Computational Statistics
Spring 2013

Petri Koistinen
Department of Mathematics and Statistics
University of Helsinki

Chapter 1

Introduction

This course gives an overview of computational methods which are useful in Bayesian statistics. Some of the methods (such as stochastic simulation or EM algorithm) are useful also for statisticians who follow the frequentist approach to inference.

1.1 Bayesian statistics: the basic components

Suppose we are going to observe **data** y in the form of a vector $y = (y_1, \dots, y_n)$. Before the observation takes place, the values $y_1, \dots, y_n$ are uncertain (due to measurement errors, the natural variation of the population or due to some other reason). To allow for this uncertainty, we consider y to be the observed value of a random vector $Y = (Y_1, \dots, Y_n)$.

We consider a **parametric model** for the distribution of Y: the distribution of Y is governed by a parameter Θ which is unknown and cannot be directly observed. Usually there are several (scalar) parameters, and then Θ is actually a vector. If $\Theta = \theta$, then the vector Y has the distribution with density

$$y \mapsto f_{Y|\Theta}(y \mid \theta). \tag{1.1}$$

This is called the **sampling distribution**. Having observed the data $Y = y$, the function

$$\theta \mapsto f_{Y|\Theta}(y \mid \theta)$$

(considered as a function of θ and with y equal to the observed value) is called the **likelihood function** (but often multiplicative constants are omitted from the likelihood).

In Bayesian statistics both observables and parameters are considered random. Bayesian inference requires that one sets up a a joint distribution for the data and the parameters (and perhaps other unknown quantities such as future observations). If the data and the parameter are jointly continuously distributed, then the density of the joint distribution can be written in the form

$$(y, \theta) \mapsto f_{Y,\Theta}(y, \theta) = f_{Y|\Theta}(y \mid \theta)\, f_\Theta(\theta),$$

where f_Θ is the density of the marginal distribution of Θ, which is called the **prior distribution**. The prior distribution represents the statistician's uncertainty about plausible values of the parameter Θ before any data has been observed.

Having observed the data $Y = y$, the statistician constructs the conditional distribution of Θ given $Y = y$, which is called the **posterior distribution**. This the result of the inference. The posterior distribution summarizes the statistician's knowledge of the parameter after the data has been observed. The main goal of Bayesian inference is to gain an understanding of the posterior distribution.

Using **Bayes' rule** (Bayes' theorem) of elementary probability theory, one can derive the posterior in a straightforward way from the prior and the likelihood,

$$\theta \mapsto f_{\Theta|Y}(\theta \mid y) = \frac{f_{Y,\Theta}(y,\theta)}{f_Y(y)} = \frac{f_{Y|\Theta}(y \mid \theta)\, f_\Theta(\theta)}{\int f_{Y|\Theta}(y \mid t)\, f_\Theta(t)\, \mathrm{d}t}. \tag{1.2}$$

Here f_Y, the density of the marginal distribution of Y, has been expressed by integrating the variable θ out from the density $f_{Y,\Theta}(y,\theta)$ of the joint distribution.

Notice that the posterior density is obtained, up to a constant of proportionality depending on the data, by multiplying the prior density by the likelihood,

$$f_{\Theta|Y}(\theta \mid y) \propto f_\Theta(\theta)\, f_{Y|\Theta}(y \mid \theta).$$

Once the full probability model has been set up, the formula of the posterior density is therefore available immediately, except for the nuisance that the normalizing constant $1/f_Y(y)$ is sometimes very hard to determine.

1.2 Remarks on notation

In Bayesian statistics one rarely uses as exact notation as we have been using up to now.

- It is customary to blur the distinction between a random variable and its observed (or possible) value by using the same symbol in both cases. This is especially handy, when the quantity is represented by such a lower-case Greek character which does not posess a useful upper-case version.

- It is customary to use the terms "distribution" and "density" interchangeably, and to use the same notation for density functions of continuous distributions and probability mass functions of discrete distributions.

- When the statistical model is complex, it very soon becomes cumbersome to differentiate all the different densities in question by subscripts. An alternative notation is to introduce a different symbol for each of the distributions of interest, e.g., in the style

  $$h(y,\theta) = g(\theta)\, f(y \mid \theta) = m(y)\, p(\theta \mid y),$$

  where h is what we previously denoted by $f_{Y,\Theta}$, g is f_Θ, f is $f_{Y|\Theta}$ and so on.

- However, many authors use a different system of notation, where one **abuses notation** to make the presentation more compact. For instance, one may use $p(\cdot)$ to stand generically for different densities, so that the argument of p shows both what random quantity is under consideration and the value it may assume. Further, it is customary to let an expression such as $g(\theta)$ denote the function g. Using such notation, e.g.,

$$p(\theta) \quad \text{means the function } f_\Theta$$

and

$$p(y) \quad \text{means the function } f_Y$$

even though f_Θ and f_Y may be quite different functions. Using such compact notation, Bayes' rule can be written as

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{p(y)}.$$

- In the sequel, we will often use such compact notation, since it is important to become familiar with notational conventions used in the field. However, we will also use more explicit (and cumbersome) notation where one uses subscripts on the densities in order to avoid misunderstandings.

1.3 Frequentist statistics versus Bayesian statistics

There are two different approaches to statistics: the Bayesian approach and the frequentist approach. The basic ideas of Bayesian statistics were introduced by the reverend Thomas Bayes in a posthumously published article in 1763. The approach was also developed and popularized by Laplace. The Bayesian approach was widely used in the 19'th century, but it was then called inverse probability. However, since the 1930's, the dominant approach to statistical inference has been what we (nowadays) call **frequentist statistics** (or **classical statistics**). This was largely due the influence of the eminent statistician R. A. Fisher and others (such as J. Neyman and E. Pearson). It is only since the 1990's that the Bayesian approach has gradually become once again widely spread largely thanks to advances in its computational methods.

In frequentist statistics the parameter is considered a deterministic, unknown quantity, whose value, say θ_0, we seek to estimate. In frequentist statistics, one does not introduce any probability distributions on the parameter space, so concepts like prior or posterior distribution do not make any sense within frequentist statistics. The typical way of estimation is by the principle of **maximum likelihood** although other methods are used, too. The maximum likelihood estimate (say, $\hat{\theta}(y)$) is that point in the parameter space which maximizes the likelihood function. In some situations, the principle of maximum likelihood needs to be supplemented with various other principles in order to avoid nonsensical results.

Frequentist statistics assess the performance of a statistical procedure by considering its performance under a large number of **hypothetical repetitions** of the observations under identical conditions. In particular, a frequentist statistician is interested in the distribution of the estimator $\hat{\theta}(Y)$, when data Y is drawn from the sampling distribution with density $f_{Y|\Theta}(y \mid \theta_0)$. (A true frequentist would not use such notation but would use something like $f_Y(y; \theta_0)$

instead.) The distribution produced for $\hat{\theta}(Y)$ is the **sampling distribution of the estimator**.

Often the estimate is supplemented with its standard error or a confidence interval. Another popular approach is to make hypothesis tests on (functions) of parameters. These concepts are based on the sampling distribution of the estimator.

In contrast to frequentists, Bayesian statisticians always condition on the observed data. Bayesians are not concerned with what would happen with data we might have observed but did not. The end result of Bayesian inference is the posterior distribution. However, since this probability distribution can be difficult to understand directly, one usually settles for summarizing the posterior distribution in some manner (e.g., by calculating its mean or mode, or by calculating posterior intervals). A Bayesian makes probability statements about the parameter given the observed data, rather than probability statements about hypothetical repetitions of the data conditional on the unknown value of the parameter.

There used to be a bitter controversy among followers of the two different schools of thought. The frequentists pointed out that the inferences made by Bayesians depend on the prior distribution chosen by the statistician. Therefore Bayesian inference is not objective but is based on the personal beliefs of the statistician. On the other hand, the Bayesians liked to poke fun at the many paradoxes one gets by adhering rigidly to the principles used in frequentist statistics and accused the field of frequentist statistics to be a hodgepodge of methods derived from questionable principles.

However, nowadays many statisticians use both Bayesian and frequentist inference. If the sample size is large, then the point estimates, confidence intervals and many other inferences using either approach are usually quite similar. However, the interpretations of these results are different. A Bayesian statistician might consider results he or she obtains using frequentist methods to be approximations to results one would obtain using proper Bayesian methodology, and vice versa.

One area where the two approaches differ clearly is hypothesis testing. In frequentist statistics it is very common to conduct a test of a sharp null hypothesis (or a point null hypothesis or a simple hypothesis) such as

$$H_0 : \theta = \theta_0 \quad \text{vs.} \quad H_1 : \theta \neq \theta_0.$$

Many Bayesians have objections to the whole idea of testing a sharp null hypothesis. What is more, in this setting one arrives at quite different results using Bayesian or frequentist methods. In contrast, the two schools of inference obtain typically very similar results in one-sided tests, e.g., of the form

$$H_0 : \theta \le \theta_0 \quad \text{vs.} \quad H_1 : \theta > \theta_0.$$

1.4 A simple example of Bayesian inference

To illustrate the basic notions, consider the following example. Suppose that conditionally on $\Theta = \theta$, the random variables $Y_i, i = 1, \dots, n$ are independently exponentially distributed with rate θ, i.e., that

$$p(y_i \mid \theta) = \mathrm{Exp}(y_i \mid \theta) = \theta \, \mathrm{e}^{-\theta y_i}, \qquad y_i > 0.$$

See Appendix A for the notation and parametrization used in these lecture notes for the standard probability distributions such as the exponential. The likelihood is given by

$$p(y \mid \theta) = \prod_{i=1}^{n} p(y_i \mid \theta) = \theta^n \, \exp(-\theta \sum_{i=1}^{n} y_i).$$

Suppose that our prior is the gamma distribution $\mathrm{Gam}(a, b)$ with known hyperparameters $a, b > 0$, i.e.,

$$p(\theta) = \mathrm{Gam}(\theta \mid a, b) = \frac{b^a}{\Gamma(a)} \, \theta^{a-1} \mathrm{e}^{-b\theta}, \qquad \theta > 0.$$

Then, as a function of $\theta > 0$,

$$\begin{aligned} p(\theta \mid y) &\propto p(y \mid \theta) \, p(\theta) \\ &\propto \theta^{a-1} \mathrm{e}^{-b\theta} \, \theta^n \, \exp(-\theta \sum_{i=1}^{n} y_i) \\ &= \theta^{a+n-1} \, \exp(-(b + \sum_{i=1}^{n} y_i)\theta) \\ &\propto \mathrm{Gam}(\theta \mid a+n, b + \sum_{i=1}^{n} y_i). \end{aligned}$$

This shows that the posterior distribution is the gamma distribution

$$\mathrm{Gam}(a+n, b + \sum_{i=1}^{n} y_i).$$

Since the gamma distribution is a well-understood distribution, we can consider the inference problem solved.

In this case the prior distribution and posterior distribution belong to the same parametric family of distributions. In such a case we speak of a conjugate family (under the likelihood under consideration). In such a case Bayesian inference amounts to finding formulas for updating the so called hyperparameters of the conjugate family.

We might also want to consider a future observable Y^* whose distribution conditionally on $\Theta = \theta$ is also exponential with rate θ but which is conditionally indpendent of the already available observations $y_1, \ldots, y_n$. Then $p(y^* \mid y)$ is called the (posterior) **predictive distribution** of the future observable. Thanks to conditional independence, the joint posterior of Θ and Y^* can be shown to factorize as follows

$$p(y^*, \theta \mid y) = p(y^* \mid \theta) \, p(\theta \mid y)$$

and therefore, by marginalizing,

$$\begin{aligned} p(y^* \mid y) &= \int p(y^*, \theta \mid y) \, \mathrm{d}\theta = \int p(y^* \mid \theta) \, p(\theta \mid y) \, \mathrm{d}\theta \\ &= \int_0^\infty \theta \, \mathrm{e}^{-\theta y^*} \, \frac{(b + \sum_1^n y_i)^{a+n}}{\Gamma(a+n)} \, \theta^{a+n-1} \, \mathrm{e}^{-(b+\sum_1^n y_i)\theta} \, \mathrm{d}\theta \end{aligned}$$

where the integral can be expressed in terms of the gamma function. Hence also the predictive distribution can be obtained explicitely.

If we are not satistifid by any gamma distribution as a representation of our prior knowledge, and we may pick our prior from another family of distributions. In this case the situation changes dramatically in that we must resort to numerical methods in order to understand the posterior distribution.

1.5 Introduction to Bayesian computations

Conceptually, Bayesian inference is simple. One simply combines the prior and the likelihood to derive the posterior. For a single parameter, this can be implemented quite simply by graphical methods or by numerical integration. However for more complex problems, Bayesian inference was traditionally extremely hard to implement except in some simple situations where it was possible to use conjugate priors and arrive at analytical solutions. In distinction, in classical statistics the conceptual underpinnings behind statistical inference are more complicated, but the calculations are simple, at least in the case of certain standard statistical models.

A breakthrough occurred in the 1980's, when people realized two things.

- Instead of an analytic expression, one can represent the posterior distribution on a computer by drawing a sequence of samples from it.

- In most situations it is easy to draw samples from the posterior using MCMC methods (Markov chain Monte Carlo methods). Such methods were introduced in the statistical physics literature already in the 1950's. Several computer programs, most notably BUGS (WinBUGS or OpenBUGS), are now available for constructing automatically MCMC algorithms for a wide variety of statistical models.

1.6 Literature

- See, e.g., Bernardo and Smith [2] for a clear exposition of the ideas of Bayesian statistics.

- Schervish [25] treats both Bayesian and frequentist statistics using a rigorous, measure theoretic formulation.

- See, e.g., Gelman et al. [10], O'Hagan and Forster [21], Rossi, Allenby and McCulloch [24], Link and Barker [17], Kadane [14] or Christensen et al. [6] for expositions of Bayesian analysis and its computational techniques.

- See, e.g., Bolstad [3], Robert and Casella [23] and Albert [1] for introductions to Bayesian computation and MCMC.

- More advanced books discussing Bayesian computation and MCMC include those by Tanner [26]; Robert and Casella [22]; Liu [18]; Chen, Shao and Ibrahim [5]; Gamerman and Lopes [9]; Liang, Liu and Carroll [16] and the handbook [4].

- Congdon [8, 7], Ntzoufras [20], Kéry [15] and Lunn *et al.* [19] discuss lots of of Bayesian models using BUGS as the computational engine for implementing the inference.
- To gain a wider picture of computational statistics, consult Gentle [11, 12] or Givens and Hoeting [13].

Bibliography

[1] Jim Albert. *Bayesian Computation with R.* Springer, 2007.

[2] José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory.* John Wiley & Sons, 2000. First published in 1994.

[3] William M. Bolstad. *Understanding Computational Bayesian Statistics.* Wiley, 2010.

[4] Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng, editors. *Handbook of Markov Chain Monte Carlo.* CRC Press, 2011.

[5] Ming-Hui Chen, Qi-Man Shao, and Joseph G. Ibrahim. *Monte Carlo Methods in Bayesian Computation.* Springer, 2000.

[6] Ronald Christensen, Wesley Johnson, Adam Branscum, and Timothy E. Hanson. *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians.* Texts in Statistical Science. CRC Press, 2011.

[7] Peter Congdon. *Applied Bayesian Modelling.* Wiley, 2003.

[8] Peter Congdon. *Bayesian Statistical Modelling.* Wiley, 2nd edition, 2006.

[9] Dani Gamerman and Hedibert F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference.* Chapman & Hall/CRC, second edition, 2006.

[10] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis.* Chapman & Hall/CRC Press, 2nd edition, 2004.

[11] James E. Gentle. *Elements of Computational Statistics.* Springer, 2002.

[12] James E. Gentle. *Computational Statistics.* Springer, 2009.

[13] Geof H. Givens and Jennifer A. Hoeting. *Computational Statistics.* Wiley-Interscience, 2005.

[14] Joseph B. Kadane. *Principles of Uncertainty.* Texts in Statistical Science. CRC Press, 2011.

[15] Marc Kéry. *Introduction to WinBUGS for ecologists: A Bayesian approach to regression, ANOVA, mixed models, and related analyses.* Academic Press, 2010.

[16] Faming Liang, Chuanhai Liu, and Reymond J. Carroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples.* Wiley, 2010.

[17] William A. Link and Richard J. Barker. *Bayesian Inference—with ecological applications.* Academic Press, 2010.

[18] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer, 2001.

[19] David Lunn, Christopher Jackson, Nicky Best, Andrew Thomas, and David Spiegelhalter. *The BUGS Book: a practical introduction to Bayesian analysis.* Texts in Statistical Science. CRC Press, 2013.

[20] Ioannis Ntzoufras. *Bayesian Modeling Using WinBUGS.* Wiley, 2009.

[21] Anthony O'Hagan and Jonathan Forster. *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics.* Arnold, second edition, 2004.

[22] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer, second edition, 2004.

[23] Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R.* Springer, 2010.

[24] Peter E. Rossi, Greg M. Allenby, and Robert McCulloch. *Bayesian Statistics and Marketing.* Wiley, 2005.

[25] Mark J. Schervish. *Theory of Statistics.* Springer series in statistics. Springer-Verlag, 1995.

[26] Martin A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* Springer Series in Statistics. Springer, 3rd edition, 1996.

Chapter 2

Review of Probability

We are going to work with random vectors. Some of their components have discrete distributions and some continuous distributions, and a random vector may have both types of components. The reader is hopefully familiar with most of the concepts used in this chapter. We use uppercase letters such as X for random variables and random vectors, and lowercase letters such as x for their possible values. When there are several random variables under consideration, we may use subscripts to differentiate between functions (such as cumulative distribution functions, densities, ...) associated with the different variables.

2.1 Random variables and random vectors

While the student needs not know measure theoretic probability theory, it useful to at least recognize some concepts. The starting point of the theory is a **probability space** (or probability triple) $(\Omega, \mathcal{A}, P)$, where

- Ω is a set called a **sample space**,
- $\mathcal{A}$ is a collection of subsets of Ω. A set $E \in \mathcal{A}$ is called an **event**.
- P is a **probability measure**, which assigns a number

  $$0 \le P(E) \le 1, \qquad E \in \mathcal{A}$$

  for each event E.

A **random variable** X is defined to be a function

$$X : \Omega \to \mathbb{R} \, .$$

Intuitively, a random variable is a number determined by chance. A **random vector** Y is a function

$$Y : \Omega \to \mathbb{R}^d$$

for some positive integer d. I.e., random vectors are vector-valued functions whose components are random variables. A random variable is a special case of a random vector (take $d = 1$). We will use the abbreviation RV to denote either a random variable or a random vector.

For technical reasons, which we will not discuss, the set of events $\mathcal{A}$ usually does not contain all subsets of Ω. Further, all RVs need to be Borel measurable. This is a technical condition, which ensures that everything is properly defined. Further, for technical reasons, all subsets of of $\mathbb{R}$ or $\mathbb{R}^d$ used in these notes are assumed to be Borel subsets, and this requirement is not going to be mentioned anymore.

If X is a random variable, then it is of interest to know how to calculate the probability that $X \in B$ for and arbitrary set $B \subset \mathbb{R}$. The function

$$B \mapsto P(X \in B), \qquad B \subset \mathbb{R}$$

is called the **distribution** of X. Here $P(X \in B)$ means the probability of the event

$$\{X \in B\} = \{\omega \in \Omega : X(\omega) \in B\}.$$

In probability theory, it is customary to suppress the argument ω whenever possible, as was done here.

The distribution of a random vector Y is defined similarly as the set function

$$B \mapsto P(X \in B), \qquad B \subset \mathbb{R}^d\,.$$

The distribution of a RV defined as a set function is an abstract concept. In applications one usually deals with more concrete representations such as cumulative distribution functions, probability mass functions or probability densities.

2.2 Cumulative distribution function, cdf

The **cumulative distribution function** (cdf) of a random variable X is defined as

$$F_X(x) = P(X \le x), \qquad x \in \mathbb{R}\,. \tag{2.1}$$

(Probabilists often use the shorter term **distribution function**.) If there is only one random variable under consideration, we may omit the symbol of that variable from the subscript. The cdf is defined for any random variable no matter what type its distribution is (discrete, continuous, or something more complicated).

The cumulative distribution function (cdf) determines the distribution. If two random variables X and Y have the same cdf's, then they have the same distributions, i.e.,

$$F_X = F_Y \quad \Leftrightarrow \quad (P(X \in B) = P(Y \in B), \quad \forall B \subset \mathbb{R})\,.$$

The cumulative distribution function of a random vector $X = (X_1, \ldots, X_d)$ is defined analogously,

$$F_X(x) = P(X \le x) = P(X_1 \le x_1, \ldots, X_d \le x_d), \qquad x = (x_1, \ldots, x_d) \in \mathbb{R}^d\,.$$

The cumulative distribution function determines the distribution also for random vectors.

2.3 Discrete distributions

A discrete RV takes values in a finite or countable set. In this case also the distribution of that quantity is called discrete. The **probability (mass) function** (pmf) of a discrete RV is defined by

$$f_X(x) = P(X = x). \tag{2.2}$$

Usually the range of a discrete random variable is a subset of the integers.

A pmf f_X has the properties

$$0 \le f_X(x) \le 1, \quad \forall x,$$

and

$$\sum_x f_X(x) = 1,$$

which follow at once from the properties of the probability measure. Here the sum extends over all the possible values of X.

2.4 Continuous distributions

A RV X is called continuous and is said to have a continuous distribution, if its distribution has a **probability density function** (pdf) (or simply density), i.e., if there exists a function $f_X \ge 0$ such that for any set B,

$$P(X \in B) = \int_B f_X(x)\,\mathrm{d}x. \tag{2.3}$$

If X is a random variable, then $B \subset \mathbb{R}$, but if X is d-dimensional random vector, then $B \subset \mathbb{R}^d$, and the integral is actually a multiple integral.

The integral over the set B is defined as

$$\int_B f_X(x)\,\mathrm{d}x = \int 1_B(x) f_X(x)\,\mathrm{d}x,$$

where on the right the integral is taken over the whole space, and 1_B is the **indicator** function of the set B,

$$1_B(x) = \begin{cases} 1, & \text{if } x \in B \\ 0, & \text{otherwise.} \end{cases}$$

With integrals we follow the convention that if the range of integration is not indicated, then the range of integration is the whole space under consideration.

By definition, a probability density f_X satisfies

$$f_X(x) \ge 0, \qquad \forall x,$$

but a density need not be bounded from above. Also

$$\int f_X(x)\,\mathrm{d}x = 1,$$

(where the integral extends over the whole space). This follows since the probability that X takes on *some* value is 1.

The requirement (2.3) does not determine the density uniquely but only modulo sets of measure zero. In applications one works with continuous or piecewise-continuous versions of the densities, and does not worry about this non-uniqueness. We say that two densities f and g are equal, and write $f = g$, if f and g are densities of the same distribution, i.e., if f and g are equal almost everywhere.

The pdf can be obtained from the cdf by differentiation. In one dimension,

$$f_X = F_X'$$

Here the derivative on the right is defined almost everywhere, and on the right we may extend the function arbitrarily to whole $\mathbb{R}$. After this we obtain a valid density function. In d dimensions one has an analogous result,

$$f_{X_1,\dots,X_d}(x_1,\dots,x_d) = \frac{\partial^d F_{X_1,\dots,X_d}(x_1,\dots,x_d)}{\partial x_1 \cdots \partial x_d},$$

almost everywhere, in the sense that the mixed derivative is defined almost everywhere and after an arbitrary extension one obtains a density for the joint distribution of $X_1, \dots, X_d$.

2.5 Notation for probability distributions

The pmfs of discrete random variables and the pdfs of continuous random variables behave in many contexts in exactly the same way. That is why we use the same notation in both cases. Sometimes we use the word 'density' to refer to the pmf of a discrete random variable or even to the analogous concept for more complicated distributions. (The key mathematical concept is the Radon-Nikodym derivative with respect to some dominating sigma-finite measure.) If it is necessary to make a distinction, we will speak of the density of a continuous distribution or the density of a continuous RV.

We will be working with many standard discrete distributions (binomial, Poisson, etc.) as well as with continuous distributions (gamma, normal, etc.). We will use the following unified notation to denote the densities of these distributions.

Each of the standard distributions has a symbol, and depends on a number of parameters. The symbols we use are listed in Appendix A. We use the symbol of the distribution to denote its probability mass function (pmf) or probability density function (pdf) writing the argument on the left-hand side of the vertical bar, and the parameters on its right-hand side.

For example, the binomial distribution with sample size parameter n and probability parameter p is denoted $\mathrm{Bin}(n, p)$, and the value of its pmf at argument x is denoted $\mathrm{Bin}(x \mid n, p)$. The normal distribution with mean μ and variance σ^2 is denoted $N(\mu, \sigma^2)$, and the value of the pdf at x is denoted by $N(x \mid \mu, \sigma^2)$.

2.6 Quantile function

A quantile function is the inverse function of the distribution function of a random variable whenever the cumulative distribution function is invertible. Otherwise the quantile function is defined as a generalized inverse function of the cumulative distribution function. Notice that quantile functions are defined only for univariate distributions.

Let us first consider the important case, where the quantile function can be obtained by inverting the cdf. Consider a random variable X whose cdf F_X is continuous and strictly increasing on an interval (a, b) such that $F_X(a) = 0$ and $F_X(b) = 1$. In other words, we assume that $X \in (a, b)$ with probability one. The values $a = -\infty$ or $b = +\infty$ are permitted, in which case $F_X(a)$ or $F_X(b)$ has to be interpreted as the corresponding limit.

In this case, the equation

$$F_X(x) = u, \qquad 0 < u < 1,$$

has a unique solution $F_X^{-1}(u) \in (a, b)$ and we call the resulting function

$$q_X(u) = F_X^{-1}(u), \qquad 0 < u < 1 \tag{2.4}$$

the quantile function of (the distribution of) X. (This is abuse of notation: we are actually using the inverse function of the cdf F_X restricted to the interval (a, b).) If a or b is finite, we could extend the domain of definition of q_X in a natural way to cover the points 0 or 1, respectively. However, we will not do this since this would lead to difficulties when $a = -\infty$ or $b = \infty$.

Since

$$P(X \le q_X(u)) = F_X(q_X(u)) = F_X(F_X^{-1}(u)) = u, \qquad 0 < u < 1,$$

a proportion of u of the distribution of X lies to the left of the point $q_X(u)$. Similarly,

$$P(X > q_X(u)) = 1 - F_X(q_X(u)) = 1 - u, \qquad 0 < u < 1,$$

which shows that a proportion $1 - u$ of the distribution of X lies to the right of the point $q_X(u)$. If the distribution of X is continuous, then the quantiles satisfy

$$\int_{-\infty}^{q_X(u)} f_X(x)\,\mathrm{d}x = u, \qquad \forall 0 < u < 1 \tag{2.5}$$

$$\int_{q_X(u)}^{\infty} f_X(x)\,\mathrm{d}x = 1 - u, \qquad \forall 0 < u < 1, \tag{2.6}$$

i.e., for any $0 < u < 1$, the area under the pdf in the left-hand tail $(-\infty, q_X(u))$ is u, and the area under the pdf in right-hand tail $(q_X(u), \infty)$ is $1 - u$.

Example 2.1. The unit exponential distribution Exp(1) has the density

$$f_X(x) = \mathrm{e}^{-x}\, 1_{[0,\infty)}(x)$$

and cdf

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\,\mathrm{d}t = \begin{cases} 1 - \mathrm{e}^{-x}, & \text{if } x \ge 0 \\ 0, & \text{otherwise.} \end{cases}$$

Hence the quantile function of this distribution is

$$q_X(u) = F_X^{-1}(u) = -\ln(1-u), \qquad 0 < u < 1.$$

△

The quantile function has important uses in simulation. Let $U \sim \mathrm{Uni}(0,1)$, which means that U has the uniform distribution on $(0,1)$. Recall that most programming environments have a random number generator for the $\mathrm{Uni}(0,1)$ distribution. Let q_X be the quantile function of a random variable X. Then

$$q_X(U) \overset{\mathrm{d}}{=} X, \tag{2.7}$$

which means that $q_X(U)$ has the same distribution as X. We will check this claim shortly.

Equation (2.7) shows how a uniformly distributed random variable U can be transformed to have a given distribution. We will refer to this method by the name **inverse transform** or **inversion**. This method has several other names in the literature: the **probability integral transform** the **inverse transformation method**, the **quantile transformation method** and others. The inverse transform is an excellent simulation method for certain distributions, whose quantile functions are easy to calculate.

Example 2.2. By the previous example, we can simulate a random draw from $\mathrm{Exp}(1)$ by generating $U \sim \mathrm{Uni}(0,1)$ and then calculating

$$-\ln(1-U).$$

This procedure can be simplified a bit by noticing that when $U \sim \mathrm{Uni}(0,1)$, then also $1-U \sim \mathrm{Uni}(0,1)$ distribution. Therefore we may as well simulate $\mathrm{Exp}(1)$ by calculating

$$-\ln(U).$$

△

We now check the claim (2.7) in the case introduced before, where F_X is continuous and strictly increasing on (a,b) and $F_X(a) = 0$ and $F_X(b) = 1$.

Recall that the inverse function of a strictly increasing function is strictly increasing. Therefore

$$\{(u,x) \in (0,1) \times (a,b) : q_X(u) \le x)\} = \{(u,x) \in (0,1) \times (a,b) : u \le F_X(x)\}.$$

(Apply F_X to both sides of the first inequality, or $q_X = F_X^{-1}$ to the second.) Hence, for any $a < x < b$,

$$P(q_X(U) \le x) = P(U \le F_X(x)) = F_X(x).$$

This proves eq. (2.7).

A more general cdf F does not admit an inverse function defined on $(0,1)$. However, one can define a generalized inverse function by using the formula

$$F^{-1}(u) = \inf\{x : F(x) \ge u\}, \qquad 0 < u < 1. \tag{2.8}$$

Here $\inf B$ is the greatest lower bound of the set $B \subset \mathbb{R}$. Since a cdf is increasing and right continuous, the set $\{x : F(x) \ge u\}$ is of the form $[t, \infty)$ for some $t \in \mathbb{R}$, and then its infimum is t.

The inverse transform principle (2.7) holds for all univariate distributions, when we define the quantile function to be the generalized inverse of the cumulative distribution function.

2.7 Joint, marginal and conditional distributions

If we are considering two RVs X and Y, then we may form a vector V by concatenating the components of X and Y,

$$V = (X, Y).$$

Then the **joint distribution** of X and Y is simply the distribution of V. If the distribution of V is discrete or continuous, then we use the following notation for the pmf or density of the joint distribution

$$f_{X,Y}(x, y),$$

which means the same thing as $f_V(v)$, when $v = (x, y)$. The distribution of X or Y alone is often called its **marginal distribution**.

Recall the elementary definition of conditional probability. Suppose that A and B are events and that $P(A) > 0$. Then the conditional probability $P(B \mid A)$ of B given A (the probability that B occurs given that A occurs) is defined by

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}. \tag{2.9}$$

If the joint distribution of RVs X and Y is discrete, then the conditional distribution of Y given $X = x$ is defined by using (2.9). Given $X = x$, Y has the pmf

$$f_{Y|X}(y \mid x) = P(Y = y \mid X = x) = \frac{P(X = x, Y = y)}{P(X = x)} = \frac{f_{X,Y}(x, y)}{f_X(x)}. \tag{2.10}$$

Here f_X, the pmf of the marginal distribution of X is obtained by summing y out from the joint pmf,

$$f_X(x) = \sum_y f_{X,Y}(x, y),$$

Naturally, definition (2.10) makes sense only for those x for which $f_X(x) > 0$. If need be, we may extend the domain of definition of the conditional pmf $f_{Y|X}(y \mid x)$ by agreeing that

$$f_{Y|X}(y \mid x) = 0, \quad \text{if } f_X(x) = 0.$$

It is useful to have in mind some such extension in order to make sense of certain formulas. However, the exact manner in which we do this extensions does not really matter.

By rearranging the definition of the conditional pmf we see that for all x and y

$$f_{X,Y}(x, y) = f_X(x)\, f_{Y|X}(y \mid x).$$

By reversing the roles of X and Y, wee see that also the following holds,

$$f_{X,Y}(x, y) = f_Y(y)\, f_{X|Y}(x \mid y).$$

Hence, the pmf of the joint distribution can be obtained by multiplying the marginal pmf with the pmf of the conditional distribution. This result is called the **multiplication rule** or the **chain rule** (or the product rule).

When RVs X and Y have a continuous joint distribution, we define the conditional density $f_{Y|X}$ of Y given X as

$$f_{Y|X}(y \mid x) = \frac{f_{X,Y}(x,y)}{f_X(x)}, \quad \text{when } f_X(x) > 0. \tag{2.11}$$

Here f_X is the density of the marginal distribution of X, which can be calculated by integrating y out from the joint distribution,

$$f_X(x) = \int f_{X,Y}(x,y)\,\mathrm{d}y,$$

Again, if need be, we may extend the definition by agreeing that $f_{Y|X}(y \mid x) = 0$ whenever $f_X(x) = 0$.

The multiplication rule holds also for jointly continuously distributed RVs. Considered as a function of x and y

$$f_{X,Y}(x,y) = f_X(x)\,f_{Y|X}(y \mid x) = f_Y(y)\,f_{X|Y}(x \mid y).$$

(Equality is here interpreted as equality of density functions, i.e., it holds almost everywhere.)

If we have a discrete RV X and a continuous RV Y, then their joint distribution can be manipulated by making use of a function $f_{X,Y}(x,y)$ which yields probabilities when its summed over x and integrated over y, i.e.,

$$P(X \in A, Y \in B) = \sum_{x \in A} \int_B f_{X,Y}(x,y)\,\mathrm{d}y$$

for arbitrary sets A and B. For convenience, we call such a representation a density (of the joint distribution). We obtain the pmf of X by integrating y out from the joint density,

$$f_X(x) = \int f_{X,Y}(x,y)\,\mathrm{d}y,$$

and the density of Y by summing x out from the joint density,

$$f_Y(y) = \sum_x f_{X,Y}(x,y).$$

The multiplication rule holds,

$$f_{X,Y}(x,y) = f_X(x)\,f_{Y|X}(y \mid x) = f_Y(y)\,f_{X|Y}(x \mid y).$$

Often a joint distribution like this is specified by giving the marginal distribution of one variable and the conditional distribution of the other variable.

Often we consider the joint distribution of more than two variables. E.g., consider three RVs X, Y and Z which have (say) continuous joint distribution. By conditioning on (X,Y) and by using the multiplication rule twice, we see that

$$f_{X,Y,Z}(x,y,z) = f_{X,Y}(x,y)\,f_{Z|X,Y}(z \mid x,y) = f_X(x)\,f_{Y|X}(y \mid x)\,f_{Z|X,Y}(z \mid x,y).$$

Of course, other factorizations are possible, too. We obtain the density of the marginal distribution of any set of variables, by integrating out the other variables from the joint density. E.g., the joint (marginal) density of X and Y is

$$f_{X,Y}(x,y) = \int f_{X,Y,Z}(x,y,z)\,\mathrm{d}z,$$

and the (marginal) density of X is

$$f_X(x) = \iint f_{X,Y,Z}(x,y,z)\,\mathrm{d}y\,\mathrm{d}z$$

The multiplication rule holds also for a random vector which has an arbitrary number of components some of which have discrete distributions and some of which continuous distributions as long as the joint distribution of the continuous components is of the continuous type. In this case the joint density of any subset of the components can be obtained by marginalizing out the rest of the components from the joint density: the discrete variables have to be summed out and the continuous ones integrated out.

The multiplication rule holds also for conditional distributions. E.g., consider three variables X, Y and Z. As functions of x and y we have

$$f_{X,Y|Z}(x,y \mid z) = f_{X|Z}(x \mid z)\, f_{Y|X,Z}(y \mid x,z) = f_{Y|Z}(y \mid z)\, f_{X|Y,Z}(x \mid y,z). \tag{2.12}$$

Notice that we use one vertical bar to indicate conditioning: on the right hand side of the bar appear the variables on which we condition, in some order, and on the left hand side those variables whose conditional distribution we are discussing, in some order. We can calculate the densities of marginals of conditional distributions using the same kind of rules as for unconditional distributions: we sum over discrete and integrate over continuous variables. E.g., if the distribution of Y is continuous, then

$$f_{X|Z}(x \mid z) = \int f_{X,Y|Z}(x,y \mid z)\,\mathrm{d}y, \tag{2.13}$$

and if Y is discrete, then

$$f_{X|Z}(x \mid z) = \sum_y f_{X,Y|Z}(x,y \mid z). \tag{2.14}$$

Once we have more than two RVs, it becomes tedious to write the RVs as subscripts and their potential values as arguments. We let p be the generic symbol of a density. The argument of $p(\cdot)$ indicates both the symbol of the RV and its potential value. Hence, e.g., $p(x,y)$ indicates, that there are two RVs X and Y under consideration, and that we are considering their joint density $f_{X,Y}(x,y)$. The multiplication rule for two variables can be written as

$$p(x,y) = p(x)\,p(y \mid x) = p(y)\,p(x \mid y).$$

However, in some other contexts this notation can be misleading. In those cases we will use subscripts to make the notation unambiguous.

2.8 Independence and conditional independence

If we have several RVs $X_1, X_2, \ldots, X_n$, then they are independent, if their joint cdf factorizes as

$$F_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = F_{X_1}(x_1)\, F_{X_2}(x_2) \ldots F_{X_n}(x_n), \tag{2.15}$$

for all $x_1, x_2, \ldots, x_n$. If we have available some sort of a joint density, this is the case, if it factorizes as

$$f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = f_{X_1}(x_1)\, f_{X_2}(x_2) \ldots f_{X_n}(x_n),$$

for all $x_1, x_2, \ldots, x_n$.

If two random variables X and Y are independent, then their joint density has to satisfy

$$f_{X,Y}(x, y) = f_X(x)\, f_{Y|X}(y \mid x) = f_Y(y)\, f_{X|Y}(x \mid y) = f_X(x)\, f_Y(y)$$

by the multiplication rule and by independence. We conclude that X and Y are independent if and only if

$$f_{X|Y}(x \mid y) = f_X(x), \qquad f_{Y|X}(y \mid x) = f_Y(y)$$

for all x and y. If X and Y are independent, then if we learn that $Y = y$, this does not give us any new information concerning the distribution of X.

Sometimes we consider an infinite sequence of RVs $X_1, X_2, \ldots$. Then the sequence is independent, if for any n, the first n RVs $X_1, X_2, \ldots, X_n$ are independent. If all the RVs X_i in a finite or infinite sequence have the same distribution, then we say that $X_1, X_2, \ldots$ is an **i.i.d.** (independent, identically distributed) sequence.

Fact. If $X_1, X_2, \ldots$ are independent, and $f_1, f_2, \ldots$ are functions, then $f_1(X_1), f_2(X_2), \ldots$ are independent.

RVs $X_1, X_2, \ldots, X_n$ are **conditionally independent** given Y, if their conditional density factorizes as

$$f_{X_1,X_2,\ldots,X_n|Y}(x_1, x_2, \ldots, x_n \mid y) = f_{X_1|Y}(x_1 \mid y)\, f_{X_2|Y}(x_2 \mid y) \ldots f_{X_n|Y}(x_n \mid y),$$

for all $x_1, x_2, \ldots, x_n$ and y. Then the joint density of $X_1, X_2, \ldots, X_n$ and Y is

$$f_{X_1,\ldots,X_n,Y}(x_1, \ldots, x_n, y) = f_Y(y)\, f_{X_1|Y}(x_1 \mid y) \ldots f_{X_n|Y}(x_n \mid y).$$

We can obtain the marginal distribution of $X_1, X_2, \ldots, X_n$ from this by integrating (or summing) y out.

If conditionally on Y, the RVs $X_1, X_2, \ldots, X_n$ are not only independent but also have the same distribution, then we say that $X_1, X_2, \ldots, X_n$ are i.i.d. given Y (or conditionally on Y). It can be shown that in this case every permutation of $(X_1, \ldots, X_n)$ has the same (marginal) distribution as any other permutation. Such a collection of RVs is called **exchangeable**.

2.9 Expectations and variances

If X is a discrete RV and h is a function such that $h(X)$ is a scalar or a vector, then the **expected value** (or **expectation** or **mean**) of $h(X)$ is

$$Eh(X) = \sum_x h(x) f_X(x).$$

On the other hand, if X is a continuous RV, then

$$Eh(X) = \int h(x) f_X(x) \,\mathrm{d}x,$$

whenever that integral can be defined and the result is finite. In particular, EX is called the mean (or expectation or expected value) of X. If X is a random vector, then the mean is also a vector.

If X is a random variable, then its variance is

$$\operatorname{var} X = E((X - EX)^2).$$

The variance is always non-negative. By expanding the square, and by the linearity of expectation,

$$\operatorname{var} X = E(X^2) - (EX)^2.$$

If X is a random vector (a column vector), then we may consider its covariance matrix (variance matrix, dispersion matrix)

$$\operatorname{Cov} X = E[(X - EX)(X - EX)^T],$$

which has dimensions $d \times d$, when X has d scalar components.

Sometimes we consider the conditional expectation of a random variable Y given the value of another random variable X. Below, we write the formulas for the case when the joint distribution of X and Y is continuous. The conditional expectation of Y given $X = x$ is defined as the expectation of the conditional distribution $y \mapsto f_{Y|X}(y \mid x)$,

$$E(Y \mid X = x) = \int y \, f_{Y|X}(y \mid x) \,\mathrm{d}y.$$

The result is a function of x, say $m(x)$. When we plug the random variable X in that function, we get a random variable $m(X)$ which is called the conditional expectation of Y given the random variable X,

$$E(Y \mid X) = m(X), \quad \text{where } m(x) = E(Y \mid X = x).$$

$E(Y \mid X)$ is a random variable.

An important property of conditional expectations is the following property (iterated expectation, tower rule),

$$EE(Y \mid X) = EY, \tag{2.16}$$

i.e., one can calculate the unconditional expectation by averaging the conditional expectation over the marginal distribution. This is valid whenever EY is a well-defined extended real number (possibly infinite). In the continuous case this follows from

$$EE(Y \mid X) = \int \left[\int y \, f_{Y|X}(y \mid x) \,\mathrm{d}y \right] f_X(x) \,\mathrm{d}x = \iint y \, f_{X,Y}(x, y) \,\mathrm{d}x \,\mathrm{d}y.$$

The conditional variance of Y given $X = x$,

$$\operatorname{var}(Y \mid X = x),$$

is defined as the variance of the conditional distribution of Y given $X = x$. The result is a function depending on x. When we substitute the random variable X for x, we get the conditional variance $\operatorname{var}(Y \mid X)$ of Y given the random variable X. We have the result

$$\operatorname{var} Y = E \operatorname{var}(Y \mid X) + \operatorname{var} E(Y \mid X). \tag{2.17}$$

This shows that conditioning decreases the variance: the variance of the conditional expectation, $\operatorname{var} E(Y \mid X)$, is less or equal to the unconditional variance $\operatorname{var} Y$.

2.10 Change of variable formula for densities

If X is a discrete RV and $Y = g(X)$ is some function X, then Y has the pmf

$$f_Y(y) = P(Y = y) = P(g(X) = y) = \sum_{x:g(x)=y} f_X(x).$$

However, for continuous distributions the situation is more complicated.

2.10.1 Univariate formula

Let us first consider the univariate situation. Suppose that X is a continuous random variable with density f_X and Y is defined by

$$Y = g(X),$$

where $g : A \to B$ is a continuously differentiable function such that

- The function $g : A \to B$ is a continuously differentiable bijection from an open interval $A \subset \mathbb{R}$ to an open interval $B \subset \mathbb{R}$.
- The inverse function $g^{-1} : B \to A$ is also continuously differentiable.
- $P(X \in A) = 1$.

Since g is a bijective function defined on an open interval, it has to be either increasing or decreasing. Suppose first that g is increasing. Suppose $a < b$ and $a, b \in B$. For convenience, let $h = g^{-1}$. Then h is increasing, and therefore

$$P(a < Y < b) = P(a < g(X) < b) = P(h(a) < X < h(b)) = \int_{h(a)}^{h(b)} f_X(x)\,\mathrm{d}x.$$

By making the change of variable

$$y = g(x) \quad \Leftrightarrow \quad x = h(y),$$

we get

$$P(a < Y < b) = \int_{h(a)}^{h(b)} f_X(x)\,\mathrm{d}x = \int_a^b f_X(h(y))\,h'(y)\,\mathrm{d}y.$$

Since this holds for all $a, b \in B$ such that $a < b$, and since $P(Y \in B) = 1$, we conclude that

$$f_Y(y) = f_X(h(y))\,h'(y), \qquad \text{when } y \in B,$$

and zero elsewhere.

On the other hand, if g is decreasing, then $h = g^{-1}$ is also decreasing, and the previous calculation holds except for a change of sign.

The end result of the calculations is that in either case Y has the density given by

$$f_Y(y) = f_X(h(y)) \, |h'(y)|, \qquad \text{when } y \in B, \tag{2.18}$$

and zero elsewhere.

A useful heuristic, which helps to keep this in mind is to note that the formula

$$f_X(x) \, |\mathrm{d}x| = f_Y(y) \, |\mathrm{d}y| \tag{2.19}$$

holds under the bijective change of variable

$$y = g(x) \quad \Leftrightarrow \quad x = h(y).$$

Solving for $f_Y(y)$, we get

$$f_Y(y) = f_X(x) \left| \frac{\mathrm{d}x}{\mathrm{d}y} \right| = f_X(h(y)) \, |h'(y)|.$$

Notice that the result holds on B, the image of A under the mapping g. Elsewhere $f_Y(y) = 0$.

The result can also be expressed by using the derivative of g instead of h, if one calculates as follows,

$$f_Y(y) = f_X(x) \frac{1}{\left| \dfrac{\mathrm{d}y}{\mathrm{d}x} \right|} = f_X(x) \frac{1}{|g'(x)|} = \frac{f_X(h(y))}{|g'(h(y))|}. \tag{2.20}$$

Also this formula holds on B and $f_Y(y) = 0$ elsewhere. Formula (2.20) is correct, since the formula

$$\frac{\mathrm{d}x}{\mathrm{d}y} = \frac{1}{\dfrac{\mathrm{d}y}{\mathrm{d}x}}$$

expresses correctly the derivative of the inverse function.

This univariate case can usually be handled more easily by calculating first the cdf of $Y = g(X)$ and then by taking the derivative of the cdf. However, in higher-dimensional settings the change of variables formula becomes indispensable.

2.10.2 Multivariate formula

Consider a two-dimensional random vector $X = (X_1, X_2)$ with continuous distribution and pdf f_X, a function $g : A \to B$, where $A, B \subset \mathbb{R}^2$, and define the two-dimensional random vector Y by

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = g(X) = \begin{bmatrix} g_1(X) \\ g_2(X) \end{bmatrix}.$$

We assume that g is a **diffeomorphism**, i.e., that g is bijective, continuously differentiable, and that its inverse function is also continuously differentiable. We make the following assumptions.

- The set A is open and $P(X \in A) = 1$. The set B is the image of A under the function g. The function g is continuously differentiable.
- B is open and the inverse function $g^{-1} : B \to A$ is also continuously differentiable.

It can be shown that the random vector Y has the density

$$f_Y(y) = f_X(h(y))\, |J_h(y)|, \qquad y \in B \tag{2.21}$$

and zero elsewhere, where h is g^{-1}, the inverse function of g, and $J_h(y)$ is the **Jacobian determinant** (or Jacobian) of the function h evaluated at the point y,

$$J_h(y) = \det \begin{bmatrix} \dfrac{\partial h_1(y)}{\partial y_1} & \dfrac{\partial h_1(y)}{\partial y_2} \\ \dfrac{\partial h_2(y)}{\partial y_1} & \dfrac{\partial h_2(y)}{\partial y_2} \end{bmatrix} \tag{2.22}$$

The matrix, whose determinant the Jacobian is, is called the Jacobian matrix or the derivative matrix of the function h. This two-variate formula can be derived in the same manner as the corresponding univariate formula by making a multivariate change of variable in a multivariate integral. Notice that we need the absolute value $|J_h(y)|$ of the Jacobian determinant in the change of variable formula (2.21).

A convenient standard notation for the Jacobian determinant is

$$J_h(y) = \frac{\partial(x_1, x_2)}{\partial(y_1, y_2)}.$$

Notice that here J_h is a function of y. On the other hand, the Jacobian determinant of g,

$$J_g(x) = \frac{\partial(y_1, y_2)}{\partial(x_1, x_2)}$$

is a function of x. When $y = g(x)$ which is the same as $x = h(y)$, then we have

$$\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} \frac{\partial(y_1, y_2)}{\partial(x_1, x_2)} = 1,$$

since the two Jacobian matrices are inverses of each other, and $\det(A^{-1}) = 1/\det(A)$ for any invertible matrix A.

There is a useful heuristic also in the two-dimensional case. The formula

$$f_X(x)\, |\partial(x_1, x_2)| = f_Y(y)\, |\partial(y_1, y_2)| \tag{2.23}$$

has to hold under the bijective change of variable

$$y = g(x) \quad \Leftrightarrow \quad x = h(y).$$

Therefore

$$f_Y(y) = f_X(x) \left| \frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} \right| = f_X(h(y))\; |J_h(y)|$$

On the other hand, we may express $f_Y(y)$ as follows,

$$f_Y(y) = f_X(x)\frac{1}{\left|\dfrac{\partial(y_1, y_2)}{\partial(x_1, x_2)}\right|} = f_X(h(y))\frac{1}{|J_g(h(y))|}, \tag{2.24}$$

where J_g is the Jacobian determinant of the function g (expressed as a function of x). These formulas for $f_Y(y)$ hold on the set B. Elsewhere $f_Y(y) = 0$.

The formulas (2.21) and (2.24) generalize also to higher dimensions, when one defines the Jacobians as

$$J_h(y) = \frac{\partial x}{\partial y} = \frac{\partial(x_1, \ldots, x_d)}{\partial(y_1, \ldots, y_d)} = \det \begin{bmatrix} \dfrac{\partial h_1(y)}{\partial y_1} & \cdots & \dfrac{\partial h_1(y)}{\partial y_d} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial h_d(y)}{\partial y_1} & \cdots & \dfrac{\partial h_d(y)}{\partial y_d} \end{bmatrix}$$

and

$$J_g(x) = \frac{\partial y}{\partial x} = \frac{\partial(y_1, \ldots, y_d)}{\partial(x_1, \ldots, x_d)} = \det \begin{bmatrix} \dfrac{\partial g_1(x)}{\partial x_1} & \cdots & \dfrac{\partial g_1(x)}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial g_d(x)}{\partial x_1} & \cdots & \dfrac{\partial g_d(x)}{\partial x_d} \end{bmatrix}.$$

As an application of these formulas, consider a RV X, which has a d-dimensional continuous distribution, and define Y as an affine function of X,

$$Y = AX + b.$$

Here A is an invertible (i.e., nonsingular) $d \times d$ matrix and b is a d-vector, and A and b are constants (non-random quantities). Now

$$g(x) = Ax + b \quad \text{and} \quad h(y) = A^{-1}(y - b).$$

The Jacobian matrix of g is simply A and the Jacobian matrix of h is A^{-1}, so $J_g(x) = \det(A)$ and $J_h(y) = \det(A^{-1})$. By (2.21) or (2.24) we have

$$f_Y(y) = f_X(A^{-1}(y - b))|\det(A^{-1})| = \frac{f_X(A^{-1}(y - b))}{|\det(A)|}.$$

Chapter 3

Simulating Random Variables and Random Vectors

In this chapter we discuss methods for producing an endless supply of random values from a specified distribution, which we call the target distribution. Actually we should speak of **pseudo-random** values, since the calculated numbers are not random, but are calculated using deterministic, iterative algorithms. For practical purposes, however, the calculated values can be used as if they were the observed values of an i.i.d. sequence of RVs.

There are many terms in use for denoting this activity. Some authors speak of random variable/variate/deviate/number generation. Some say that they draw/generate/produce samples from a distribution. Some say that they simulate random variables/variates/deviates/numbers.

The aim of this chapter is not to present good (or the best) simulation methods for particular distributions. Rather, the emphasis is on explaining general principles on which such methods are based.

3.1 Simulating the uniform distribution

One speaks of **random numbers** especially when the target distribution is either the uniform distribution $\mathrm{Uni}(0, 1)$ on the unit interval $(0, 1)$ or the discrete uniform distribution on the set $\{0, \ldots, m-1\}$, where m is a large integer. Other distributions can be obtained from the uniform distribution by using a large variety of techniques.

Most programming languages and mathematical or statistical computing environments have available a generator for the uniform distribution $\mathrm{Uni}(0, 1)$. The successive values $u_1, u_2, \ldots, u_n$ returned by a good uniform random number generator can be used as if they were the observed values of and i.i.d. sequence of random variables $U_1, U_2, \ldots, U_n$ having the uniform distribution $\mathrm{Uni}(0, 1)$.

During the years, several tests have been devised for testing these key properties: uniformity and independence. (One famous test suite is the Diehard battery of tests assembled by G. Marsaglia.) Good uniform random number generators are well documented and pass all the usual tests. Good quality mathematical and statistical computing environments have such good generators, but the reader is warned that some lower quality generators remain in use in some circles.

Mathematically, a uniform random number generator is of the form

$$s_i = g(s_{i-1}), \quad u_i = h(s_i), \qquad i = 1, 2, \dots,$$

where s_i is the state of the generator at the ith step. (Typically, the state is either a scalar or a vector of a fixed dimension.) Notice that s_i is a deterministic function of the previous state s_{i-1}. The ith value returned by the generator is u_i, and it is obtained by applying a deterministic function to the state s_i. One needs an initial state s_0 to start the iteration. The initial state is usually called the seed state or the **seed**.

A random number generator usually provides means for

- querying and setting the seed (or state) of the generator,
- generating one or several random numbers.

If the random number generator is started on two different occasions from the same seed, one obtains exactly the same sequences of random numbers. Therefore it is important to be aware how one sets the seed and what happens if the seed is not explicitly set.

E.g., in the C programming language, there is available the uniform random number generator `random()` whose seed can be set with the functions `srandom()` or `initstate()`. If a program uses the function `random()` without setting the seed, then the seed is set to its default initial value with the consequence that different runs of the program make use of exactly the same "random" values.

From now on, it is assumed that the reader has available a uniform random number generator. Next we discuss how one can simulate i.i.d. random variables having some specified non-uniform target distribution. Basically, all methods are based on just two tricks, which are sometimes applied in a series,

- apply (one or several) deterministic transformations to uniform random numbers,
- apply a probabilistic transformation (such as random stopping in the accept–reject method) to an i.i.d. sequence of random numbers drawn from some distribution, the simulation of which is ultimately based on i.i.d. uniform random numbers.

3.2 The inverse transform

Let F be a univariate df, and let q be the corresponding quantile function. Recall from section 2.5 that if $U \sim \text{Uni}(0, 1)$, then the random variable X defined by

$$X = q(U) \tag{3.1}$$

has the distribution function F. This is the *inverse transform* method or *inversion* (of the distribution function). Some other names for the method include the probability integral transform and the quantile transform(ation) method).

If $U_1, \dots, U_n$ are i.i.d. and follow the $\mathrm{Uni}(0, 1)$ distribution, then also

$$X_1 = q(U_1), \dots, X_n = q(U_n) \tag{3.2}$$

are i.i.d. with the distribution function F. Independence follows, since (deterministic) functions of independent random variables are themselves independent.

The inverse transform is a good choice if the quantile function of the target distribution is easy to calculate. This is the case, e.g., for

- the exponential distribution,
- the Weibull distribution,
- the Pareto distribution,
- the Cauchy distribution (which is same as the t_1 distribution); also the t_2 distribution.

Even though there may be available an iterative routine for calculating the quantile function of some given complicated target distribution, simulating it may be computationally more efficient with some other approach.

If one uses the inverse transform for simulating the general discrete distribution with pmf

$$f(i) = p_i, \qquad i = 1, 2, \dots, k$$

with $\sum_{i=1}^{k} p_i = 1$, and remembers to use the generalized inverse function of the distribution function as the quantile function, then one obtains the following obvious algorithm.

Algorithm 1: The inverse transform method for the general discrete distribution.

```
Input: The pmf $p_1, p_2, \dots, p_k$ of the target distribution.
Result: One sample $I$ from the target distribution.
Generate $U \sim \mathrm{Uni}(0, 1)$;
Return $I$, if
    $\sum_{j=1}^{I-1} p_j \le U < \sum_{j=1}^{I} p_j$.
```

This algorithm works by dividing the unit interval into n pieces whose lengths are $p_1, \dots, p_k$ from left to right. Having generated U, the algorithm checks, into which of the intervals U falls, and returns the number of the interval. Notice that this algorithm requires a search, which may be time-consuming if k is large.

There are available more efficient algorithms such as the alias method for simulating the general discrete distribution. However, they require an initialization step. If one needs to generate just one value from a discrete distribution, then this simple method may well be the most efficient one.

3.3 Transformation methods

If we already know how to simulate a random vector $Y = (Y_1, \dots, Y_k)$ with a known distribution, and we calculate (the scalar or vector) X as some function of Y,

$$X = T(Y),$$

then X has *some* distribution. With careful choices for the distribution of Y and for the transformation T, we can obtain a wide variety of distributions for X. Of course, the inverse transform is an example of a transformation method.

Notice that if we apply the transformation T to an i.i.d. sequence $Y^{(1)}, Y^{(2)}, \dots$ with the distribution of Y, then we obtain an i.i.d. sequence

$$X^{(1)} = T(Y^{(1)}),\ X^{(2)} = T(Y^{(2)}), \dots$$

from the distribution of X.

In simulation settings one uses certain conventions, which are rarely explained in the literature. The main convention is the following. **If one generates several values in an algorithm, then they are generated independently.** This a natural convention, since the successive calls of the usual random number generators indeed do return values which can be considered independent (more pedantically: which can be considered to be observed values of independent random variables). E.g., a valid way of describing the preceding simulation of the i.i.d. sequence would be the following.

```
1. for $i = 1, 2, \dots, n$ do
       • Generate $Y$ from the appropriate distribution and set $X^{(i)} = T(Y)$.
   end
2. Return $(X^{(1)}, X^{(2)}, \dots, X^{(n)})$.
```

In some contexts one may want to denote the generated values by lower case letters (since the actual numbers should be considered to be the observed values of random variables) and in some other contexts it is more convenient to use the corresponding upper case letters (especially when one is interested in the distributional aspects of the generated numbers). This should not cause serious confusion.

Sometimes we can use known connections between distributions to find the distribution of Y and the transformation T.

Example 3.1. The log-normal distribution. Random variable X has the log-normal distribution with parameters (μ, σ^2) if and only if its logarithm is normally distributed with mean μ and variance σ^2, i.e., if

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

Therefore once we know how to simulate the normal distribution, we know how to simulate the log-normal distribution:

```
1. Generate $Y \sim N(\mu, \sigma^2)$.
2. Return $X = \exp(Y)$.
```

△

3.3.1 Scaling and shifting

If Y has a continuous distribution with the density g, and X is obtained from Y by scaling and shifting,

$$X = m + sY, \qquad m \in \mathbb{R}, s > 0, \tag{3.3}$$

then (by the change of variable formula for densities) X has the density

$$f(x \mid s, m) = g\left(\frac{x-m}{s}\right)\frac{1}{s}. \tag{3.4}$$

The density g is obtained with $s = 1$ and $m = 0$. If we know, how to simulate Y from the density g, then we can simulate from the density $f(\cdot \mid s, m)$ as follows.

1. Generate Y from the density g.
2. Return $X = m + sY$.

Many well-known families of continuous distributions have a scale parameter, i.e., their densities can be written in the form

$$x \mapsto g\left(\frac{x}{s}\right)\frac{1}{s}, \qquad s > 0. \tag{3.5}$$

In this case s is called a **scale parameter** of the family (and the family of distributions can be called a scale family). The density g is obtained, when $s = 1$. In this case we have the situation of (3.4) with $m = 0$, so simulation from the density with scale parameter s can be implemented as follows.

1. Generate Y from the density g.
2. Return $X = sY$.

Many families of distributions have a rate parameter, i.e., their densities can be represented as

$$x \mapsto \lambda\, g(\lambda x), \qquad \lambda > 0,$$

where g is a density. This means that the family is a scale family, with scale parameter $s = 1/\lambda$, i.e., the scale is the reciprocal of the rate.

As an example, consider the family of exponential distributions, which is usually parametrized using the rate parameter $\lambda > 0$. The density function of the $\mathrm{Exp}(\lambda)$ distribution (exponential with rate λ) is

$$\mathrm{Exp}(x \mid \lambda) = \lambda \exp(-\lambda x) 1_{(0,\infty)}(x)$$

We see that $s = 1/\lambda$ is a scale parameter. Recall that we already know how to simulate the $\mathrm{Exp}(1)$ distribution (the unit exponential distribution) using the inverse transform. Therefore we can simulate the $\mathrm{Exp}(\lambda)$ distribution as follows.

1. Generate Y from the unit exponential distribution $\mathrm{Exp}(1)$.
2. Return $X = Y/\lambda$.

This simulation algorithm can also be derived directly using the inverse transform.

Some families of continuous distributions have both a scale and a location parameter, i.e., their densities can be written in the form (3.4). Such a family is called a location-scale family, and s is called the scale parameter and m the location parameter of the family. A familiar example is the family of normal distributions

$$\{N(\mu, \sigma^2), \mu \in \mathbb{R}, \sigma^2 > 0\}.$$

$N(\mu, \sigma^2)$, the normal distribution with mean μ and variance σ^2, has the density

$$N(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) = \frac{1}{\sigma}\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right).$$

Therefore μ is a location parameter, and the standard deviation (square root of variance) σ is a scale parameter of (univariate) normal distributions.

As a consequence, we can generate $X \sim N(\mu, \sigma^2)$ as follows.

1. Generate $Y \sim N(0, 1)$.
2. Return $X = \mu + \sigma Y$.

For another example of a location-scale family of distributions, consider $\mathrm{Uni}(a, b)$, the uniform distribution on the interval (a, b), where $a < b$. This distribution has the density

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{if } a < x < b, \\ 0, & \text{otherwise.} \end{cases}$$

A moments reflection shows that one can simulate the $\mathrm{Uni}(a, b)$ distribution as follows.

1. Generate $U \sim \mathrm{Uni}(0, 1)$.
2. Return $X = a + (b - a)U$.

3.3.2 Polar coordinates

Consider the transformation from polar coordinates (r, ϕ) to the Cartesian coordinates (x, y),

$$x = r\cos(\phi), \qquad y = r\sin(\phi). \tag{3.6}$$

Here r is the radius (or radial coordinate) and ϕ is the polar angle in radians. The mapping (3.6) is defined for all $r \geq 0$ and for all angles ϕ. However, if we want to use the change of variable formula with this mapping, we first have to restrict its domain so that the mapping becomes a bijection between its domain and its range. We obtain a bijective correspondence between (r, ϕ) and (x, y), if the domain of the mapping is selected so that $r > 0$ and ϕ is allowed to have values in any fixed open interval of length 2π.

We will use the following domain domain for the polar angle ϕ,

$$-\pi < \phi < \pi.$$

With this choice, the mapping (3.6) defines a bijective correspondence between the following open sets

$$(r,\phi) \in (0,\infty) \times (-\pi,\pi) \quad \to \quad (x,y) \in \mathbb{R}^2 \setminus \{(x,y) : x \le 0, y = 0\}. \tag{3.7}$$

Here the image of the domain $(0,\infty) \times (-\pi,\pi)$ is the coordinate plane cut along the negative x-axis. The Jacobian of the mapping $(r,\phi) \mapsto (x,y)$ is

$$\frac{\partial(x,y)}{\partial(r,\phi)} = \det \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial x}{\partial \phi} \\ \dfrac{\partial y}{\partial r} & \dfrac{\partial y}{\partial \phi} \end{bmatrix} = \det \begin{bmatrix} \cos\phi & -r\sin\phi \\ \sin\phi & r\cos\phi \end{bmatrix} = r.$$

The inverse function of the mapping (3.6) is a bit tricky to express. Many books state (not correctly) that we get r and ϕ form x and y by the formulas

$$r = \sqrt{x^2+y^2}, \qquad \phi = \arctan(y/x),$$

but if not an outright error, at least this is an instance of misuse of notation. If you have to program your own routines for the rectangular to polar conversion, do not use those formulas!

The formula for r is correct, and it is true that one has to select the value of ϕ so that $\tan(\phi) = y/x$. There is, however, a problem with the formula $\phi = \arctan(y/x)$, which stems from the fact, that the tangent function does not have a unique inverse function. Usually, the notation arctan means the principal branch of the (multivalued) inverse tangent function with the range

$$-\pi/2 < \arctan(u) < \pi/2, \qquad u \in \mathbb{R}\,.$$

If you use this convention and the formula $\phi = \arctan(y/x)$, then your polar coordinate point (r,ϕ) is never in the second or third quadrant even when the original Cartesian coordinate point (x,y) is.

So, care is needed with the Cartesian to polar coordinate formula $(x,y) \mapsto (r,\phi)$. One expression, which is correct and easy to program, is given by

$$r = \sqrt{x^2+y^2}, \qquad \phi = \operatorname{atan2}(y,x), \tag{3.8}$$

where $\operatorname{atan2}(y,x)$ is the arc tangent function of two variables, which is defined for all $(x,y) \neq (0,0)$. It returns the counterclockwise (signed) angle in radians in the range $(-\pi,\pi]$ between the positive x axis and the vector (x,y). The function atan2 is available in most programming languages (but the order of the arguments is reversed in some programming environments). If (x,y) does not fall on the negative x-axis, then r and ϕ calculated by (3.8) satisfy $r > 0$ and $-\pi < \phi < \pi$.

The polar to Cartesian conversion formula (3.6) and the Cartesian to polar conversion formula (3.8) define a diffeomorphism between the sets in eq. (3.7).

After this preparation, suppose the two-dimensional random vector (X,Y) has a continuous density, and we want to express this distribution by means of polar coordinates (R,Φ) using the conversion formula (3.8). Now the probability that (X,Y) is exactly on the negative x-axis, $P(X \le 0, Y = 0) = 0$, since the joint distribution is continuous. Furthermore, we have a diffeomorphism between the coordinates (r, ϕ) and (x, y) given by formulas (3.6) and (3.8). Hence, we can apply the change of variables formula with the result

$$f_{R,\Phi}(r, \phi) = f_{X,Y}(x, y) \left| \frac{\partial(x, y)}{\partial(r, \phi)} \right| = r f_{X,Y}(r \cos\phi, r \sin\phi), \qquad r > 0, -\pi < \phi < \pi. \tag{3.9}$$

Actually, the same formula for $f_{R,\Phi}$ is valid, if we choose *any* open interval of length 2π as the domain of ϕ. This follows, since in that case one can define a diffeomorphism between rotated versions of the sets in eq. (3.7), and the Jacobian needed in the change of variables formula is still r.

Suppose in particular that the density $f_{X,Y}(x, y)$ is invariant under rotations about the origin, i.e., that

$$f_{X,Y}(x, y) = g(r), \quad \text{with } r = \sqrt{x^2 + y^2}. \tag{3.10}$$

Then the polar coordinates of (X, Y) have the density

$$f_{R,\Phi}(r, \phi) = r g(r) = 2\pi r g(r) \, \frac{1}{2\pi}, \qquad r > 0, -\pi < \phi < \pi.$$

This shows that R and Φ are independent, the polar angle Φ has the uniform distribution on its domain of length 2π (and this is obvious because of the rotational symmetry!), and the density of R can be read off from the previous formula. I.e., under the assumption (3.10), we have

$$R \perp\!\!\!\perp \Phi, \tag{3.11}$$

$$\Phi \sim \mathrm{Uni}(-\pi, \pi), \qquad f_R(r) = 2\pi r g(r), \quad r > 0. \tag{3.12}$$

On the other hand, suppose we start with a density for the polar coordinates (R, Φ),

$$f_{R,\Phi}(r, \phi), \qquad r > 0, -\pi < \phi < \pi$$

and let (X, Y) be (R, Φ) in Cartesian coordinates (formula (3.6)). By the change of variables formula,

$$f_{X,Y}(x, y) = \frac{f_{R,\Phi}(r, \phi)}{\left| \frac{\partial(x,y)}{\partial(r,\phi)} \right|} = \frac{f_{R,\Phi}(\sqrt{x^2 + y^2}, \mathrm{atan2}(y, x))}{\sqrt{x^2 + y^2}}, \tag{3.13}$$

where, initially, it is forbidden that (x, y) is on the negative x-axis. However, any continuous joint density for (X, Y) implies that

$$P(X \le 0, Y = 0) = 0,$$

and so we can let x and y to have any real values in (3.13). An exception is the origin $(x, y) = (0, 0)$, since the formula (3.13) is not defined at the origin, but one can use any value for $f_{X,Y}$ there, and the result remains correct.

As an application of the formulas in this section, consider the joint distribution of two independent variables, X and Y, having the standard normal distribution $N(0, 1)$. Their joint density is

$$f_{X,Y}(x, y) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} \, \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-y^2/2} = \frac{1}{2\pi} \exp(-r^2/2), \qquad \text{with } r^2 = x^2 + y^2,$$

and so it is invariant under rotations about the origin. Let (R, Φ) be (X, Y) in polar coordinates. According to formulas (3.11) and (3.12), R and Φ are independent, $\Phi \sim \mathrm{Uni}(-\pi, \pi)$, and the density of R is

$$f_R(r) = r \exp(-r^2/2), \qquad r > 0.$$

The distribution of R belongs to the family of Rayleigh distributions. A statistician recognizes more easily the distribution of $Z = R^2$. A change of variables gives

$$f_Z(z) = f_R(\sqrt{z}) \frac{1}{2\sqrt{z}} = \frac{1}{2} \exp(-\frac{1}{2} z), \qquad z > 0,$$

so $Z \sim \mathrm{Exp}(1/2)$, the exponential distribution with rate $1/2$.

As a side product, we have obtained a way to simulate two independent samples X and Y from the standard normal distribution $N(0, 1)$. We have actually rediscovered the the famous method of Box and Muller, first published in 1958. (Notice: the name is Muller, not Müller.)

Algorithm 2: The method of Box and Muller, initial version.

```
Result: Two independent samples $X$ and $Y$ from $N(0,1)$.
Generate independently $Z \sim \mathrm{Exp}(1/2)$ and $\Phi \sim \mathrm{Uni}(-\pi,\pi)$;
$X \leftarrow \sqrt{Z}\cos(\Phi), \quad Y \leftarrow \sqrt{Z}\sin(\Phi)$.
```

Of course, since we know how to simulate the $\mathrm{Exp}(1/2)$ and $\mathrm{Uni}(-\pi, \pi)$ distributions using the uniform distribution $\mathrm{Uni}(0, 1)$, we can implement the method of Box and Muller also as follows.

Algorithm 3: The method of Box and Muller, second version.

```
Result: Two independent samples $X$ and $Y$ from $N(0,1)$.
Generate $U$ and $V$ independently from the $\mathrm{Uni}(0,1)$ distribution;
$X \leftarrow \sqrt{-2\ln U}\cos(\pi(2V-1)), \quad Y \leftarrow \sqrt{-2\ln U}\sin(\pi(2V-1))$.
```

If you did not know about the explanation involving polar coordinates, these formulas would probably seem totally mysterious to you.

Actually, Box and Muller stated their method in the following form.

Algorithm 4: The method of Box and Muller, original version.

```
Result: Two independent samples $X$ and $Y$ from $N(0,1)$.
Generate $U$ and $V$ independently from the $\mathrm{Uni}(0,1)$ distribution;
$X \leftarrow \sqrt{-2\ln U}\cos(2\pi V), \quad Y \leftarrow \sqrt{-2\ln U}\sin(2\pi V)$.
```

This form uses the same idea, but corresponds to the convention that the polar angle belongs to the interval $(0, 2\pi)$.

There are also other methods for generating two independent draws from the standard normal, which are based on the use of polar coordinates (look up the Marsaglia polar method in Wikipedia). If one uses a bad uniform random number generator, then the method of Box and Muller leads to certain practical difficulties, although the method is exact if one uses uniform random variables.

3.3.3 The ratio of uniforms method

A nonnegative function $h \geq 0$ defined on some Euclidean space is called an **unnormalized density**, if its integral over the whole space is finite and nonzero. An unnormalized density can be converted to a density function f by normalizing it,

$$f(x) = h(x) \Big/ \int h(t)\,\mathrm{d}t, \qquad x \in \mathbb{R}\,.$$

Unnormalized densities occur quite frequently in Bayesian statistics in the form

$$\text{prior} \times \text{likelihood}.$$

Truncated distributions (defined in the next section) provide other examples of unnormalized densities.

For still another example, consider the following definition for the uniform distribution on a set $A \subset \mathbb{R}^d$. Let $m(A)$ be the Lebesgue measure of $A \subset \mathbb{R}^d$, given by

$$m(A) = \int 1_A(x)\,\mathrm{d}x.$$

If $A \in \mathbb{R}$, then $m(A)$ is the length of set A; if $A \in \mathbb{R}^2$, then $m(A)$ is the area of A; if $A \in \mathbb{R}^3$, then $m(A)$ is the volume of A, and if $A \in \mathbb{R}^d$, we can call $m(A)$ the d-dimensional volume of A. Let $A \subset \mathbb{R}^d$. We assume that A has nonzero, finite d-dimensional volume, $0 < m(A) < \infty$. The **uniform distribution on the set** A, which we can denote by $\mathrm{Uni}(A)$, is the continuous distribution having the unnormalized density 1_A. The corresponding normalized density is, of course, $1_A/m(A)$.

Suppose that we want to generate samples from a distribution having a given unnormalized density h on the real line. Define the set $C \in \mathbb{R}^2$ by

$$C = \{(u, v) : 0 < u < \sqrt{h(v/u)}\}, \tag{3.14}$$

Kinderman and Monahan (1977) noticed that if we are able to generate the pair (U, V) from the uniform distribution on C, then V/U has the distribution corresponding to the unnormalized density h.

```
Algorithm 5: The ratio of uniforms method.
  Assumption: We know how to simulate $\mathrm{Uni}(C)$, see eq. (3.14).
  Result: One sample $X$ from the distribution with unnormalized density
          $h$.
  Generate $(U, V) \sim \mathrm{Uni}(C)$;
  $X \leftarrow V/U$
```

The correctness of the algorithm can be proved by first completing the transformation by (e.g.) defining $Y = U$, after which we have a bijective correspondence between (U, V) and (X, Y), and then by calculating the density of X from the joint density of (X, Y). The joint density can be calculated easily by the change of variables formula. The details are left as an exercise for the reader. The uniform distribution on the set C can often be simulated in the manner described in the next section.

3.4 Naive simulation of a truncated distribution

Suppose that RV X has a continuous distribution with density f_X. Suppose A a set such that $P(X \in A) > 0$. Then we can consider the distribution of X

truncated (or restricted) to the set A, which has the unnormalized density given by

$$y \mapsto f_X(y) 1_A(y). \tag{3.15}$$

This is also called the distribution of X conditionally on $X \in A$ (or given $X \in A$).

We can simulate this truncated distribution with the following, obvious method. Notice that we follow the usual convention: in the following algorithm, the successive draws within the repeat–unitl loop from the distribution with density f_X are supposed to be independent.

Algorithm 6: Naive method for simulating from a truncated distribution.

```
Input: Set $A$ and simulation method for $f_X$.
Result: A sample $Y$ from $f_X$ truncated to the set $A$.
repeat
    Simulate $X$ from the density $f_X$
until $X \in A$ ;
$Y \leftarrow X$ (i.e., accept $X$, if it is in $A$).
```

The correctness of this method follows from the following calculation,

$$P(Y \in B) = P(X \in B \mid X \in A) = \frac{\int_{A \cap B} f_X(x) \, \mathrm{d}x}{P(X \in A)} = \int_B f_Y(y) \, \mathrm{d}y,$$

where

$$f_Y(y) = \frac{1}{P(X \in A)} f_X(y) 1_A(y).$$

The efficiency of this method depends on the acceptance probability

$$p = P(X \in A). \tag{3.16}$$

The number of simulations needed in order to get one acceptance has the geometric distribution on $1, 2, \ldots$ with success probability p. The mean of this distribution is $1/p$.

For example, suppose that we simulate the standard normal $N(0, 1)$ truncated to the set $A = (5, \infty)$ using this naive method. Then the acceptance probability p turns out to be about $2.9 \cdot 10^{-7}$. With sample size of ten million from the $N(0, 1)$ distribution, the expected number of accepted values would be 2.9. On the other hand, should we be interested in simulating $N(0, 1)$ truncated to the complementary set $(-\infty, 5]$, then practically every point of the sample would be accepted by the naive method.

One important application for this naive simulation method is simulation of the uniform distribution on some complicated set A. Suppose that we are able to find a set B, such that $A \subset B$, and we already know how to simulate the uniform distribution on the set B. Then the uniform distribution on B truncated to the set A is the uniform distribution on A. This obvious fact can be proved by noting that the uniform distribution on B truncated to the set A has the unnormalized density

$$1_B 1_A = 1_{A \cap B} = 1_A,$$

where the last step follows from the inclusion $A \subset B$. As a consequence, we can simulate $Y \sim \mathrm{Uni}(A)$ as follows.

- Generate $X \sim \mathrm{Uni}(B)$ until $X \in A$, and then return $Y = X$.

Often we are interested a set $A \subset \mathbb{R}^2$, which can be enclosed in a rectangle $B = (a, b) \times (c, d)$. The uniform distribution on the rectangle B can simulated by generating independently the first coordinate from $\mathrm{Uni}(a, b)$ and the second coordinate from $\mathrm{Uni}(c, d)$.

Sometimes it is costly to test whether $x \in A$. In such a case we can save some computational effort, if we can find a simpler set S such that $S \subset A$. So, now we have the inclusions

$$S \subset A \subset B, \tag{3.17}$$

and we know how to simulate $\mathrm{Uni}(B)$. If now $X \in S$ with reasonable probability, and it is less costly to test, whether $x \in S$ than whether $x \in A$, then we can, on average, save some computational effort with the following algorithm.

Algorithm 7: Simulating from Uni(A), with a pretest.

```
Assumption: The inclusions $S \subset A \subset B$ hold, and we know how to
            simulate Uni($B$)
Result: One sample $Y$ from Uni($A$).
repeat
    Generate $X \sim$ Uni($B$);
    if $X \in S$ then accept $\leftarrow$ true;
    else if $X \in A$ then accept $\leftarrow$ true;
    else accept $\leftarrow$ false
until accept ;
$Y \leftarrow X$
```

The algorithm uses a Boolean variable `accept` to keep track of whether the proposed value X has been accepted or not.

If we use the naive method repeatedly (using an i.i.d. sequence of X's) to generate several values $Y_1, Y_2, \ldots, Y_n$, then they are i.i.d. On first thought this may seem obvious. After further thought this may, however, seem not so obvious anymore. The independence of the generated Y's can be proved either by elementary means or by appealing to the strong Markov property of i.i.d. sequences, but we skip the proof. The basic idea is that the sequence of X's starts afresh after each (random) time when a freshly generated Y is accepted.

3.5 Accept–reject method

In this section $f^* : \mathbb{R}^d \to [0, \infty)$ is an unnormalized density of some continuous target distribution. The corresponding normalized density function is

$$f(x) = f^*(x) \Big/ \int f^*(t)\,\mathrm{d}t.$$

In most of the applications of the method $d = 1$, but the method can be used in any dimension.

3.5.1 The fundamental theorem

Suppose $d = 1$ and consider **the set under the graph of** f^*, i.e., the set bounded by the x-axis and the graph of the function f^*,

$$A = \{(x, y) : 0 < y < f^*(x)\}. \tag{3.18}$$

The area of A is

$$m(A) = \int \left(\int_0^{f^*(x)} 1 \, \mathrm{d}y \right) \mathrm{d}x = \int f^*(x) \, \mathrm{d}x.$$

The same calculation for $m(A)$ holds for other values of d, too.

Suppose (X, Y) is uniformly distributed in the set A (3.18), and let us calculate (a) the marginal density of X and (b) the conditional density of Y given $X = x$. The joint density of (X, Y) is given by

$$f_{X,Y}(x, y) = \begin{cases} 1/m(A), & \text{if } (x, y) \in A, \\ 0, & \text{otherwise} \end{cases}$$

By the following calculation, the marginal density of X is simply f

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) \, \mathrm{d}y = \int_0^{f^*(x)} \frac{1}{m(A)} \, \mathrm{d}y = \frac{f^*(x)}{m(A)} = f(x).$$

If x is such that $f^*(x) > 0$ and y is such that $0 < y < f^*(x)$, we have

$$f_{Y|X}(y \mid x) = \frac{f_{X,Y}(x, y)}{f_X(x)} = \frac{1}{f^*(x)},$$

while for other values of y, the conditional density is zero. In other words, given $X = x$, the random variable Y has the uniform distribution on the interval $(0, f^*(x))$.

We have incidentally proved the following theorem, which Robert and Casella call the fundamental theorem of simulation.

Theorem 1 (Fundamental theorem of simulation.)**.** *Suppose f^* is an unnormalized density on $\mathbb{R}^d$ and let f bet the corresponding normalized density. Let A be the set under the graph of f^*, i.e.,*

$$A = \{(x, y) : 0 < y < f^*(x)\}.$$

Then we have the following

1. *If $(X, Y) \sim \mathrm{Uni}(A)$, then $X \sim f$.*

2. *If $X \sim f$ and, conditionally on $X = x$, Y has the distribution $\mathrm{Uni}(0, f^*(x))$, then $(X, Y) \sim \mathrm{Uni}(A)$.*

3.5.2 Deriving the accept–reject method

Suppose that f^* is defined on the real line and that the set where $f^* > 0$ is a finite interval (a, b). Further, suppose f^* is bounded, $f^* \le K$. Then we can enclose the set A in the rectangle $(a, b) \times (0, K)$, whose uniform distribution is simple to simulate. Hence we can simulate the uniform distribution on A by the naive method for truncated distributions. But not all pdfs of interest are supported on a finite interval. What to do in that case?

The solution is to apply the fundamental theorem twice. Suppose that we are able to find a (normalized) density function g such that

1. Mg majorizes (or envelopes) the unnormalized target density f^*, where $M > 0$ is a known (majorizing) constant, i.e.,
$$f^*(x) \le Mg(x) \qquad \text{for all } x. \tag{3.19}$$
2. We know how simulate from g.

Then

$$A = \{(x, y) : 0 < y < f^*(x)\} \subset B = \{(x, y) : 0 < y < Mg(x)\}.$$

By the fundamental theorem, we can simulate (X, Y) from the uniform distribution on B as follows,

$$\text{Generate } X \sim g \text{ and } U \sim \text{Uni}(0, 1); \text{ set } Y = Mg(X)U.$$

Therefore we can use the naive method for a truncated distribution to simulate the uniform distribution on A: we simulate $(X, Y) \sim \text{Uni}(B)$ until (X, Y) falls under the graph of f^*. Combining these ideas, we get the following algorithm.

Algorithm 8: The accept–reject method.

```
Assumption: The unnormalized $f^*$ is majorized by $Mg$
Result: One sample $X$ from $f$.
repeat
    Generate $Z \sim g$ and $U \sim \text{Uni}(0, 1)$.
until $Mg(Z)U < f^*(Z)$ ;
$X \leftarrow Z$ (i.e., accept the proposal $Z$).
```

Remarks

- Notice carefully that we don't need to know the normalizing constant of the unnormalized density f^*; we only need a valid majorant.
- Some people call the method acceptance sampling or the acceptance method; some others call it rejection sampling or the rejection method.
- The majorizing function $Mg(z)$ is also called the (upper) envelope of $f^*(z)$.
- The method can also described so that one accepts the proposal $Z \sim g$ with probability $f^*(Z)/(Mg(Z))$.
- The accept–reject method was originally published by John von Neumann in 1951.

- Although the method works in any dimension, finding useful envelopes in high-dimensional cases is very challenging.

The efficiency of the method depends crucially on the acceptance probability. Notice that the joint density Z and U before the acceptance test is

$$f_{Z,U}(z,u) = g(z) 1_{(0,1)}(u).$$

Therefore the acceptance probability is

$$\begin{aligned} p &= P\left(U < \frac{f^*(Z)}{Mg(Z)}\right) \\ &= \int \mathrm{d}z \int_0^{f^*(z)/(Mg(z))} \mathrm{d}u \, g(z) 1_{(0,1)}(u) \\ &= \int g(z) \frac{f^*(z)}{Mg(z)} \, \mathrm{d}z = \frac{\int f^*(z) \, \mathrm{d}z}{M}. \end{aligned} \tag{3.20}$$

If $d = 1$, this is the same as

$$\frac{\text{Area under } f^*}{\text{Area under the envelope } Mg.}$$

(Here, e.g., "area under f^*" actually means the area of the set bounded by the graph of f^* and the x-axis.) In order to get high efficiency, we need as high acceptance probability as possible. This is achieved by using a tightly fitting envelope Mg.

For a fixed g, if the majorizing condition

$$f^* \le Mg$$

holds for $M = M_0$, then it holds for all $M \ge M_0$. In order to achieve the best efficiency, one should choose the least possible value for M such that the majorizing condition holds. However, the accept–reject method is valid for any choice of M such that the majorizing condition is true.

3.5.3 An example of accept–reject

Consider the unnormalized target density

$$f^*(x) = \exp(-x^2/2)(1 + 2\cos^2(x)\sin^2(4x)), \tag{3.21}$$

which is majorized by the function

$$Mg(x) = 3\exp(-x^2/2).$$

Here $Mg(x)$ is an unnormalized density of the $N(0,1)$ distribution, so g is the density of $N(0,1)$. Based on this fact, we could (but now need not) give an expression for M. The implied value of M is valid, but is not the best possible.

The following fragment coded in the R-language calculates $n = 1000$ independent values from the distribution corresponding to f^* using the accept–reject method and stores them in the vector `x`. The acceptance condition $Mg(Z)U < f^*(Z)$ has been converted to the equivalent condition

$$U < \frac{f^*(Z)}{Mg(Z)},$$

which now simplifies a bit.

```
n <- 1000;
x <- numeric(n)  # create a vector with n entries to store the results
for (i in 1:n) { # generate x[i]
  while (TRUE) {
    z <- rnorm(1); u <- runif(1)
    if (u < (1 + 2 * cos(z)^2 * sin(4 * z)^2) / 3) { # accept!
      x[i] <- z
      break
    }
  }
}
```

3.5.4 Further developments of the method

Sometimes the function f^* is costly to evaluate, but we can find a simpler function $s \geq 0$ which minorizes it,

$$s(x) \leq f^*(x) \leq Mg(x), \qquad \text{(all } x\text{)}. \tag{3.22}$$

Then we can say that f^* has been squeezed between the lower envelope s and the upper envelope Mg. Sometimes such a function s is called a squeeze.

If s is less costly to evaluate than f^*, then we can save computation by using the following algorithm instead of the original version of accept–reject.

Algorithm 9: Accept–reject with squeezing.

```
Assumption: Inequality (3.22) holds
Result: One sample $X$ from $f$.
repeat
    Generate $Z \sim g$ and $U \sim \mathrm{Uni}(0, 1)$;
    $Y \leftarrow Mg(Z)U$;
    if $Y < s(Z)$ then accept ← true;
    else if $Y < f^*(Z)$ then accept ← true;
    else accept ← false;
until accept ;
$X \leftarrow Z$
```

Here the test $Y < s(Z)$ is now the pretest. If it succeeds, then certainly $Y < f^*(Z)$ and there is no need to evaluate $f^*(Z)$.

Many familiar univariate continuous distributions have log-concave densities. A function is called log-concave, if its logarithm is a concave function. We are now interested in the case, where the density f is defined on an open interval (a, b), and f is strictly positive and twice differentiable on that interval. Then f is log-concave, if and only if

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2} \log f(x) \leq 0, \qquad a < x < b.$$

The graph of a concave function lies below each of its tangents. Also, the graph of a concave function lies above each of its chords (secants). Therefore it is easy to find piecewise linear upper and lower envelopes for concave functions. If one constructs piecewise linear envelopes for $\log f$, then, by exponentiation, one gets piecewise exponential envelopes $s \leq f \leq g^*$. It turns out to be relatively easy to generate values from the distribution, which has the piecewise exponential unnormalized density g^*. After this has been accomplished, we can immediately use the accept–reject method with squeezing to simulate from the log-concave density f.

It is even possible to construct iteratively better and better upper and lower envelopes for a log-concave density, so that the bounds get tighter every time a new value is generated from the density. This is called **adaptive rejection sampling (ARS)**, but there exist several different implementations of this basic idea.

3.6 Using the multiplication rule for multivariate distributions

Suppose we want to simulate the joint distribution of three variables X, Y and Z. The multiplication rule (i.e., the chain rule) gives us a decomposition of the joint distribution of the form

$$f_{X,Y,Z}(x, y, z) = f_X(x)\, f_{Y|X}(y \mid x)\, f_{Z|X,Y}(z \mid x, y).$$

If all the distributions on the right are available in the sense that we know how to simulate from them, then we can be interpret the multiplication rule as a recipe for simulating the joint distribution. This is often called the **composition method** for simulating a multivariate distribution.

Algorithm 10: Composition method, pedantic version

```
Generate the value $x$ from $f_X$;
Generate the value $y$ from $f_{Y|X}(\cdot \mid x)$;
Generate the value $z$ from $f_{Z|X,Y}(\cdot \mid x, y)$.
```

If we repeat the process, we get i.i.d. samples

$$(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), \ldots$$

from the joint distribution of (X, Y, Z). Of course, one can generalize this to as many components as are needed. The components need not be scalars, but they may as well be vectors or even matrices.

Many people tend to describe the same algorithm more informally, e.g., as follows.

Algorithm 11: Composition method, informal version

```
Generate $x \sim p(x)$;
Generate $y \sim p(y \mid x)$;
Generate $z \sim p(z \mid x, y)$.
```

This is acceptable, if both the writer and the reader understand what this is supposed to mean. However, the danger of misunderstanding (or rather, not understanding anything) is great.

3.7 Mixtures

It is instructive to consider the special case of the multiplication rule, when there are just two components. It is useful to the check what the marginal distribution of the first component looks like.

Suppose X is continuous and J is discrete with values $1, 2, \dots, k$. Then their joint distribution has the density

$$f_{X,J}(x, j) = f_{X \mid J}(x \mid j) f_J(j).$$

Let us denote

$$p_j = f_J(j), \quad \text{and} \quad f_j = f_{X \mid J}(\cdot \mid j), \qquad j = 1, 2, \dots, k.$$

Then the marginal density of X is a convex combination of the densities f_j,

$$f_X(x) = \sum_{j=1}^{k} p_j f_j(x), \quad \text{where} \quad p_j \geq 0 \; \forall j, \quad \sum_{j=1}^{k} p_j = 1. \tag{3.23}$$

If we have a representation of the form (3.23), where the functions f_j are densities, then we say that the density of X is a (finite) mixture of the densities $f_1, \dots, f_k$. The numbers $p_1, \dots, p_k$ can be called mixing weights. We can simulate such a finite mixture distribution as follows.

Algorithm 12: Simulating from a finite mixture of distributions

```
Generate $J$ from the pmf $(p_1, p_2, \dots, p_k)$;
Generate $X$ from density $f_J$;
Return $X$ (and ignore $J$).
```

Example 3.2. [The Laplace distribution considered as a finite mixture] The (standard) Laplace distribution has the pdf

$$f(x) = \frac{1}{2} \, \mathrm{e}^{-|x|} = \frac{1}{2} \, 1_{(-\infty,0)}(x) \, \mathrm{e}^{x} + \frac{1}{2} \, 1_{[0,\infty)}(x) \, \mathrm{e}^{-x}$$

Here the mixing weights are $\frac{1}{2}$ and $\frac{1}{2}$, and the second of the densities is the familiar exponential density $\mathrm{Exp}(x \mid 1)$. Furthermore, it is easy to see that the first density is the density of the random variable $-X$, when $X \sim \mathrm{Exp}(1)$. These observations yield the following simulation recipe.

1. Generate $X \sim \mathrm{Exp}(1)$.
2. Generate a random sign J such that $P(J = -1) = P(J = 1) = \frac{1}{2}$.
3. Return JX.

△

Similarly, if the distribution of (X, Y) is continuous, then the marginal distribution of X is

$$f_X(x) = \int f_{X \mid Y}(x \mid y) \, f_Y(y) \, \mathrm{d}y. \tag{3.24}$$

If we have a representation of the form (3.24), then we say that the distribution of X is a (continuous) mixture of the densities $f_{X \mid Y}$. In such a case, simulation can be implemented as follows.

Algorithm 13: Simulating from a continuous mixture of distributions

```
Generate $y$ from density $f_Y$;
Generate $X \sim f_{X \mid Y}(\cdot \mid y)$;
Return $X$ (and ignore $y$).
```

Some important distributions can be represented in the form (3.24) so that y is the scale parameter of the family of distributions

$$\{f_{X|Y}(\cdot \mid y) : y > 0\}.$$

In this case we can say that the distribution of X is a scale mixture of the distributions $f_{X|Y}$.

Example 3.3. [Simulating the multivariate t distribution] Let $\nu > 0$, $\mu \in \mathbb{R}^d$ and let Σ be a symmetric, positive definite $d \times d$ matrix. The multivariate t distribution $t_d(\nu, \mu, \Sigma)$ can be represented hierarchically as a scale mixture of multivariate normal distributions

$$X \mid Y \sim N_d(\mu, \Sigma/Y), \quad \text{where} \quad Y \sim \mathrm{Gam}(\nu/2, \nu/2).$$

Therefore it can be simulated as follows

1. Generate $Y \sim \mathrm{Gam}(\nu/2, \nu/2)$.
2. Generate $Z \sim N_d(0, \Sigma)$.
3. Return $X = \mu + Z/\sqrt{Y}$.

We will discuss methods for simulating the multivariate normal distribution in the next Section.

The multivariate t distribution has become popular in Monte Carlo studies since its location and shape can be adjusted (by varying μ and Σ) and since it has heavier tails than the corresponding multivariate normal distribution. △

3.8 Affine transformations

Affine transformations of random vectors are multivariate analogs of scaling and shifting of univariate random variables. If d-dimensional Z has density f_Z and X is defined by

$$X = b + AZ,$$

where $b \in \mathbb{R}^d$ is a constant vector, and A is an invertible, constant $d \times d$ matrix, then X has the density

$$f_X(x) = \frac{f_Z(A^{-1}(x - b))}{|\det(A)|}. \tag{3.25}$$

Example 3.4. [Another multivariate generalization of the t distribution] Let $Z_1, \ldots Z_d$ be independent random variables so that Z_i has the univariate t distribution with $\nu_i > 0$ degrees of freedom. Then the vector $Z = (Z_1, \ldots, Z_d)$ has the pdf

$$f_Z(z) = \prod_{i=1}^{d} t(z_i \mid \nu_i),$$

which is **not** the same as the elliptically contoured multivariate t density we discussed earlier (not even when $\nu_1 = \cdots = \nu_d$). The random vector $X = b + AZ$ has the density given by (3.25). Confusingly, some authors call the resulting distribution of X the multivariate t distribution. △

We next apply affine transformations in order to simulate the multivariate normal distribution $N_d(\mu, \Sigma)$. Here $\mu \in \mathbb{R}^d$ is the mean (vector) of the distribution, and Σ, the covariance matrix of the distribution, is a $d \times d$ matrix. Σ is always symmetric and positive semidefinite. We now assume that Σ is positive definite, in which case it is also invertible. Then the $N_d(\mu, \Sigma)$ distribution has a density given by

$$f_X(x) = N_d(x \mid \mu, \Sigma) = (2\pi)^{-d/2} (\det \Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right). \tag{3.26}$$

For any symmetric, positive definite matrix Σ it is possible to find a matrix A such that

$$\Sigma = AA^T, \quad A \text{ is } d \times d \text{ and invertible} \tag{3.27}$$

One method for finding A is to use the Cholesky decomposition $\Sigma = LL^T$, where L (the lower triangular Cholesky factor of Σ) is a lower triangular matrix. Another possible choice is to use the symmetric, positive definite square root of Σ, often denoted by $\Sigma^{1/2}$, as the matrix A.

Let us consider, what is the density of the vector $Z = (Z_1, \ldots, Z_d)$, when $Z_i \sim N(0,1)$ independently $i = 1, \ldots, d$. Then

$$f_Z(z) = \prod_{i=1}^d \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}z_i^2} = (2\pi)^{-d/2} \exp\left(-\frac{1}{2} z^T z\right).$$

This is the d-dimensional standard normal distribution $N_d(0, I)$.

Suppose we have available the decomposition (3.27) and calculate as follows.

1. Generate $Z \sim N_d(0, I)$.
2. Return $X = \mu + AZ$.

Then it can be proved that $X \sim N_d(\mu, \Sigma)$ either directly from eq. (3.25) or by using familiar properties of the multivariate normal distribution (i.e., an affine transform of a multivariate normal rv also has a multivariate normal distribution).

Sometimes one has to simulate a high-dimensional normal distribution $N_d(\mu, \Sigma)$ whose covariance matrix Σ is not explicitely available but whose precision matrix $Q = \Sigma^{-1}$ (inverse covariance matrix) is known. In some statistical models the precision matrix is a huge sparse matrix (i.e., most of its entries are zeros) whereas the covariance matrix is a full matrix (i.e., practically all entries are non-zero). In some situations there is not enough computer memory available to store the covariance matrix, whereas the non-zero entries of the precision matrix can be stored easily.

Suppose that one is able to obtain a decomposition

$$Q = BB^T$$

for the precision matrix. Then one can simulate the distribution as follows

1. Generate $Z \sim N_d(0, I)$.
2. Solve Y from the linear equation $B^T Y = Z$, and return $X = \mu + Y$.

This follows since Y now has the normal distribution $N_d(0, (B^T)^{-1}((B^T)^{-1})^T)$, where

$$(B^T)^{-1}((B^T)^{-1})^T = (B^T)^{-1}B^{-1} = (BB^T)^{-1} = Q^{-1}.$$

Notice that solving high-dimensional linear equation is computationally much easier than matrix inversion.

Another possibility is that one is able to generate efficiently from the normal distribution $N_d(0, Q)$ whose covariance matrix Q is the precision matrix of the target distribution. Then one can do as follows

1. Generate $Z \sim N_d(0, Q)$.

2. Solve Y from $QY = Z$, and return $X = \mu + Y$.

3.9 Literature

The following text books are good references for the topics of this chapter.

Bibliography

[1] Luc Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, 1986.

[2] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods.* Wiley, 2011.

[3] Averll M. Law and W. David Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, Inc., 2nd edition, 1991.

[4] Brian D. Ripley. *Stochastic Simulation.* John Wiley & Sons, 1987.

[5] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer, second edition, 2004.

[6] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo method.* Wiley, 2nd edition, 2008.

Chapter 4

Classical Monte Carlo

In this chapter we discuss methods, which use an i.i.d. sample $X_1, X_2, \dots$ from some distribution. Then one speaks of classical Monte Carlo, or i.i.d. Monte Carlo or good old-fashioned Monte Carlo. In a later chapter we will discuss MCMC methods, where the underlying random variables are not independent and where they do not have identical distributions.

Monte Carlo methods are computational methods, which depend on the use of random or pseudo random numbers. The name Monte Carlo refers to the famous casino located in Monaco. Like casino games, Monte Carlo methods are highly repetitive and depend on randomness.

4.1 Limit theorems

When the underlying sample is i.i.d., one can use the two most important limit theorems of probability theory to analyze the behavior of arithmetic means.

Theorem 2 (Strong law of large numbers, SLLN)**.** *Let $Y_1, Y_2, \dots$ be i.i.d. random variables such that $E|Y_i| < \infty$. Denote $\mu = EY_i$. Then*

$$\frac{1}{n} \sum_{i=1}^{n} Y_i \to \mu,$$

almost surely, as $n \to \infty$.

Remark. The condition $E|Y_i| < \infty$ guarantees that the expectation EY_i is defined and finite. If $E|Y_i| = \infty$, then it can be shown that

$$\limsup_{n \to \infty} \left| \frac{1}{n} \sum_{i=1}^{n} Y_i \right| \to \infty$$

almost surely, which means that the sample mean oscillates wildly and therefore diverges.

Theorem 3 (Central Limit Theorem, CLT)**.** *Let $Y_1, Y_2, \dots$ be i.i.d. random variables such that $EY_i^2 < \infty$. Denote*

$$\mu = EY_i, \qquad \sigma^2 = \operatorname{var} Y_i,$$

and assume that $\sigma^2 > 0$. *Then*

$$\frac{\frac{1}{n}\sum_{i=1}^n Y_i - \mu}{\sigma/\sqrt{n}} \xrightarrow{\mathrm{d}} N(0,1), \tag{4.1}$$

as $n \to \infty$.

In the CLT the arrow $\xrightarrow{\mathrm{d}}$ denotes convergence in distribution. Random variables $Z_1, Z_2, \dots$ converge in distribution to a limit distribution with df F, if

$$P(Z_n \le z) \to F(z), \quad \text{as } n \to \infty$$

at all points of continuity z of F. Since in the CLT the df of the limit distribution $N(0,1)$ is continuous, in the CLT the convergence of the distribution functions holds at each point.

In CLT the quantity in (4.1) which has a limit distribution is the standardized mean of the n first random variables. I.e., if we denote

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n Y_i,$$

then

$$E\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n \mu = \mu$$

and

$$\begin{aligned}\operatorname{var}\bar{Y}_n &= E\left[\left(\frac{1}{n}\sum_{i=1}^n (Y_i - \mu)\right)^2\right] = \frac{1}{n^2} E\left[\sum_{i=1}^n (Y_i - \mu)\sum_{j=1}^n (Y_j - \mu)\right] \\ &= \frac{1}{n^2} n\sigma^2 = \frac{1}{n}\sigma^2.\end{aligned}$$

Therefore the numerator is $\bar{Y}_n$ minus its expectation, and the denominator is the standard deviation of $\bar{Y}_n$.

4.2 Confidence intervals for means

Let $Y_1, \dots, Y_n$ be i.i.d. random variables with mean μ and finite variance $\sigma^2 > 0$. If the sample size n is large, then we can pretend that the standardized mean already follows its limit distribution, i.e., we can pretend that

$$\frac{\bar{Y}_n - \mu}{\sigma/\sqrt{n}} \overset{\mathrm{d}}{=} N(0,1).$$

This an example of normal approximation.

Suppose we know σ but do not know μ. Then we can calculate a confidence limit for μ as follows. We seek a central $100(1-\alpha)\%$ confidence interval, for some $0 < \alpha < 1$. Let $z_{1-\alpha/2}$ be the value of the quantile function of the standard normal $N(0,1)$ at $1 - \alpha/2$, i.e., a proportion $1 - \alpha/2$ of the probability mass of $N(0,1)$ lies to the left of $z_{1-\alpha/2}$. E.g., a 95 % confidence interval corresponds to $\alpha = 0.05$ and $z_{0.975} \approx 1.96$. We start from the normal approximation

$$P\left(\left|\frac{\bar{Y}_n - \mu}{\sigma/\sqrt{n}}\right| \le z_{1-\alpha/2}\right) \approx 1 - \alpha.$$

When we solve the inequality for μ, we see that approximately with probability $1-\alpha$ we have

$$\mu \in \bar{Y}_n \pm z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}.$$

Usually not only μ but also σ would be unknown. However, we can still apply the preceding confidence interval, when we plug in a reasonable estimate $\hat{\sigma}$ of the standard deviation σ. Usually one uses the sample standard deviation of the Y_i values,

$$\hat{\sigma} = s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_n\right)^2},$$

which is a consistent estimate of σ. With this choice, we get the approximate $(1-\alpha)100\%$ confidence interval for the mean μ

$$\mu \in \bar{Y}_n \pm z_{1-\alpha/2} \frac{s}{\sqrt{n}}. \tag{4.2}$$

Here the quantity

$$\frac{s}{\sqrt{n}} = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} \left(Y_i - \bar{Y}_n\right)^2},$$

is called the standard error of the mean.

Instead of the critical values of the standard normal, one often uses the critical values of the t distribution with $(n-1)$ degrees of freedom in the previous construction. If the sample size is large, then the resulting confidence interval is in practice the same as (4.2).

There is nothing probabilistic about the coverage a single confidence interval: the interval either contains μ or does not. However, if one constructs a large number of $(1-\alpha)100\%$ confidence intervals (4.2), where n is large, then approximately proportion $(1-\alpha)$ of them covers μ and proportion α does not cover μ.

Our confidence interval (4.2) is valid only for a fixed sample size n. It is also possible to develop confidence bands for *the running mean plot*, which plots $\bar{Y}_m$ against m, see the books by Robert and Casella [6, 7].

4.3 Confidence intervals for ratios of means

Sometimes we need an estimate of the ratio

$$r = \frac{EX}{EY}, \tag{4.3}$$

when we have an i.i.d. sample $(X_1, Y_1), \dots, (X_n, Y_n)$ from the joint distribution of X and Y. A natural estimator is the ratio of the averages,

$$\hat{r} = \frac{\bar{X}}{\bar{Y}}, \qquad \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i, \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i. \tag{4.4}$$

This estimator is consistent but biased. Now we develop an approximate confidence interval for it.

Consider the random variable $Z = X - rY$, which has mean zero and variance

$$\sigma^2 = \operatorname{var}(Z) = \operatorname{var}(X) - 2r \operatorname{cov}(X, Y) + r^2 \operatorname{var}(Y). \tag{4.5}$$

Let $Z_i = X_i - rY_i$. Then the Z_i are i.i.d. random variables and therefore the CLT ensures that

$$\frac{\bar{Z}}{\sigma/\sqrt{n}} = \frac{\bar{X} - r\bar{Y}}{\sigma/\sqrt{n}} = \frac{\hat{r} - r}{\sigma/(\sqrt{n}\bar{Y})}$$

converges in distribution to $N(0, 1)$ as $n \to \infty$. Normal approximation gives now

$$P\left(\left|\frac{\hat{r} - r}{\sigma/(\sqrt{n}\,\bar{Y})}\right| \le z_{1-\alpha/2}\right) \approx 1 - \alpha$$

Therefore, an approximate $(1-\alpha)100\%$ confidence interval for the ratio EX/EY is

$$\hat{r} \pm z_{1-\alpha/2} \frac{S}{\sqrt{n}\,\bar{Y}} \tag{4.6}$$

where

$$S^2 = S_X^2 - 2\hat{r}\, S_{XY}^2 + 2\hat{r}^2\, S_Y^2$$

is the estimator of σ^2 where the unkwnown population parameters have been replaced by their consistent estimators, and

$$S_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad S_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

$$S_{XY}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

We see from (4.6) that in large samples $\hat{r}$ is approximately normally distributed with mean r and with variance

$$\operatorname{var}(\hat{r}) \approx \frac{1}{n}\frac{S^2}{(\bar{Y})^2}. \tag{4.7}$$

Remark. This simple derivation of the results for the ratio estimator has been borrowed from [8, Sec. 4.3.2.2]. The same results can be derived based on a multivarate version of the CLT and the delta method.

4.4 Basic principles of Monte Carlo integration

Suppose f is a density, which we are able to to simulate from, and that we are interested in the expectation

$$I = \int h(x) f(x) \,\mathrm{d}x = Eh(X). \tag{4.8}$$

Suppose that we simulate $X_1, X_2, \ldots$ independently from the density f and set $Y_i = h(X_i)$. Then the sequence $Y_1, Y_2, \ldots$ is i.i.d. and

$$EY_i = Eh(X_i) = \int h(x) f(x) \,\mathrm{d}x = I.$$

If we calculate the mean of the N values $h(X_1), \dots, h(X_N)$, then we obtain the estimate

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i). \tag{4.9}$$

By the SLLN, $\hat{I}_N$ converges to I as N increases, provided that the condition $E|h(X)| < \infty$ holds. In Monte Carlo simulations we are free to select N as large as our budget (available computer time) allows.

We have

$$E\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} Eh(X_i) = I,$$

and therefore the estimate $\hat{I}_N$ is unbiased. It is also easy to express the variance and the standard error of the estimator. If the variance of a single term $h(X)$ is finite, then the variance of the average is

$$\operatorname{var} \hat{I}_N = \frac{1}{N} \operatorname{var} h(X). \tag{4.10}$$

This can be called the sampling variance, simulation variance or Monte Carlo variance of the estimator $\hat{I}_N$.

A more meaningful quantity for measuring the accuracy of $\hat{I}_N$ is the square root of the variance. Recall that the square root of the variance of an estimator (i.e., its standard deviation) is called its **standard error**. (This term is commonly used also for the estimate of the (theoretical) standard error.) The standard error of a Monte Carlo estimate can be called its sampling standard error, simulation standard error of Monte Carlo standard error. The Monte Carlo standard error is of the order $1/\sqrt{N}$, since

$$\sqrt{\operatorname{var} \hat{I}_N} = \frac{1}{\sqrt{N}} \sqrt{\operatorname{var} h(X)}. \tag{4.11}$$

The theoretical variance (population variance) $\operatorname{var} h(X)$, which is needed in both (4.10) and (4.11), is usually unknown. However, it it can be estimated by the sample variance of the $h(X_i)$ values,

$$s^2 = \widehat{\operatorname{var}}\, h(X) = \frac{1}{N-1} \sum_{i=1}^{N} \left(h(X_i) - \hat{I}_N \right)^2.$$

We get an approximate $100(1-\alpha)\%$ confidence interval for I from (4.2), namely

$$\hat{I}_N \pm z_{1-\alpha/2} \frac{s}{\sqrt{N}}. \tag{4.12}$$

Example 4.1. Calculating the 95 % confidence interval (4.12) with R. We assume that the sample from the density f is generated with the call `rname(N)`. We also assume that we have available a function `h`, which applies the function h element-by-element to its vector argument.

```
x <- rname(N)
# Calculate vector y so that y[i] = h(x[i]) for all i.
y <- h(x)
```

```
Ihat <- mean(y)
se <- sqrt(var(y) / N)
# or: se <- sd(y) / sqrt(N)
z <- qnorm(1 - 0.05/2)
ci <- c(Ihat - z * se, Ihat + z * se)
```

△

The accuracy of Monte Carlo integration goes to zero like $1/\sqrt{N}$ as N increases. To get an extra decimal place of accuracy it is necessary to increase N by a factor of 100. In practice, one usually achieves moderate accuracy with a moderate simulation sample size N. However, in order to achieve high accuracy, one usually needs an astronomical simulation sample size. Notice, however that Monte Carlo integration works equally well in a space of any dimensionality. In contrast, the classical quadrature rules of numerical analysis become prohibitively expensive in high dimensional spaces.

Notice, how versatile Monte Carlo integration is. If one wants to estimate several expectations $Eh_1(X), Eh_2(X), \ldots, Eh_k(X)$, then a single sample $X_1, \ldots, X_N$ from the density f suffices, since

$$Eh_j(X) \approx \frac{1}{N} \sum_{i=1}^{N} h_j(X_i), \qquad j = 1, \ldots, k.$$

In that case one uses *common random numbers* to estimate the different expectations.

4.5 Empirical quantiles

Often one wants to estimate the quantile function of a random variable X, when one has available a sample $X_1, \ldots, X_N$ (i.i.d. or not) from its distribution. Then one speaks of the **empirical quantile function**. This problem can be approached via Monte Carlo integration. One wants to solve x from the equation

$$E1_{(-\infty,x]}(X) = u, \qquad 0 < u < 1,$$

for various values of u. One can approximate the expectation by the Monte Carlo method. However, the resulting equation does not have a unique solution, as we will see in a moment.

Let $X_{(j)}$ be the j'th smallest observation, which is also called the j'th order statistic of the sample. I.e., the observations sorted from lowest to highest are

$$X_{(1)} \le X_{(2)} \le \cdots \le X_{(N)}.$$

If

$$X_{(j)} < x < X_{(j+1)}$$

for some $j = 1, \ldots, N$, then by Monte Carlo

$$E1_{(-\infty,x]}(X) \approx \frac{1}{N} \sum_{i=1}^{N} 1_{(-\infty,x]}(X_i) = \frac{j}{N}.$$

Therefore a reasonable value for the empirical quantile function at $u = j/N$ is some value between $X_{(j)}$ and $X_{(j+1)}$, and one can use various interpolation methods to extend the definition to all values $0 < u < 1$.

Different statistical computer packages use slightly different formulas to define the empirical quantile function. There is latitude in selecting the exact point at which the empirical quantile function takes on the j'th order statistic and latitude in how one interpolates in between. E.g., in R the empirical quantile function is calculated by the function `quantile()`, and the user can choose between nine definitions of the empirical quantile function. For a large sample from a continuous distribution, all the definitions calculate approximately the same results.

4.6 Techniques for variance reduction

It is always possible to estimate the unknown integral by using different representations of the form

$$\int h(x)f(x)\,\mathrm{d}x.$$

A clever choice may imply a significantly lower variance for the Monte Carlo estimator. Then one speaks of **variance reduction** methods.

E.g., to reduce variance, it is always a good idea to try to carry out the computation analytically as far as possible, and then use Monte Carlo integration only as a last resort.

Suppose that we have two Monte Carlo methods for estimating the same integral. Let the variance in method i be

$$\frac{v_i}{N}, \qquad i = 1, 2,$$

where N is the sample size employed. Then, in order to achieve the same accuracy (e.g., the same variance or the same standard error), we should use in method two the sample size

$$\frac{v_2}{v_1} N,$$

where N is the sample size used in method one.

4.6.1 Conditioning

Conditioning decreases variance in the sense that

$$\operatorname{var} E(Z \mid Y) \le \operatorname{var} Z$$

for any random variables Y and Z. In Monte Carlo integration it is therefore advantageous to use the conditional expectation of the integrand instead of the original integrand, whenever that is possible. Conditioning performs part of the original integration analytically, and the rest by Monte Carlo.

Conditioning is often called **Rao-Blackwellization**. (Explanation: in the celebrated Rao-Blackwell theorem one conditions on a sufficient statistic.)

To exemplify conditioning, suppose we want to estimate the integral

$$I = Eh(X, Y) = EE(h(X, Y) \mid Y),$$

and are able to compute the conditional expectation

$$m(y) = E[h(X,Y) \mid Y = y].$$

Then we can estimate I either by simulating $(X_i, Y_i), i = 1, \ldots, N$ from the joint distribution of (X, Y) and by calculating

$$\hat{I}_N^{(1)} = \frac{1}{N} \sum_{i=1}^{N} h(X_i, Y_i)$$

or by calculating

$$\hat{I}_N^{(2)} = \frac{1}{N} \sum_{i=1}^{N} m(Y_i).$$

Supposing that the computational effort required for evaluating $h(X_i, Y_i)$ or $m(Y_i)$ is about the same, the second method is better since its variance is lower.

One case where this idea can be used is in estimating posterior predictive expectations. We have often the situation, where in addition to the observed data we want to consider a future observation Y^*. The distribution of Y^* conditionally on the observed data $Y = y$ is its **(posterior) predictive distribution**. Typically, the data Y and future observation Y^* are modeled as conditionally independent given the parameter Θ. Then the joint posterior of Θ and Y^* factorizes as follows

$$p(y^*, \theta \mid y) = p(\theta \mid y)\, p(y^* \mid y, \theta) = p(\theta \mid y)\, p(y^* \mid \theta),$$

where the first identity follows by the multiplication rule for conditional distributions, and the second by conditional independence. Therefore we can simulate the joint posterior distribution of Y^* and Θ by first simulating θ_i from the posterior distribution $p(\theta \mid y)$ and then y_i^* from the sampling distribution of Y^* conditionally on the simulated value θ_i. We can estimate the mean $E[Y^* \mid Y = y]$ of the posterior predictive distribution by straightforward Monte Carlo as follows

$$\hat{I}_N^{(1)} = \frac{1}{N} \sum_{i=1}^{N} y_i^*.$$

However, in a typical situation we would know the mean of Y^* given the value of the parameter Θ, i.e., the mean of the sampling distribution of Y^*,

$$m(\theta) = E[Y^* \mid \Theta = \theta] = \int y^* p(y^* \mid \theta) \, \mathrm{d}y^*.$$

In this case we obtain a better estimator of $E[Y^* \mid Y]$ by conditioning,

$$\hat{I}_N^{(2)} = \frac{1}{N} \sum_{i=1}^{N} m(\theta_i).$$

The same approach applies also, when we want to estimate the expectation

$$E[h(Y^*) \mid Y = y],$$

where h is a function for which we know

$$\int h(y^*)\, p(y^* \mid \theta) \, \mathrm{d}y^*,$$

which is the expectation of $h(Y^*)$ given $\Theta = \theta$.

4.6.2 Control variates

Sometimes we want estimate the expectation $I = Eh(X)$ and know that

$$\mu = Em(X),$$

where m is a known function and μ is a known constant. By defining

$$W = h(X) - \beta(m(X) - \mu), \tag{4.13}$$

where β is a constant, we get a RV W, whose expectation is I. Since

$$\operatorname{var} W = \operatorname{var} h(X) - 2\beta \operatorname{cov}(h(X), m(X)) + \beta^2 \operatorname{var} m(X),$$

the lowest possible variance for W is obtained by selecting for β the value

$$\beta^* = \frac{\operatorname{cov}(h(X), m(X))}{\operatorname{var} m(X)}. \tag{4.14}$$

Here we must have $\operatorname{var}(m(X)) > 0$. If we use $\beta = \beta^*$ in (4.13), then

$$\operatorname{var} W = \operatorname{var} h(X) - \frac{\operatorname{cov}^2(h(X), m(X))}{\operatorname{var} m(X)}.$$

Notice that $\operatorname{var} W < \operatorname{var} h(X)$, if the RVs $h(X)$ and $m(X)$ are correlated, i.e., if $\operatorname{cov}(h(X), m(X)) \neq 0$. The stronger the correlation, the greater the variance reduction.

If we manage to select the value β so that $\operatorname{var} W < \operatorname{var} h(X)$, then we should estimate I as the mean of values W_i which are simulated from the distribution of W,

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} \left[h(X_i) - \beta(m(X_i) - \mu) \right]. \tag{4.15}$$

Here $X_1, \dots, X_N$ is an i.i.d. sample with the distribution of X. Here $m(X)$ is the **control variate**, whose expectation we know. The variance of the control variate estimator (4.15) is less than the variance of the naive Monte Carlo estimator, which just averages the values $h(X_i)$.

To understand, why this happens, suppose that $\operatorname{cov}(h(X), m(X))$ is positive. Then also β should be selected positive. In this case an unusually high outcome for $\bar{h}$, the sample average of the $h(X_i)$ values, tends to be associated with an unusually high outcome for $\bar{m}$ the sample average of the $m(X_i)$ values In that case the control variate estimate adjusts the naive Monte Carlo estimate $\bar{h}$ of $Eh(X)$ downward, i.e.,

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} [h(X_i) - \beta(m(X_i) - \mu)] = \bar{h} - \beta(\bar{m} - \mu),$$

where

$$\bar{h} = \frac{1}{N} \sum_{i=1}^{N} h(X_i), \qquad \bar{m} = \frac{1}{N} \sum_{i=1}^{N} m(X_i).$$

Similar explanation works also when the correlation is negative.

The optimal β^* depends on the moments of RVs $h(X)$ and $m(X)$, and these are usually unknown. However, we can estimate the optimal β by using a pilot sample X_i', $i = 1, \ldots, n$. We then divide the sample covariance of $h(X_i')$ and $m(X_i')$ with the sample variance of $m(X_i')$. This is then our estimate of β^*, which is then used in eq. (4.15) with a fresh sample $X_1, \ldots, X_N$.

Somewhat surprisingly, the same calculation can be done by fitting a linear model, as follows. We fit the linear model

$$h(X_i') = \alpha + \beta m(X_i') + \epsilon_i, \qquad i = 1, \ldots, n.$$

by least squares

$$\sum_{i=1}^{n} \left(h(X_i') - \alpha - \beta m(X_i')\right)^2 = \min!,$$

and this can be done by using any statistical package. Here the errors ϵ_i are definitely not normally distributed as would be required for linear models. We are just using the available software for linear models for our own purposes. This approach works, since the least squares estimate of β happens to be the same as calculated in the previous approach for estimating β^*. The estimated slope, $\hat{\beta}$, is then used in eq. (4.15) and the estimated intercept $\hat{\alpha}$ is ignored.

Example 4.2. Suppose that `rname(n)` simulates n values from the distribution of X and that `hfunc(x)` and `mfunc(x)` calculates the functions h and m for each value of its vector argument. Then the following code fragments demonstrates the two ways of estimating β^*.

```
x.pilot <- rname(n.pilot)
h <- hfunc(x.pilot); m <- mfunc(x.pilot)
beta <- cov(m, h) / var(m)
# Alternative; here the function lm() fits the linear model.
model <- lm(h ~ m)
# Select beta as the estimated coefficient for m:
beta <- coef(model)[2]

# Estimate the integral and the simulation standard error
x <- rname(n)
h <- hfunc(x); m <- mfunc(x)
w <- h - beta * (m - mu)
Ihat <- mean(w)
se <- sd(w) / sqrt(n)
```

△

If one knows several expectations

$$\mu_j = E m_j(X), \qquad j = 1, \ldots, k,$$

then it is possible to use several control variates $m_1(X), \ldots, m_k(X)$. The values of the optimal coefficients can, again, be estimated using a pilot sample and by fitting a linear model.

4.6.3 Common random numbers

Often one wants to compare two expectations

$$I_1 = E_f h_1(X), \qquad \text{and} \qquad I_2 = E_f h_2(X),$$

where the functions h_1 and h_2 resemble one another. Suppose we estimate the expectations by the Monte Carlo estimators $\hat{I}_1$ and $\hat{I}_2$. We are interested in the sign of the difference $I_1 - I_2$. Since

$$\operatorname{var}\left(\hat{I}_1 - \hat{I}_2\right) = \operatorname{var}(\hat{I}_1) + \operatorname{var}(\hat{I}_2) - 2\operatorname{cov}(\hat{I}_1, \hat{I}_2),$$

it is worthwhile to use estimators, which have positive correlation. This is typically achieved by basing the estimators $\hat{I}_1$ and $\hat{I}_2$ on common random numbers, i.e., by using a single sample $X_1, \dots, X_N$ instead of separate samples for the two estimators.

Using common random numbers is even more important in the case, where one tries to estimate a parametrized expectation

$$I(\alpha) = E_f h(X, \alpha)$$

for various values of the parameter α. Then the estimator using common random numbers produces a much smoother approximation

$$\alpha \mapsto \hat{I}(\alpha)$$

then what would be obtained by using separate samples at each α. Besides, by using common random numbers one saves a lot of computational effort.

4.7 Importance sampling

Suppose we want to estimate the integral

$$I = E_f[h(X)] = \int h(x)\, f(x)\, \mathrm{d}x, \tag{4.16}$$

where the density f might be difficult to sample from. We can rewrite the integral as

$$I = \int_{\{g>0\}} h(x)\frac{f(x)}{g(x)}\, g(x)\, \mathrm{d}x = E_g\left[h(X)\frac{f(X)}{g(X)}\right]. \tag{4.17}$$

Here the subscript of the expectation symbol shows, under what distribution the expectation is calculated. Robert and Casella [6] call this the importance sampling **fundamental identity**. In importance sampling, one draws a sample from g and uses the fundamental identity for developing a Monte Carlo estimate of the integral. This idea was used already in the early 1950's.

The new density g can be selected otherwise quite freely, but we must be certain that

$$g(x) = 0 \quad \Rightarrow \quad h(x)f(x) = 0,$$

since otherwise the integrals (4.16) and (4.17) are not guaranteed to be equal. In other words, the support of the function hf must be included in the support of the function g.

Of course, the fundamental identity can be formulated for other types of distributions, too. If X has a discrete distribution with pmf f, then we may estimate the sum

$$I = E_f[h(X)] = \sum_x h(x)\, f(x)$$

by drawing a sample from another pmf g with the same support, since

$$I = \sum_x h(x)\, \frac{f(x)}{g(x)}\, g(x).$$

Again, the support of the function hf must be included in the support of the function g.

4.7.1 Unbiased importance sampling

We assume the setting (4.17), where f is the pdf of a continuous distribution. We assume that we know f completely, including its normalizing constant.

We select a density g, which is easy to sample from. Then we generate a sample $X_1, \ldots, X_N$ from g and calculate

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i)\, \frac{f(X_i)}{g(X_i)} \tag{4.18}$$

Let us call the ratio of the densities

$$w(x) = \frac{f(x)}{g(x)}$$

the importance ratio (or likelihood ratio), and the weights

$$w_i = w(X_i) = \frac{f(X_i)}{g(X_i)}, \qquad i = 1, \ldots, N \tag{4.19}$$

the **importance weights**. Then the importance sampling estimate (4.18) can be written as

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} w_i h(X_i).$$

Importance sampling gives more weight for those sample points X_i for which $f(X_i) > g(X_i)$ and downweights the other sample points, in order to form an unbiased estimate of $I = E_f[h(X)]$, given a sample $X_1, \ldots, X_N$ from g.

Different authors use different names for g such as the importance sampling density, the approximation density, proposal density and so on. Following Robert and Casella [6], we call g the **instrumental** density.

We can interpret the procedure as producing a **weighted sample**

$$(w_1, X_1), \ldots, (w_N, X_N),$$

where the weights are needed in order to correct for the fact that the sample is produced from the wrong density. Since the estimator (4.18) is the arithmetic mean of terms $w_i\, h(X_i)$ each with mean I,

$$E_g[w_i\, h(X_i)] = E_g\left[\frac{f(X_i)}{g(X_i)}\, h(X_i)\right] = \int h(x) f(x)\, \mathrm{d}x = I,$$

the estimator is unbiased. Its variance can be estimated in the same way as the variance of the basic Monte Carlo estimator.

In importance sampling we should strive for low variance. In particular, the variance should be finite. This is the case, if and only if the expectation of the square of one term is finite, i.e., we should have

$$E_g \left[h^2(X) \frac{f^2(X)}{g^2(X)} \right] = \int h^2(x) \frac{f^2(x)}{g(x)} \, \mathrm{d}x < \infty.$$

If this condition is not satisfied, then the estimator behaves erratically.

One pair of conditions which guarantees finite variance is

$$\mathrm{var}_f[h(X)] < \infty, \quad \text{and} \quad \frac{f(x)}{g(x)} \le M, \quad \forall x,$$

for some $M > 0$. The second of these conditions means that the tails of g should be at least as heavy as the tails of f.

In order to achieve minimal variance, one can show that it is optimal to choose the instrumental density g proportional to $|h|f$. Then the variance of the importance sampling estimator is smaller (or equal to) the variance of the naive Monte Carlo estimator, which uses samples from f. While the optimal choice

$$g \propto |h| f$$

can hardly ever be used in practice, it can still provide some guidance in choosing the form of g: the shape of the instrumental density should resemble the product $|h|f$ as closely as possible. One should focus sampling on the regions of interest where $|h|f$ is large in order to save computational resources.

On the other hand, if the integrand h is not fixed in advance (e.g., one wants to estimate expectations for many functions h) then the instrumental density g should be selected so that $f(x)/g(x) = w(x)$ is nearly constant. If g is a good approximation to f, then all the importance weights will be roughly equal. If, on the other hand, g is a poor approximation to f, then most of the weights will be close to zero, and thus a few of the X_i's will dominate the sum, and the estimate will be inaccurate. Therefore it is a good idea to inspect the importance weights, e.g., by examining their variance or histogram.

Notice that the importance weights can be utilized to form a control variate. Denoting the importance weight w_i by $w(X_i)$, we have

$$E_g w(X_i) = \int \frac{f(x)}{g(x)} \, g(x) \, \mathrm{d}x = 1.$$

Therefore the average of the importance weights can be used as a control variate, whose expectation is known to be one.

4.7.2 Self-normalized importance sampling

It is possible to apply importance sampling also in the situation, where we want to estimate $I = E_f[h(X)]$, but only know an unnormalized version f^* of the density f. Here

$$f(x) = \frac{1}{c} f^*(x),$$

but the normalizing constant c is unknown. Of course, c can be expressed as the integral

$$c = \int f^*(x)\, \mathrm{d}x.$$

Such a situation is common in Bayesian statistics, but also when f^* corresponds to a truncated density. In these cases we cannot calculate (4.18) directly. However, we can express the integral as

$$I = \int h(x) f(x)\, \mathrm{d}x = \frac{\int h(x) f^*(x)\, \mathrm{d}x}{\int f^*(x)\, \mathrm{d}x},$$

and then estimate the numerator and denominator separately using importance sampling.

We sample $X_1, \ldots, X_N$ from an instrumental density g. We estimate the denominator by

$$\int f^*(x)\, \mathrm{d}x = \int \frac{f^*(x)}{g(x)}\, g(x)\, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^N \frac{f^*(X_i)}{g(X_i)} = \frac{1}{N} \sum_{i=1}^N w_i,$$

where we use the importance weights w_i corresponding to the unnormalized density f^*, given by

$$w_i = \frac{f^*(X_i)}{g(X_i)}.$$

Our estimate of the numerator is

$$\int h(x) f^*(x)\, \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^N h(X_i) \frac{f^*(X_i)}{g(X_i)} = \frac{1}{N} \sum_{i=1}^N w_i h(X_i).$$

Canceling the common factor $1/N$, we obtain the following self-normalized importance sampling estimator (which is usually called just the importance sampling estimator without any further qualification).

1. Generate $X_1, X_2, \ldots, X_N$ from density g.
2. Calculate the importance weights

$$w_i = \frac{f^*(X_i)}{g(X_i)}$$

3. Estimate I by the weighted average

$$\hat{I} = \frac{\sum_{i=1}^N w_i h(X_i)}{\sum_{j=1}^N w_j}. \tag{4.20}$$

The same method can be described so that having calculated the (raw) importance weights w_i, one calculates the **normalized importance weights**,

$$\tilde{w}_i = \frac{w_i}{s}, \quad \text{where} \quad s = \sum_{j=1}^n w_j,$$

by dividing the raw weights by their sum, and then calculates the (self-normalized) importance sampling estimate as

$$\bar{I} = \sum_{i=1}^{N} \tilde{w}_i \, h(X_i).$$

Unlike the unbiased estimator (4.18), the self-normalized estimator (4.20) is not unbiased. Its bias is, however, negligible when N is large.

In both forms of importance sampling it is a good idea to inspect the importance weights w_i. If only few of the weights are large and others are negligible, then the estimate is likely not accurate. In self-normalized importance sampling one can examine the histogram or the coefficient of variation (which is the sample standard deviation divided by the sample mean) of the importance weights (standardized or not).

4.7.3 Variance estimator for self-normalized importance sampling

One should view the self-normalized estimator (4.20) as the ratio of two averages

$$\hat{I} = \frac{\frac{1}{N}\sum_{i=1}^{N} h(x_i) w_i}{\frac{1}{N}\sum_{j=1}^{N} w_j} = \frac{\bar{U}}{\bar{w}},$$

where $\bar{U}$ is the average of N RVs $U_i = h(X_i) w_i$ and $\bar{w}$ is the average of N raw importance weights w_i, and the pairs (U_i, w_i) are i.i.d. random vectors. Then we can apply the formula (4.7) for the approximate variance of the ratio,

$$\frac{1}{N} \frac{1}{\bar{w}^2} S^2,$$

where

$$\begin{aligned} (N-1)S^2 &= \sum (U_i - \bar{U})^2 - 2\frac{\bar{U}}{\bar{w}} \sum (U_i - \bar{U})(V_i - \bar{V}) + \frac{\bar{U}^2}{\bar{w}^2} \sum (w_i - \bar{w})^2 \\ &= \sum (h(X_i) - \hat{I})^2 w_i^2, \end{aligned}$$

after some straightforward calculations.

This implies the following formula for the Monte Carlo variance of the self-normalized importance sampling estimator,

$$\begin{aligned} \widehat{\operatorname{var}}\, \hat{I} &= \frac{1}{N(N-1)} \frac{1}{\bar{w}^2} \sum_{i=1}^{N} (h(X_i) - \hat{I})^2 \, w_i^2 \qquad (4.21) \\ &= \frac{N}{N-1} \frac{\sum_{i=1}^{N} (h(X_i) - \hat{I})^2 \, w_i^2}{(\sum_{j=1}^{N} w_j)^2}. \end{aligned}$$

Geweke [2] omits the term $N/(N-1)$ which corresponds to using denominator N instead of $N-1$ in the formulas for the sample variances and covariances.

A $(1-\alpha)100\%$ confidence interval for I is then given by

$$\hat{I} \pm z_{1-\alpha/2} \sqrt{\widehat{\operatorname{var}}\, \hat{I}}.$$

4.7.4 SIR: Sampling importance resampling

Importance sampling can be interpreted so that it produces a weighted sample $(p_1, X_1), \ldots, (p_N, X_N)$, where now $(p_1, \ldots, p_N)$ is a probability vector (i.e., a probability mass function on $1, \ldots, N$). Then $I = E_f[h(X)]$ is approximated by

$$\sum_{i=1}^{N} p_i h(X_i).$$

The probability vector is here the vector of normalized importance weights.

However, for some purposes one needs a true sample; a weighted sample does not suffice. Such a sample can be produced approximately by sampling with replacement from the sequence

$$X_1, \ldots, X_N$$

with probabilities given by the vector $(p_1, \ldots, p_N)$. This is called SIR (sampling/importance resampling). Following Smith and Gelfand [9], this approach is sometimes called the weighted bootstrap.

If one samples without replacement, then one obtains an i.i.d. sample, which comes from an approximation to the target distribution. The approximation improves as the size of the initial sample N increases. (Sampling with replacement does not here result in an i.i.d. sample.)

4.8 Literature

Monte Carlo integration and variance reduction methods are discussed in the simulation literature, see e.g., Law and Kelton [4], Rubinstein and Kroese [8], the handbook [3] and Asmussen and Glynn [1]. Ripley [5] demonstrates how one can reduce the simulation variance by a factor of 10^8 by using variance reduction techniques cleverly.

Bibliography

[1] Søren Asmussen and Peter W. Glynn. *Stochastic Simulation.* Springer, 2007.

[2] John Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57:1317–1339, 1989.

[3] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods.* Wiley, 2011.

[4] Averll M. Law and W. David Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, Inc., 2nd edition, 1991.

[5] Brian D. Ripley. *Stochastic Simulation.* John Wiley & Sons, 1987.

[6] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer, second edition, 2004.

[7] Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R.* Springer, 2010.

[8] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo method*. Wiley, 2nd edition, 2008.

[9] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

Appendix A

Probability distributions

This appendix contains a summary of certain common distributions. Each distribution has a symbol, and depends on a number of parameters. We use the symbol of the distribution to denote its probability mass function (pmf) or probability density function (pdf) writing the argument on the left-hand side of the vertical bar, and the parameters on its right-hand side. For instance, the binomial distribution with sample size parameter n and probability parameter p is denoted $\text{Bin}(n, p)$, and its pmf at argument x is denoted $\text{Bin}(x \mid n, p)$. The normal distribution with mean μ and variance σ^2 is denoted $N(\mu, \sigma^2)$, and its pdf at x is denoted by $N(x \mid \mu, \sigma^2)$. Notice that different authors and different computing environments use different parametrizations for the distributions. We illustrate the distributions using the R language.

A.1 Probability distributions in the R language

R is an open-source general purpose statistical package, where one uses the R language. It is very handy for experimenting with various distributions.

The R language has available facilities for calculating the density function, the distribution function, the quantile function and for simulating the distribution for a wide variety of univariate distributions. For a discrete distribution, density function means the probability mass function. The values of the functions are calculated by calling functions, which all have the same naming conventions. Each built-in distribution of the R language has an R name, which is an abbreviation of the name of the distribution. For each R name `name`, there are four functions:

- `dname` calculates the density,
- `pname` calculates the distribution function,
- `qname` calculates the quantile function,
- `rname` simulates the distribution.

E.g., the univariate normal distribution has the R name `norm`, so R has the functions `dnorm`, `pnorm`, `qnorm` and `rnorm`. For the uniform distribution on an interval, the R name is `unif` and R has the functions `dunif`, `punif`, `qunif` and `runif`, and so on for other distributions.

One can ask the d-, p- and q-forms of theses functions to calculate the natural logarithms instead of the original quantities For instance, `dnorm(x)` calculates $\phi(x)$, the value of the standard normal density at the point x, whereas `dnorm(x, log = TRUE)` calculates $\ln(\phi(x))$. In the p- and q-functions one can specify that the quantities pertain to the upper tail instead of the lower tail. E.g., `pnorm(x)` calculates $\Phi(x)$, the value of the standard normal cdf at the point x, `prnorm(x, log = TRUE)` calculates $\ln(\Phi(x))$, whereas `pnorm(x, lower.tail = FALSE)` calculates the complement of the cdf,

$$1 - \Phi(x) = P(Z > x), \quad \text{when} \quad Z \sim N(0,1).$$

The R names for some standard univariate discrete distributions are

`binom, nbinom, pois, geom, hyper.`

The R names for some standard univariate continuous distributions are

`unif, norm, lnorm, chisq, t, f, exp, gamma, weibull, cauchy, beta.`

You can read the documentation of the functions, e.g., by giving the command `?dname`, where `name` is the R name of the distribution. There you can find information on how R parametrizes the distributions. In R, the parameters of functions can have default values, and you do not need to give those function parameters, whose default values are what you want.

The support for multivariate distributions is not as systematic as for univariate distributions. For many multivariate distributions there are only functions for calulating its pdf/pmf and for drawing random values from it. For some multivariate distributions a function is available for calculating the multivariate cumulative distribution function. Notice that the quantile function is only defined for univariate distributions.

A.2 Gamma and beta functions

Gamma and beta functions are special functions which are needed for the normalizing constants of some of the standard distributions.

Gamma function can be defined by the integral

$$\Gamma(z) = \int_0^\infty x^{z-1} \mathrm{e}^{-x} \, \mathrm{d}x, \qquad z > 0. \tag{A.1}$$

It satisifies the functional equation

$$\Gamma(z+1) = z\,\Gamma(z), \qquad \text{for all } z > 0, \tag{A.2}$$

and besides $\Gamma(1) = 1$, from which it follows that

$$\Gamma(n) = (n-1)!, \quad \text{when } n = 1, 2, 3, \ldots.$$

Therefore the gamma function is a generalization of the factorial. The value of, $\Gamma(z)$ for half-integer arguments can be calculated using its functional equation and the value $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.

Evaluating $\Gamma(z)$ with R:

`gamma(z)`

Evaluating $\ln(\Gamma(z))$ with R:

`lgamma(z)`

Beta function can be defined by the integral

$$B(a, b) = \int_0^1 u^{a-1}(1-u)^{b-1}\,\mathrm{d}u, \qquad a, b > 0. \tag{A.3}$$

It has the following connection with the gamma function,

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}. \tag{A.4}$$

Evaluating $B(a, b)$ with R:

`beta(a, b)`

Evaluating $\ln(B(a, b))$ with R:

`lbeta(a, b)`

A.3 Univariate discrete distributions

Binomial distribution $\mathrm{Bin}(n, p)$, n positive integer, $0 \le p \le 1$, has pmf

$$\mathrm{Bin}(x \mid n, p) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, \ldots, n.$$

If $X \sim \mathrm{Bin}(n, p)$, then

$$EX = np, \qquad \operatorname{var} X = np(1-p).$$

Evaluating $\mathrm{Bin}(x \mid n, p)$ and simulating k independent draws from $\mathrm{Bin}(n, p)$:

```
dbinom(x, n, p)
rbinom(k, n, p)
```

When $n = 1$, the binomial is also called the **Bernoulli** distribution.

Geometric distribution $\mathrm{Geom}(p)$ with probability parameter $0 < p < 1$ has pmf

$$\mathrm{Geom}(x \mid p) = p\,(1-p)^x, \qquad x = 0, 1, 2, \ldots$$

If $X \sim \mathrm{Geom}(p)$, then

$$EX = \frac{1-p}{p}, \qquad \operatorname{var} X = \frac{1-p}{p^2}.$$

Evaluating $\mathrm{Geom}(x \mid p)$ and simulating n independent draws from $\mathrm{Geom}(p)$:

```
dgeom(x, p)
rgeom(n, p)
```

Negative binomial distribution $\mathrm{NegBin}(r, p)$ with "size" parameter $r > 0$ and probability parameter $0 < p < 1$ has pmf

$$\mathrm{NegBin}(x \mid r, p) = \frac{\Gamma(r+x)}{\Gamma(r)\, x!}\, p^r\, (1-p)^x, \qquad x = 0, 1, 2, \ldots$$

If $X \sim \mathrm{NegBin}(r, p)$, then

$$EX = r\,\frac{1-p}{p}, \qquad \operatorname{var} X = r\,\frac{1-p}{p^2}$$

Evaluating $\mathrm{NegBin}(x \mid r, p)$ and simulating n independent draws from $\mathrm{NegBin}(r, p)$:

```
dnbinom(x, r, p)
rnbinom(n, r, p)
```

Geometric distribution $\mathrm{Geom}(p)$ is the same as $\mathrm{NegBin}(1, p)$.

Poisson distribution $\mathrm{Poi}(\theta)$ with parameter $\theta > 0$ has pmf

$$\mathrm{Poi}(x \mid \theta) = \mathrm{e}^{-\theta}\,\frac{\theta^x}{x!}, \quad x = 0, 1, 2, \ldots$$

If $X \sim \mathrm{Poi}(\theta)$, then

$$EX = \theta, \qquad \operatorname{var} X = \theta$$

Evaluating $\mathrm{Poi}(x \mid \theta)$ and simulating n independent draws from $\mathrm{Poi}(\theta)$:

```
dpois(x, theta)
rpois(n, theta)
```

A.4 Univariate continuous distributions

Beta distribution $\mathrm{Be}(a, b)$ with parameters $a > 0, b > 0$ has pdf

$$\mathrm{Be}(x \mid a, b) = \frac{1}{B(a, b)}\, x^{a-1}(1-x)^{b-1}, \quad 0 < x < 1.$$

$B(a, b)$ is the beta function with arguments a and b. If $X \sim \mathrm{Be}(a, b)$, then

$$EX = \frac{a}{a+b}, \qquad \operatorname{var} X = \frac{ab}{(a+b)^2\,(a+b+1)}.$$

Evaluating $\mathrm{Be}(x \mid a, b)$ and simulating n independent draws from $\mathrm{Be}(a, b)$:

```
dbeta(x, a, b)
rbeta(n, a, b)
```

The uniform distribution $\mathrm{Uni}(0, 1)$ is the same as $\mathrm{Be}(1, 1)$.

Cauchy distribution $\mathrm{Cau}(\mu, \sigma)$ with location parameter μ and scale parameter $\sigma > 0$ has the pdf

$$\mathrm{Cau}(x \mid \mu, \sigma) = \frac{1}{\sigma\pi\left(1 + \frac{(x-\mu)^2}{\sigma^2}\right)}.$$

For this distribution neither the mean nor the variance exist. Cauchy distribution is the same as the t distribution with one degree of freedom. Evaluating $\mathrm{Cau}(x \mid \mu, \sigma)$ and simulating n independent draws from $\mathrm{Cau}(\mu, \sigma)$:

$$\texttt{dcauchy}(\texttt{x}, \texttt{mu}, \texttt{sigma})$$
$$\texttt{rcauchy}(\texttt{n}, \texttt{mu}, \texttt{sigma})$$

Chi squared distribution χ^2_ν with $\nu > 0$ degrees of freedom is the same as the gamma distribution

$$\mathrm{Gam}(\frac{\nu}{2}, \frac{1}{2}).$$

If $X \sim \chi^2_\nu$, then $EX = \nu$ and $\operatorname{var} X = 2\nu$. The R name is `chisq`.

Exponential distribution $\mathrm{Exp}(\lambda)$ with rate $\lambda > 0$ has pdf

$$\mathrm{Exp}(x \mid \lambda) = \lambda\, \mathrm{e}^{-\lambda x}, \quad x > 0.$$

If $X \sim \mathrm{Exp}(\lambda)$, then

$$EX = \frac{1}{\lambda}, \qquad \operatorname{var} X = \frac{1}{\lambda^2}.$$

Evaluating $\mathrm{Exp}(x \mid \lambda)$ and simulating n independent draws from $\mathrm{Exp}(\lambda)$:

$$\texttt{dexp}(\texttt{x}, \texttt{lambda})$$
$$\texttt{rexp}(\texttt{n}, \texttt{lambda})$$

Gamma distribution $\mathrm{Gam}(a, b)$ with parameters $a > 0, b > 0$ has pdf

$$\mathrm{Gam}(x \mid a, b) = \frac{b^a}{\Gamma(a)}\, x^{a-1} \mathrm{e}^{-bx}, \quad x > 0.$$

$\Gamma(a)$ is the gamma function. If $X \sim \mathrm{Gam}(a, b)$, then

$$EX = \frac{a}{b}, \qquad \operatorname{var} X = \frac{a}{b^2}.$$

Evaluating $\mathrm{Gam}(x \mid a, b)$ and simulating n independent draws from $\mathrm{Gam}(a, b)$:

$$\texttt{dgamma}(\texttt{x}, \texttt{a}, \texttt{b})$$
$$\texttt{rgamma}(\texttt{n}, \texttt{a}, \texttt{b})$$

Exponential distribution $\mathrm{Exp}(\lambda)$ is the same as $\mathrm{Gam}(1, \lambda)$.

Generalized gamma distribution with parameters $a, b > 0$ and $r \neq 0$ has pdf

$$f(x \mid a, b, r) = \frac{rb}{\Gamma(a)} (bx)^{ra-1} \exp(-(bx)^r), \quad x > 0.$$

This is the distribution of $X = Y^{1/r}/b$ when $Y \sim \text{Gam}(a, 1)$. (Here $Y = (bX)^r$.)

Laplace distribution (or **Double exponential** distribution) $\text{Laplace}(\mu, \sigma^2)$ with mean μ and dispersion parameter $\sigma^2 > 0$ has pdf

$$\text{Laplace}(x \mid \mu, \sigma^2) = \frac{1}{2\sigma} \exp\left(-\frac{|x-\mu|}{\sigma}\right)$$

Laplace (without parameters) is the standard Laplace distribution $\text{Laplace}(0, 1)$. If $X \sim \text{Laplace}(\mu, \sigma^2)$, then

$$EX = \mu, \qquad \operatorname{var} X = 2\sigma^2.$$

Logistic distribution $\text{Logistic}(\mu, \sigma^2)$ with mean μ and dispersion parameter $\sigma^2 > 0$ has pdf

$$\text{Logistic}(x \mid \mu, \sigma^2) = \frac{\exp\left(-\frac{x-\mu}{\sigma}\right)}{\sigma\left[1 + \exp\left(-\frac{x-\mu}{\sigma}\right)\right]^2}$$

Logistic (without parameters) is the standard logistic distribution $\text{Logistic}(0, 1)$. If $X \sim \text{Logistic}(\mu, \sigma^2)$, then

$$EX = \mu, \qquad \operatorname{var} X = \frac{\pi^2}{3} \sigma^2.$$

Normal distribution $N(\mu, \sigma^2)$ with mean μ and variance $\sigma^2 > 0$ has pdf

$$N(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right).$$

Notice that R parametrizes the normal distribution by the mean and the standard deviation (square root of variance). Evaluating $N(x \mid \mu, \sigma^2)$ and simulating n independent draws from $N(\mu, \sigma^2)$:

$$\texttt{dnorm}(\texttt{x}, \texttt{mu}, \texttt{sigma})$$
$$\texttt{rnorm}(\texttt{n}, \texttt{mu}, \texttt{sigma})$$

Student's t distribution $t(\nu, \mu, \sigma)$ with $\nu > 0$ degrees of freedom, location μ and scale parameter $\sigma > 0$ has pdf

$$t(x \mid \nu, \mu, \sigma) = \frac{\Gamma((\nu+1)/2)}{\sigma\sqrt{\pi\nu}\,\Gamma(\nu/2)} \left(1 + \frac{1}{\nu}\frac{(x-\mu)^2}{\sigma^2}\right)^{-(\nu+1)/2}.$$

$t(\nu)$ or t_ν is short for $t(\nu, 0, 1)$. If $X \sim t(\nu, \mu, \sigma)$, then

$$EX = \mu, \qquad \text{when } \nu > 1$$
$$\operatorname{var} X = \sigma^2 \frac{\nu}{\nu - 2}, \qquad \text{when } \nu > 2.$$

If $0 < \nu \leq 1$, then neither the mean nor the variance exists. If $1 < \nu \leq 2$, then the mean equals μ but the variance is infinite. Evaluating $t(x \mid \nu) = t(x \mid \nu, 0, 1)$ in R:

$$\texttt{dt(x,nu)}$$

Evaluating $t(x \mid \nu, \mu, \sigma)$ and simulating n independent draws from $t(\nu, \mu, \sigma)$:

$$\texttt{dt((x - mu)/sigma, nu)/sigma}$$
$$\texttt{mu + sigma * rt(n, nu)}$$

Representation as a scale mixture of normals: if $\nu > 0$ and $Y \sim \mathrm{Gam}(\nu/2, \nu/2)$ and $[X \mid Y = y] \sim N(0, 1/y)$, then $X \sim t(\nu)$. Cauchy distribution $\mathrm{Cau}(\mu, \sigma)$ is the same as $t(1, \mu, \sigma)$.

Uniform distribution $\mathrm{Uni}(a, b)$ on the interval (a, b), where $a < b$, has pdf

$$\mathrm{Uni}(x \mid a, b) = \frac{1}{b - a}, \qquad a < x < b.$$

If $X \sim \mathrm{Uni}(a, b)$, then

$$EX = \frac{1}{2}(a + b), \qquad \operatorname{var} X = \frac{1}{12\,(b - a)^2}.$$

Evaluating $\mathrm{Uni}(x \mid a, b)$ and simulating n independent draws from $\mathrm{Uni}(a, b)$:

$$\texttt{dunif(x, a, b)}$$
$$\texttt{runif(n, a, b)}$$

Weibull distribution $\mathrm{Weib}(\alpha, \sigma)$ with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ has pdf

$$\mathrm{Weib}(x \mid \alpha, \beta) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha - 1} \exp\left(-\left(\frac{x}{\beta}\right)^{\alpha}\right), \qquad x > 0.$$

If $X \sim \mathrm{Weib}(\alpha, \beta)$, then

$$EX = b\,\Gamma\left(1 + \frac{1}{a}\right), \qquad \operatorname{var} X = b\left[\Gamma\left(1 + \frac{2}{a}\right) - \Gamma\left(1 + \frac{1}{a}\right)^2\right]$$

Evaluating $\mathrm{Weib}(x \mid \alpha, \beta)$ and simulating n independent draws from $\mathrm{Weib}(\alpha, \beta)$:

$$\texttt{dweibull(x, alpha, beta)}$$
$$\texttt{rweibull(n, alpha, beta)}$$

A.5 Multivariate discrete distributions

Multinomial distribution $\mathrm{Mult}(n, (p_1, p_2, \ldots, p_k))$ with sample size n and probability vector parameter $(p_1, \ldots, p_k)$ has pmf

$$\mathrm{Mult}(x_1, \ldots, x_k \mid n, (p_1, \ldots, p_k)) = \frac{n!}{\prod_{i=1}^{k} x_i!} \prod_{j=1}^{k} p_j^{x_j},$$

when $x_1, \ldots, x_k \geq 0$ are integers summing to n (and the pmf is zero otherwise). The numbers p_i form a probability vector, which means that all $p_i \geq 0$ and $p_1 + \cdots + p_n = 1$. If $X \sim \mathrm{Mult}(n, p)$, then $EX_i = np_i$, $\operatorname{var} X_i = np_1(1 - p_i)$, and $\operatorname{cov}(X_i, X_j) = -np_ip_j$, when $i \neq j$. Evaluating $\mathrm{Mult}(x_1, \ldots, x_k \mid n, (p_1, \ldots, p_k))$ in R, when `x` is a k-vector containing the components x_i and `p` is a k-vector containing the components p_i (`p` need not be normalized):

$$\texttt{dmultinom(x,p)}$$

Simulating m independent draws from the distribution: the call

$$\texttt{rmultinom(m,size = n,p)}$$

returns a $k \times m$ matrix whose column vectors are the simulated draws. If $(X, Y) \sim \mathrm{Mult}(n, (p, 1 - p))$, then $X \sim \mathrm{Bin}(n, p)$.

A.6 Multivariate continuous distributions

Dirichlet distribution $\mathrm{Dir}(a_1, \ldots, a_{d+1})$ with parameters $a_1, \ldots, a_{d+1} > 0$ is the d-dimensional distribution with the pdf

$$\mathrm{Dir}(x \mid a) = \mathrm{Dir}(x \mid a_1, \ldots, a_{d+1}) =$$
$$\frac{\Gamma(a_1 + \cdots + a_{d+1})}{\Gamma(a_1) \cdots \Gamma(a_{d+1})} x_1^{a_1 - 1} x_2^{a_2 - 1} \ldots x_d^{a_d - 1} (1 - x_1 - x_2 - \cdots - x_d)^{a_{d+1} - 1},$$

when

$$x_1, \ldots, x_d > 0, \quad \text{and} \quad x_1 + \cdots + x_d < 1,$$

and zero otherwise. If $X \sim \mathrm{Dir}(a_1, \ldots, a_{d+1})$, and $s = \sum a_i$, then

$$EX_i = \frac{a_i}{s}, \qquad \operatorname{var} X_i = \frac{a_i(s - a_i)}{s^2(s + 1)}$$
$$\operatorname{cov}(X_i, X_j) = \frac{-a_i a_j}{s^2(s + 1)}, \quad \text{when } i \neq j.$$

Evaluation of the pdf in R is easy to program; generating random draws can be accomplished by generating $d + 1$ independent gamma variates $Y_i \sim \mathrm{Gam}(a_i, 1)$, and then calculating $X_i = Y_i/S$, where S is the sum $S = Y_1 + \cdots + Y_{d+1}$. Such random draws could be simulated as follows in R,

```
d1 <- length(a)
y <- matrix(rgamma(d1 * n, a), ncol = d1, byrow = TRUE)
x <- sweep(y, 1, rowSums(y), FUN = '/')
```

the draws are now row vectors of matrix `x[ ,1:d]` (the last column of the matrix `x` should be deleted).

The univariate Dirichlet $\mathrm{Dir}(a_1, a_2)$ is the same as the beta distribution $\mathrm{Be}(a_1, a_2)$.

Multivariate normal distribution (in d dimensions), $N_d(\mu, \Sigma)$ with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix Σ (a symmetric, positive definite $d \times d$ matrix) has pdf

$$N_d(x \mid \mu, \Sigma) = (2\pi)^{-d/2} (\det \Sigma)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right).$$

In terms of the mean vector and the precision matrix $Q = \Sigma^{-1}$, the pdf is given by

$$N_d(x \mid \mu, Q^{-1}) = (2\pi)^{-d/2} (\det Q)^{1/2} \exp\left(-\frac{1}{2}(x-\mu)^T Q (x-\mu) \right).$$

Evaluating $N_d(x \mid \mu, \Sigma)$ in R using the library `mnormt` (which may have to be installed first):

```
library(mnormt)
dmnorm(x, mu, Sigma)
```

Above, `x` may be a matrix and then the x-vectors have to be given as row vectors of the matrix. Simulating n independent draws from $N_d(\mu, \Sigma)$: the call

```
rmnorm(n, mu, Sigma)
```

returns a $n \times d$ matrix whose row vectors are the simulated draws (using the library `mnormt`). Alternatively, the draws can be simulated with the function `mvrnorm` from library `MASS`. It is also possible to compute the Cholesky factor of the covariance matrix first and then produce simulations using d independent draws form the univariate standard normal.

Multivariate t distribution (in d dimensions), $t_d(\nu, \mu, \Sigma)$ with $\nu > 0$ degrees of freedom, location parameter $\mu \in \mathbb{R}^d$ and dispersion parameter Σ (a symmetric, positive definite $d \times d$ matrix) has pdf

$$t_d(x \mid \nu, \mu, \Sigma) = \frac{\Gamma((\nu+d)/2)}{\nu^{d/2}\, \pi^{d/2}\, \Gamma(\nu/2)} (\det \Sigma)^{-1/2} \left(1 + \frac{1}{\nu}(x-\mu)^T \Sigma^{-1} (x-\mu) \right)^{-(\nu+d)/2}$$

If $X \sim t_d(\nu, \mu, \Sigma)$, then

$$\begin{aligned} EX &= \mu, \qquad \text{when } \nu > 1 \\ \operatorname{Cov} X &= \frac{\nu}{\nu-2}\, \Sigma, \qquad \text{when } \nu > 2. \end{aligned}$$

For other values of ν the mean (the covariance matrix) does not exist. Evaluating $t_d(x \mid \nu, \mu, \Sigma)$ in R using the library `mnormt` (which may have to be installed first):

```
library(mnormt)
dmt(x, nu, mu, Sigma)
```

Above, `x` may be a matrix and then the x-vectors have to be given as row vectors of the matrix. Simulating n independent draws from $t_d(\nu, \mu, \Sigma)$: the function

```
rmt(n, nu, mu, Sigma)
```

returns a $n \times d$ matrix whose row vectors are the simulated draws (using the library `mnormt`).

Multivariate t can also be simulated using the mixture representation

$$X \mid Y \sim N(\mu, \frac{1}{Y}\Sigma), \quad \text{where} \quad Y \sim \text{Gam}(\nu/2, \nu/2).$$

A.7 Matrix-valued distributions

The Wishart distribution and the inverse Wishart distribution are examples of distributions for a random symmetric $d \times d$ matrix X, which is also constrained to be positive definite (i.e., all the eigenvalues of X are constrained to be strictly positive). Since $X = [X_{ij}]$ is assumed to be symmetric, it has only $\frac{1}{2}d(d+1)$ unique elements, which can be taken to be the elements on and below the main diagonal, i.e., X_{ij} such that $i \geq j$. The matrix-valued distributions described below are continuous joint distributions for such a set of $\frac{1}{2}d(d+1)$ matrix elements.

In the pdf formulas we need the generalized gamma function (or multivariate gamma function), see [3, Ch. 35]. It is the univariate function defined by the multivariate integral

$$\Gamma_d(a) = \int_{X>0} \exp\left(-\operatorname{tr}(X)\right) \det(X)^{a-(d+1)/2} \, \mathrm{d}X, \qquad a > \frac{1}{2}(d-1) \qquad \text{(A.5)}$$

where the restriction $X > 0$ means that the integral is taken over symmetric positive definite $d \times d$ matrices X, and so the integral is actually a $\frac{1}{2}d(d+1)$-fold integral. It is known that this integral can be expressed using the ordinary gamma function in the form

$$\Gamma_d(a) = \pi^{d(d-1)/4} \prod_{j=1}^{d} \Gamma\left[a + (1-j)/2\right], \qquad a > \frac{1}{2}(d-1). \qquad \text{(A.6)}$$

Wishart distribution with parameters $\alpha > \frac{1}{2}(d-1)$ and B (a symmetric positive definite matrix) has pdf

$$\text{Wish}_d(X \mid \alpha, B) = \frac{\det(B)^{\alpha}}{\Gamma_d(\alpha)} \det(X)^{\alpha-(d+1)/2} \exp(-\operatorname{tr}(B\,X)), \qquad X > 0$$

The qualification $X > 0$ means that this expression applies when X is symmetric and positive definite and that the pdf is zero otherwise. This is the parametrization used by Bernardo and Smith [1]. This parametrization tries to emphasize that the Wishart distribution is a matrix-variate generalization of the gamma distribution. It is more usual to employ a parametrization (see, e.g., Gelman *et al.* [2]) which tries to show the connection to the chi-squared distribution. Then the parameters are the degrees of freedom parameter ν and the scale matrix S given by

$$\nu = 2\alpha, \qquad S = (2B)^{-1},$$

The expected value of the $\text{Wish}_d(\alpha, B)$ distribution is

$$EX = \alpha\, B^{-1} = \nu\, S$$

Inverse Wishart distribution with parameters $\alpha > \frac{1}{2}(d-1)$ and B (a symmetric positive definite matrix) has pdf

$$\mathrm{InvWish}_d(X \mid \alpha, B) = \frac{\det(B)^\alpha}{\Gamma_d(\alpha)} \det(X)^{-\alpha-(d+1)/2} \exp(-\operatorname{tr}(B\,X^{-1})), \qquad X > 0$$

The qualification $X > 0$ means that this expression applies when X is symmetric and positive definite and that the pdf is zero otherwise. Once again, this is the Bernardo-Smith parametrization which is different from the conventional parametrization. If $Y \sim \mathrm{Wish}_d(\alpha, B)$, then its inverse matrix has the corresponding inverse Wishart distribution

$$Y^{-1} \sim \mathrm{InvWish}_d(\alpha, B).$$

Bibliography

[1] José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory.* John Wiley & Sons, 2000. First published in 1994.

[2] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis.* Chapman & Hall/CRC Press, 2nd edition, 2004.

[3] Frank W. Olver, Daniel W. Lozier, Ronald F. Boisvert, and Charles W. Clark, editors. *NIST Handbook of Mathematical Functions.* Cambridge University Press, 2010.

Appendix B

R tools

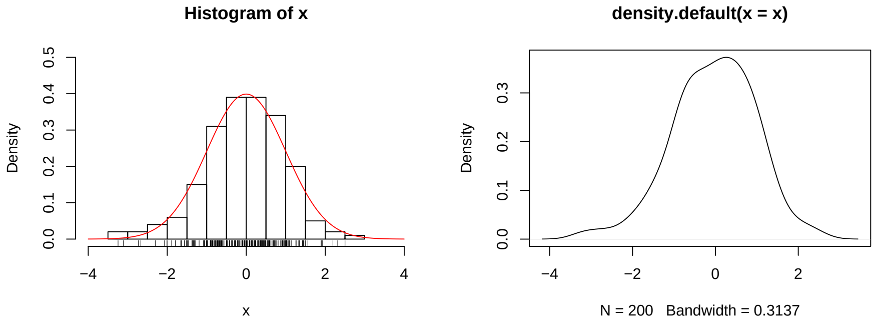
B.1 Combining the histogram and the pdf

When we have a sample from some known continuous distribution, we can plot both the histogram of the sample and the pdf of the distribution in the same figure. In order to have a meaningful comparison between the two results, it is necessary to use a version of the histogram which is normalized to have total area of one (probability density histogram), instead of the ordinary frequency histogram. The R function `hist` with argument `freq = FALSE` (or alternatively, with argument `probability = TRUE`) plots a probability density histogram. Also the `truehist` function of the `MASS` library does the same. In the following example we draw a histogram of values simulated from the $N(0, 1)$ distribution and plot the pdf of the distribution in the same figure. We set the axis limits in the call of `hist` so that both plots fit nicely in the same figure. Finding proper axis limits may require trial and error. Additionally, we specify that the number of histogram bins should be determined using Scott's rule instead of the default Sturges' rule (which usually selects too few bins when the sample size is large).

We can visualize the indivdual sample points with a rug plot. Instead of the histogram, one can also plot another nonparametric probability density estimate, namely the kernel density estimate, which can be calculated with the function `density()`. Notice that this function has several arguments which influence the result.

```
> n <- 200
> x <- rnorm(n) # the default values correspond to N(0, 1)
> hist(x, freq = FALSE, breaks = 'Scott', xlim = c(-4, 4), ylim = c(0, 0.5))
> # Add the graph of the N(0, 1) density drawn in red:
> t <- seq(-4, 4, len = 401)
> lines(t, dnorm(t), col = 'red')
> rug(x)
```

```
> plot(density(x))
```



(a)

(b)

Figure B.1: (a) Probability density function and a probability density histogram, (b) kernel density estimate. The estimates are based on a sample of size $n = 200$ from the standard normal distribution.

B.2 Simulating a discrete distribution with a finite range

Suppose $w = (w_1, w_2, \ldots, w_k)$ is a vector of nonnegative numbers stored in the variable `w`. One can simulate an i.i.d. sample of size n from the corresponding pmf with probabilities

$$p_i = \frac{w_i}{\sum_{j=1}^k w_j}, \qquad i = 1, \ldots, k$$

with the following call

```
x <- sample(1:k, size = n, prob = w, replace = TRUE)
```

See the documentation of `sample` for the details. Notice that the default value of the argument `replace` is `FALSE`, and this corresponds to sampling without replacement. The specifaction `replace = TRUE` yields an i.i.d. sample.

In the following example we draw a sample and calculate the frequencies of the sample.

```
> n <- 100
> w <- c(2, 3, 5)
> x <- sample(1:3, size = n, prob = w, replace = TRUE)
> # calculate frequencies:
> table(x)
```

```
x
 1  2  3
20 27 53
```

If one only needs to simulate the frequencies, not each individual draw, then this can be achieved directly with the call `rmultinom(1, n, prob = w)`.

B.3 Vectorized computations and matrix operations

R has separate data types for representing vectors and for representing matrices. Vectors are often created using the `c()` function or using the `seq()` function. The colon operator is shorthand for a special case of `seq()`.

```
> v <- c(0.1, 0.2, 0.3)
> v

[1] 0.1 0.2 0.3

> ind <- 1:3
> ind

[1] 1 2 3

> seq(1, 3, length = 6)

[1] 1.0 1.4 1.8 2.2 2.6 3.0
```

In R, the basic arithmetic opetators +, -, *, / can be applied to vectors or matrices of the same size, and the operate element by element. E.g., if `A` and `B` are matrices of the same size, then `A * B` does not calculate the matrix product but calculates the matrix whose element (i, j) is equal to `A[i, j] * B[i, j]`. That is, all the basic arithmetic operations for vectors or matrices are performed element by element (and one of the two operands can be a scalar). For clarity and for efficiency, one should avoid using explicit for loops such as

```
> for (i in 1:m)
+   for (j in 1:n) C[i, j] <- A[i, j] * B[i, j]
```

when a simple arithmetic operation `C <- A * B` suffices. However, sometimes one really needs to use explicit loops in R code, e.g., when one tries to implement a Metropolis–Hastings algorithm.

In R, vectors and matrices are distinct data structures; e.g., a $d \times 1$ matrix is for some purposes different from a d component vector. Therefore one sometimes needs to convert such matrices to vectors or vice versa. R function `as.numeric(A)` converts the matrix `A` to a vector by concatenting its columns from left to right. One can convert the vector `v` to a $m \times n$ matrix by using call of the form `matrix(v, m, n)`.

Elements of matrices can be extracted using brackets, e.g., `A[i, j]` extracts the element at position (i, j). Also submatrices can be extracted using brackets. An important special case is that `A[i, ]` extracts the i'th row and `A[, j]` extracts the j'th column of matrix A. In either case the result will be a vector. To prevent that from happening, one can add the argument `drop = FALSE` inside the brackets. This precaution is sometimes needed in order to prevent R code from crashing.

```
> print(A <- matrix(1:6, 2, 3))

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> A[1, 2]

[1] 3

> A[1, ]

[1] 1 3 5

> A[1, , drop = FALSE]

     [,1] [,2] [,3]
[1,]    1    3    5

> A[, 2]

[1] 3 4

> A[, 2, drop = FALSE]

     [,1]
[1,]    3
[2,]    4
```

Some of the often needed matrix operations are

- `matrix(v, nrow = m, ncol = n)` forms a $m \times n$ matrix out of the elements of the vector v which should have mn entries.
- `A %*% B` calculates the matrix product of matrices A and B.
- `t(A)` calculates the transpose of the matrix A.
- `rowSums(A)` and `colSums(A)` calculate the row sums and column sums of matrix A.
- `apply(A, 1, FUN)` applies the function `FUN` to each of the rows of matrix A; `apply(A, 1, sum)` is an alternative way of calculating the row sums of matrix A.
- `apply(A, 2, FUN)` applies the function `FUN` to each of the columns of matrix A; `apply(A, 2, sum)` is an alternative way of calculating the column sums of matrix A.
- `chol(S)` accepts as its argument a symmetric positive definite matrix S and calculates its upper triangular Cholesky factor R, i.e., an upper triangular matrix R such that $S = R^T R$. The lower triangular Cholesky factor L is then $L = R^T$ and it satisfies $S = LL^T$.
- `solve(A)` calculates the inverse of the square matrix A. Notice that `solve(A, b)` solves the linear system of equations $Ax = b$.
- `det(S)` calculates the determinant of the square matrix S.
- `eigen(S)` calculates the eigenvalues and eigenvectors of the square matrix S.

- `svd(A)` calculates the singular value decomposition of a rectangular matrix A.

To illustrate, consider the simulation of the multivariate normal distribution with a given mean vector m and covariance matrix S. If L is the lower triangular Cholesky factor of S, i.e. $S = LL^T$ and Z has the d dimensional standard normal distribution $N_d(0, I)$, then $LZ \sim N(0, LL^T) = N(0, S)$, and $X = LZ + m \sim N(m, S)$. It is customary to store multivariate observations as row vectors of a data matrix, and we decide to do likewise. Transposing,

$$X^T = Z^T L^T + m^T = Z^T R + m^T,$$

where R is the upper triangular Cholesky factor. The following code fragment shows how one can simulate n vectors from the multivariate normal $N(m, S)$ and store them as row vectors of the matrix `x.s` using the above idea.

```
> m <- c(-1.3, 2.2)
> S <- matrix(c(2.1, -1.4, -1.4, 2.1), nrow = 2)
> R <- chol(S)
> n <- 1000
> d <- length(m)
> zz <- matrix(rnorm(n * d), ncol = d)
> x.s <- sweep(zz %*% R, 2, m, '+')
```

We only needed to generate nd random numbers from the standard normal distribution and store them as row vectors of the matrix `zz` in order to get n random draws from the $N_d(0, I)$ distribution. When these row vectors are multiplied from the right by the upper triangular Cholesky factor R, we get n (row vector) draws from the $N_d(0, S)$ distribution. We still need to add the vector m to each of the resulting row vectors. This is done via a call to the function `sweep`, which is a very useful function despite of its obscure documentation:

- `sweep(A, 1, v)` returns a matrix where vector v has been subtracted from each of the columns of matrix A.
- `sweep(A, 2, v)` returns a matrix where vector v has been subtracted from each of the rows of matrix A.
- `sweep(A, 1, v, FUN)` returns a matrix where the ith column of A has been replaced by the result of `FUN(A[ , i], v)`
- `sweep(A, 2, v, FUN)` does a similar transformation operating on the rows of matrix A.

As a second example, consider the following code for calculating the probability density function of a multivariate normal. The call `mydmnrom(x, m, S)` evaluates the density of $N(m, S)$ for each row vector of matrix x and likewise `mydmnrom(x, m, precmat = Q)` evaluates the density of $N(m, Q^{-1})$.

```
> mydmnorm <- function(x, m, covmat, precmat = solve(covmat), log = FALSE) {
+   d <- length(m)
+   if (d > 1 & is.vector(x)) x <- matrix(x, nrow = 1)
+   yy <- sweep(x, 2, m)
+   log.pdf <- (
+     (-d/2) * log(2 * pi) + 0.5 * log(det(precmat))
+     - 0.5 * rowSums((yy %*% precmat) * yy)
+   )
+   if (log) log.pdf else exp(log.pdf)
+ }
```

Here the challenge is to evaluate the quadratic forms $y^t Q y$ efficiently, when $y = x - m$ and x a column vector containing the elements of one the rows of the input matrix. In the code we do this for all rows at the same time by using matrix multiplication followed by element by element multiplication and summation.

B.4 Contour plots

Contour plots can be drawn by calling the function `contour`, e.g., using the arguments `contour(x, y, z)`. Here `x` and `y` are vectors containing the values at which the function $f(x, y)$ whose contour lines we want to draw have been evaluated, and `z` is a matrix such that

$$\mathrm{z[i, j]} = f(\texttt{x[i]}, \texttt{y[j]}) \tag{B.1}$$

Those parts of the contour plot where `z` has the value `NA` are omitted.

In the following example we first define a function `ddiri` which evaluates the Dirichlet density (or its logarithm) at the points given as row vectors of the argument matrix `x`. (Argument `x` can also be a vector.) The function returns the value `NA` for those points which fall outside the valid domain.

```
> ddiri <- function(x, a, log = FALSE) {
+   if (is.vector(x)) x <- matrix(x, nrow = 1)
+   d <- dim(x)[2]
+   n <- dim(x)[1]
+   d1 <- length(a)
+   stopifnot(d1 == d + 1)
+   x <- cbind(x, 1 - rowSums(x))
+   valid <- apply(x > 0, 1, all)
+   log.pdf <- numeric(n)
+   log.pdf[!valid] <- NA
+   log.pdf[valid] <- (lgamma(sum(a)) - sum(lgamma(a)) +
+     rowSums(sweep(log(x[valid, , drop = FALSE]), 2, a - 1, '*'))
+   )
+   if (log) log.pdf else exp(log.pdf)
+ }
```

In order to make a contour plot of the a bivariate Dirichlet density, we first define its parameters and set up grids `x` and `y` for the two axes.

```
> a <- c(15.3, 11.2, 13.0)
> n <- 400
> eps <- 0.0001
> x <- seq(0 + eps, 1 - eps, length = 101)
> y <- x
```

Next we calculate a matrix z such that (B.1) holds for all i, j, when f stands for the probability density. The safe but dull solution is to use two nested for loops as follows

```
> z <- matrix(0, nrow = length(x), ncol = length(y))
> for (i in seq_along(x))
+   for (j in seq_along(y))
+     z[i, j] <- ddiri(c(x[i], y[j]), a)
```

A fancier and more efficient solution uses only a single call of the function ddiri. We first form matrices xx and yy such that xx[i, j] = x[i] for each j and yy[i, j] = y[j] for each i. Here we use the function outer which calculates outer products of vectors: the outer product of column vectors u and v is the matrix uv^T. These two matrices xx and yy are then converted to vectors which are then combined to form the first argument to the function ddiri. The function returns a vector which is reshaped into the appropriate form.

```
> xx <- outer(x, rep(1, length(y)))
> yy <- outer(rep(1, length(x)), y)
> xarg <- cbind(as.vector(xx), as.vector(yy))
> zz <- matrix(ddiri(xarg, a), nrow = length(x))
```

It would be easier to define a version of ddiri() which has separate arguments for the two coordinates and do the following

```
> ddiri2 <- function(x, y, a, log = FALSE) ddiri(cbind(x, y), a, log)
> z <- outer(x, y, ddiri2, a)
```

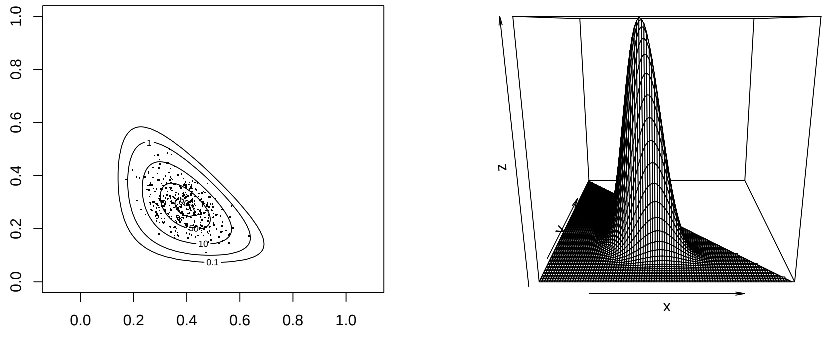
This calculates the result using the same calculation as detailed in the previous code snippet. In particular, the result will be calculated using just a single call to the function ddiri2() (and therefore using just a single call of the function ddiri()).

Before calling contour we scale the result so that the maximum value of z becomes 100. Then it is easy to define meaningful levels at which to draw the contour lines. The default values of the levels would produce a more crowded figure which would not show the tail behaviour of the density as well. This way of selecting the contour levels works well when the density is bounded and all of the contours have roughly the same shape. To finish the plot, we draw a random sample from the density and plot it on top of the contour plot.

```
> maxz <- max(z, na.rm = TRUE)
> z <- 100 * z / maxz
> contour(x, y, z, levels = c(90, 50, 10, 1, 0.1), asp=1)
> d1 <- 2 + 1
> yy.s <- matrix(rgamma(d1 * n, a), ncol = d1, byrow = TRUE)
> xx.s <- sweep(yy.s, 1, rowSums(yy.s), FUN = '/')
> points(xx.s[,1], xx.s[,2], pch = '.')
```

An alternative way to visualize a two-dimensional density is to draw a perspective plot using the function persp.

```
> persp(x, y, z)
```



(a)

(b)

Figure B.2: Probability density function of a two-dimensional Dirichlet distribution: (a) contour plot and sample points, (b) perspective plot.

B.5 Numerical integration

The function `integrate` calculates numerically the integral of a given function over a given (univariate) interval. It returns a list from which the approximate value of the integral can be extracted for further use.

```
> f <- function(x) x^(100) * (1 - x)^(110)
> print(v <- integrate(f, 0, 1))

6.63829e-65 with absolute error < 8.4e-66

> print(v$value)

[1] 6.63829e-65
```

B.6 Root finding

R function `polyroot` is able to find the (complex) roots of a polynomial. The real and imaginary parts of complex numbers can be extracted with the functions `Re` and `Im`, respectively.

R function `uniroot` can find the root of a continuous function, when it is given an interval such that the function has values of opposite signs at the endpoints.

B.7 Optimization

The function `optim` optimizes iteratively a given multivariate objective function using one of several methods. The default method is derivative free in that one does need to write a function for calculating the gradient of the objective function. The function is also able to calculate an approximation to the Hessian matrix of the objective function the optimum point. This can be used, e.g., in order to calculate an Laplace approximation to some interesting integral.

For the sake of demonstration we use `optim` to approximate the integral of an unnormalized from of the Dirichlet density discussed earlier. Since `optim` likes to minimize (instead of maximize) we use as the target function the negative of the logarithm of an unnormalized verision of the density function. Then the calculated Hessian is the negative Hessian needed in the approximation. (It is possible to make `optim` to maximize by specifying a suitable value for its argument `control`.) Notice that our objective function is coded to return plus infinity, if its argument is outside the valid domain.

```
> objective.f <- function(x, a) {
+   x <- c(x, 1 - sum(x))
+   if (any(x < 0)) return(Inf)
+   return(sum((1 - a) * log(x)))
+ }
> a <- c(15.3, 11.2, 13.0)
> ini.x <- c(0.3, 0.4)
> r <- optim(ini.x, objective.f, gr = NULL, a, hessian = TRUE)
> print(opt.x <- r$par)

[1] 0.3918050 0.2794669

> print(Q <- r$hessian)

         [,1]     [,2]
[1,] 204.2033 111.0493
[2,] 111.0493 241.6515

> d <- 2
> I.Lap <- (2 * pi)^(d/2) * exp(-objective.f(opt.x, a)) / sqrt(det(Q))
> I.exact <- prod(gamma(a)) / gamma(sum(a))
> print(c(Laplace = I.Lap, exact = I.exact))

     Laplace        exact
1.776269e-19 1.669990e-19
```