1

INTRODUCTION

1.1 UNCERTAINTY IN ROBOTICS

Robotics is the science of perceiving and manipulating the physical world through computer-controlled mechanical devices. Examples of successful robotic systems include mobile platforms for planetary exploration [], robotics arms in assembly lines [], cars that travel autonomously on highways [], actuated arms that assist surgeons []. Robotics systems have in common that they are are situated in the physical world, perceive their environments through sensors, and manipulate their environment through things that move.

While much of robotics is still in its infancy, the idea of "intelligent" manipulating devices has an enormous potential to change society. Wouldn't it be great if all our cars were able to safely steer themselves, making car accidents a notion of the past? Wouldn't it be great if robots, and not people, would clean up nuclear disasters sites like Chernobyl? Wouldn't it be great if our homes were populated by intelligent service robots that would carry out such tedious tasks as loading the dishwasher, and vacuuming the carpet, or walking our dogs? And lastly, a better understanding of robotics will ultimately lead to a better understanding of animals and people.

Tomorrows application domains differ from yesterdays, such as manipulators in assembly lines that carry out the identical task day-in day-out. The most striking characteristic of the new robot systems is that they operate in increasingly unstructured environments, environments that are inherently unpredictable. An assembly line is orders of magnitude more predictable and controllable than a private home. As a result, robotics is moving into areas where sensor input becomes increasingly important, and where robot software has to be robust enough to cope with a range of situations—often too many to anticipate them all. Robotics, thus, is increasingly becoming a software

science, where the goal is to develop robust software that enables robots to withstand the numerous challenges arising in unstructured and dynamic environments.

This book focuses on a key element of robotics: *Uncertainty*. Uncertainty arises if the robot lacks critical information for carrying out its task. It arises from five different factors:

1. **Environments.** Physical worlds are inherently unpredictable. While the degree of uncertainty in well-structured environments such assembly lines is small, environments such as highways and private homes are highly dynamic and unpredictable.
2. **Sensors.** Sensors are inherently limited in what they can perceive. Limitations arise from two primary factors. First, range and resolution of a sensor is subject to physical laws. For example, Cameras can't see through walls, and even within the perceptual range the spatial resolution of camera images is limited. Second, sensors are subject to noise, which perturbs sensor measurements in unpredictable ways and hence limits the information that can be extracted from sensor measurements.
3. **Robots.** Robot actuation involves motors that are, at least to some extent, unpredictable, due effects like control noise and wear-and-tear. Some actuators, such as heavy-duty industrial robot arms, are quite accurate. Others, like low-cost mobile robots, can be extremely inaccurate.
4. **Models.** Models are inherently inaccurate. Models are abstractions of the real world. As such, they only partially model the underlying physical processes of the robot and its environment. Model errors are a source of uncertainty that has largely been ignored in robotics, despite the fact that most robotic models used in state-or-the-art robotics systems are rather crude.
5. **Computation.** Robots are real-time systems, which limits the amount of computation that can be carried out. Many state-of-the-art algorithms (such as most of the algorithms described in this book) are approximate, achieving timely response through sacrificing accuracy.

All of these factors give rise to uncertainty. Traditionally, such uncertainty has mostly been ignored in robotics. However, as robots are moving away from factory floors into increasingly unstructured environments, the ability to cope with uncertainty is critical for building successful robots.

Robotics is mostly software based now.
We focus on the algorithms to model & handle uncertainty in robotics in this course.

1.2 PROBABILISTIC ROBOTICS

This book provides a comprehensive overview of probabilistic algorithms for robotics. Probabilistic robotics is a new approach to robotics that pays tribute to the uncertainty in robot perception and action. They key idea of probabilistic robotics is to represent uncertainty explicitly, using the calculus of probability theory. Put differently, instead of relying on a single "best guess" as to what might be the case in the world, probabilistic algorithms represent information by probability distributions over a whole space of possible hypotheses. By doing so, they can represent ambiguity and degree of belief in a mathematically sound way, enabling them to accommodate all sources of uncertainty listed above. Moreover, by basing control decisions on probabilistic information, these algorithms degrade nicely in the face of the various sources of uncertainty described above, leading to new solutions to hard robotics problems.

Let us illustrate the probabilistic approach with a motivating example: mobile robot localization. Localization is the problem of estimating a robot's coordinates in an external reference frame from sensor data, using a map of the environment. Figure 1.1 illustrates the probabilistic approach to mobile robot localization. The specific localization problem studied here is known as *global localization*, where a robot is placed somewhere in the environment and has to localize itself from scratch. In the probabilistic paradigm, the robot's momentary estimate (also called *belief*) is represented by a probability density function over the space of all locations. This is illustrated in the first diagram in Figure 1.1, which shows a uniform distribution (the *prior*) that corresponds to maximum uncertainty. Suppose the robot takes a first sensor measurement and observes that it is next to a door. The resulting belief, shown in the second diagram in Figure 1.1, places high probability at places next to doors and low probability elsewhere. Notice that this distribution possesses three peaks, each corresponding to one of the (indistinguishable) doors in the environment. Furthermore, the resulting distribution assigns high probability to three distinct locations, illustrating that the probabilistic framework can handle multiple, conflicting hypotheses that naturally arise in ambiguous situations. Finally, even non-door locations possess non-zero probability. This is accounted by the uncertainty inherent in sensing: With a small, non-zero probability, the robot might err and actually not be next to a door. Now suppose the robot moves. The third diagram in Figure 1.1 shows the effect of robot motion on its belief, assuming that the robot moved as indicated. The belief is shifted in the direction of motion. It is also smoothed, to account for the inherent uncertainty in robot motion. Finally, the fourth and last diagram in Figure 1.1 depicts the belief after observing another door. This observation leads our algorithm to place most of the probability mass on a location near one of the doors, and the robot is now quite confident as to where it is.

2

RECURSIVE STATE ESTIMATION

2.1 INTRODUCTION

At the core of probabilistic robotics is the idea of estimating state from sensor data. State estimation addresses the problem of estimating quantities from sensor data that are not directly observable, but that can be inferred. In most robotic applications, determining what to do is relatively easy if one only knew *certain* quantities. For example, moving a mobile robot is relatively easy if the exact location of the robot and all nearby obstacles are known. Unfortunately, these variables are not directly measurable. Instead, a robot has to rely on its sensors to gather this information. Sensors carry only partial information about those quantities, and their measurements are corrupted by noise. State estimation seeks to recover state variables from the data. Probabilistic state estimation algorithms compute belief distributions over possible world states. An example of probabilistic state estimation was already encountered in the introduction to this book: mobile robot localization.

The goal of this chapter is to introduce the basic vocabulary and mathematical tools for estimating state from sensor data.

- Section 2.2 introduces basic probabilistic concepts and notations used throughout the book.
- Section 2.3 describes our formal model of robot environment interaction, setting forth some of the key terminology used throughout the book.
- Section 2.4 introduces *Bayes filters*, the recursive algorithm for state estimation that forms the basis of virtually every technique presented in this book.

- Section 2.5 discusses representational and computational issues that arise when implementing Bayes filters.

2.2 BASIC CONCEPTS IN PROBABILITY

This section familiarizes the reader with the basic notation and probabilistic facts and notation used throughout the book. In probabilistic robotics, quantities such as sensor measurements, controls, and the states a robot and its environment might assume are all modeled as random variables. Random variables can take on multiple values, and they do so according to specific probabilistic laws. Probabilistic inference is the process of calculating these laws for random variables that are derived from other random variables, such as those modeling sensor data.

Let X denote a random variable and x denote a specific event that X might take on. A standard example of a random variable is that of a coin flip, where X can take on the values head or tail. If the space of all values that X can take on is discrete, as is the case if X is the outcome of a coin flip, we write

$$p(X = x) \tag{2.1}$$

to denote the probability that the random variable X has value x. For example, a fair coin is characterized by $p(X = \text{head}) = p(X = \text{tail}) = \frac{1}{2}$. Discrete probabilities sum to one, that is,

$$\sum_x p(X = x) \quad = \quad 1\,, \tag{2.2}$$

and of course, probabilities are always non-negative, that is, $p(X = x) \geq 0$. To simplify the notation, we will usually omit explicit mention of the random variable whenever possible, and instead use the common abbreviation $p(x)$ instead of writing $p(X = x)$.

Most techniques in this book address estimation and decision making in continuous spaces. Continuous spaces are characterized by random variables that can take on a continuum of values. Throughout this book, we assume that all continuous random variables possess probability density functions (PDFs). A common density function is that of the one-dimensional *normal distribution* with mean μ and variance σ^2. This

distribution is given by the following Gaussian function:

$$p(x) \quad = \quad \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \; \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\} \tag{2.3}$$

Normal distributions play a major role in this book. We will frequently abbreviate them as $\mathcal{N}(x; \mu, \sigma^2)$, which specifies the random variable, its mean, and its variance.

The Normal distribution (2.3) assumes that x is a scalar value. Often, x will be a multi-dimensional vector. Normal distributions over vectors are called *multivariate*. Multivariate normal distributions are characterized by density functions of the following form:

$$p(x) \quad = \quad \det\left(2\pi\Sigma\right)^{-\frac{1}{2}} \; \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{2.4}$$

Here μ is the mean vector and Σ a (positive semidefinite) symmetric matrix called *covariance matrix*. The superscript ^{T} marks the transpose of a vector. The reader should take a moment to realize that Equation (2.4) is a strict generalization of Equation (2.3); both definitions are equivalent if x is a scalar value. The PDFs of a one- and a two-dimensional normal distribution are graphically depicted in Figure 5.6.

Equations (2.3) and (2.4) are examples of PDFs. Just as discrete probability distributions always sums up to one, a PDF always integrates to 1:

$$\int p(x)\, dx \quad = \quad 1\,. \tag{2.5}$$

However, unlike a discrete probability, the value of a PDF is not bounded above by 1. Throughout this book, we will use the terms *probability*, *probability density* and *probability density function* interchangeably. We will silently assume that all continuous random variables are measurable, and we also assume that all continuous distributions actually possess densities.

The *joint distribution* of two random variables X and Y is given by

$$p(x, y) \quad = \quad p(X = x \text{ and } Y = y)\,. \tag{2.6}$$

This expression describes the probability of the event that the random variable X takes on the value x *and* that Y takes on the value y. If X and Y are *independent*, we have

$$p(x, y) = p(x)\, p(y)\,. \tag{2.7}$$

Often, random variables carry information about other random variables. Suppose we already know that Y's value is y, and we would like to know the probability that X's value is x conditioned on that fact. Such a probability will be denoted

$$p(x \mid y) = p(X = x \mid Y = y) \tag{2.8}$$

and is called *conditional* probability. If $p(y) > 0$, then the conditional probability is defined as

$$p(x \mid y) = \frac{p(x, y)}{p(y)}\,. \tag{2.9}$$

If X and Y are independent, we have

$$p(x \mid y) = \frac{p(x)\, p(y)}{p(y)} = p(x)\,. \tag{2.10}$$

In other words, if X and Y are independent, Y tells us nothing about the value of X. There is no advantage of knowing Y if our interest pertains to knowing X. Independence, and its generalization known as conditional independence, plays a major role throughout this book.

An interesting fact, which follows from the definition of conditional probability and the axioms of probability measures, is often referred to as *theorem of total probability*:

$$p(x) = \sum_y p(x \mid y)\, p(y) \qquad \text{(discrete case)} \tag{2.11}$$

$$p(x) = \int p(x \mid y)\, p(y)\, dy \qquad \text{(continuous case)} \tag{2.12}$$

If $p(x \mid y)$ or $p(y)$ are zero, we define the product $p(x \mid y)\, p(y)$ to be zero, regardless of the value of the remaining factor.

Equally important is *Bayes rule*, which relates conditionals of the type $p(x \mid y)$ to their "inverse," $p(y \mid x)$. The rule, as stated here, requires $p(y) > 0$:

$$p(x \mid y) = \frac{p(y \mid x)\, p(x)}{p(y)} = \frac{p(y \mid x)\, p(x)}{\sum_{x'} p(y \mid x')\, p(x')} \quad \text{(discrete)} \tag{2.13}$$

$$p(x \mid y) = \frac{p(y \mid x)\, p(x)}{p(y)} = \frac{p(y \mid x)\, p(x)}{\int p(y \mid x')\, p(x')\, dx'} \quad \text{(continuous)} \tag{2.14}$$

Bayes rule plays a predominant role in probabilistic robotics. If x is a quantity that we would like to infer from y, the probability $p(x)$ will be referred to as *prior probability distribution*, and y is called the *data* (e.g., a sensor measurement). The distribution $p(x)$ summarizes the knowledge we have regarding X prior to incorporating the data y. The probability $p(x \mid y)$ is called the *posterior probability distribution* over X. As (2.14) suggests, Bayes rule provides a convenient way to compute a posterior $p(x \mid y)$ using the "inverse" conditional probability $p(y \mid x)$ along with the prior probability $p(x)$. In other words, if we are interested in inferring a quantity x from sensor data y, Bayes rule allows us to do so through the inverse probability, which specifies the probability of data y assuming that x was the case. In robotics, this inverse probability is often coined "generative model," since it describes, at some level of abstraction, how state variables X *cause* sensor measurements Y.

An important observation is that the denominator of Bayes rule, $p(y)$, does not depend on x. Thus, the factor $p(y)^{-1}$ in Equations (2.13) and (2.14) will be the same for any value x in the posterior $p(x \mid y)$. For this reason, $p(y)^{-1}$ is often written as a *normalizer* variable, and generically denoted η:

$$p(x \mid y) = \eta\, p(y \mid x)\, p(x)\,. \tag{2.15}$$

If X is discrete, equations of this type can be computed as follows:

$$\forall x : \text{aux}_{x|y} = p(y \mid x)\, p(x) \tag{2.16}$$

$$\text{aux}_y = \sum_x \text{aux}_{x|y} \tag{2.17}$$

$$\forall x : p(x \mid y) = \frac{\text{aux}_{x|y}}{\text{aux}_y}\,, \tag{2.18}$$

where $\text{aux}_{x|y}$ and aux_y are auxiliary variables. These instructions effectively calculate $p(x \mid y)$, but instead of explicitly computing $p(y)$, they instead just normalize the

result. The advantage of the notation in (2.15) lies in its brevity. Instead of explicitly providing the exact formula for a normalization constant—which can grow large very quickly in some of the mathematical derivations to follow—we simply will use the normalizer "η" to indicate that the final result has to be normalized to 1. Throughout this book, normalizers of this type will be denoted η (or η', η'', ...). We will freely use the same η in different equations to denote normalizers, even if their actual values are different.

The *expectation* of a random variable X is given by

$$\begin{aligned} E[X] &= \sum_x x\, p(x)\,, \\ E[X] &= \int x\, p(x)\, dx\,. \end{aligned} \tag{2.19}$$

Not all random variables possess finite expectations; however, those that do not are of no relevance to the material presented in this book. The expectation is a linear function of a random variable. In particular, we have

$$E[aX + b] = aE[X] + b \tag{2.20}$$

for arbitrary numerical values a and b. The covariance of X is obtained as follows

$$\mathrm{Cov}[X] = E[X - E[X]]^2 = E[X^2] - E[X]^2 \tag{2.21}$$

The covariance measures the squared expected deviation from the mean. As stated above, the mean of a multivariate normal distribution $\mathcal{N}(x; \mu, \Sigma)$ is μ, and its covariance is Σ.

Another important characteristic of a random variable is its *entropy*. For discrete random variables, the entropy is given by the following expression:

$$H(P) = E[-\log_2 p(x)] = -\sum_x p(x)\, \log_2 p(x)\,. \tag{2.22}$$

The concept of entropy originates in information theory. The entropy is the expected information that the value of x carries: $-\log_2 p(x)$ is the number of bits required

to encode x using an optimal encoding, and $p(x)$ is the probability at which x will be observed. In this book, entropy will be used in robotic information gathering, to express the information a robot may receive upon executing specific actions.

Finally, we notice that it is perfectly fine to condition any of the rules discussed so far on arbitrary other random variables, such as the variable Z. For example, conditioning Bayes rule on $Z = z$ gives us:

$$p(x \mid y, z) \quad = \quad \frac{p(y \mid x, z)\, p(x \mid z)}{p(y \mid z)} \tag{2.23}$$

Similarly, we can condition the rule for combining probabilities of independent random variables (2.7) on other variables z:

$$p(x, y \mid z) \quad = \quad p(x \mid z)\, p(y \mid z)\,. \tag{2.24}$$

Such a relation is known as *conditional independence*. As the reader easily verifies, (2.24) is equivalent to

$$\begin{aligned} p(x \mid z) &= p(x \mid z, y) \\ p(y \mid z) &= p(y \mid z, x) \end{aligned} \tag{2.25}$$

Conditional independence plays an important role in probabilistic robotics. It applies whenever a variable y carries no information about a variable x if another variable's value z is known. Conditional independence does not imply (absolute) independence, that is,

$$p(x, y \mid z) \;=\; p(x \mid z)\, p(y \mid z) \quad \not\Rightarrow \quad p(x, y) \;=\; p(x)\, p(y) \tag{2.26}$$

The converse is also in general untrue: absolute independence does not imply conditional independence:

$$p(x, y) \;=\; p(x)\, p(y) \quad \not\Rightarrow \quad p(x, y \mid z) \;=\; p(x \mid z)\, p(y \mid z) \tag{2.27}$$

In special cases, however, conditional and absolute independence may coincide.

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

Appendix A

Mathematical Background

A.1 Joint, Marginal and Conditional Probability

Let the n (discrete or continuous) random variables $y_1, \ldots, y_n$ have a *joint* probability $p(y_1, \ldots, y_n)$, or $p(\mathbf{y})$ for short.[1] Technically, one ought to distinguish between probabilities (for discrete variables) and probability densities for continuous variables. Throughout the book we commonly use the term "probability" to refer to both. Let us partition the variables in $\mathbf{y}$ into two groups, $\mathbf{y}_A$ and $\mathbf{y}_B$, where A and B are two disjoint sets whose union is the set $\{1, \ldots, n\}$, so that $p(\mathbf{y}) = p(\mathbf{y}_A, \mathbf{y}_B)$. Each group may contain one or more variables.

joint probability

The *marginal* probability of $\mathbf{y}_A$ is given by

marginal probability

$$p(\mathbf{y}_A) = \int p(\mathbf{y}_A, \mathbf{y}_B)\, d\mathbf{y}_B. \tag{A.1}$$

The integral is replaced by a sum if the variables are discrete valued. Notice that if the set A contains more than one variable, then the marginal probability is itself a joint probability—whether it is referred to as one or the other depends on the context. If the joint distribution is equal to the product of the marginals, then the variables are said to be *independent*, otherwise they are *dependent*.

independence

The *conditional* probability function is defined as

conditional probability

$$p(\mathbf{y}_A|\mathbf{y}_B) = \frac{p(\mathbf{y}_A, \mathbf{y}_B)}{p(\mathbf{y}_B)}, \tag{A.2}$$

defined for $p(\mathbf{y}_B) > 0$, as it is not meaningful to condition on an impossible event. If $\mathbf{y}_A$ and $\mathbf{y}_B$ are independent, then the marginal $p(\mathbf{y}_A)$ and the conditional $p(\mathbf{y}_A|\mathbf{y}_B)$ are equal.

[1] One can deal with more general cases where the density function does not exist by using the distribution function.

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

Bayes' rule

Using the definitions of both $p(\mathbf{y}_A|\mathbf{y}_B)$ and $p(\mathbf{y}_B|\mathbf{y}_A)$ we obtain *Bayes' theorem*

$$p(\mathbf{y}_A|\mathbf{y}_B) = \frac{p(\mathbf{y}_A)p(\mathbf{y}_B|\mathbf{y}_A)}{p(\mathbf{y}_B)}. \tag{A.3}$$

Since conditional distributions are themselves probabilities, one can use all of the above also when further conditioning on other variables. For example, in supervised learning, one often conditions on the inputs throughout, which would lead e.g. to a version of Bayes' rule with additional conditioning on X in all four probabilities in eq. (A.3); see eq. (2.5) for an example of this.

A.2 Gaussian Identities

Gaussian definition

The multivariate Gaussian (or Normal) distribution has a joint probability density given by

$$p(\mathbf{x}|\mathbf{m},\Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2}\exp\left(-\tfrac{1}{2}(\mathbf{x}-\mathbf{m})^\top\Sigma^{-1}(\mathbf{x}-\mathbf{m})\right), \tag{A.4}$$

where $\mathbf{m}$ is the *mean* vector (of length D) and Σ is the (symmetric, positive definite) *covariance* matrix (of size $D \times D$). As a shorthand we write $\mathbf{x} \sim \mathcal{N}(\mathbf{m},\Sigma)$.

Let $\mathbf{x}$ and $\mathbf{y}$ be jointly Gaussian random vectors

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}A & C\\ C^\top & B\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\tilde{A} & \tilde{C}\\ \tilde{C}^\top & \tilde{B}\end{bmatrix}^{-1}\right), \tag{A.5}$$

conditioning and marginalizing

then the *marginal* distribution of $\mathbf{x}$ and the *conditional* distribution of $\mathbf{x}$ given $\mathbf{y}$ are

$$\begin{aligned}\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, A), \quad \text{and} \quad \mathbf{x}|\mathbf{y} &\sim \mathcal{N}\big(\boldsymbol{\mu}_x + CB^{-1}(\mathbf{y}-\boldsymbol{\mu}_y),\ A - CB^{-1}C^\top\big)\\ \text{or} \quad \mathbf{x}|\mathbf{y} &\sim \mathcal{N}\big(\boldsymbol{\mu}_x - \tilde{A}^{-1}\tilde{C}(\mathbf{y}-\boldsymbol{\mu}_y),\ \tilde{A}^{-1}\big).\end{aligned} \tag{A.6}$$

See, e.g. von Mises [1964, sec. 9.3], and eqs. (A.11 - A.13).

products

The product of two Gaussians gives another (un-normalized) Gaussian

$$\begin{aligned}\mathcal{N}(\mathbf{x}|\mathbf{a},A)\mathcal{N}(\mathbf{x}|\mathbf{b},B) &= Z^{-1}\mathcal{N}(\mathbf{x}|\mathbf{c},C)\\ \text{where} \quad \mathbf{c} &= C(A^{-1}\mathbf{a}+B^{-1}\mathbf{b}) \quad \text{and} \quad C = (A^{-1}+B^{-1})^{-1}.\end{aligned} \tag{A.7}$$

Notice that the resulting Gaussian has a precision (inverse variance) equal to the sum of the precisions and a mean equal to the convex sum of the means, weighted by the precisions. The normalizing constant looks itself like a Gaussian (in $\mathbf{a}$ or $\mathbf{b}$)

$$Z^{-1} = (2\pi)^{-D/2}|A+B|^{-1/2}\exp\left(-\tfrac{1}{2}(\mathbf{a}-\mathbf{b})^\top(A+B)^{-1}(\mathbf{a}-\mathbf{b})\right). \tag{A.8}$$

To prove eq. (A.7) simply write out the (lengthy) expressions by introducing eq. (A.4) and eq. (A.8) into eq. (A.7), and expand the terms inside the exp to

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

verify equality. Hint: it may be helpful to expand C using the matrix inversion lemma, eq. (A.9), $C = (A^{-1}+B^{-1})^{-1} = A - A(A+B)^{-1}A = B - B(A+B)^{-1}B$.

generating multivariate Gaussian samples

To generate samples $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, K)$ with arbitrary mean $\mathbf{m}$ and covariance matrix K using a scalar Gaussian generator (which is readily available in many programming environments) we proceed as follows: first, compute the Cholesky decomposition (also known as the "matrix square root") L of the positive definite symmetric covariance matrix $K = LL^\top$, where L is a lower triangular matrix, see section A.4. Then generate $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I)$ by multiple separate calls to the scalar Gaussian generator. Compute $\mathbf{x} = \mathbf{m} + L\mathbf{u}$, which has the desired distribution with mean $\mathbf{m}$ and covariance $L\mathbb{E}[\mathbf{u}\mathbf{u}^\top]L^\top = LL^\top = K$ (by the independence of the elements of $\mathbf{u}$).

In practice it may be necessary to add a small multiple of the identity matrix ϵI to the covariance matrix for numerical reasons. This is because the eigenvalues of the matrix K can decay very rapidly (see section 4.3.1 for a closely related analytical result) and without this stabilization the Cholesky decomposition fails. The effect on the generated samples is to add additional independent noise of variance ϵ. From the context ϵ can usually be chosen to have inconsequential effects on the samples, while ensuring numerical stability.

A.3 Matrix Identities

matrix inversion lemma

The *matrix inversion lemma*, also known as the Woodbury, Sherman & Morrison formula (see e.g. Press et al. [1992, p. 75]) states that

$$(Z + UWV^\top)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + V^\top Z^{-1}U)^{-1}V^\top Z^{-1}, \qquad \text{(A.9)}$$

assuming the relevant inverses all exist. Here Z is $n \times n$, W is $m \times m$ and U and V are both of size $n \times m$; consequently if Z^{-1} is known, and a low rank (i.e. $m < n$) perturbation is made to Z as in left hand side of eq. (A.9), considerable speedup can be achieved. A similar equation exists for determinants

determinants

$$|Z + UWV^\top| = |Z|\,|W|\,|W^{-1} + V^\top Z^{-1}U|. \qquad \text{(A.10)}$$

inversion of a partitioned matrix

Let the invertible $n \times n$ matrix A and its inverse A^{-1} be partitioned into

$$A = \begin{pmatrix} P & Q \\ R & S \end{pmatrix}, \qquad A^{-1} = \begin{pmatrix} \tilde{P} & \tilde{Q} \\ \tilde{R} & \tilde{S} \end{pmatrix}, \qquad \text{(A.11)}$$

where P and $\tilde{P}$ are $n_1 \times n_1$ matrices and S and $\tilde{S}$ are $n_2 \times n_2$ matrices with $n = n_1 + n_2$. The submatrices of A^{-1} are given in Press et al. [1992, p. 77] as

$$\left.\begin{aligned} \tilde{P} &= P^{-1} + P^{-1}QMRP^{-1} \\ \tilde{Q} &= -P^{-1}QM \\ \tilde{R} &= -MRP^{-1} \\ \tilde{S} &= M \end{aligned}\right\} \text{where } M = (S - RP^{-1}Q)^{-1}, \qquad \text{(A.12)}$$