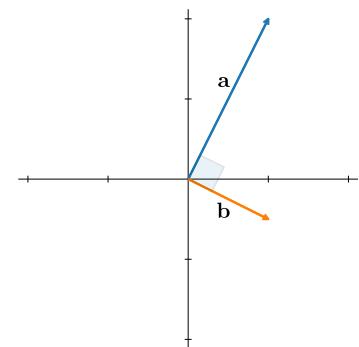


Figure A.22: Geometric illustration of two orthogonal vectors \mathbf{a} and \mathbf{b} .

dot product

orthogonal

L_2 -norm

$$\|\mathbf{a}\|_1 = \sum_{j=1}^p |a_j|.$$

Let \mathbf{A} be a $p \times r$ matrix, i.e. and matrix with p rows and r columns:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1r} \\ a_{21} & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pr} \end{pmatrix}.$$

The **identity matrix** \mathbf{I}_p is the $p \times p$ matrix

$$\mathbf{I}_p = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix},$$

identity matrix

which plays the role of 1 in the world of matrices so that $\mathbf{A}\mathbf{I}_p = \mathbf{I}_p\mathbf{A} = \mathbf{A}$ for any $p \times p$ matrix \mathbf{A} .

The **matrix-vector product** of an $p \times r$ matrix \mathbf{A} and r -element vector $\mathbf{b} = (b_1, b_2, \dots, b_r)^\top$ is

$$\mathbf{Ab} = \begin{pmatrix} \sum_{j=1}^r a_{1j}b_j \\ \sum_{j=1}^r a_{2j}b_j \\ \vdots \\ \sum_{j=1}^r a_{pj}b_j \end{pmatrix}.$$

matrix-vector product

Defining \mathbf{a}_i^\top to be the i th row of \mathbf{A} we can write

$$\mathbf{Ab} = \begin{pmatrix} \mathbf{a}_1^\top \mathbf{b} \\ \mathbf{a}_2^\top \mathbf{b} \\ \vdots \\ \mathbf{a}_p^\top \mathbf{b} \end{pmatrix},$$

where $\mathbf{a}_i^\top \mathbf{b} = \sum_{j=1}^r a_{ij}b_j$ is a simple vector (dot) product.

Similarly, the **matrix-matrix product** of the $p \times q$ matrix \mathbf{A} and the $q \times r$ matrix \mathbf{B} is defined as

$$\mathbf{AB} = \begin{pmatrix} \mathbf{a}_1^\top \mathbf{b}_1 & \mathbf{a}_1^\top \mathbf{b}_2 & \cdots & \mathbf{a}_1^\top \mathbf{b}_r \\ \mathbf{a}_2^\top \mathbf{b}_1 & \mathbf{a}_2^\top \mathbf{b}_2 & \cdots & \mathbf{a}_2^\top \mathbf{b}_r \\ \vdots & & & \\ \mathbf{a}_p^\top \mathbf{b}_1 & \mathbf{a}_p^\top \mathbf{b}_2 & \cdots & \mathbf{a}_p^\top \mathbf{b}_r \end{pmatrix}.$$

matrix-matrix product

Note the the number of columns in \mathbf{A} must equal the number of rows in \mathbf{B} and the end result of the product is a matrix with dimensions

$p \times r$. We use the terminology that **A pre-multiplies B** in the product **AB**, or, equivalently, that **B post-multiplies A**.

The **matrix transpose** of $p \times r$ matrix **A**, denoted by \mathbf{A}^\top , is the $r \times p$ matrix where the i th column is the i row of **A**. Let **A** be a matrix with p rows and r columns

$$\mathbf{A}^\top = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{1p} \\ a_{12} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1p} & a_{2p} & \cdots & a_{rp} \end{pmatrix}.$$

matrix transpose

Determinant and inverse matrix

The **determinant** of a square 2×2 matrix **A** is the scalar (i.e. single number)

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \quad (31)$$

determinant

and for a 3×3 matrix

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}, \quad (32)$$

and increasingly more complex expressions for higher dimensional matrices. The exact expressions are less important here however. It is enough to remember that a determinant of a matrix **A** is a scalar that represent the *volume* of the matrix, in the sense that the absolute value of the determinant of **A** is the volume of a parallelepiped formed by the columns of **A**; see Figure A.2 for an illustration.

We will most often see the determinant of a covariance matrix Σ for a random vector **x**, where $|\Sigma|$ can then be taken as a measure of *total variance* of **x**. Let us for concreteness consider the bivariate case with a bivariate normal with mean vector $\mu = (\mu_1, \mu_2)$ and covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

which has determinant $|\Sigma| = \sigma_1^2\sigma_2^2(1 - \rho^2)$. Consider first the case with no correlation, $\rho = 0$, where the total variance is $|\Sigma| = \sigma_1^2\sigma_2^2$. As $\rho \rightarrow 1$ and the variables are increasing correlated and the total variance decreases. When $\rho = 1$ the two variables are perfectly correlated and the total variance is zero. The same is true when $\rho \rightarrow -1$ where the variables are perfectly negatively correlated, the total variance becomes smaller and smaller.

Some rules of determinants are worth noting. First, $|c\mathbf{A}| = c^p|\mathbf{A}|$ for any scalar c and $p \times p$ matrix **A**. Second, the determinant of a

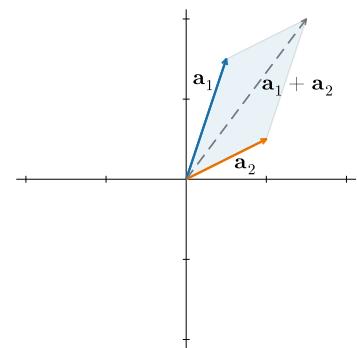


Figure A.23: Geometric illustration of the determinant as the area of the parallelogram formed by the 2×2 matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2)$.

diagonal matrix is just the product of the diagonal elements

$$\begin{vmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{pp} \end{vmatrix} = a_{11}a_{22} \cdots a_{pp}.$$

The same is true for a lower diagonal matrix, i.e. a matrix where all the elements above the diagonal are zero, but some elements on the diagonal and/or below the diagonal may be non-zero. Finally, for the product of two square matrices \mathbf{A} and \mathbf{B} we have

$$|\mathbf{AB}| = |\mathbf{A}| \cdot |\mathbf{B}|. \quad (33)$$

The same type of result holds for a product of three matrices $|\mathbf{ABC}| = |\mathbf{A}| \cdot |\mathbf{B}| \cdot |\mathbf{C}|$ and so on.

The **matrix inverse** of a square $p \times p$ matrix \mathbf{A} is the matrix \mathbf{A}^{-1} such that

$$\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = I_p. \quad (34)$$

Not every square matrix has an inverse, but when it exists it is unique. A sufficient and necessary condition for a square matrix \mathbf{A} to have an inverse is that its column are linearly independent, i.e. that $\sum_{j=1}^p \alpha_j \mathbf{a}_j = \mathbf{0}$ only for $\alpha_1 = \alpha_2 = \dots = \alpha_p = 0$, where \mathbf{a}_j is the j th column of \mathbf{A} and $\mathbf{0}$ is the zero vector. Invertible matrices are also called non-singular. Here are two useful rules for inverses:

$$|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|}$$

and if both \mathbf{A} and \mathbf{B} are invertible then

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1},$$

where you should note the reverse order of the matrices. The same type of result holds for a product of three matrices $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$.

The **matrix trace** of a matrix \mathbf{A} is simply the sum of its diagonal elements

$$\text{tr}(\mathbf{A}) = \sum_{j=1}^p a_{jj}. \quad (35)$$

The trace has the following circular property

$$\text{tr}(\mathbf{ABC}) = \text{tr}(\mathbf{CAB}) = \text{tr}(\mathbf{BCA}), \quad (36)$$

for any square matrices \mathbf{A}, \mathbf{B} and \mathbf{C} with the same dimensions.

*Partitioned matrices**

Consider a *partitioned matrix* of dimensions $p \times p$

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}, \quad (37)$$

where \mathbf{A}_{11} is of dimensions $p_1 \times p_1$, \mathbf{A}_{22} is of dimensions $p_2 \times p_2$, \mathbf{A}_{12} and \mathbf{A}_{21} are of dimensions $p_1 \times p_2$ and $p_2 \times p_1$ respectively. Hence, $p = p_1 + p_2$. The determinant can be then be expressed

$$|\mathbf{A}| = |\mathbf{A}_{11}| |\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}| = |\mathbf{A}_{22}| |\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}|.$$

and the inverse

$$\mathbf{A}^{-1} = \begin{pmatrix} \mathbf{A}^{(11)} & -\mathbf{A}^{(11)}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}^{(11)} & (\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} \end{pmatrix},$$

where $\mathbf{A}^{(11)} = (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1}$.

*Linear transformation, eigendecomposition and principal components**

Consider a linear transformation $\mathbf{y} = \mathbf{m} + \mathbf{Ax}$ from \mathbf{x} to \mathbf{y} , where \mathbf{y} and \mathbf{m} are p -dimensional vectors, \mathbf{x} is an q -dimensional vector, and \mathbf{A} is a $p \times q$ matrix. If \mathbf{x} is a random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ then

$$\mathbb{E}(\mathbf{y}) = \mathbf{m} + \mathbf{A}\boldsymbol{\mu} \quad (38)$$

$$\mathbb{V}(\mathbf{y}) = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top \quad (39)$$

Let $p = 1$ so that $\mathbf{A} = \mathbf{a}^\top$ is a r -dimensional row vector. Then $y = m + \mathbf{a}^\top \mathbf{x} = m + \sum_{i=1}^r a_i x_i$ is a scalar, and $\mathbb{V}(y) = \mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a}$. Since we require a variance to be positive we must require that the covariance matrix $\boldsymbol{\Sigma}$ satisfies $\mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a} > 0$ for all $\mathbf{a} \neq \mathbf{0}$. We say that $\boldsymbol{\Sigma}$ must be **positive definite**. A matrix $\boldsymbol{\Sigma}$ is positive definite if and only if $|\boldsymbol{\Sigma}| > 0$. If we allow that the variance can also be exactly zero, then we require $\boldsymbol{\Sigma}$ to be positive semidefinite, sometimes abbreviated by psd or p.s.d.

An **eigenvector** \mathbf{v} of an invertible matrix \mathbf{A} is a vector that keeps its direction when transformed by \mathbf{A} , i.e.

$$\mathbf{Av} = \lambda \mathbf{v},$$

where λ is the **eigenvalue** associated with the eigenvector \mathbf{v} . Note how the transformation only leads to a scaling of \mathbf{v} by λ , but the direction of the vector remains the same. A non-singular $p \times p$ matrix \mathbf{A} has p linearly independent eigenvectors, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p$ each associated with its own eigenvalue $\lambda_1, \lambda_2, \dots, \lambda_p$. Eigenvectors are normalized to have unit length, i.e. $\mathbf{v}_j^\top \mathbf{v}_j = 1$ for $j = 1, \dots, p$ and to be

positive definite

eigenvector

eigenvalue

orthogonal to each other, i.e. $\mathbf{v}_i^\top \mathbf{v}_j = 0$ for $i \neq j$. We can therefore collect all eigenvectors into a $p \times p$ orthonormal matrix $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_p)$ with the property $\mathbf{V}^\top \mathbf{V} = \mathbf{V}\mathbf{V}^\top = \mathbf{I}_p$; note that the inverse of an orthonormal matrix is simply its transpose. We can now write

$$\mathbf{A}\mathbf{V} = \mathbf{V}\Lambda, \quad (40)$$

where $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_p)$ is a diagonal matrix of eigenvalues. We therefore obtain the **spectral decomposition** of the invertible matrix \mathbf{A} by post-multiplying both sides of (40) with \mathbf{V}^\top (since $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_p$)

spectral decomposition

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top. \quad (41)$$

The spectral decomposition gives us connection between the determinant and inverse of a matrix and its eigenvalues and eigenvectors. The determinant can be written

$$|\mathbf{A}| = |\mathbf{V}\Lambda\mathbf{V}^\top| = |\mathbf{V}||\Lambda||\mathbf{V}^\top| = |\Lambda||\mathbf{V}\mathbf{V}^\top| = \prod_{j=1}^p \lambda_j,$$

since the determinant of a diagonal matrix is the product of its diagonal elements and $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_p$ so $|\mathbf{V}\mathbf{V}^\top| = 1$. Given that a matrix is positive definite if its determinant is non-zero, this shows that a matrix is positive definite if and only if all of its eigenvalues are positive.

Since the inverse of an orthonormal matrix is its transpose, we can use the product rule for inverses to express the inverse of \mathbf{A} as

$$\mathbf{A}^{-1} = (\mathbf{V}^\top)^{-1}\Lambda^{-1}\mathbf{V}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}^\top,$$

and $\Lambda^{-1} = \text{Diag}(1/\lambda_1, \dots, 1/\lambda_p)$. There are more general decompositions of matrices, also for non-square and non-invertible matrices, the most famous being the singular value decomposition (Harville, 1998).

Finally, using the circular property of the trace in (36), we see that the trace of matrix is the sum of its eigenvalues

$$\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{V}\Lambda\mathbf{V}^\top) = \text{tr}(\mathbf{V}^\top\mathbf{V}\Lambda) = \text{tr}(\mathbf{I}_p\Lambda) = \text{tr}(\Lambda) = \sum_{j=1}^p \lambda_j.$$

Consider now the spectral value decomposition $\Sigma = \mathbf{V}\Lambda\mathbf{V}^\top$ on a covariance matrix Σ of a random vector \mathbf{x} . The linear transformation $\mathbf{y} = \mathbf{V}^\top \mathbf{x}$ has an interesting covariance matrix

$$\mathbb{V}(\mathbf{y}) = \mathbf{V}^\top \Sigma \mathbf{V} = \mathbf{V}^\top (\mathbf{V}\Lambda\mathbf{V}^\top) \mathbf{V} = \Lambda. \quad (42)$$

Hence, the new variables in $y_j = \mathbf{v}_j^\top \mathbf{x}$ for $j = 1, \dots, p$ are uncorrelated and have the eigenvalues as variances: $\mathbb{V}(y_j) = \lambda_j$. These variables are called the **principal components** of \mathbf{x} . If we order the eigenvalues

principal components

in descending order $\lambda_1 \geq \dots \geq \lambda_p$ then the first principal component $y_1 = \mathbf{v}_1^\top \mathbf{x}$ is the linear combination of the variables in \mathbf{x} with maximal variance, the second principal component $y_2 = \mathbf{v}_2^\top \mathbf{x}$ is the linear combination with maximal variance subject to being uncorrelated with y_1 and so on. Summarizing a possibly high-dimensional correlated \mathbf{x} with the $r < p$ largest principal components is therefore a useful way to compress the data while retaining most of the variance. Figure A.24 illustrates the transformation of sampled data into uncorrelated principal components.

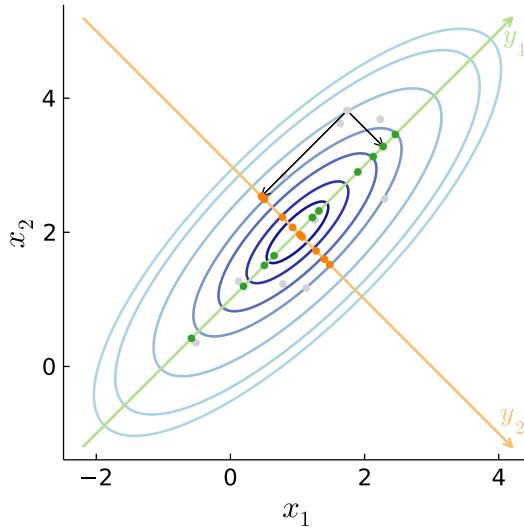


Figure A.24: Illustration of principal components from data points sampled from a multivariate normal distribution with mean $\mu = (1, 2)^\top$ and correlation $\rho = 0.8$. The sampled data points are shown in light gray and their projections onto the first principal components axis (y_1) are shown as green points and as orange points when projected against the second principal component axis (y_2); this projection is illustrated by arrows for one of the data points. The larger variability of the green points along the y_1 axis compared to the variability of the orange points along the y_2 is reflected in the eigenvalues $\lambda_1 = 1.8 > \lambda_2 = 0.2$.

Matrix powers and the Cholesky decomposition*

The spectral decomposition is useful for defining powers of a matrix. Let \mathbf{A} be a square non-singular matrix with spectral decomposition $\mathbf{V}\Lambda\mathbf{V}^\top$. Then since \mathbf{V} is orthonormal we have

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top\mathbf{V}\Lambda\mathbf{V}^\top = \mathbf{V}\Lambda^2\mathbf{V}^\top,$$

where $\Lambda^2 = \text{Diag}(\lambda_1^2, \dots, \lambda_p^2)$. Continuing by multiplying with additional \mathbf{A} factors we have for any positive integer k the **matrix power**

$$\mathbf{A}^k = \mathbf{V}\Lambda^k\mathbf{V}^\top.$$

We can extend this to any power k , not necessarily a positive integer, and in particular to $k = 1/2$ to define a **matrix square root** $\mathbf{A}^{1/2} = \mathbf{V}\Lambda^{1/2}\mathbf{V}^\top$ with the property $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{A}$. This construction can be used to simulate $\mathbf{x} \sim N(\mu, \Sigma)$ by

$$\mathbf{x} = \mu + \Sigma^{1/2}\mathbf{z}, \quad (43)$$

matrix power

matrix square root

where \mathbf{z} is a p -dimensional vector with independent standard normal variables. Since linear transformations of normal variables are normal, \mathbf{x} is multivariate normal with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbb{V}(\mathbf{x}) = \boldsymbol{\Sigma}^{1/2}\mathbb{V}(\mathbf{z})\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}^{1/2}\mathbf{I}_p\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}$ as required. The spectral decomposition is just one way of defining a matrix square root. Another commonly used matrix square root is the **Cholesky decomposition**

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top, \quad (44)$$

Cholesky decomposition

where

$$\mathbf{L} = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{p1} & l_{p2} & \cdots & l_{p,p-1} & l_{pp} \end{pmatrix}$$

is a lower triangular matrix. The Cholesky square root can equally well be used for multivariate normal simulation: if $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$ then $\mathbf{x} = \boldsymbol{\mu} + \mathbf{L}\mathbf{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where again \mathbf{z} is a p -dimensional vector with independent standard normal variables. The Cholesky decomposition makes it possible to compute the multivariate normal density cheaply since

$$\begin{aligned} p(\mathbf{x}) &= |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \\ &= (2\pi)^{-p/2} |\mathbf{L}\mathbf{L}^\top|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top (\mathbf{L}\mathbf{L}^\top)^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \\ &= (2\pi)^{-p/2} |\mathbf{L}|^{-1} \exp\left(-\frac{1}{2}\mathbf{y}^\top \mathbf{y}\right), \end{aligned} \quad (45)$$

where $\mathbf{y} = \mathbf{L}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ and $|\mathbf{L}| = \prod_{j=1}^p l_{jj}$ since \mathbf{L} is lower triangular. We can compute $\mathbf{y} = \mathbf{L}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ without explicitly inverting \mathbf{L} by solving the system of equations $\mathbf{Ly} = \mathbf{x} - \boldsymbol{\mu}$ for \mathbf{y} . Since \mathbf{L} is lower triangular this can be solved quickly using forward/backward substitution. Note that we have used several of the above mentioned results for determinants and inverses in (45), so verifying this derivation is a useful exercise.

Vector differentiation*

Let $f(\mathbf{x})$ be a scalar valued function of an p -dimensional vector \mathbf{x} . The gradient of $f(\mathbf{x})$ with respect to \mathbf{x} is the p -dimensional vector with partial derivatives

$$\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_p} f(\mathbf{x}) \end{pmatrix}$$

The gradient is sometimes written $\nabla_{\mathbf{x}} f(\mathbf{x})$. For a linear function $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ for some p -dimensional vector \mathbf{a} the gradient is easily seen to be

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{a}^\top \mathbf{x} = \mathbf{a},$$

matching up with the one-dimensional case $\frac{d}{dx} ax = a$. For a quadratic function $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x}$ for some square matrix \mathbf{A} , often called a quadratic form, we have the gradient

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x},$$

which also matches the one-dimensional case $\frac{d}{dx} ax^2 = 2ax$.

Consider now a *multi-output* function $\mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x}))^\top$ with p -dimensional output \mathbf{y} and q -dimensional input \mathbf{x} . The $p \times q$ matrix of partial derivatives

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f_1(\mathbf{x}) & \frac{\partial}{\partial x_2} f_1(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_q} f_1(\mathbf{x}) \\ \vdots & & & \\ \frac{\partial}{\partial x_1} f_p(\mathbf{x}) & \frac{\partial}{\partial x_2} f_p(\mathbf{x}) & \cdots & \frac{\partial}{\partial x_q} f_p(\mathbf{x}) \end{pmatrix}.$$

is called the **Jacobian matrix**. For a linear multi-output function $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ we have $\frac{\partial}{\partial \mathbf{x}} \mathbf{A}\mathbf{x} = \mathbf{A}$.

Recall that the **chain rule** for differentiation of the function composition $f(x) = g(h(x))$ is the product of the so called outer and inner derivatives: $\frac{d}{dx} f(x) = \frac{d}{dz} g(z) \frac{d}{dx} h(x)$. The chain rule for a multi-dimensional function composition $f(\mathbf{x}) = g(h(\mathbf{x}))$, where $h : \mathbb{R}^p \rightarrow \mathbb{R}^q$ and $g : \mathbb{R}^q \rightarrow \mathbb{R}$, is similar

$$\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \left(\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}) \right)^\top \frac{\partial}{\partial \mathbf{z}} g(\mathbf{z}),$$

where $\mathbf{z} = h(\mathbf{x})$ is in general a mapping $\mathbf{x} \rightarrow \mathbf{z}$ from \mathbb{R}^p to \mathbb{R}^q , so that $\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x})$ is a $q \times p$ Jacobian matrix when both $p > 1$ and $q > 1$.

As an example on how to use the above rules for differentiation, consider deriving the least squares estimator in linear regression obtained by minimizing the residual sum of squares

$$Q(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{e}(\boldsymbol{\beta})^\top \mathbf{e}(\boldsymbol{\beta}),$$

where $\mathbf{e}(\boldsymbol{\beta}) = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ is the vector of residuals. The least squares estimate is therefore the solution to $\frac{\partial}{\partial \boldsymbol{\beta}} Q(\boldsymbol{\beta}) = \mathbf{0}$ where

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q(\boldsymbol{\beta}) = \left(\frac{\partial}{\partial \boldsymbol{\beta}} \mathbf{e}(\boldsymbol{\beta}) \right)^\top \frac{\partial}{\partial \mathbf{e}} \mathbf{e}^\top \mathbf{e} = \left(\frac{\partial}{\partial \boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)^\top 2\mathbf{e} = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Hence the least squares estimator is the solution to $\mathbf{X}^\top \mathbf{y} = \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$. If the columns of \mathbf{X} are linearly independent then the inverse $(\mathbf{X}^\top \mathbf{X})^{-1}$ exist and we can multiply both sides with it to get the least squares solution $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$.

Jacobian matrix

chain rule

A.3 Taylor approximation

The Taylor approximation is a tailored polynomial approximation of a function $f(x)$. The **Taylor series** of an infinitely differentiable function $f(x)$ is

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x - a)^k, \quad (46)$$

where $f^{(k)}(a)$ is the k th derivative of f evaluated in the point $x = a$.

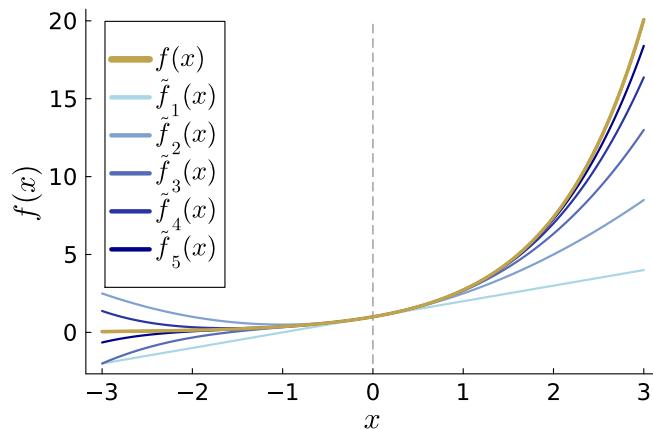
The classical example of a Taylor series is that of the exponential function. The derivatives of the exponential function $f(x) = e^x$ are the exponential function itself, i.e. $f^{(k)}(x) = e^x$ for all k . The Taylor series expansion of the exponential function around $x = 0$ is therefore

$$\begin{aligned} e^x &= e^0 + \frac{1}{1!} e^0 (x - 0) + \frac{1}{2!} e^0 (x - 0)^2 + \frac{1}{3!} e^0 (x - 0)^3 + \dots \\ &= 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \\ &= \sum_{k=0}^{\infty} \frac{x^k}{k!}. \end{aligned}$$

A **Taylor approximation** of $f(x)$ uses only a small number of terms in the Taylor series

$$f(x) \approx \sum_{k=0}^K \frac{f^{(k)}(a)}{k!} (x - a)^k, \quad (47)$$

for some finite and typically small K . Figure A.31 shows how the Taylor approximation of e^x improves as higher order polynomial terms are included in the approximation. Taylor's theorem can be used to bound the approximation error of a k th order Taylor approximation using the $(k+1)$ th derivative of the function.



Taylor series

Taylor approximation

Figure A.31: Taylor approximation of the exponential function for different polynomial orders.

The Taylor expansion is a local approximation around the expansion point $x = a$, and the approximation is most accurate in a neighborhood around a . This point is illustrated in Figure A.32 where the function $\log(1 + x)$ is well approximated only in the neighborhood around the expansion point $x = 0$.

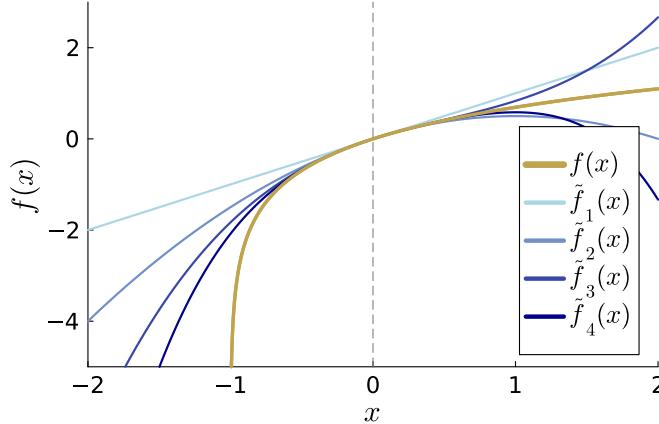


Figure A.32: Taylor approximation of $\log(1 + x)$ around $x = 0$ for different approximation orders.

There is a multi-dimensional version of the Taylor approximation for functions $f(\mathbf{x}) = f(x_1, \dots, x_d)$ of several variables. We will only make use of the first and second order versions. The second order Taylor approximation of the function $f(\mathbf{x})$ around the point $\mathbf{x} = \mathbf{a}$ is

$$f(\mathbf{x}) \approx f(\mathbf{a}) + \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{a}}(\mathbf{x} - \mathbf{a}) + \frac{1}{2}(\mathbf{x} - \mathbf{a})^\top \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}|_{\mathbf{x}=\mathbf{a}}(\mathbf{x} - \mathbf{a}),$$

where

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_d} \right),$$

is the **gradient** row vector with partial derivatives of $f(\mathbf{x})$ with respect to each of the input coordinates x_1, \dots, x_d . The notation $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{a}}$ means that this vector of derivatives is evaluated in the point $\mathbf{x} = \mathbf{a}$. The $d \times d$ matrix $\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}$ is the **Hessian** matrix

$$\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_d} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_d} \\ \vdots & & \ddots & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_d \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_d \partial x_2} & & \frac{\partial^2 f(\mathbf{x})}{\partial x_d^2} \end{pmatrix},$$

with second derivatives $\frac{\partial^2 f(\mathbf{x})}{\partial x_j^2}$ and cross-derivatives $\frac{\partial^2 f(\mathbf{x})}{\partial x_j \partial x_k}$.

To see the multidimensional Taylor approximation in action, consider the following two-dimensional function

$$f(x_1, x_2) = \exp(x_1) \sin(x_2).$$

gradient

Hessian

To compute a second order Taylor approximation around $\mathbf{x} = (0, 0)^\top$ we need to compute the gradient vector and Hessian matrix. The gradient vector is

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \left(\exp(x_1) \sin(x_2), \exp(x_1) \cos(x_2) \right),$$

which evaluates to $(0, 1)$ at $\mathbf{x} = (0, 0)^\top$. The Hessian matrix is

$$\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} = \begin{pmatrix} \exp(x_1) \sin(x_2) & \exp(x_1) \cos(x_2) \\ \exp(x_1) \cos(x_2) & -\exp(x_1) \sin(x_2) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

at $\mathbf{x} = (0, 0)^\top$. The second order Taylor approximation is therefore

$$f(x_1, x_2) \approx 0 + (0, 1)(x_1, x_2)^\top + \frac{1}{2}(x_1, x_2)^\top \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} (x_1, x_2) = x_2 + 2x_1 x_2.$$

Figure A.33 plots the second order Taylor approximation of $\exp(x_1) \sin(x_2)$.

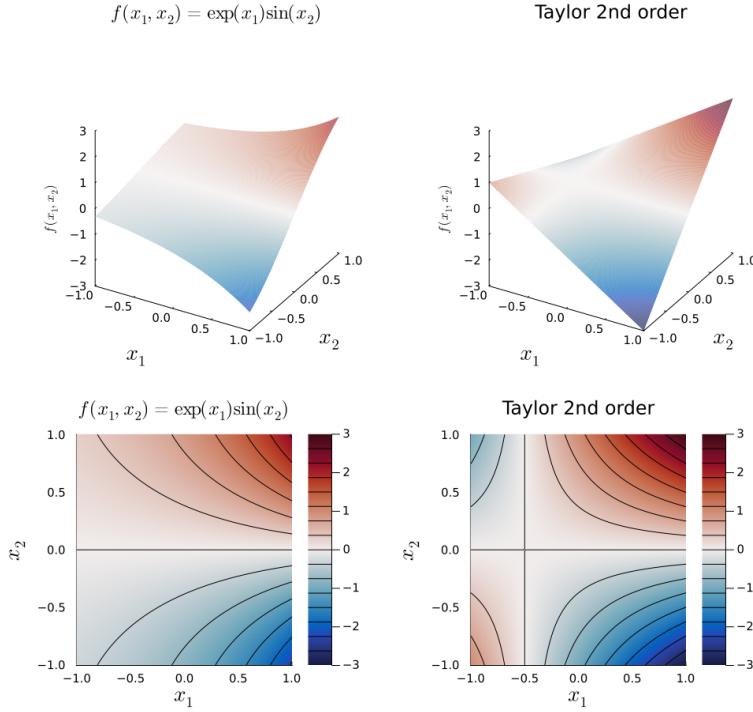


Figure A.33: Taylor approximation of $f(x_1, x_2) = \exp(x_1) \sin(x_2)$ around $\mathbf{x} = (0, 0)$. The graphs in the first row show function surface plots and the second row displays corresponding heatmaps and contours of the functions.

Index

- L_1 -norm, 342
 L_2 -norm, 342
 \mathcal{M} -closed, 265
 \mathcal{M} -open, 271
- action, 117
automatic differentiation, 135
autoregressive model, 80, 181
auxillary particle filter, 333
- basis functions, 231
batch learning, 37
Bayes estimator, 32
Bayes factor, 266
Bayes' theorem, 21
Bayesian coverage property, 47
Bayesian histogram, 258
Bayesian Lasso, 225
Bayesian updating, 24
Bayesian variable selection, 278
Bernoulli distribution, 15
Bernoulli trials, 15
Bernstein-von Mises theorem, 130
Beta distribution, 27
beta function, 341
bias, 91
bike share dataset, 101
binary response variable, 141
binomial coefficient, 341
Binomial distribution, 18
binomial regression, 179
Birnbaum's theorem, 49
block Gibbs sampler, 172
bootstrap filter, 333
- Categorical data, 65
central limit theorem, 65
chain rule, 350
Cholesky decomposition, 349
- class-conditional distribution, 161
classes, 141
coefficient of variation, 65
component allocation variable, 242
conditional probability, 21
confidence interval, 42
conjugate prior, 29, 41
continuous random variables, 17
control variables, 314
convergence in distribution, 65
convergence in probability, 63
covariance function, 286
covariance kernel, 286
covariates, 91
credibility interval, 42
cross-sectional, 98
- data generating process, 265
decision making under uncertainty, 117
degrees of freedom for linear fits, 236
dependent observations, 79
determinant, 344
digamma function, 132
Dirac delta function, 332
Dirichlet distribution, 67
Dirichlet process, 259
discrete finite mixtures, 239
discrete mixture, 240
discriminative model, 161
dot product, 342
dummy variables, 98
dutch book argument, 21
- eBayCoin dataset, 39
effective sample size, 170, 195
eigenvalue, 346
eigenvector, 346
empirical Bayes, 299
- equal tail credibility interval, 42
equivariance, 93
estimate, 17
evidence, 266
exponential family, 51
- factorial, 341
Factorization criterion, 50
features, 91
filtering posterior, 315
Fisher information, 75
Fisher information matrix, 76
fitted values, 92
forward filtering backward sampling algorithm, 325
forward selection, 278
frequentist probability, 20
full conditional posterior, 165
- Gamma distribution, 38
gamma function, 341
Gaussian kernel, 286
Gaussian process, 286
Gaussian process logistic regression, 303
Gaussian process regression, 283
generalized linear models, 157
generative model, 161
GLM, 157
global shrinkage, 82
global-local shrinkage prior, 226
gradient, 352
- Hessian, 352
heteroscedastic, 158
hierarchical prior, 83
Highest Posterior Density (HPD) region, 42
histogram estimator, 255

histogram probability model, 256
homoscedastic, 91
horseshoe prior, 226
hyperparameters, 33

identity matrix, 343
iid, 14
imaginary prior sample, 29
importance sampling, 192
improper prior, 89
independent and identically distributed, 14
incidence rate ratio, 154
inefficiency factor, 170
intercept, 91
Internet speed dataset, 35, 61, 63
intractable posterior, 126
invariant prior, 87
inverse Gamma distribution, 59
inverse Wishart distribution, 73

Jacobian matrix, 350
Jeffreys' prior, 87
joint posterior distribution, 57
joint smoothing distribution, 325

K-fold cross-validated LPS, 277
Kalman gain, 319
knots, 232

L₂ regularization, 213
lag length, 82
lagged value, 80, 181
Laplace distribution, 222
Lasso estimator, 222
latent variables, 309
law of iterated expectation, 112
law of large numbers, 63
law of total probability, 22
law of total variance, 112
least squares estimator, 92
length scale, 286
license, 2
Lidar dataset, 296
likelihood function, 15
Likelihood principle, 49
likelihood surface, 57
lin-lin utility, 121
linear Gaussian regression model, 91
linear Gaussian state-space model, 313

linear predictor, 157
linear utility, 121
link function, 157
local level model, 311
local linear spline, 232
local polynomial spline basis, 232
local quadratic spline, 232
log predictive score, 275
log-normal distribution, 99
logistic function, 143
Logistic regression, 143
long-run properties, 18
longitudinal, 98

marginal likelihood, 265
marginalization, 58
Markov Chain, 115
Markov process, 116
matrix inverse, 345
matrix power, 348
matrix square root, 348
matrix trace, 345
matrix transpose, 344
matrix-matrix product, 343
matrix-vector product, 92, 343
maximin rule, 119
maximum likelihood estimator, 17
mean square convergence, 290
Mean-square continuity, 290
Mean-square convergence, 290
Mean-square differentiability, 290
measurement errors, 313
measurement variables, 310
mixture components, 239
mixture of experts, 254
mixture of linear Gaussian regressions, 253
mixture of normals, 239
mixture of Poisson, 248
mixture weights, 239
mobile phone survey data, 66
modified Bessel function of the first kind, 341
modified Bessel function of the second kind, 341
Monte Carlo estimator, 188
multi-class, 66
Multi-class classification, 142
multicollinearity, 99
multinomial distribution, 66
multivariate normal distribution, 71

Multivariate student-*t*, 96
natural parameter, 51
negative binomial distribution, 48, 111
negative binomial regression, 159
negative class, 141
Newton's method, 135
non-identified, 150
nonparametric regression, 283
nugget, 288
nuisance parameters, 58

observed information, 75
observed information matrix, 76
odds ratio, 144
one-hot encoding, 66, 98
online learning, 36
optimal Bayesian decision, 119
order statistics, 224
ordinal data, 66
orthogonal, 342
outliers, 104
over-dispersion, 159
overfitting, 211

parameter space, 14
particle filter, 332
particle weights, 332
particles, 332
pdf, 17
penalizing, 213
percentile, 122
personal degree of belief, 20
piecewise constant basis function, 232
point estimate, 121
point estimation, 17
point mass distribution, 278, 332
point prediction, 110
Poisson distribution, 38
Poisson regression, 152
polygamma function, 132
polynomial regression, 231
positive class, 141
positive definite, 346
posterior, 22
posterior density, 23
posterior draws, 62
posterior expected utility, 119
posterior median, 121
posterior mode, 121

- precision, 34
- predictive distribution, 109
- predictive interval, 110
- principal components, 347
- prior, 22
- prior density, 23
- prior elicitation, 27
- prior hyperparameters, 28
- prior inclusion probability, 280
- prior predictive distribution, 84, 266, 272
- probability density function, 17
- probability mass function (pmf), 15
- Probit regression, 143, 173
- proposal distribution, 191
- Pólya-Gamma, 178
- quadratic utility, 121
- reference category, 99
- reference class, 151
- reference prior, 89
- regression, 91
- regression coefficients, 91
- Regularization priors, 82
- reinforcement learning, 314
- Rejection sampling, 197
- residuals, 92
- response variable, 91
- ridge regression, 213, 214
- salaries dataset, 97
- sampling distribution, 18
- sampling variance, 18
- scaled inverse chi-squared distribution, 59
- self-normalized importance sampling, 195
- sensitivity, 21
- separation principle, 120
- sequential learning, 36
- shrinkage factor, 214
- shrinking, 214
- simulation consistent, 63
- smoothness beliefs, 82
- SpamBase dataset, 29
- sparse model, 215
- specificity, 21
- spectral decomposition, 347
- spike-and-slab, 278
- squared exponential kernel, 286
- state innovations, 313
- state variables, 309
- stationary, 80
- steady-state form, 80
- stochastic process, 79
- student- t distribution, 60
- subjective consensus, 25
- subjective probability, 19
- Sufficiency principle, 50
- Sufficient statistic, 50
- summary statistics, 31
- target distribution, 192
- Taylor approximation, 351
- Taylor series, 351
- time series, 79
- titanic dataset, 147
- trigamma function, 132
- truncated normal distribution, 175
- unbiased, 18
- underfitting, 211
- uniform distribution, 28
- uniform distribution on the unit simplex, 67
- unit information prior, 97
- unit simplex, 67
- utility function, 118
- vector transpose, 342
- von Mises, 140
- weight degeneracy, 333
- weighted particle system, 332
- weights, 91
- zero sample prior, 86
- zero-inflated Poisson, 248
- zero-inflation, 248
- zero-one utility, 121