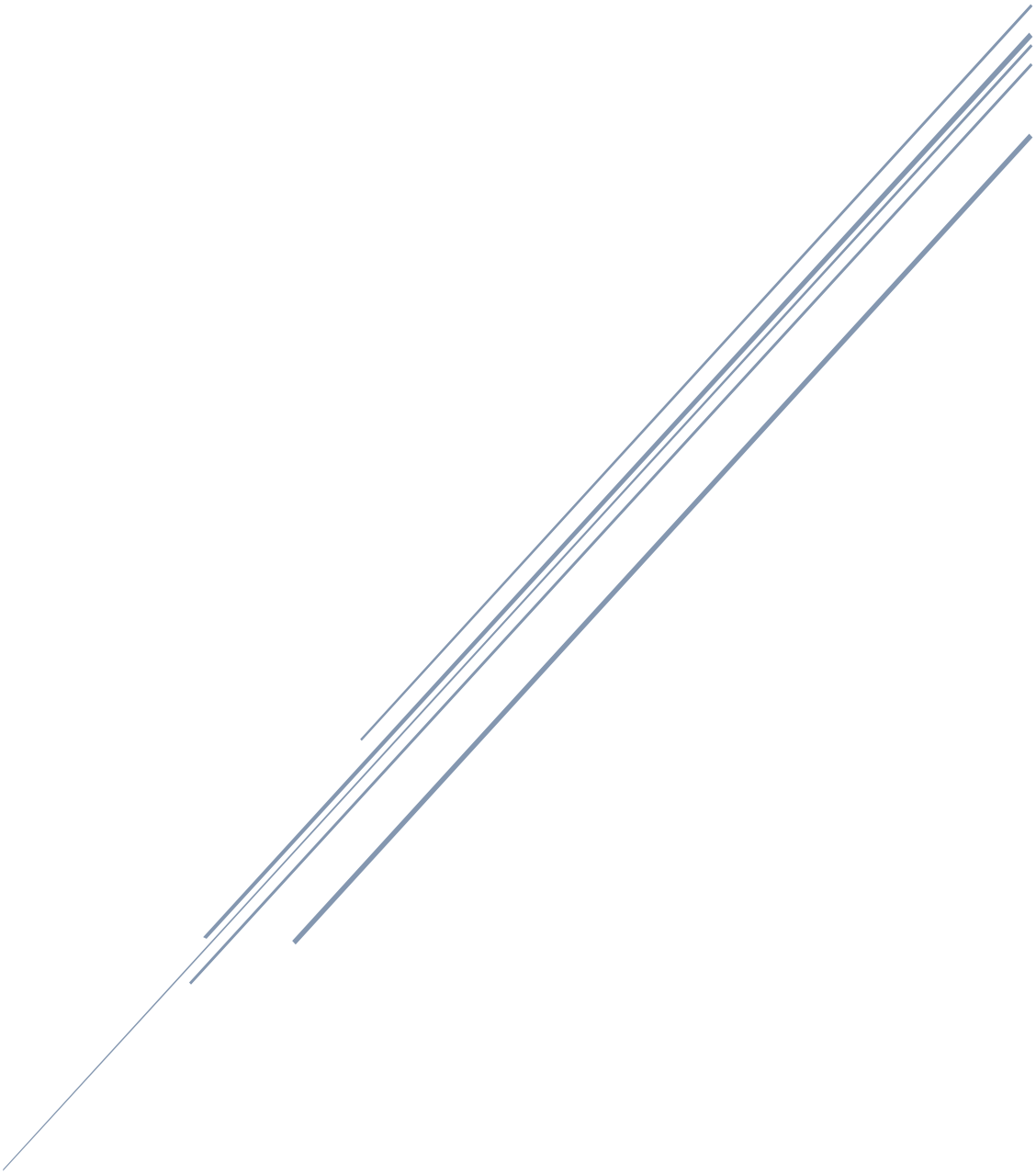
CHAPTER 1

Mathematical Foundations



1.1 Probability and statistics

1.1.1 Construction of a probability space

The sample space Ω: The *sample space* is the set of all possible outcomes of the experiment, usually denoted by Ω. For example, two successive coin tosses have a sample space of {hh, tt, ht, th}, where "h" denotes "heads" and "t" denotes "tails".

The event space A: The *event space* is the space of potential results of the experiment. A subset A of the sample space Ω is in the event space A if at the end of the experiment we can observe whether a particular outcome $\omega \in \Omega$ is in A. The event space A is obtained by considering the collection of subsets of Ω, and for discrete probability distributions A is often the power set of Ω.

The probability P: With each event $A \in A$, we associate a number P(A) that measures the probability or degree of belief that the event will occur. P(A) is called the *probability* of A.

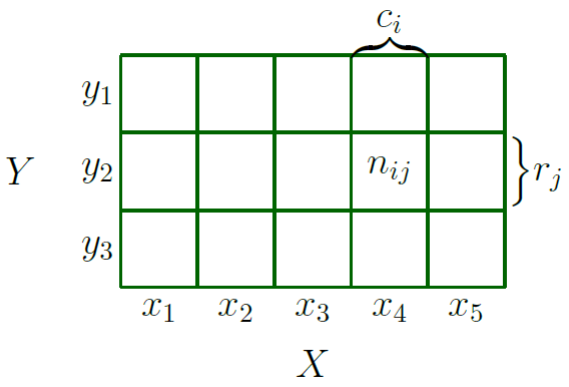
1.1.2 Discrete and Continuous Probabilities

Depending on whether the target space is discrete or continuous, the natural way to refer to distributions is different. When the target space T is discrete, we can specify the probability that a random variable X takes a particular value $x \in T$, denoted as P(X = x). The expression P(X = x) for a discrete random variable X is known as the probability mass function.

When the target space T is continuous, e.g., function the real line R, it is more natural to specify the probability that a random variable X is in an interval, denoted by $P(a \leqslant X \leqslant b)$ for $a < b$. By convention, we specify the probability that a random variable X is less than a particular value x, denoted by $P(X \leqslant x)$. The expression $P(X \leqslant x)$ for a continuous random variable X is known as the cumulative distribution function.

a) Discrete Probabilities

Consider two random variables X and Y, where X has five possible states and Y has three possible states. We denote by n_{ij} the number of events with state $X = x_i$ and $Y = y_j$, and denote by N the total number of events. The value c_i is the sum of the individual frequencies for the i^{th} column. Similarly, the value r_j is the row sum. Using these definitions, we can compactly express the distribution of X and Y.



- The probability distribution of each random variable, the marginal probability, can be seen as the sum over a row or column.

$$P(X = x_i) = \frac{c_i}{N} = \frac{\sum_{j=1}^{3} n_{ij}}{N}$$

And

$$P(Y = y_j) = \frac{r_j}{N} = \frac{\sum_{i=1}^{5} n_{ij}}{N}$$

where c_i and r_j are the i^{th} column and j^{th} row of the probability table, respectively. By convention, for discrete random variables with a finite number of events, we assume that probabilities sum up to one, that is,

$$\sum_{i=1}^{5} P(X = x_i) = 1 \quad \text{and} \quad \sum_{j=1}^{3} P(Y = y_j) = 1$$

- The conditional probability is the fraction of a row or column in a particular cell. For example, the conditional probability of Y given X is

$$P(Y = y_j \mid X = x_i) = \frac{n_{ij}}{c_i}$$

and the conditional probability of X given Y is

$$P(X = x_i \mid Y = y_j) = \frac{n_{ij}}{r_j}$$

In machine learning, we use discrete probability distributions to model *categorical variables*, i.e., variables that take a finite set of unordered values. They could be categorical labels, such as letters of the alphabet when doing handwriting recognition.

b) Continuous Probabilities

We consider real-valued random variables, i.e., we consider target spaces that are intervals of the real line R.

Definition (Probability Density Function)**:** A function $f : \mathrm{R}^{\mathrm{D}} \rightarrow \mathrm{R}$ is called a *probability density function* (*pdf*) if

1. $\forall \boldsymbol{x} \in \mathbb{R}^D : f(\boldsymbol{x}) \geqslant 0$
2. Its integral exists and

$$\int_{\mathbb{R}^D} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1 \,.$$

For probability mass functions (pmf) of discrete random variables, the integral is replaced with a sum. Observe that the probability density function is any function f that is non-negative and integrates to one. We associate a random variable X with this function f by

$$P(a \leqslant X \leqslant b) = \int_a^b f(x) \mathrm{d}x$$

where $a, b \in \mathrm{R}$ and $x \in \mathrm{R}$ are outcomes of the continuous random variable X. States $x \in \mathrm{R}^{\mathrm{D}}$ are defined analogously by considering a vector of $x \in \mathrm{R}$. This association is called the *law* or *distribution* of the random variable X.

In contrast to discrete random variables, the probability of a continuous random variable X taking a particular value P(X = x) is zero. This is like trying to specify an interval where a = b.

Definition (Cumulative Distribution Function)**.** A *cumulative distribution function* (cdf) of a multivariate real-valued random variable X with distribution function states $x \in R^D$ is given by

$$F_X(\boldsymbol{x}) = P(X_1 \leqslant x_1, \ldots, X_D \leqslant x_D)$$

where $X = [X_1, \ldots, X_D]^T$, and the right-hand side represents the probability that random variable Xi takes the value smaller than or equal to x_i.
The cdf can be expressed also as the integral of the probability density function f(x) so that

$$F_X(\boldsymbol{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_D} f(z_1, \ldots, z_D) \mathrm{d}z_1 \cdots \mathrm{d}z_D$$

c) Contrasting Discrete and Continuous Distributions

Probabilities are positive and the total probability sums up to one. For discrete random variables, this implies that the probability of each state must lie in the interval [0, 1]. However, for continuous random variables the normalization does not imply that the value of the density is less than or equal to 1. We illustrate this in Figure 1.2 using the *uniform distribution* for both discrete and continuous random variables.

Example: We consider two examples of the uniform distribution, where each state is equally likely to occur. This example illustrates some differences between discrete and continuous probability distributions.
Let Z be a discrete uniform random variable with three states $\{z = -1.1, z = 0.3, z = 1.5\}$. The probability mass function can be represented as a table of probability values:

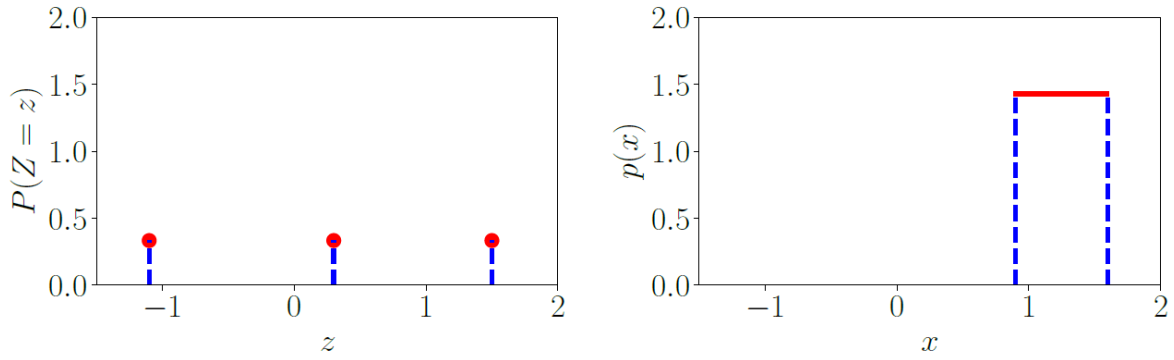
z	-1.1	0.3	1.5
$P(Z = z)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Alternatively, we can think of this as a graph (Figure 1.1(a)), where we use the fact that the states can be located on the x-axis, and the y-axis represents the probability of a particular state.

Let X be a continuous random variable taking values in the range $0.9 \leqslant X \leqslant 1.6$, as represented by Figure 1.1(b). Observe that the height of the density can be greater than 1. However, it needs to hold that

$$f(x) = \frac{1}{b-a}, \quad \text{for } x \in [a, b]$$

$$\int_{0.9}^{1.6} p(x)\mathrm{d}x = 1$$



(a) Discrete distribution (b) Continuous distribution

Figure 1.1

Type	"Point probability"	"Interval probability"
Discrete	$P(X = x)$ Probability mass function	Not applicable
Continuous	$p(x)$ Probability density function	$P(X \leqslant x)$ Cumulative distribution function

1.1.3 Sum Rule, Product Rule, and Bayes' Theorem

Given that p(x, y) is the joint distribution of the two random variables x, y. The distributions p(x) and p(y) are the corresponding marginal distributions, and p(y | x) is the conditional distribution of y given x. Given the definitions of the marginal and conditional probability for discrete and continuous random variables, we can now present the two fundamental rules in probability theory.

- The first rule, the *sum rule*, states that

$$p(\boldsymbol{x}) = \begin{cases} \sum_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) & \text{if } \boldsymbol{y} \text{ is discrete} \\ \int_{\mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} & \text{if } \boldsymbol{y} \text{ is continuous} \end{cases}$$

where Y are the states of the target space of random variable Y. This means that we sum out (or integrate out) the set of states y of the random variable Y. The sum rule is also known as the *marginalization property*. The sum rule relates the joint distribution to a marginal distribution. In general, when the joint distribution contains more than two random variables, the sum rule can be applied to any subset of the random variables, resulting in a marginal distribution of potentially more than one random variable.

- The second rule, known as the *product rule*, relates the joint distribution to the conditional distribution via

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x})$$

The product rule can be interpreted as the fact that every joint distribution of two random variables can be factorized (written as a product) of two other distributions. The two factors are the marginal distribution of the first random variable p(x), and the conditional distribution of the second random variable given the first p(y | x). Since the ordering of random variables is arbitrary in p(x, y), the product

rule also implies p(x, y) = p(x | y)p(y).

In machine learning and Bayesian statistics, we are often interested in making inferences of unobserved (latent) random variables given that we have observed other random variables. Let us assume we have some prior knowledge p(x) about an unobserved random variable x and some relationship p(y|x) between x and a second random variable y, which we can observe. If we observe y, we can use Bayes' theorem to draw some conclusions about x given the observed values of y. *Bayes' theorem* (also Bayes' theorem *Bayes' rule* or *Bayes' law*)

$$\underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y} \mid \boldsymbol{x})}^{\text{likelihood}} \overbrace{p(\boldsymbol{x})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y})}_{\text{evidence}}}$$

It is a direct consequence of the product rule since

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x} \mid \boldsymbol{y})p(\boldsymbol{y})$$

And

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x})$$

so that

$$p(\boldsymbol{x} \mid \boldsymbol{y})p(\boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x}) \iff p(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}$$

p(x) is the *prior*, which encapsulates our subjective prior knowledge of the unobserved (latent) variable x before observing any data. We can choose any prior that makes sense to us, but it is critical to ensure that the prior has a nonzero pdf (or pmf) on all plausible x, even if they are very rare.
The *likelihood* p(y | x) describes how x and y are related, and in the case of discrete probability distributions, it is the probability of the data y if we were to know the latent variable x. Note that the likelihood is not a distribution in x, but only in y. We call p(y | x) either the "likelihood of x (given y)" or the "probability of y given x" but never the likelihood of y.
The *posterior* p(x | y) is the quantity of interest in Bayesian statistics because it expresses exactly what we are interested in, i.e., what we know about x after having observed y.
The quantity

$$p(\boldsymbol{y}) := \int p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \mathbb{E}_X[p(\boldsymbol{y} \mid \boldsymbol{x})]$$

is the *marginal likelihood/evidence*.

1.1.4 Summary Statistics and Independence

The summary statistics of a distribution provide one useful view of how a random variable behaves, and as the name suggests, provide numbers that summarize and characterize the distribution. We describe the mean and the variance, two well-known summary statistics. Then we discuss how to say that two random variables are independent.

a) Means and Covariances

Mean and (co)variance are often useful to describe properties of probability distributions (expected values and spread). We will see later that there is a useful family of distributions (called the exponential family), where the statistics of the random variable capture all possible information.

Definition 6.3 (Expected Value)**.** The *expected value* of a function g : R → R of a univariate continuous random variable X ~ p(x) is given by

$$\mathbb{E}_X[g(x)] = \int_{\mathcal{X}} g(x)p(x)\mathrm{d}x$$

Correspondingly, the expected value of a function g of a discrete random variable X ~ p(x) is given by

$$\mathbb{E}_X[g(x)] = \sum_{x\in\mathcal{X}} g(x)p(x)$$

where X is the set of possible outcomes (the target space) of the random variable X.

Remark. We consider multivariate random variables X as a finite vector of univariate random variables $[X_1, \ldots, X_D]^\top$. For multivariate random variables, we define the expected value element wise

$$\mathbb{E}_X[g(\boldsymbol{x})] = \begin{bmatrix} \mathbb{E}_{X_1}[g(x_1)] \\ \vdots \\ \mathbb{E}_{X_D}[g(x_D)] \end{bmatrix} \in \mathbb{R}^D$$

where the subscript E_{Xd} indicates that we are taking the expected value with respect to the d^{th} element of the vector x.
The definition of the mean, is a special case of the expected value, obtained by choosing g to be the identity function.

In one dimension, there are two other intuitive notions of "average", which are the *median* and the *mode*.

The *median* is the "middle" value if we sort the values, i.e., 50% of the values are greater than the median and 50% are smaller than the median. This idea can be generalized to continuous values by considering the value where the cdf is 0.5. For distributions, which are asymmetric or have long tails, the median provides an estimate of a typical value that is closer to human intuition than the mean value. Furthermore, the median is more robust to outliers than the mean. The generalization of the median to higher dimensions is non-trivial as there is no obvious way to "sort" in more than one dimension.

The *mode* is the most frequently occurring value. For a discrete random variable, the mode is defined as the value of x having the highest frequency of occurrence. For a continuous random variable, the mode is defined as a peak in the density p(x). A particular density p(x) may have more than one mode, and furthermore there may be a very large number of modes in high-dimensional distributions.

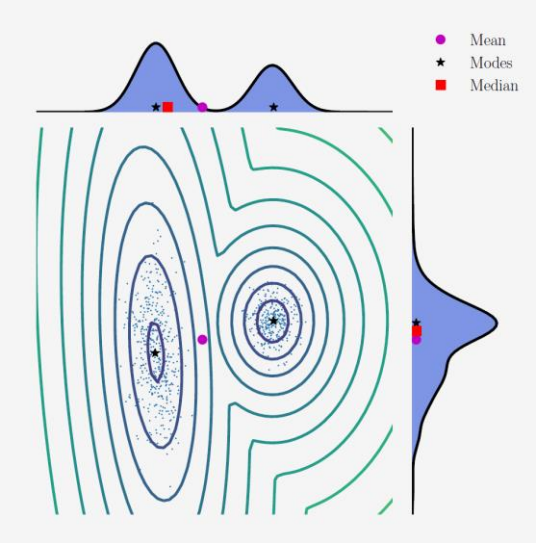


Figure 1.2 Illustration of the mean, mode, and median for a two-dimensional dataset, as well as its marginal densities.

Definition (Covariance (Univariate))**.** The *covariance* between two univariate random variables X, Y $\in$ R is given by the expected product of their deviations from their respective means, i.e.,

$$\mathrm{Cov}_{X,Y}[x, y] := \mathbb{E}_{X,Y}\left[(x - \mathbb{E}_X[x])(y - \mathbb{E}_Y[y])\right] .$$

By using the linearity of expectations, the expression in Definition 6.5 can be rewritten as the expected value of the product minus the product of the expected values, i.e.,

$$\mathrm{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] .$$

The covariance of a variable with itself Cov[x, x] is called the *variance* and is denoted by $V_X[x]$. The square root of the variance is called the *standard deviation* and is often denoted by $\sigma(x)$. The notion of covariance can be generalized to multivariate random variables.

Definition (Covariance (Multivariate))**.** If we consider two multivariate random variables X and Y with states x $\in$ R^D and y $\in$ R^E respectively, the *covariance* between X and Y is defined as

$$\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^\top] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{y}]^\top = \mathrm{Cov}[\boldsymbol{y}, \boldsymbol{x}]^\top \in \mathbb{R}^{D\times E}$$

This definition can be applied with the same multivariate random variable in both arguments, which results in a useful concept that intuitively captures the "spread" of a random variable. For a multivariate random variable, the variance describes the relation between individual dimensions of the random variable.

Definition (Variance)**.** The *variance* of a random variable X with states x $\in$ R^D and a mean vector μ $\in$ R^D is defined as

$$\begin{aligned}\mathbb{V}_X[\boldsymbol{x}] &= \mathrm{Cov}_X[\boldsymbol{x},\boldsymbol{x}] \\ &= \mathbb{E}_X[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top] = \mathbb{E}_X[\boldsymbol{x}\boldsymbol{x}^\top] - \mathbb{E}_X[\boldsymbol{x}]\mathbb{E}_X[\boldsymbol{x}]^\top \\ &= \begin{bmatrix} \mathrm{Cov}[x_1,x_1] & \mathrm{Cov}[x_1,x_2] & \dots & \mathrm{Cov}[x_1,x_D] \\ \mathrm{Cov}[x_2,x_1] & \mathrm{Cov}[x_2,x_2] & \dots & \mathrm{Cov}[x_2,x_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[x_D,x_1] & \dots & \dots & \mathrm{Cov}[x_D,x_D] \end{bmatrix}.\end{aligned}$$

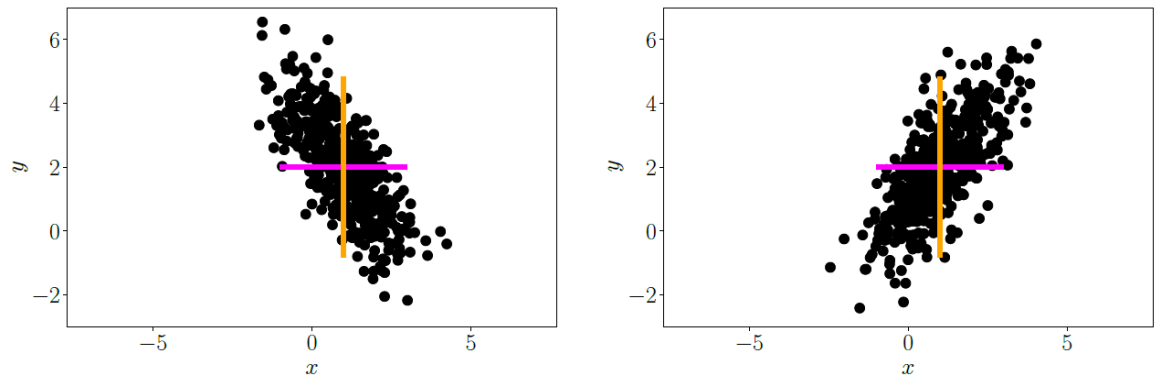
The D×D matrix is called the *covariance matrix* of the multivariate random variable X. The covariance matrix is symmetric and positive semidefinite and tells us something about the spread of the data. On its diagonal, the covariance matrix contains the variances of the *marginals*, where "i" denotes "all variables but i". The off-diagonal entries are the cross-covariance terms $\mathrm{Cov}[x_i, x_j]$ for i, j = 1, . . . ,D, i ≠ j.

Definition (Correlation)**.** The *correlation* between two random variables X, Y is given by

$$\mathrm{corr}[x,y] = \frac{\mathrm{Cov}[x,y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}} \in [-1,1].$$

The correlation matrix is the covariance matrix of standardized random variables, $x/\sigma(x)$. In other words, each random variable is divided by its standard deviation (the square root of the variance) in the correlation matrix.

The covariance (and correlation) indicate how two random variables are related; see Figure 1.3. Positive correlation corr[x, y] means that when x grows, then y is also expected to grow. Negative correlation means that as x increases, then y decreases.



(a) x and y are negatively correlated. (b) x and y are positively correlated.

Figure 1.3 Two-dimensional datasets with identical means and variances along each axis (colored lines) but with different covariances.

b) Empirical Means and Covariances

The definitions in the previous section are often also called the *population mean and covariance*, as it refers to the true statistics for the population. In machine learning, we need to learn from empirical observations of data. Consider a random variable X. There are two conceptual steps to go from

population statistics to the realization of empirical statistics. First, we use the fact that we have a finite dataset (of size N) to construct an empirical statistic that is a function of a finite number of identical random variables, $X_1, \ldots, X_N$. Second, we observe the data, that is, we look at the realization $x_1, \ldots, x_N$ of each of the random variables and apply the empirical statistic.

Definition (Empirical Mean and Covariance)**.** The *empirical mean* vector is the arithmetic average of the observations for each variable, and it is defined as

$$\bar{\boldsymbol{x}} := \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\,,$$

where $x_n \in R^D$.

Similar to the empirical mean, the *empirical covariance* matrix is a D×D matrix.

$$\boldsymbol{\Sigma} := \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^\top$$

To compute the statistics for a particular dataset, we would use the realizations (observations) $x_1, \ldots, x_N$ and use the empirical mean and covariance. Empirical covariance matrices are symmetric, positive semidefinite.

c) Statistical Independence

Definition (Independence)**.** Two random variables X, Y are *statistically independent* if and only if

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

Intuitively, two random variables X and Y are independent if the value of y (once known) does not add any additional information about x (and vice versa). If X, Y are (statistically) independent, then

- $p(\boldsymbol{y} \mid \boldsymbol{x}) = p(\boldsymbol{y})$
- $p(\boldsymbol{x} \mid \boldsymbol{y}) = p(\boldsymbol{x})$
- $\mathbb{V}_{X,Y}[\boldsymbol{x} + \boldsymbol{y}] = \mathbb{V}_X[\boldsymbol{x}] + \mathbb{V}_Y[\boldsymbol{y}]$
- $\mathrm{Cov}_{X,Y}[\boldsymbol{x}, \boldsymbol{y}] = \boldsymbol{0}$

In machine learning, we often consider problems that can be modeled as *independent and identically distributed* (*i.i.d.*) random variables, $X_1, \ldots, X_N$. For more than two random variables, the word "independent" usually refers to mutually independent random variables, where all subsets are independent. The phrase "identically distributed" means that all the random variables are from the same distribution.

Definition (Conditional Independence)**.** Two random variables X and Y are *conditionally independent* given Z if and only if

$$p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{y} \mid \boldsymbol{z}) \quad \text{for all} \quad \boldsymbol{z} \in \mathcal{Z}\,,$$

where Z is the set of states of random variable Z. We write $X \perp\!\!\!\perp Y \mid Z$ to denote that X is conditionally independent of Y given Z.

1.1.5 Gaussian Distribution

The Gaussian distribution is the most well-studied probability distribution for continuous-valued random variables. It is also referred to as the *normal distribution*. Its importance originates from the fact that it has many computationally convenient properties, which we will be discussing in the following. In particular, we will use it to define the likelihood and prior for linear regression.

There are many other areas of machine learning that also benefit from using a Gaussian distribution, for example Gaussian processes, variational inference, and reinforcement learning. It is also widely used in other application areas such as signal processing (e.g., Kalman filter), control (e.g., linear quadratic regulator), and statistics (e.g., hypothesis testing).

For a univariate random variable, the Gaussian distribution has a density that is given by

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

The *multivariate Gaussian distribution* is fully characterized by a *mean* vector μ and a covariance matrix Σ and defined as

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

where $x \in R^D$. We write $p(x) = N(x \mid \mu, \Sigma)$ or $X \sim N(\mu, \Sigma)$. Figure 1.4 shows a bivariate Gaussian (mesh), with the corresponding contour plot. Figure 1.5 shows a univariate Gaussian and a bivariate Gaussian with corresponding samples. The special case of the Gaussian with zero mean and identity covariance, that is, $\mu = 0$ and $\Sigma = I$, is referred to as the *standard normal distribution*.

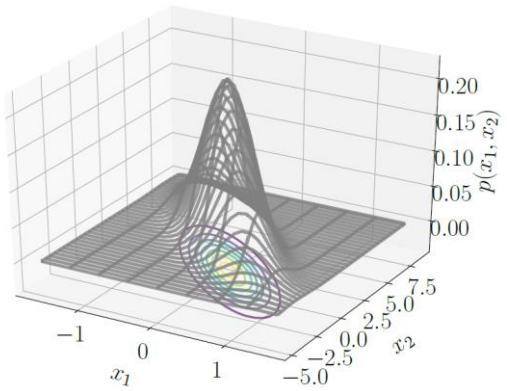
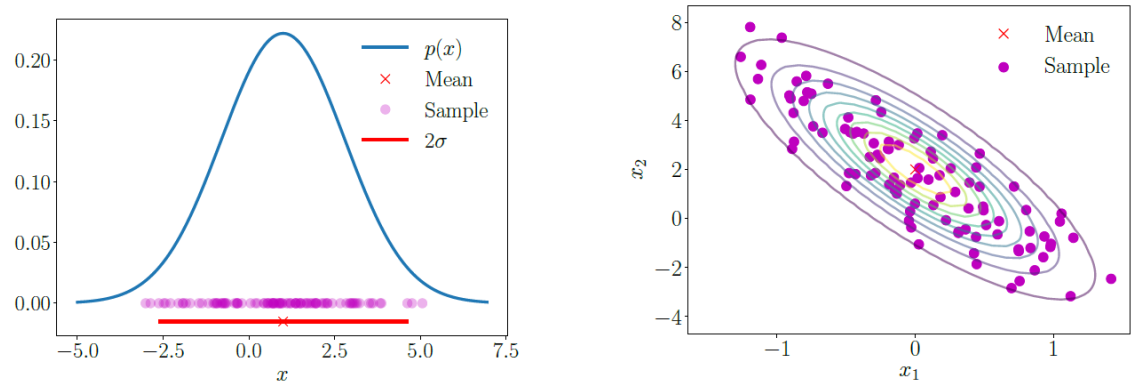


Figure 1.4 Gaussian distribution of two random variables x_1 and x_2.



(a) Univariate (one-dimensional) Gaussian; The red cross shows the mean and the red line shows the extent of the variance.

(b) Multivariate (two-dimensional) Gaussian, viewed from top. The red cross shows the mean and the colored lines show the contour lines of the density.

Figure 1.5 Gaussian distributions overlaid with 100 samples. (a) One dimensional case; (b) two-dimensional case.

Gaussians are widely used in statistical estimation and machine learning as they have closed-form expressions for marginal and conditional distributions. We will use these closed-form expressions extensively for linear regression. A major advantage of modeling with Gaussian random variables is that variable transformations are often not needed. Since the Gaussian distribution is fully specified by its mean and covariance, we often can obtain the transformed distribution by applying the transformation to the mean and covariance of the random variable.

It is worth recalling at this point the desiderata for manipulating probability distributions in the machine learning context:

1. There is some "closure property" when applying the rules of probability, e.g., Bayes' theorem. By closure, we mean that applying a particular operation returns an object of the same type.
2. As we collect more data, we do not need more parameters to describe the distribution.
3. Since we are interested in learning from data, we want parameter estimation to behave nicely.

It turns out that the class of distributions called the *exponential family*. Before we introduce the exponential family, let us see three more members of "named" probability distributions, the Bernoulli, Binomial, and Beta distributions.

1.1.6 Bernoulli distribution

The *Bernoulli distribution* is a distribution for a single binary random variable X with state $x \in \{0, 1\}$. It is governed by a single continuous parameter $\mu \in [0, 1]$ that represents the probability of X = 1. The Bernoulli distribution Ber(μ) is defined as

$$
\begin{aligned}
p(x \mid \mu) &= \mu^x (1-\mu)^{1-x}, \quad x \in \{0, 1\}, \\
\mathbb{E}[x] &= \mu, \\
\mathbb{V}[x] &= \mu(1-\mu),
\end{aligned}
$$

where E[x] and V[x] are the mean and variance of the binary random variable X.

1.1.7 Binomial Distribution

The *Binomial distribution* is a generalization of the Bernoulli distribution to a distribution over integers. In particular, the Binomial can be used to describe the probability of observing m occurrences of X = 1 in a set of N samples from a Bernoulli distribution where p(X = 1) = μ ∈ [0, 1]. The Binomial distribution Bin(N, μ) is defined as

$$p(m \mid N, \mu) = \binom{N}{m} \mu^m (1-\mu)^{N-m},$$
$$\mathbb{E}[m] = N\mu,$$
$$\mathbb{V}[m] = N\mu(1-\mu),$$

where E[m] and V[m] are the mean and variance of m, respectively.

1.1.8 Exponential Family

There are three possible levels of abstraction we can have when considering distributions (of discrete or continuous random variables). At level one (the most concrete end of the spectrum), we have a particular named distribution with fixed parameters, for example a univariate Gaussian N(0, 1) with zero mean and unit variance. In machine learning, we often use the second level of abstraction, that is, we fix the parametric form (the univariate Gaussian) and infer the parameters from data. For example, we assume a univariate Gaussian $N(\mu, \sigma^2)$ with unknown mean μ and unknown variance σ^2, and use a maximum likelihood fit to determine the best parameters (μ, σ^2). We will see an example of this when considering linear regression. A third level of abstraction is to consider families of distributions, we consider the exponential family. The univariate Gaussian is an example of a member of the exponential family. Many of the widely used statistical models, are members of the exponential family.

1.2 Vector calculus

Many machine learning algorithms optimize an objective function by adjusting model parameters that determine how well the model fits the data. This process can be framed as an optimization problem.
A function f is a quantity that relates two quantities to each other. These quantities are typically inputs $x \in \mathbb{R}^D$ and targets (function values) $f(x)$, which we assume are real-valued. Here $\mathbb{R}^D$ is the *domain* of f, and the function values $f(x)$ are the *image/codomain* of f.

$$f: \mathbb{R}^D \to \mathbb{R}$$
$$x \to f(x)$$

1.2.1 Differentiation of Univariate Functions

Definition (Difference Quotient): The difference quotient computes the slope of the secant line through two points on the graph of f. In Figure 1.1, these are the points with x-coordinates x_0 and $x_0 + \delta x$.

$$\frac{\delta y}{\delta x} := \frac{f(x+\delta x) - f(x)}{\delta x}$$

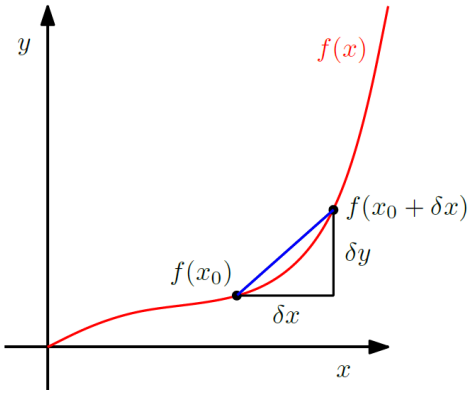


Figure 1.6 Difference quotient

In the limit for $\delta x \to 0$, we obtain the tangent of f at x, if f is differentiable. The tangent is then the derivative of f at x.

Definition (Derivative): More formally, for $h>0$ the *derivative* of f derivative at x is defined as the limit.

$$\frac{\mathrm{d}f}{\mathrm{d}x} := \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

and the secant in Figure 1.1 becomes a tangent. The derivative of f points in the direction of steepest ascent of f.

1.2.2 Differentiation Rules

In the following, we briefly state basic differentiation rules, where we denote the derivative of f by f'.

$$\begin{aligned}
&\text{Product rule:} && (f(x)g(x))' = f'(x)g(x) + f(x)g'(x) \\
&\text{Quotient rule:} && \left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2} \\
&\text{Sum rule:} && (f(x) + g(x))' = f'(x) + g'(x) \\
&\text{Chain rule:} && \big(g(f(x))\big)' = (g \circ f)'(x) = g'(f(x))f'(x)
\end{aligned}$$

Here, $g \circ f$ denotes function composition $x \mapsto f(x) \mapsto g(f(x))$.

1.2.3 Partial Differentiation and Gradients

Differentiation as discussed in the previous section applies to functions f of a scalar variable $x \in R$. In the following, we consider the general case where the function f depends on one or more variables $x \in R^n$, e.g., f(x) = f(x1, x2). The generalization of the derivative to functions of several variables is the *gradient*.

We find the gradient of the function f with respect to x by *varying one variable at a time* and keeping the others constant. The gradient is then the collection of these *partial derivatives*.

Definition (Partial Derivative): For a function $f : R^n \to R$, $x \to f(x)$, $x \in$ partial derivative R^n of n variables $x_1, \ldots, x_n$ we define the *partial derivatives* as:

$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \ldots, x_n) - f(\boldsymbol{x})}{h}$$
$$\vdots$$
$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{n-1}, x_n + h) - f(\boldsymbol{x})}{h}$$

and collect them in the row vector

$$\nabla_{\boldsymbol{x}} f = \operatorname{grad} f = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1 \times n}$$

where n is the number of variables and 1 is the dimension of the image/range/codomain of f. Here, we defined the column vector $x = [x_1, \ldots, x_n]^\top \in R^n$. The row vector is called the *gradient* of f.

1.2.4 Basic Rules of Partial Differentiation

In the multivariate case, where $x \in R^n$, the basic differentiation rules that we know (e.g., sum rule, product rule, chain rule) still apply. However, when we compute derivatives with respect to vectors $x \in R^n$ we need to pay attention: Our gradients now involve vectors and matrices, and matrix multiplication is not commutative, i.e., the order matters.

Here are the general product rule, sum rule, and chain rule:

Product rule: $$\frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x})g(\boldsymbol{x})) = \frac{\partial f}{\partial \boldsymbol{x}} g(\boldsymbol{x}) + f(\boldsymbol{x}) \frac{\partial g}{\partial \boldsymbol{x}}$$

Sum rule: $$\frac{\partial}{\partial \boldsymbol{x}}(f(\boldsymbol{x}) + g(\boldsymbol{x})) = \frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{x}}$$

Chain rule: $$\frac{\partial}{\partial \boldsymbol{x}}(g \circ f)(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}(g(f(\boldsymbol{x}))) = \frac{\partial g}{\partial f} \frac{\partial f}{\partial \boldsymbol{x}}$$

1.2.5 Chain rule

Consider a function f: $R^2 \to R$ of two variables x_1, x_2. Furthermore, $x_1(t)$ and $x_2(t)$ are themselves functions of t. To compute the gradient of f with respect to t, we need to apply the chain rule for multivariate functions as:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1(t)}{\partial t} \\ \frac{\partial x_2(t)}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}$$

where d denotes the gradient and ∂ partial derivatives.

If $f(x_1, x_2)$ is a function of x_1 and x_2, where $x_1(s, t)$ and $x_2(s, t)$ are themselves functions of two variables s and t, the chain rule yields the partial derivatives.

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial s}$$
$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}$$

and the gradient is obtained by the matrix multiplication

$$\frac{\mathrm{d}f}{\mathrm{d}(s,t)} = \frac{\partial f}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial (s,t)} = \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}}_{= \frac{\partial f}{\partial \boldsymbol{x}}} \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial s} & \frac{\partial x_2}{\partial t} \end{bmatrix}}_{= \frac{\partial \boldsymbol{x}}{\partial (s,t)}}$$

This compact way of writing the chain rule as a matrix multiplication only makes sense if the gradient is defined as a row vector. Otherwise, we will need to start transposing gradients for the matrix dimensions to match. This may still be straightforward as long as the gradient is a vector or a matrix; however, when the gradient becomes a tensor (we will discuss this in the following), the transpose is no longer a triviality.

1.2.6 Gradients of Vector-Valued Functions

Thus far, we discussed partial derivatives and gradients of functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ mapping to the real numbers. In the following, we will generalize the concept of the gradient to vector-valued functions (vector fields) $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, where $n \geqslant 1$ and $m > 1$.

For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and a vector $x = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$, the corresponding vector of function values is given as:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^m$$

Writing the vector-valued function in this way allows us to view a vector-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as a vector of functions $[f_1, \ldots, f_m]^\top$, $f_i : \mathbb{R}_n \rightarrow \mathbb{R}$ that map onto R. The differentiation rules for every f_i are exactly the ones we discussed previously.

Therefore, the partial derivative of a vector-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with respect to $x_i \in \mathbb{R}$, $i = 1, \ldots n$, is given as the vector.

$$\frac{\partial \boldsymbol{f}}{\partial x_i} = \begin{bmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_m}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \lim_{h \to 0} \frac{f_1(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots x_n) - f_1(\boldsymbol{x})}{h} \\ \vdots \\ \lim_{h \to 0} \frac{f_m(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots x_n) - f_m(\boldsymbol{x})}{h} \end{bmatrix} \in \mathbb{R}^m$$

We know that the gradient of f with respect to a vector is the row vector of the partial derivatives. Every partial derivative $\partial f / \partial x_i$ is itself a column vector. Therefore, we obtain the gradient of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with respect to $x \in \mathbb{R}^n$ by collecting these partial derivatives:

$$\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \left[\boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1}} \cdots \boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_n}}\right] = \begin{bmatrix} \boxed{\frac{\partial f_1(\boldsymbol{x})}{\partial x_1}} & \cdots & \boxed{\frac{\partial f_1(\boldsymbol{x})}{\partial x_n}} \\ \vdots & & \vdots \\ \boxed{\frac{\partial f_m(\boldsymbol{x})}{\partial x_1}} & \cdots & \boxed{\frac{\partial f_m(\boldsymbol{x})}{\partial x_n}} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Definition (Jacobian)**:** The collection of all first-order partial derivatives of a vector-valued function f : $R^n \rightarrow R^m$ is called the *Jacobian*. The Jacobian J is an m × n matrix, which we define and arrange as follows:

$$\boldsymbol{J} = \nabla_{\boldsymbol{x}} \boldsymbol{f} = \frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{bmatrix},$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad J(i,j) = \frac{\partial f_i}{\partial x_j}.$$

1.2.7 Gradient of matrices

We will encounter situations where we need to take gradients of matrices with respect to vectors (or other matrices), which results in a multidimensional tensor. We can think of this tensor as a multidimensional array that collects partial derivatives. For example, if we compute the gradient of an m × n matrix A with respect to a p × q matrix B, the resulting Jacobian would be (m×n)×(p×q), i.e., a four-dimensional tensor J, whose entries are given as $J_{ijkl} = \partial A_{ij}/\partial B_{kl}$.

Example: (Gradient of Vectors with Respect to Matrices)

Let us consider the following example, where

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{f} \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M \times N}, \quad \boldsymbol{x} \in \mathbb{R}^N$$

and where we seek the gradient df/dA. Let us start again by determining the dimension of the gradient as

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} \in \mathbb{R}^{M \times (M \times N)}$$

By definition, the gradient is the collection of the partial derivatives:

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{A}} \\ \vdots \\ \frac{\partial f_M}{\partial \boldsymbol{A}} \end{bmatrix}, \quad \frac{\partial f_i}{\partial \boldsymbol{A}} \in \mathbb{R}^{1 \times (M \times N)}$$

To compute the partial derivatives, it will be helpful to explicitly write out the matrix vector multiplication:

$$f_i = \sum_{j=1}^{N} A_{ij} x_j, \quad i = 1, \ldots, M$$

and the partial derivatives are then given as

$$\frac{\partial f_i}{\partial A_{iq}} = x_q$$

This allows us to compute the partial derivatives of fi with respect to a row of A, which is given as

$$\frac{\partial f_i}{\partial A_{i,:}} = \boldsymbol{x}^\top \in \mathbb{R}^{1 \times 1 \times N}$$

where we have to pay attention to the correct dimensionality. Since f_i maps onto R and each row of A is of size 1 × N, we obtain a 1 × 1 × N-sized tensor as the partial derivative of fi with respect to a row of A.

$$\frac{\partial f_i}{\partial A_{k \neq i,:}} = \mathbf{0}^\top \in \mathbb{R}^{1 \times 1 \times N}$$

We stack the partial derivatives and get the desired gradient via:

$$\frac{\partial f_i}{\partial \boldsymbol{A}} = \begin{bmatrix} \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \\ \boldsymbol{x}^\top \\ \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \end{bmatrix} \in \mathbb{R}^{1 \times (M \times N)}$$

1.3 Data, Models, and Learning

There are three major components of a machine learning system: data, models, and learning.

1.3.1 Data as Vectors

We assume that our data can be read by a computer, and represented adequately in a numerical format. If the Data is tabular (Figure 2.1), where we think of each row of the table as representing a particular instance or example, and each column to be a particular feature. In recent years, machine learning has been applied to many types of data that do not obviously come in the tabular numerical format, for example genomic sequences, text and image contents of a webpage, and social media graphs. Identifying good features depends on domain expertise and require careful engineering, and, in recent years, they have been put under the umbrella of data science.

Table 1.1 Example data from a fictitious human resource database that is not in a numerical format.

Name	Gender	Degree	Postcode	Age	Annual salary
Aditya	M	MSc	W21BG	36	89563
Bob	M	PhD	EC1A1BA	47	123543
Chloé	F	BEcon	SW1A1BH	26	23989
Daisuke	M	BSc	SE207AT	68	138769
Elisabeth	F	MBA	SE10AA	33	113888

Even when we have data in tabular format, there are still choices to be made to obtain a numerical representation. For example, in Table 2.1, the gender column (a categorical variable) may be converted into numbers 0 representing "Male" and 1 representing "Female". Alternatively, the gender could be represented by numbers −1, +1, respectively (as shown in Table 2.2). Furthermore, it is often important to use domain knowledge when constructing the representation, such as knowing that university degrees progress from bachelor's to master's to PhD or realizing that the postcode provided is not just a string of characters but actually encodes an area in London. In Table 2.2, we converted the data from Table 8.1 to a numerical format, and each postcode is represented as two numbers, a latitude and longitude.

Table 2.2 Example data from a fictitious human resource database (see Table 2.1), converted to a numerical format.

Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

Even numerical data that could potentially be directly read into a machine learning algorithm should be carefully considered for units, scaling, and constraints. Without additional information, one should shift and scale all columns of the dataset such that they have an empirical mean of 0 and an empirical variance of 1. We assume that a domain expert already converted data appropriately, i.e., each input x_n is a D-dimensional vector of real numbers, which are called *features*, *attributes*, or *covariates*. Observe that we have dropped the Name column of Table 2.2 in the new numerical representation. There are two main reasons why this is desirable: (1) we do not expect the identifier (the Name) to be informative for a machine learning task; and (2) we may wish to anonymize the data to help protect the privacy of the employees.

We will use N to denote the number of examples in a dataset and index the examples with lowercase n = 1, . . . , N. We assume that we are given a set of numerical data, represented as an array of vectors (Table 2.2). Each row is a particular individual x_n, often referred to as an *example* or *data point* in machine learning. The subscript example n refers to the fact that this is the n^{th} example out of a total of N examples in the dataset. Each column represents a particular feature of interest about the example, and we index the features as d = 1, . . . , D. Recall that data is represented as vectors, which means that each example (each data point) is a D-dimensional vector.

Let us consider the problem of predicting annual salary from age, based on the data in Table 2.2. This is called a supervised learning problem where we have a label y_n (the salary) associated with each example x_n label (the age). The label y_n has various other names, including *target*, *response variable*, and *annotation*. A dataset is written as a set of example-label pairs $\{(x_1, y_1), \ldots, (x_n, y_n), \ldots, (x_N, y_N)\}$. The table of examples $\{x_1, \ldots, x_N\}$ is often concatenated, and written as $X \in R^{N \times D}$.

1.3.2 Models as Functions

Once we have data in an appropriate vector representation, we can get to the business of constructing a predictive function (known as a model). We present two major approaches for the prediction: a model as a function, and a model as a probabilistic model.

A model is a function that, when given a particular input example (in our case, a vector of features),

produces an output. For now, consider the output to be a single number, i.e., a real-valued scalar output. This can be written as

$$f : \mathbb{R}^D \to \mathbb{R},$$

Example: A possible function that can be used to compute the value of the prediction for input values x is the linear function.

$$f(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + \theta_0$$

For unknown θ and θ_1. Figure 2.1 illustrates a possible function that can be used to compute the value of the prediction for input values x.

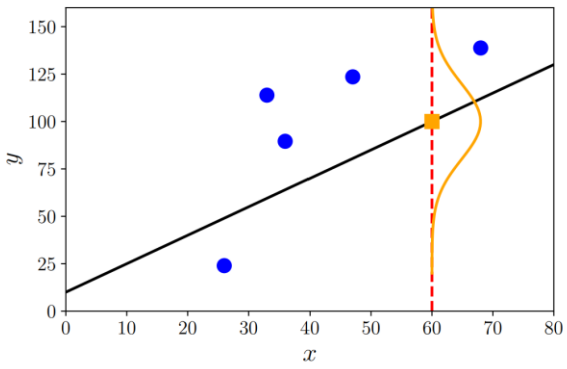


Figure 1.7 Example function (black solid diagonal line) and its predictive uncertainty at x = 60 (drawn as a Gaussian).

1.3.3 Models as Probability Distributions

We often consider data to be noisy observations of some true underlying effect, and hope that by applying machine learning we can identify the signal from the noise. This requires us to have a language for quantifying the effect of noise. We often would also like to have the predictions of the models that express some sort of uncertainty, e.g., to quantify the confidence we have about the value of the prediction for a particular test data point. Figure 2.1 illustrates the predictive uncertainty of the function as a Gaussian distribution.

1.3.4 Learning is Finding Parameters

The goal of learning is to find a model and its corresponding parameters such that the resulting predictor will perform well on unseen data. There are conceptually three distinct algorithmic phases when discussing machine learning algorithms:

1) *Training or parameter estimation:* The training or parameter estimation phase is when we adjust our predictive model based on training data. We would like to find good model given training data. We use numerical methods to find good parameters that "fit" the data, and most training methods can be thought of as hill-climbing approaches to find the maximum of an objective. We are interested in learning a model based on data such that it performs well on future data. It is not enough for the model to only fit the training data well; the model needs to perform well on unseen data. We simulate the behavior of our model on future unseen data using validation. To achieve the goal of performing well on unseen data, we will need to balance between fitting well on training

data and finding "simple" explanations of the phenomenon.

2) *Hyperparameter tuning or model selection:* We often need to make high-level modeling decisions about the structure of the predictor, such as the number of components to use or the class of probability distributions to consider. The choice of the number of components is an example of a hyperparameter, and this choice can affect the performance of the model significantly. The hyperparameters are often optimized on the validation set.

   The distinction between parameters and hyperparameters is somewhat arbitrary, and is mostly driven by the distinction between what can be numerically optimized versus what needs to use search techniques.

3) *Prediction or inference:* The prediction phase is when we use a trained model on previously unseen test data. In other words, the parameters and model choice is already fixed and the model is applied to new vectors representing new input data points.