2 Mathematical Foundation

Before we dig into any particular machine learning method, we will first review some important subjects in mathematics and statistics because they form the foundation for almost all machine learning methods. In particular, we will cover some relevant topics in *linear algebra*, *probability and statistics*, *information theory*, and *mathematical optimization*. This chapter stresses the mathematical knowledge that is required to understand the following chapters, and meanwhile, it presents many examples to prepare readers for the notation used in this book. Moreover, the coverage in this chapter is intended to be as self-contained as possible so that readers can study it without referring to other materials. All readers are encouraged to go over this chapter first so as to become acquainted with the mathematical background as well as the notation used in the book.

2.1 Linear Algebra

2.1.1 Vectors and Matrices

A *scalar* is a single number, often denoted by a lowercase letter, such as x or n. We also use $x \in \mathbb{R}$ to indicate that x is a real-valued scalar and $n \in \mathbb{N}$ for that n is a natural number. A *vector* is a list of numbers arranged in order, denoted by a lowercase letter in bold, such as $\mathbf{x}$ or $\mathbf{y}$. All numbers in a vector can be aligned in a row or column, called a *row vector* or *column vector*, accordingly. We use $\mathbf{x} \in \mathbb{R}^n$ to indicate that $\mathbf{x}$ is an n-dimensional vector containing n real numbers. This book adopts the convention of writing a vector in a column, such as the following:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

A *matrix* is a group of numbers arranged in a two-dimensional array, often denoted by an uppercase letter in bold, such as $\mathbf{A}$ or $\mathbf{B}$. For example, a matrix containing m rows and n columns is called an m $\times$ n *matrix*,

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

represented as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Along the same lines, we can arrange a group of numbers in a three-dimensional or higher-dimensional array, which is often called a *tensor*.

We use $\mathbf{A} \in \mathbb{R}^{m \times n}$ to indicate that $\mathbf{A}$ is an $m \times n$ matrix containing all real numbers.

2.1.2 Linear Transformation as Matrix Multiplication

A common question that beginners have is why we need vectors and matrices and what we can do with them. We can easily spot that vectors may be viewed as special matrices. However, it must be noted that vectors and matrices represent very different concepts in mathematics. An n-dimensional vector can be viewed as a point in an n-dimensional space if we interpret each number in the vector as the coordinate along an axis. Each axis in turn can be viewed as some measurement of one particular characteristic of an object. In other words, vectors can be viewed as an abstract way to represent objects in mathematics. On the other hand, a matrix represents a motion of all points in a space (i.e., one particular way to move any point in a space into a different position in another space). Alternatively, a matrix can be viewed as a particular way to transform the representations of objects from one space to another. More importantly, the exact algorithm to implement such motion is to take advantage of a matrix operation, called *matrix multiplication*, which is defined as shown in Figure 2.1.

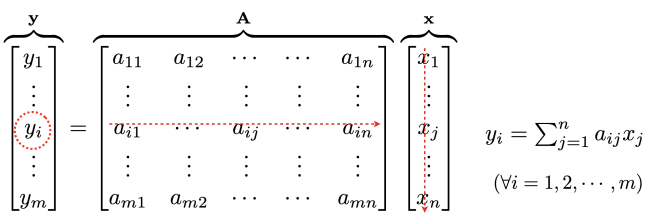


Figure 2.1: An illustration of how to implement linear transformation using matrix multiplication.

We denote this as $\mathbf{y} = \mathbf{A}\mathbf{x}$ for short. Using the matrix multiplication, any point $\mathbf{x}$ in the first space $\mathbb{R}^n$ is transformed into another point $\mathbf{y}$ in a different space $\mathbb{R}^m$. The exact mapping between $\mathbf{x}$ and $\mathbf{y}$ depends on all numbers in the matrix $\mathbf{A}$. If $\mathbf{A}$ is a square matrix in $\mathbb{R}^{n \times n}$, this mapping can also be viewed as transforming one point $\mathbf{x} \in \mathbb{R}^n$ into another point $\mathbf{y}$ in the same space $\mathbb{R}^n$.

However, this matrix multiplication cannot implement any arbitrary mapping between two spaces. The matrix multiplication actually can only

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

implement a small subset of all possible mappings called *linear transformations*. As shown in Figure 2.2, a linear transformation is a mapping from the first space $\mathbb{R}^n$ to another space $\mathbb{R}^m$ that must satisfy two conditions: (i) the origin in $\mathbb{R}^n$ is mapped to the origin in $\mathbb{R}^m$; (ii) every straight line in $\mathbb{R}^n$ is always mapped to a straight line (or a single point) in $\mathbb{R}^m$. Other mappings that do not satisfy these two conditions are called *nonlinear transformations*, which must be implemented by other methods rather than matrix multiplication.

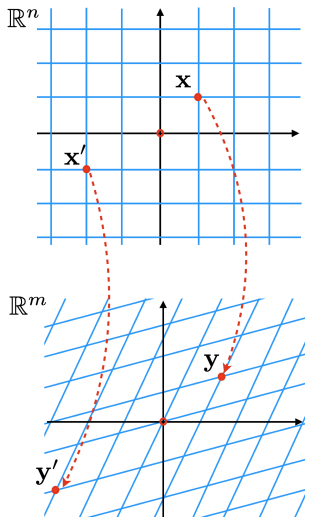
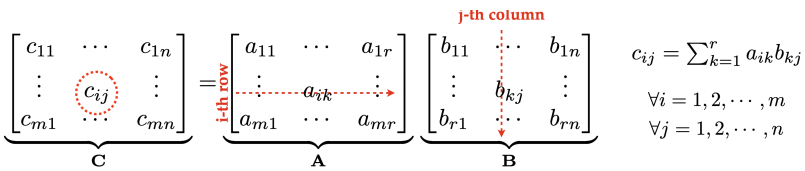


Figure 2.2: An illustration of mapping a point from one space $\mathbb{R}^n$ to another space $\mathbb{R}^m$ through a linear transformation.

This matrix multiplication method can be done between two matrices. For example, we can have the following:



$$c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}$$
$$\forall i = 1, 2, \cdots, m$$
$$\forall j = 1, 2, \cdots, n$$

We denote this as $\mathbf{C} = \mathbf{AB}$ for short. Note that the column number of the first matrix $\mathbf{A}$ must match the row number of the second matrix $\mathbf{B}$ so that they can be multiplied together.

Conceptually speaking, this matrix multiplication corresponds to a composition of two linear transformations. As shown in Figure 2.3, $\mathbf{A}$ represents a linear transformation from the first space $\mathbb{R}^n$ to the second space $\mathbb{R}^r$, and $\mathbf{B}$ represents another linear transformation from the second space $\mathbb{R}^r$ to the third space $\mathbb{R}^m$. The matrix multiplication $\mathbf{C} = \mathbf{AB}$ composes these two transformations to derive a direct linear transformation from the first space $\mathbb{R}^n$ to the third one $\mathbb{R}^m$. Because this process has to go through the same space in the middle, these two matrices must match each other in their dimensions, as described previously.

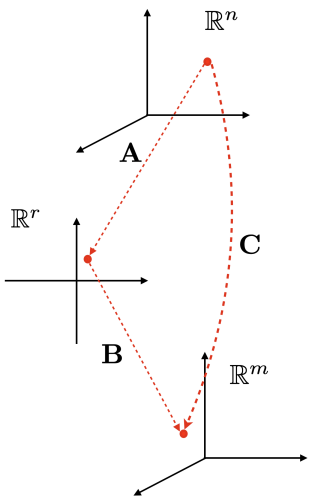


Figure 2.3: An illustration of composing two linear transformations into another linear transformation by matrix multiplication.

2.1.3 Basic Matrix Operations

The transpose of a matrix $\mathbf{A}$ is an operator that flips the matrix over its diagonal so that all rows become columns, and vice versa. The new matrix is denoted as $\mathbf{A}^\intercal$. If $\mathbf{A}$ is an $m \times n$ matrix, then $\mathbf{A}^\intercal$ will be an $n \times m$ matrix.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix} \Longrightarrow \mathbf{A}^\intercal = \begin{bmatrix} a_{11} & a_{21} & \cdots & \cdots & a_{m1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1i} & \cdots & a_{ji} & \cdots & a_{mi} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \cdots & \cdots & a_{mn} \end{bmatrix}$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

We have

$$(\mathbf{A}^\top)^\top = \mathbf{A}$$
$$(\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top$$
$$(\mathbf{A} \pm \mathbf{B})^\top = \mathbf{A}^\top \pm \mathbf{B}^\top$$

A square matrix $\mathbf{A}$ is symmetric if and only if

$$\mathbf{A}^\top = \mathbf{A}.$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

For any $\mathbf{A} \in \mathbb{R}^{n\times n}$, we have

$$\mathbf{AI} = \mathbf{IA} = \mathbf{A}.$$

We can verify that

$$|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|}.$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad \Longrightarrow \quad \mathbf{w}^\top = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}.$$

For any square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we can compute a real number for it, called the *determinant*, denoted as $|\mathbf{A}|$ ($\in \mathbb{R}$). As we know, a square matrix $\mathbf{A}$ represents a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^n$, and it will transform any unit hypercube in the original space into a polyhedron in the new space. The determinant $|\mathbf{A}|$ represents the volume of the polyhedron in the new space.

We often use $\mathbf{I}$ to represent a special square matrix, called an *identity* matrix, that has all 1s in its diagonal and 0s everywhere else. For a square matrix $\mathbf{A}$, if we can find another square matrix, denoted as $\mathbf{A}^{-1}$, that satisfies

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I},$$

we call $\mathbf{A}^{-1}$ the inverse matrix of $\mathbf{A}$. We say $\mathbf{A}$ is invertible if its inverse matrix $\mathbf{A}^{-1}$ exists.

The inner product between any two n-dimensional vectors (e.g., $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$) is defined as the sum of all element-wise multiplications between them, denoted as $\mathbf{w} \cdot \mathbf{x}$ ($\in \mathbb{R}$) . We can further represent the inner product using the matrix transpose and multiplication as follows:

$$\mathbf{w} \cdot \mathbf{x} \stackrel{\Delta}{=} \sum_{i=1}^{n} w_i x_i = \mathbf{w}^\top\mathbf{x} = \mathbf{x}^\top\mathbf{w}.$$

The norm of a vector $\mathbf{w}$ (a.k.a. the L_2 norm), denoted as $\|\mathbf{w}\|$, is defined as the square root of the inner product with itself. The meaning of the norm $\|\mathbf{w}\|$ represents the length of the vector $\mathbf{w}$ in the Euclidean space:

$$\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w} = \sum_{i=1}^{n} w_i^2 = \mathbf{w}^\top\mathbf{w}.$$

Example 2.1.1 Given two n-dimensional vectors, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^n$, and an $n \times n$ matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, reparameterize the following norms using matrix multiplication:

$$\|\mathbf{z} - \mathbf{x}\|^2 \qquad \text{and} \qquad \|\mathbf{z} - \mathbf{Ax}\|^2.$$

$$\|\mathbf{z} - \mathbf{x}\|^2 = (\mathbf{z} - \mathbf{x})^\top(\mathbf{z} - \mathbf{x}) = (\mathbf{z}^\top - \mathbf{x}^\top)(\mathbf{z} - \mathbf{x}) = \mathbf{z}^\top\mathbf{z} + \mathbf{x}^\top\mathbf{x} - 2\,\mathbf{z}^\top\mathbf{x}.$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

$$\begin{aligned}\|\mathbf{z}-\mathbf{A}\mathbf{x}\|^2 &= (\mathbf{z}-\mathbf{A}\mathbf{x})^\top(\mathbf{z}-\mathbf{A}\mathbf{x}) = (\mathbf{z}^\top-\mathbf{x}^\top\mathbf{A}^\top)(\mathbf{z}-\mathbf{A}\mathbf{x})\\ &= \mathbf{z}^\top\mathbf{z}+\mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\mathbf{x}-2\,\mathbf{z}^\top\mathbf{A}\mathbf{x}\end{aligned}$$

♦

We can verify:

$$\mathbf{z}^\top\mathbf{x} = \mathbf{x}^\top\mathbf{z}$$

$$\mathbf{z}^\top\mathbf{A}\mathbf{x} = \mathbf{x}^\top\mathbf{A}^\top\mathbf{z}$$

because we have the following:

1. Both sides of each question are symmetric to each other because transposing the left-hand side leads to the right.
2. All of them are actually scalars.

Example 2.1.2 Given an n-dimensional vector, $\mathbf{x} \in \mathbb{R}^n$, compare $\mathbf{x}^\top\mathbf{x}$ with $\mathbf{x}\,\mathbf{x}^\top$.

We can first show that

$$\mathbf{x}^\top\mathbf{x} = \begin{bmatrix}x_1 & x_2 & \cdots & x_n\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix} = \sum_{i=1}^{n} x_i^2.$$

On the other hand, we have

$$\mathbf{x}\,\mathbf{x}^\top = \begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\begin{bmatrix}x_1 & x_2 & \cdots & x_n\end{bmatrix} = \begin{bmatrix}x_1^2 & x_1x_2 & \cdots & x_1x_n\\ x_1x_2 & x_2^2 & \cdots & x_2x_n\\ \vdots & \vdots & \ddots & \vdots\\ x_1x_n & x_2x_n & \cdots & x_n^2\end{bmatrix}.$$

Therefore, $\mathbf{x}\,\mathbf{x}^\top$ is actually an $n \times n$ symmetric matrix. ♦

The trace of a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is defined to be the sum of all elements on the main diagonal of $\mathbf{A}$, denoted as $\mathrm{tr}(\mathbf{A})$; we thus have

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}.$$

We can verify that $\mathbf{x}^\top\mathbf{x} = \mathrm{tr}(\mathbf{x}\,\mathbf{x}^\top)$ in this example.

For any two matrices, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$, we can verify that

$$\begin{aligned}\mathrm{tr}(\mathbf{A}^\top\mathbf{B}) &= \mathrm{tr}(\mathbf{A}\mathbf{B}^\top)\\ &= \mathrm{tr}(\mathbf{B}\mathbf{A}^\top) = \mathrm{tr}(\mathbf{B}^\top\mathbf{A})\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij}.\end{aligned}$$

For any two square matrices (i.e., $\mathbf{X} \in \mathbb{R}^{n\times n}$ and $\mathbf{Y} \in \mathbb{R}^{n\times n}$), we can also verify

$$\mathrm{tr}(\mathbf{X}\mathbf{Y}) = \mathrm{tr}(\mathbf{Y}\mathbf{X}).$$

2.1.4 Eigenvalues and Eigenvectors

Given a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we can find a nonzero vector $\mathbf{u} \in \mathbb{R}^n$ that satisfies

$$\mathbf{A}\,\mathbf{u} = \lambda\,\mathbf{u},$$

where λ is a scalar. We call $\mathbf{u}$ an *eigenvector* of $\mathbf{A}$, and λ is an *eigenvalue* corresponding to $\mathbf{u}$. As we have learned, a square matrix $\mathbf{A}$ can be viewed as a linear transformation that maps any point in a space $\mathbb{R}^n$ into another point in the same space. An eigenvector $\mathbf{u}$ represents a special point in the space whose direction is not changed by this linear transformation. Depending on the corresponding eigenvalue λ, it can be stretched or contracted along the original direction. If the eigenvalue λ is negative, it

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

is flipped into the opposite direction after the mapping. The eigenvalues and eigenvectors are completely determined by matrix $\mathbf{A}$ itself and are considered as an inherent characteristic of matrix $\mathbf{A}$.

Example 2.1.3 Given $\mathbf{A} \in \mathbb{R}^{n\times n}$, assume we can find n orthogonal eigenvectors $\mathbf{u}_i$ $(i = 1, 2, \cdots, n)$ as follows:

$$\mathbf{A}\,\mathbf{u}_i = \lambda_i\,\mathbf{u}_i \quad \text{(assuming } \|\mathbf{u}_i\|^2 = 1),$$

where λ_i is the eigenvalue corresponding to $\mathbf{u}_i$. Show that the matrix $\mathbf{A}$ can be factorized.

Any two vectors, $\mathbf{u}_i$ and $\mathbf{u}_j$, are orthogonal if and only if

$$\mathbf{u}_i \cdot \mathbf{u}_j = 0.$$

First, we align both sides of the equations column by column:

$$\begin{bmatrix} | & | & & | \\ \mathbf{A}\mathbf{u}_1 & \mathbf{A}\mathbf{u}_2 & \cdots & \mathbf{A}\mathbf{u}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \lambda_1\mathbf{u}_1 & \lambda_2\mathbf{u}_2 & \cdots & \lambda_n\mathbf{u}_n \\ | & | & & | \end{bmatrix}.$$

Next, we can move $\mathbf{A}$ out in the left-hand side and arrange the right-hand side into two matrices according to the multiplication rule:

$$\mathbf{A}\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}}_{\mathbf{U}} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\mathbf{\Lambda}},$$

A diagonal matrix has nonzero elements only on the main diagonal.

where the matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ is constructed by using all eigenvectors as its columns, and $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with all eigenvalues aligned on the main diagonal. Because all eigenvectors are normed to 1 and they are orthogonal to each other, we have

$$\mathbf{u}_i^\top \mathbf{u}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

Therefore, we can show that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. This means that $\mathbf{U}^{-1} = \mathbf{U}^\top$. If we multiply the previous equation by this from the right, we finally derive

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^\top.$$

♦

The idea of eigenvalues can be extended to nonsquare matrices, leading to the so-called *singular values*. A nonsquare matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ can be similarly factorized using the *singular value decomposition (SVD)* method. (See Section 7.3 for more.)

A square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is said to be *positive definite* (or *positive semidefinite*) if $\mathbf{x}^\top\mathbf{A}\mathbf{x} > 0$ (or ≥ 0) holds for any $\mathbf{x} \in \mathbb{R}^n$, denoted as $\mathbf{A} > 0$ (or $\mathbf{A} \geq 0$). A symmetric matrix $\mathbf{A}$ is positive definite (or semidefinite) if and only if all of its eigenvalues are positive (or nonnegative).

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

2.1.5 Matrix Calculus

In mathematics, matrix calculus is a specialized notation to conduct multivariate calculus with respect to vectors or matrices. If y is a function involving all elements of a vector $\mathbf{x}$ (or a matrix $\mathbf{A}$), then $\partial y/\partial \mathbf{x}$ (or $\partial y/\partial \mathbf{A}$) is defined as a vector (or a matrix) in the same size as $\mathbf{x}$ (or $\mathbf{A}$), where each element is defined as a partial derivative of y with respect to the corresponding element in $\mathbf{x}$ (or $\mathbf{A}$).

Assuming we are given

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},$$

then we have

$$\frac{\partial y}{\partial \mathbf{x}} \triangleq \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} \quad \text{and} \quad \frac{\partial y}{\partial \mathbf{A}} \triangleq \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \frac{\partial y}{\partial a_{12}} & \cdots & \frac{\partial y}{\partial a_{1n}} \\ \frac{\partial y}{\partial a_{21}} & \frac{\partial y}{\partial a_{22}} & \cdots & \frac{\partial y}{\partial a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial a_{m1}} & \frac{\partial y}{\partial a_{m2}} & \cdots & \frac{\partial y}{\partial a_{mn}} \end{bmatrix}.$$

Example 2.1.4 Given $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$, show the following identities:

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\top \mathbf{A}\mathbf{x}\right) = \mathbf{A}\mathbf{x} + \mathbf{A}^\top \mathbf{x} \qquad \frac{\partial}{\partial \mathbf{A}}\left(\mathbf{x}^\top \mathbf{A}\mathbf{x}\right) = \mathbf{x}\mathbf{x}^\top.$$

Let's denote $y = \mathbf{x}^\top \mathbf{A}\mathbf{x}$; we thus have

$$y = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i a_{ij} x_j.$$

For any $t \in \{1, 2, \cdots, n\}$, we have

$$\frac{\partial y}{\partial x_t} = \underbrace{\sum_{j=1}^{n} a_{tj} x_j}_{\text{when } i=t} + \underbrace{\sum_{i=1}^{n} x_i a_{it}}_{\text{when } j=t}.$$

If we denote $\mathbf{A}\mathbf{x} + \mathbf{A}^\top \mathbf{x}$ as a column vector:

$$\mathbf{A}\mathbf{x} + \mathbf{A}^\top \mathbf{x} = \begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix}^\top,$$

$$\mathbf{A}\mathbf{x} + \mathbf{A}^\top \mathbf{x} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

then for any $t \in \{1, 2, \cdots, n\}$, we can compute

$$z_t = \sum_{j=1}^{n} a_{tj} x_j + \sum_{i=1}^{n} x_i a_{it}.$$

Therefore, we have proved that $(\partial/\partial \mathbf{x})(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{A}^\intercal \mathbf{x}$.

Similarly, we can compute

$$\frac{\partial y}{\partial a_{ij}} = x_i x_j \qquad (\forall i, j \in \{1, 2, \cdots, n\}).$$

Then we have

$$\frac{\partial y}{\partial \mathbf{A}} = \begin{bmatrix} x_1^2 & x_1 x_2 & \cdots & x_1 x_n \\ x_1 x_2 & x_2^2 & \cdots & x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1 x_n & x_2 x_n & \cdots & x_n^2 \end{bmatrix}.$$

As shown in Example 2.1.2, this matrix equals to $\mathbf{x}\mathbf{x}^\intercal$. Therefore, we have shown that $(\partial/\partial \mathbf{A})(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}) = \mathbf{x}\mathbf{x}^\intercal$ holds. ♦

The following box lists all matrix calculus identities that will be used in the remainder of this book. Readers are encouraged to examine them for future reference.

Matrix Calculus Identities for Machine Learning

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{x}\right) = 2\mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{y}\right) = \mathbf{y}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}\right) = \mathbf{A}\mathbf{x} + \mathbf{A}^\intercal \mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}\right) = 2\mathbf{A}\mathbf{x} \quad \text{(symmetric } \mathbf{A}\text{)}$$

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathbf{x}^\intercal \mathbf{A}\mathbf{y}\right) = \mathbf{x}\mathbf{y}^\intercal$$

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathbf{x}^\intercal \mathbf{A}^{-1}\mathbf{y}\right) = -(\mathbf{A}^\intercal)^{-1}\mathbf{x}\mathbf{y}^\intercal(\mathbf{A}^\intercal)^{-1} \quad \text{(square } \mathbf{A}\text{)}$$

$$\frac{\partial}{\partial \mathbf{A}}\left(\ln|\mathbf{A}|\right) = (\mathbf{A}^{-1})^\intercal = (\mathbf{A}^\intercal)^{-1} \quad \text{(square } \mathbf{A}\text{)}$$

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathrm{tr}(\mathbf{A})\right) = \mathbf{I} \quad \text{(square } \mathbf{A}\text{)}$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

2.2 Probability and Statistics

Probability theory is a mathematical tool to deal with uncertainty. Probability is a real number between 0 and 1, assigned to indicate how likely an event is to occur in an experiment. The higher the probability of an event, the more likely it is the event will occur. A sample space is defined as a collection of all possible outcomes in an experiment. Any subset of the sample space can be viewed as an event. The probability of a null event is 0, and that of the full sample space is 1. For example, in an experiment of tossing a fair six-faced dice only once, the sample space includes six outcomes in total: $\{1, 2, \cdots, 6\}$. We can define many events for this experiment, such as A = "observing an even number," B = "observing the digit 6," C = "observing a natural number," and D = "observing a negative number". We can easily calculate the probabilities for these events as $\Pr(A) = 1/2$, $\Pr(B) = 1/6$, $\Pr(C) = 1$, and $\Pr(D) = 0$.

2.2.1 Random Variables and Distributions

Random variables are a formal tool to study a random phenomenon in mathematics. A *random variable* is defined as a variable whose values depend on the outcomes of a random experiment. In other words, a random variable could take different values in different probabilities based on the experimental outcomes. Depending on all possible values a random variable can take, there are two types of random variables: discrete and continuous. A *discrete random variable* can take on only a finite set of distinct values. For example, if we define a random variable X to indicate the digit observed in the previous dice-tossing experiment, X is a discrete random variable that can take only six different values. On the other hand, a *continuous random variable* may take an infinite number of possible values. For example, if we define another random variable Y to indicate a temperature measurement by a thermometer, Y is continuous because it may take any real number.

If we want to fully specify a random variable, we have to state two ingredients for it: (i) its *domain*, the set of all possible values that the random variable can take, and (ii) its *probability distribution*, how likely it is that the random variable may take each possible value. In probability theory, these two ingredients are often characterized by a probability function.

For any discrete random variable X, we can specify these two ingredients with the so-called *probability mass function (p.m.f.)*, which is defined on the domain of X (i.e., $\{x_1, x_2, \cdots\}$) as follows:

$$p(x) = \Pr(X = x) \quad \text{for all} \quad x \in \{x_1, x_2, \cdots\}.$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

x	x_1	x_2	x_3	x_4
$p(x)$	0.4	0.3	0.2	0.1

A p.m.f. of a random variable X that takes four distinct values (i.e. $\{x_1, x_2, x_3, x_4\}$) in the probabilities specified in the table.

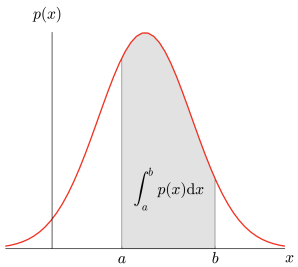


Figure 2.4: An illustration of a simple probability density function (p.d.f.) of a continuous random variable taking values in $(-\infty, +\infty)$.

By definition, we have

$$\begin{aligned} p(x) &= \lim_{\Delta x \to 0} \frac{\Pr\left(x \le X \le x + \Delta x\right)}{\Delta x} \\ &= \frac{\text{probability}}{\text{interval}} \\ &= \text{probability density}. \end{aligned}$$

In addition to p.d.f., we can also define another probability function for any continuous random variable X as

$$F(x) = \Pr\left(X \le x\right) \quad (\forall x),$$

which is often called the *cumulative distribution function (c.d.f.)*. By definition, we have

$$\lim_{x \to -\infty} F(x) = 0 \quad \text{and} \quad \lim_{x \to +\infty} F(x) = 1,$$

and

$$F(x) = \int_{-\infty}^{x} p(x)\, dx$$

$$p(x) = \frac{d}{dx} F(x).$$

If we sum $p(x)$ over all values in the domain, it satisfies the sum-to-1 constraint:

$$\sum_x p(x) = 1. \tag{2.1}$$

A p.m.f. can be conveniently represented in a table. The table shown in the left margin represents a simple p.m.f. of a random variable that can take four distinct values.

For any continuous random variable, we cannot define its probability of taking a single value. This probability will end up with 0s for all values because a continuous random variable can take an infinite number of different values. In probability theory, we instead consider the probability for a continuous random variable to fall within any interval. For example, given a continuous random variable X and any interval $[a, b]$ inside its domain, we try to measure the probability $\Pr(a \le X \le b)$, which is often nonzero. As shown in Figure 2.4, we define a function $p(x)$ in such a way that this probability equals to the area of the shaded region under the function $p(x)$ between a and b. In other words, we have

$$\Pr(a \le X \le b) = \int_a^b p(x)\, dx,$$

which holds for any interval $[a, b]$ inside the domain of the random variable. We usually call $p(x)$ the *probability density function (p.d.f.)* of X (see the margin note for an explanation). If we choose the entire domain as the interval, by definition, the probability must be 1. Therefore, we have the sum-to-1 constraint

$$\int_{-\infty}^{+\infty} p(x)\, dx = 1, \tag{2.2}$$

which holds for any probability density function.

2.2.2 Expectation: Mean, Variance, and Moments

As we know, a random variable is fully specified by its probability function. In other words, the probability function gives the full knowledge on the random variable, and we are able to compute any statistics of it from the probability function. Here, let us look at how to compute some important statistics for random variables from a p.d.f. or p.m.f. Thereafter, we will use $p(x)$ to represent the p.m.f. for a discrete random variable and the p.d.f. for a continuous random variable.

Given a continuous random variable X, for any function $f(X)$ of the

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

random variable, we can define the *expectation* of $f(X)$ as follows:

$$\mathbb{E}\big[f(X)\big] = \int_{-\infty}^{+\infty} f(x)\, p(x)\, dx.$$

If X is a discrete random variable, we replace the integral with summation as follows:

$$\mathbb{E}\big[f(X)\big] = \sum_{x} f(x)\, p(x).$$

Because X is a random variable, the function $f(X)$ also yields different values in different probabilities. The expectation $\mathbb{E}\big[f(X)\big]$ gives the average of all possible values of $f(X)$. Relying on the expectation, we may define some statistics for random variables. For example, the *mean* of a random variable is defined as the expectation of the random variable itself (i.e., $\mathbb{E}\big[X\big]$). The rth moment of a random variable is defined as the expectation of its rth power (i.e., $\mathbb{E}\big[X^r\big]$; for any $r \in \mathbb{N}$). The *variance* of a random variable is defined as follows:

$$\mathrm{var}(X) = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big].$$

Intuitively speaking, the mean of a random variable indicates the center of its distribution, and the variance tells how much it may deviate from the center on average.

Example 2.2.1 For any random variable X, show that

$$\mathrm{var}(X) = \mathbb{E}\big[X^2\big] - (\mathbb{E}[X])^2.$$

$$\begin{aligned}
\mathrm{var}(X) &= \mathbb{E}\big[(X - \mathbb{E}[X])^2\big] = \mathbb{E}\Big[X^2 - 2 \cdot X \cdot \mathbb{E}[X] + (\mathbb{E}[X])^2\Big] \\
&= \mathbb{E}\big[X^2\big] - 2\mathbb{E}[X] \cdot \mathbb{E}[X] + (\mathbb{E}[X])^2 \\
&= \mathbb{E}\big[X^2\big] - (\mathbb{E}[X])^2.
\end{aligned}$$

♦

For any constant c irrelevant of X, it is easy to show

$$\mathbb{E}[c] = c$$

$$\mathbb{E}[c \cdot X] = c \cdot \mathbb{E}[X].$$

$\mathbb{E}[X]$ can be viewed as a constant because it is a fixed value for any random variable X.

Next, let us revisit the general principle of the *bias–variance trade-off* discussed in the previous chapter. In any machine learning problem, we basically need to estimate a model from some training data. The true model is usually unknown but fixed, denoted as f. Hence, we can treat the true model f as an unknown constant. Imagine that we can repeat the model estimation many times. At each time, we randomly collect some training data and run the same learning algorithm to derive an estimate, denoted as $\hat{f}$. The estimate $\hat{f}$ can be viewed as a random variable because we may derive a different estimate each time depending on the training data used, which differ from one collection to another. Generally speaking, we are interested in the average learning error between an estimate $\hat{f}$ and

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

the true model f:

$$\text{error} = \mathbb{E}\left[(\hat{f} - f)^2\right].$$

The bias of a learning method is defined as the difference between the true model and the mean of all possible estimates derived from this method:

$$\text{bias} = \left|f - \mathbb{E}\left[\hat{f}\right]\right|.$$

The variance of an estimate is as defined previously:

$$\text{variance} = \text{var}(\hat{f}) = \mathbb{E}\left[(\hat{f} - \mathbb{E}[\hat{f}])^2\right].$$

Example 2.2.2 The Bias-Variance Trade-Off

Show that the bias and variance decomposition holds as follows:

$$\text{error} = \text{bias}^2 + \text{variance}.$$

$$\begin{aligned}
\text{error} &= \mathbb{E}\left[(f - \hat{f})^2\right] = \mathbb{E}\left[(f - \mathbb{E}[\hat{f}] - \hat{f} + \mathbb{E}[\hat{f}])^2\right] \\
&= \mathbb{E}\left[(f - \mathbb{E}[\hat{f}])^2\right] + \mathbb{E}\left[(\hat{f} - \mathbb{E}[\hat{f}])^2\right] \\
&\quad -2 \cdot \mathbb{E}\left[(f - \mathbb{E}[\hat{f}])(\hat{f} - \mathbb{E}[\hat{f}])\right] \\
&= \underbrace{(f - \mathbb{E}[\hat{f}])^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[(\hat{f} - \mathbb{E}[\hat{f}])^2\right]}_{\text{variance}}.
\end{aligned}$$

♦

Because both f and $\mathbb{E}[\hat{f}]$ are constants, we have

$$\begin{aligned}
& \mathbb{E}\left[(f - \mathbb{E}[\hat{f}])(\hat{f} - \mathbb{E}[\hat{f}])\right] \\
= \; & (f - \mathbb{E}[\hat{f}])\mathbb{E}\left[\hat{f} - \mathbb{E}[\hat{f}]\right] \\
= \; & (f - \mathbb{E}[\hat{f}])(\mathbb{E}[\hat{f}] - \mathbb{E}[\hat{f}]) \quad (= 0) \\
= \; & 0.
\end{aligned}$$

2.2.3 Joint, Marginal, and Conditional Distributions

We have discussed the probability functions for a single random variable. If we need to consider multiple random variables at the same time, we can similarly define some probability functions for them in the product space of their separate domains.

If we have multiple discrete random variables, a multivariate function can be defined in the product space of their domains as follows:

$$p(x, y) = \Pr(X = x, Y = y) \quad \forall x \in \{x_1, x_2, \cdots\}, y \in \{y_1, y_2, \cdots\},$$

where $p(x, y)$ is often called the *joint distribution* of two random variables X and Y. The joint distributions of discrete random variables can also be represented with some multidimensional tables. For example, a joint distribution $p(x, y)$ of two discrete random variables, X and Y, is shown in the left margin, where each entry indicates the probability for X and

Assume the domains of two random variables, X and Y, are

$$X \in \{x_1, x_2\} \quad \text{and} \quad Y \in \{y_1, y_2\}.$$

The product space of X and Y includes all pairs like the following:

$$\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}.$$

$y \setminus x$	x_1	x_2	x_3
y_1	0.03	0.24	0.17
y_2	0.23	0.11	0.22

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Y to take the corresponding values. If we sum over all entries in a joint distribution, it must satisfy the sum-to-1 constraint $\sum_x \sum_y p(x, y) = 1$.

For multiple continuous random variables, we can follow the same idea of the p.d.f. to define a joint distribution, as in Figure 2.5, to ensure that the probability for them to fall into any region Ω in their product space can be computed by the following multiple integral:

$$\Pr\Big((x, y) \in \Omega\Big) = \int \cdots \int_{\Omega} p(x, y)\, dxdy.$$

Similarly, if we integrate the joint distribution over the entire space, it satisfies the sum-to-1 constraint: $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x, y)\, dxdy = 1$.

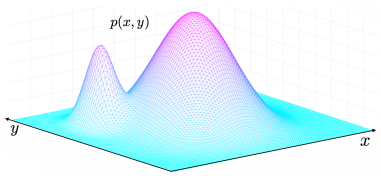


Figure 2.5: An illustration of a joint distribution (p.d.f.) of two continuous random variables $p(x, y)$.

A joint distribution fully specifies all underlying random variables. From a joint distribution, we should be able to derive any information regarding each underlying random variable. From a joint distribution of multiple random variables, we can derive the distribution function of any subset of these random variables by an operation called *marginalization*. The derived distribution of a subset is often called a *marginal distribution*. A marginal distribution is derived by marginalizing all irrelevant random variables, namely, integrating out each continuous random variable or summing out each discrete random variable. For example, given a joint distribution of two random variables $p(x, y)$, we can derive the marginal distribution of one random variable by marginalizing out the other one:

$$p(x) = \int_{-\infty}^{+\infty} p(x, y)dy,$$

if y is a continuous random variable, or

$$p(x) = \sum_{y} p(x, y),$$

if y is a discrete random variable. This marginalization can be applied to any joint distribution to derive a marginal distribution of any subset of random variables that we are interested in. This marginalization is often called the *rule of sum* in probability.

Moreover, we can further define the so-called *conditional distributions* among multiple random variables. For example, the conditional distribution of x given y is defined as follows:

$$p(x \mid y) \triangleq \frac{p(x, y)}{p(y)} = \frac{p(x, y)}{\int p(x, y)\, dx}.$$

The conditional distribution $p(x \mid y)$ is a function of x, and it only describes how x is distributed when y is known to take a particular value. Using a conditional distribution, we can compute the conditional expectation of

If x is a discrete random variable, we have

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

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

$f(X)$ when Y is given as $Y = y_0$:

$$\mathbb{E}_X\Big[f(X)\,\Big|\,Y = y_0\Big] = \int_{-\infty}^{+\infty} f(x)\cdot p(x\mid y_0)\,dx.$$

If X is discrete, we have

$$\mathbb{E}_X\Big[f(X)\,\Big|\,Y = y_0\Big] = \sum_x f(x)\cdot p(x\mid y_0).$$

Example 2.2.3 Assuming the joint distribution of two continuous random variables, X and Y, is given as $p(x, y)$, compare the regular mean of X (i.e., $\mathbb{E}[X]$) and a conditional mean of X when $Y = y_0$ (i.e., $\mathbb{E}_X\left[X\,\middle|\,Y = y_0\right]$).

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} x\cdot p(x)\,dx = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x\cdot p(x,y)\,dxdy$$

$$\begin{aligned}\mathbb{E}_X\left[X\,\middle|\,Y = y_0\right] &= \int_{-\infty}^{+\infty} x\cdot p(x\mid y_0)\,dx = \int_{-\infty}^{+\infty} x\cdot\frac{p(x,y_0)}{p(y_0)}\,dx\\ &= \frac{\int_{-\infty}^{+\infty} x\cdot p(x,y_0)\,dx}{\int_{-\infty}^{+\infty} p(x,y_0)\,dx}.\end{aligned}$$

From this, we can see that both means can be computed from the joint distribution, but they are two different quantities. ♦

Two random variables, X and Y, are said to be *independent* if and only if their joint distribution $p(x, y)$ can be factorized as a product of their own marginal distributions:

$$p(x,y) = p(x)\,p(y) \quad (\forall x, y).$$

From the previous definition of conditional distributions, we can see that X and Y are independent if and only if $p(x|y) = p(x)$ holds for all y.

For any two random variables, X and Y, we can define the *covariance* between them as

$$\begin{aligned}\operatorname{cov}(X,Y) &= \mathbb{E}\Big[\big(X-\mathbb{E}[X]\big)\big(Y-\mathbb{E}[Y]\big)\Big]\\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\big(x-\mathbb{E}[X]\big)\big(y-\mathbb{E}[Y]\big)\,p(x,y)\,dxdy.\end{aligned}$$

If X and Y are discrete,

$$\operatorname{cov}(X,Y) = \sum_x\sum_y\big(x-\mathbb{E}[X]\big)\big(y-\mathbb{E}[Y]\big)\,p(x,y).$$

If $\operatorname{cov}(X,Y) = 0$, we say that two random variables, X and Y, are *uncorrelated*. Note that *uncorrelatedness* is a much weaker condition than independence. If two random variables are independent, we can show from the previous definition that they must be uncorrelated. However, it is generally not true the other way around.

Relying on the concept of the conditional distribution, we can factorize any joint distribution involving many random variables by following a

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

particular order of these variables. For example, we have

$$p(x_1, x_2, x_3, x_4, \cdots) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_1, x_2)\, p(x_4|x_1, x_2, x_3) \cdots$$

For example, we can also do the following:

$$p(x_1, x_2, x_3, x_4, \cdots) = p(x_3)\, p(x_1|x_3)$$
$$p(x_4|x_1, x_3)\, p(x_2|x_1, x_3, x_4) \cdots$$

Note that there exist many different ways to correctly factorize any joint distribution as long as the probability of each variable is conditioned on all previous variables prior to the current one in the order. In probability theory, this factorization rule is often called the *multiplication rule* of probability, also known as the *general product rule* of probability.

When we have a joint distribution of a large number of random variables, for notational convenience, we often group some related random variables into random vectors so that we represent it as a joint distribution of random vectors:

$$p\big(\underbrace{x_1, x_2, x_3}_{\mathbf{x}}, \underbrace{y_1, y_2, y_3, y_4}_{\mathbf{y}}\big) = p(\mathbf{x}, \mathbf{y}).$$

We can use the same rule as previously to similarly derive the marginal and conditional distributions for random vectors, as follows:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y})\, d\mathbf{y}$$

$$p(\mathbf{x} \mid \mathbf{y}) \triangleq \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}.$$

If $\mathbf{y}$ is discrete, we have

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

The mean of a random vector $\mathbf{x}$ is a vector, denoted as $\mathbb{E}[\mathbf{x}]$:

$$\mathbb{E}\big[\mathbf{x}\big] = \int \mathbf{x}\, p(\mathbf{x})\, d\mathbf{x} = \int\int \mathbf{x}\, p(\mathbf{x}, \mathbf{y})\, d\mathbf{x} d\mathbf{y}.$$

If $\mathbf{x}$ and $\mathbf{y}$ are both discrete, we have

$$\mathbb{E}\big[\mathbf{x}\big] = \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbf{x}\, p(\mathbf{x}, \mathbf{y}).$$

The covariance between two random vectors, $\mathbf{x}$ and $\mathbf{y}$, becomes a matrix, which is often called the *covariance matrix*:

$$\begin{aligned} \operatorname{cov}(\mathbf{x}, \mathbf{y}) &= \mathbb{E}\Big[(\mathbf{x} - \mathbb{E}[\mathbf{x}])\,(\mathbf{y} - \mathbb{E}[\mathbf{y}])^{\mathsf{T}}\Big] \\ &= \int\int (\mathbf{x} - \mathbb{E}[\mathbf{x}])\,(\mathbf{y} - \mathbb{E}[\mathbf{y}])^{\mathsf{T}}\, p(\mathbf{x}, \mathbf{y})\, d\mathbf{x} d\mathbf{y}. \end{aligned}$$

$$\operatorname{cov}(\mathbf{x}, \mathbf{y}) =$$
$$\sum_{\mathbf{x}} \sum_{\mathbf{y}} (\mathbf{x} - \mathbb{E}[\mathbf{x}])\,(\mathbf{y} - \mathbb{E}[\mathbf{y}])^{\mathsf{T}}\, p(\mathbf{x}, \mathbf{y}).$$

Finally, the general product rule of probability can be equally applied to factorize a joint distribution of random vectors as well.

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})\, p(\mathbf{y}|\mathbf{x})\, p(\mathbf{z}|\mathbf{x}, \mathbf{y}).$$

2.2.4 Common Probability Distributions

Here, let us review some popular probability functions often used to represent the distributions of random variables. For each of these probability

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

functions, we need to know not only its functional form but also what physical phenomena it can be used to describe. Moreover, we need to clearly distinguish parameters from random variables in the mathematical formula and correctly identify the domain of the underlying random variables (a.k.a. the *support* of the distribution), as well as the valid range of the parameters.

Binomial Distribution

The binomial distribution is the discrete probability distribution of the number of outcomes in a sequence of N independent binary experiments. Each binary experiment has two different outcomes. We use the binomial distribution to compute the probabilities of observing r ($r \in \{0, 1, \cdots, N\}$) times of one particular outcome from all N experiments—for example, the probability of seeing r heads when a coin is tossed N times in a row. When we use a discrete random variable X to represent the number of an outcome, and assuming the probability of observing this outcome is $p \in [0, 1]$ in one experiment, the binomial distribution takes the following formula:

$$\mathrm{B}(r \,|\, N, p) \triangleq \Pr(X = r) = \frac{N!}{r!\,(N-r)!}\, p^r (1-p)^{N-r},$$

where N and p denote two parameters of the distribution.

We summarize some key properties for the binomial distribution as follows:

- Parameters: $N \in \mathbb{N}$ and $p \in [0, 1]$.
- Support: The domain of the random variable is $r \in \{0, 1, \cdots N\}$.
- Mean and variance: $\mathbb{E}[X] = Np$ and $\mathrm{var}(X) = Np(1-p)$.
- The sum-to-1 constraint: $\sum_{r=0}^{N} \mathrm{B}(r \,|\, N, p) = 1$.

Figure 2.6 shows an example of the binomial distribution for $p = 0.7$ and $N = 20$.

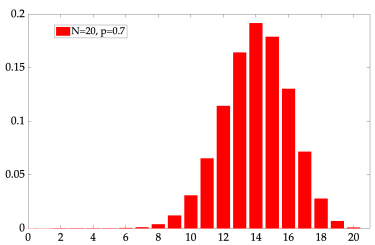


Figure 2.6: An illustration the binomial distribution $\mathrm{B}(r \,|\, N, p)$ with $p = 0.7$ and $N = 20$.

When only one binary experiment is done ($N = 1$),

$$\mathrm{B}(r \,|\, N = 1, p) = p^r (1-p)^{1-r}$$

is also called the Bernoulli distribution, where $r \in \{0, 1\}$.

Multinomial Distribution

The multinomial distribution can be viewed as an extension of the binomial distribution when each experiment is not binary but has m distinct outcomes. In each experiment, the probabilities of observing all possible outcomes are denoted as $\{p_1, p_2, \cdots, p_m\}$, where we have the sum-to-1 constraint $\sum_{i=1}^{m} p_i = 1$. When we independently repeat the experiment N times, we introduce m different random variables to represent the number of each outcome from all N experiments (i.e., $\{X_1, X_2, \cdots, X_m\}$). The joint

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

distribution of these m random variables is the multinomial distribution, computed as follows:

$$\begin{aligned} & \text{Mult}(r_1, r_2, \cdots, r_m \,|\, N, p_1, p_2, \cdots, p_m) \\ \triangleq\ & \Pr(X_1 = r_1, X_2 = r_2, \cdots, X_m = r_m) \\ =\ & \frac{N!}{r_1!\, r_2! \cdots r_m!}\, p_1^{r_1}\, p_2^{r_2} \cdots p_m^{r_m}, \end{aligned}$$

where $\sum_{i=1}^{m} r_i = N$ holds because N experiments are conducted in total.

We summarize some properties for the multinomial distribution as follows:

- Parameters: $N \in \mathbb{N}; 0 \le p_i \le 1\ (\forall i = 1, 2, \cdots, m)$ and $\sum_{i=1}^{m} p_i = 1$.
- Support (the domain of m random variables): $r_i \in \{0, 1, \cdots N\}\ (\forall i = 1, \cdots, m)$ and $\sum_{i=1}^{m} r_i = N$.
- Means, variances, and covariances:

$$\mathbb{E}\big[X_i\big] = N p_i \ \text{ and } \ \text{var}(X_i) = N p_i (1 - p_i) \qquad (\forall i)$$

$$\text{cov}(X_i, X_j) = -N p_i p_j \qquad (\forall i, j).$$

- The sum-to-1 constraint:

$$\sum_{r_1 \cdots r_m} \text{Mult}(r_1, r_2, \cdots, r_m \,|\, N, p_1, p_2, \cdots, p_m) = 1.$$

When we conduct only one experiment ($N = 1$),

$$\begin{aligned} & \text{Mult}(r_1, \cdots, r_m \,|\, N = 1, p_1, \cdots, p_m) \\ & = p_1^{r_1} p_2^{r_2} \cdots p_m^{r_m} \end{aligned}$$

is also called the categorical distribution, where we have $r_i \in \{0, 1\}\ (\forall i)$ and

$$\sum_{i=1}^{m} r_i = 1.$$

As we will learn, the multinomial distribution is the main building block for constructing any statistical model for discrete random variables in machine learning.

Beta Distribution

The beta distribution is used to describe a continuous random variable, X, that takes a probability-like value $x \in \mathbb{R}$ and $0 \le x \le 1$.The beta distribution takes the following functional form:

$$\text{Beta}(x \,|\, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha - 1} (1 - x)^{\beta - 1},$$

where $\Gamma(\cdot)$ denotes the gamma function, and α and β are two positive parameters of the beta distribution. Similarly, we can summarize some key properties for the beta distribution as follows:

- Parameters: $\alpha > 0$ and $\beta > 0$.
- Support (the domain of the continuous random variable):

$$x \in \mathbb{R} \quad \text{and} \quad 0 \le x \le 1.$$

The gamma function is defined as follows:

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}\, e^{-t}\, dt \quad (\forall x > 0).$$

$\Gamma(x)$ is often considered as a generalization of the factorial to noninteger numbers because of the following property:

$$\Gamma(x + 1) = x\, \Gamma(x)$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

- Mean and variance:

$$\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta} \qquad \text{var}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

- The sum-to-1 constraint:

$$\int_0^1 \text{Beta}(x \mid \alpha, \beta)\, dx = 1.$$

We can recognize that the beta distribution shares the same functional form as the binomial distribution. They differ only in terms of swapping the roles of the parameters and random variables. Therefore, these two distributions are said to be *conjugate* to each other. In this sense, the beta distribution can be viewed as a distribution of the parameter p in the binomial distribution. As we will learn, this viewpoint plays an important role in Bayesian learning (refer to Chapter 14).

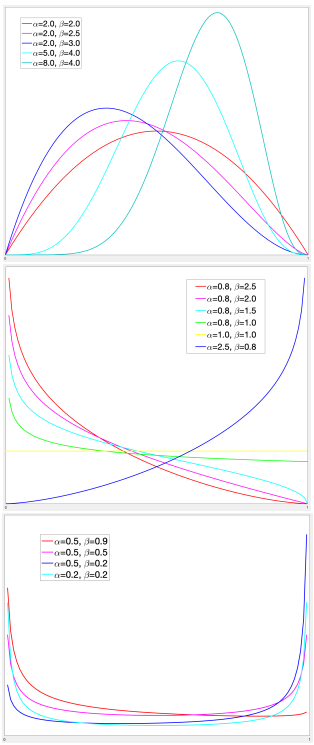


Figure 2.7: An illustration of some beta distributions when two parameters α and β take different values: (i) $\alpha > 1$ and $\beta > 1$; (ii) $0 < \alpha < 1 < \beta$ or $0 < \beta < 1 < \alpha$; (iii) $0 < \alpha, \beta < 1$.

Depending on the choices of the two parameters α and β, the beta distribution behaves quite differently. As shown in Figure 2.7, when both parameters are larger than 1, the beta distribution is a unimodal bell-shaped distribution between 0 and 1. The mode of the distribution can be computed as $(\alpha-1)/(\alpha+\beta-2)$ in this case. It becomes a monotonic distribution when one parameter is larger than 1 and the other is smaller than 1, particularly monotonically decaying if $0 < \alpha < 1 < \beta$ and monotonically increasing if $0 < \beta < 1 < \alpha$. At last, if both parameters are smaller than 1, the beta distribution is bimodal between 0 and 1, peaking at the two ends.

Dirichlet Distribution

The Dirichlet distribution is a multivariate generalization of the beta distribution that is used to describe multiple continuous random variables $\{X_1, X_2, \cdots, X_m\}$, taking values on the probabilities of observing a complete set of mutually exclusive events. As a result, the values of these random variables are always summed to 1 because these events are complete. For example, if we use some biased dice in a tossing experiment, we can define six random variables, each of which represents the probability of observing each digit when tossing a die. For each biased die, these six random variables take different probabilities, but they always sum to 1 for each die. These six random variables from all biased dice can be assumed to follow the Dirichlet distribution.

In general, the Dirichlet distribution takes the following functional form:

$$\text{Dir}(p_1, p_2, \cdots, p_m \,|\, r_1, r_2, \cdots, r_m)$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

$$= \frac{\Gamma(r_1 + \cdots + r_m)}{\Gamma(r_1) \cdots \Gamma(r_m)} \, p_1^{r_1 - 1} p_2^{r_2 - 1} \cdots p_m^{r_m - 1},$$

where $\{r_1, r_2, \cdots, r_m\}$ denote m positive parameters of the distribution. We can similarly summarize some key properties for the Dirichlet distribution as follows:

- Parameters: $r_i > 0$ $(\forall i = 1, \cdots, m)$.
- Support: The domain of m random variables is an m-dimensional simplex that can be represented as

$$0 < p_i < 1 \;\; (\forall i = 1, \cdots, m) \quad \text{and} \quad \sum_{i=1}^{m} p_i = 1.$$

  For example, Figure 2.8 shows a three-dimensional simplex for the Dirichlet distribution of three random variables $\{p_1, p_2, p_3\}$ when $m = 3$.
- Means, variances, and covariances:

$$\mathbb{E}[X_i] = \frac{r_i}{r_0} \qquad \text{var}(X_i) = \frac{r_i(r_0 - r_i)}{r_0^2(r_0 + 1)}$$

$$\text{cov}(X_i, X_j) = -\frac{r_i r_j}{r_0^2(r_0 + 1)},$$

  where we denote $r_0 = \sum_{i=1}^{m} r_i$.
- The sum-to-1 constraint holds inside the simplex:

$$\int \cdots \int_{p_1 \cdots p_m} \text{Dir}(p_1, p_2, \cdots p_m \,|\, r_1, r_2, \cdots, r_m) \, dp_1 \cdots dp_m = 1.$$

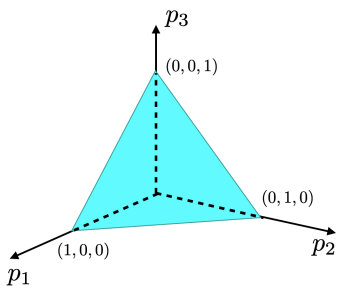


Figure 2.8: An illustration of the three-dimensional simplex of the Dirichlet distribution of three random variables.

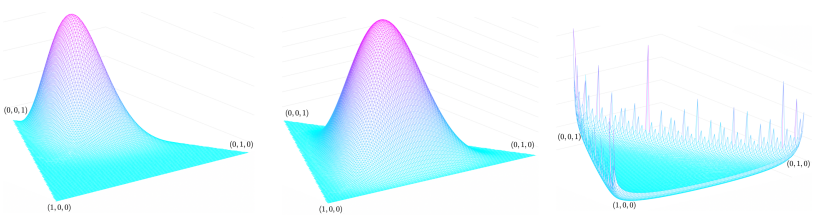


Figure 2.9: An illustration of three Dirichlet distributions in the three-dimensional simplex with various choices of parameters:
1. Regular: $r_1 = 2.0, r_2 = 4.0, r_3 = 10.0$
2. Symmetric: $r_1 = r_2 = r_3 = 4.0$
3. Sparse: $r_1 = 0.7, r_2 = 0.8, r_3 = 0.9$

The shape of a Dirichlet distribution also heavily depends on the choice of its parameters. Figure 2.9 plots the Dirichlet distribution in the triangle simplex for three typical choices of its parameters. Generally speaking, if we choose all parameters to be larger than 1, the Dirichlet distribution is a unimodal distribution centering somewhere in the simplex. In this case, the mode of the distribution is located at $[\hat{p}_1 \;\; \hat{p}_2 \;\; \cdots \;\; \hat{p}_m]^\top$, where $\hat{p}_i = (r_i - 1)/(r_0 - m)$ for all $i = 1, 2, \cdots, m$. If we force all parameters to be the same value, it leads to a symmetric distribution centering at the center of the simplex. On the other hand, if we choose all parameters to be smaller than 1, it results in a distribution that yields a large probability

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

mass only near the vertices and edges of the simplex. It is easy to verify that the vertices or edges correspond to the cases where some random variables p_i take 0 values. In other words, this choice of parameters favors sparse choices of random variables, leading to the so-called *sparse* Dirichlet distribution.

Moreover, we can also identify that the Dirichlet distribution shares the same functional form as the multinomial distribution. Therefore, these two distributions are also conjugate to each other. Similarly, the Dirichlet distribution can be viewed as a distribution of all parameters of a multinomial distribution. Because the multinomial distribution is the main building block for any statistical model of discrete random variables, the Dirichlet distribution is often said to be a distribution of all distributions of discrete random variables. Similar to the beta distribution, the Dirichlet distribution also plays an important role in Bayesian learning for multinomial-related models (see Chapter 14).

Gaussian Distribution

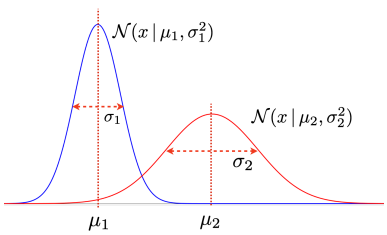


Figure 2.10: An illustration of two univariate Gaussian distributions with various parameters ($\sigma_2 > \sigma_1$).

The univariate Gaussian distribution (a.k.a. the *normal distribution*) is often used to describe a continuous random variable X that can take any real value in $\mathbb{R}$. The general form of a Gaussian distribution is

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

where μ and σ^2 are two parameters. We can summarize some key properties for the univariate Gaussian distribution as follows:

- Parameters: $\mu \in \mathbb{R}$ and $\sigma^2 > 0$.
- Support: The domain of the random variable is $x \in \mathbb{R}$.
- Mean and variance:

$$\mathbb{E}[X] = \mu \quad \text{and} \quad \text{var}(X) = \sigma^2.$$

- The sum-to-1 constraint:

$$\int_{-\infty}^{+\infty} \mathcal{N}(x \mid \mu, \sigma)\, dx = 1.$$

The Gaussian distribution is the well-known unimodal bell-shaped curve. As shown in Figure 2.10, the first parameter μ equals to the mean, indicating the center of the distribution, whereas the second parameters σ equals to the standard deviation, indicating the spread of the distribution.

Several important identities related to the univariate Gaussian distribution are as follows:

$$\int_{-\infty}^{+\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = \sqrt{2\pi\sigma^2},$$

$$\int_{-\infty}^{+\infty} x\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = \mu\sqrt{2\pi\sigma^2},$$

$$\int_{-\infty}^{+\infty} x^2\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx = (\sigma^2 + \mu^2)\sqrt{2\pi\sigma^2}.$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Multivariate Gaussian Distribution

The multivariate Gaussian distribution extends the univariate Gaussian distribution to represent a joint distribution of multiple continuous random variables $\{X_1, X_2, \cdots, X_n\}$, each of which can take any real value in $\mathbb{R}$. If we arrange these random variables as an n-dimensional random vector, the multivariate Gaussian distribution takes the following compact form:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}{2}},$$

where the vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and the symmetric matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ denote two parameters of the distribution. Note that the exponent in the multivariate Gaussian distribution is computed as follows:

$$\big[(\mathbf{x}-\boldsymbol{\mu})^\top\big]_{1\times d} \left[\begin{array}{c} \\ \boldsymbol{\Sigma}^{-1} \\ \\ \end{array}\right]_{d\times d} \left[\begin{array}{c} \\ \mathbf{x}-\boldsymbol{\mu} \\ \\ \end{array}\right]_{d\times 1} = \big[\,\cdot\,\big]_{1\times 1}.$$

We can summarize some key properties for the multivariate Gaussian distribution as follows:

- Parameters: $\boldsymbol{\mu} \in \mathbb{R}^n$; $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n} \succ 0$ is symmetric, positive definite, and invertible.
- Support: The domain of all random variables: $\mathbf{x} \in \mathbb{R}^n$.
- Mean vector and covariance matrix:

  $$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad \text{and} \quad \operatorname{cov}(\mathbf{x}, \mathbf{x}) = \boldsymbol{\Sigma}.$$

  Therefore, the first parameter $\boldsymbol{\mu}$ is called the *mean vector*, and the second parameter $\boldsymbol{\Sigma}$ is called the *covariance matrix*. The inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ is often called the *precision* matrix.
- The sum-to-1 constraint:

  $$\int \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} = 1.$$

- Any marginal distribution or conditional distribution of these n random variables is also Gaussian. (See Exercise Q2.8.)

As shown in Figure 2.11, the multivariate Gaussian is a unimodal distribution in the n-dimensional space, centering at the mean vector $\boldsymbol{\mu}$. The shape of the distribution depends on the eigenvalues (all positive) of the covariance matrix $\boldsymbol{\Sigma}$.

In order not to make this section further lengthy, the description of other probability distributions, including the uniform, Poisson, gamma, inverse-Wishart, and von Mises–Fisher distributions, are provided in Appendix A as a reference for readers.

Some important identities related to the multivariate Gaussian distribution are as follows:

$$\int \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} = 1,$$

$$\int \mathbf{x}\, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} = \boldsymbol{\mu},$$

$$\int \mathbf{x}\mathbf{x}^\top\, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\, d\mathbf{x} = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top.$$

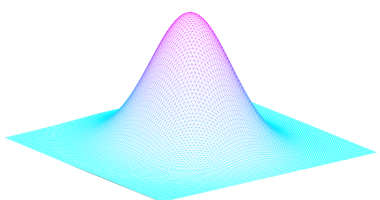


Figure 2.11: An illustration of a unimodal multivariate Gaussian distribution in a two-dimensional space.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

2.2.5 Transformation of Random Variables

Assume we have a set of n continuous random variables, denoted as $\{X_1, X_2, \cdots, X_n\}$. If we arrange their values as a vector $\mathbf{x} \in \mathbb{R}^n$, we can represent their joint distribution (p.d.f.) as $p(\mathbf{x})$. We can apply some transformations to convert them into another set of n continuous random variables as follows:

$$\begin{aligned} Y_1 &= f_1(X_1, X_2, \cdots, X_n) \\ Y_2 &= f_2(X_1, X_2, \cdots, X_n) \\ &\vdots \\ Y_n &= f_n(X_1, X_2, \cdots, X_n). \end{aligned}$$

We similarly arrange the values of the new random variables $\{Y_1, Y_2, \cdots, Y_n\}$ as another vector $\mathbf{y} \in \mathbb{R}^n$, and we further represent the transformations as a single vector-valued and multivariate function:

$$\mathbf{y} = f(\mathbf{x}) \qquad (\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n).$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \xrightarrow{f} \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \xrightarrow{f^{-1}} \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

If this function is continuously differentiable and invertible, we can represent the inverse function as $\mathbf{x} = f^{-1}(\mathbf{y})$. Under these conditions, we are able to conveniently derive the joint distribution for these new random variables, that is, $p(\mathbf{y})$.

We first need to define the so-called Jacobian matrix for these inverse transformations $\mathbf{x} = f^{-1}(\mathbf{y})$, as follows:

$$\mathbf{J}(\mathbf{y}) = \left[\frac{\partial x_i}{\partial y_j} \right]_{n \times n} = \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \cdots & \frac{\partial x_1}{\partial y_n} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} & \cdots & \frac{\partial x_2}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial y_1} & \frac{\partial x_n}{\partial y_2} & \cdots & \frac{\partial x_n}{\partial y_n} \end{bmatrix}.$$

According to Bertsekas [21], the joint distribution of the new random variables can be derived as

$$p(\mathbf{y}) = \left|\mathbf{J}(\mathbf{y})\right| p(\mathbf{x}) = \left|\mathbf{J}(\mathbf{y})\right| p\big(f^{-1}(\mathbf{y})\big), \tag{2.3}$$

where $\left|\mathbf{J}(\mathbf{y})\right|$ denotes the determinant of the Jacobian matrix.

Example 2.2.4 Assume the joint distribution (p.d.f.) of n continuous random variables is given as $p(\mathbf{x})$ ($\mathbf{x} \in \mathbb{R}^n$), and we use an $n \times n$ orthogonal matrix $\mathbf{U}$ to linearly transform $\mathbf{x}$ into another set of n random variables as $\mathbf{y} = \mathbf{U}\mathbf{x}$. Show that $p(\mathbf{y}) = p(\mathbf{x})$ in this case.

$$\mathbf{y} = \mathbf{U}\mathbf{x} \implies \mathbf{x} = \mathbf{U}^{-1}\mathbf{y}.$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

According to the definition of an orthogonal matrix, we know that $\mathbf{U}^{-1} = \mathbf{U}^\top$. Moreover, because the inverse function is linear, we can verify that the Jacobian matrix

$$\mathbf{J}(\mathbf{y}) = \mathbf{U}^{-1} = \mathbf{U}^\top.$$

Because $\mathbf{U}$ is an orthogonal matrix, we have $|\mathbf{U}^\top| = |\mathbf{U}| = 1$. According to the previous result, we can derive $p(\mathbf{y}) = p(\mathbf{x})$ due to $\left|\mathbf{J}(\mathbf{y})\right| = 1$. ♦

An interesting conclusion from this example is that any orthogonal linear transformation of some random variables does not affect their joint distribution.

An *orthogonal matrix* (a.k.a. *orthonormal matrix*) $\mathbf{U}$ is a real square matrix whose column (or row) vectors are normalized to 1 and orthogonal to each other. That is,

$$\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I}.$$

An orthogonal matrix represents a special linear transformation of rotating the coordinate system.

2.3 Information Theory

Information theory was founded by Claude Shannon in 1948 as a discipline to study the quantification, storage, and communication of information. In the past decades, it has played a critical role in modern communication as well as many other applications in engineering and computer science. This section reviews information theory from the perspective of machine learning and emphasizes only the concepts and results relevant to machine learning, particularly *mutual information* [222] and the *Kullback–Leibler (KL) divergence* [137].

2.3.1 Information and Entropy

The first fundamental problem in information theory is how to quantitatively measure information. The most significant progress to address this issue is attributed to Shannon's brilliant idea of using probabilities. The amount of information that a message delivers solely depends on the probability of observing this message rather than its real content or anything else. This treatment allows us to establish a general mathematical framework to handle information independent of application domains. According to Shannon, if the probability of observing an event A is $\Pr(A)$, the amount of information delivered by this event A is calculated as follows:

$$I(A) = \log_2\left(\frac{1}{\Pr(A)}\right) = -\log_2\left(\Pr(A)\right).$$

When we use the binary logarithm $\log_2(\cdot)$, the unit of the calculated information is the bit. Shannon's definition of information is intuitive and consistent with our daily experience. A small-probability event will surprise us because it contains more information, whereas a common event that happens every day is not telling us anything new.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Shannon's idea can be extended to measure information for random variables. As we know, a random variable may take different values in different probabilities, and we can define the so-called *entropy* for a discrete random variable X as the expectation of the information for it to take different values:

$$H(X) = \mathbb{E}\big[-\log_2 \Pr(X = x)\big] = -\sum_x p(x)\log_2 p(x),$$

where $p(x)$ is the p.m.f. of X. Intuitively speaking, the entropy $H(X)$ represents the amount of uncertainty associated with the random variable X, namely, the amount of information we need to fully resolve this random variable.

The entropy of a continuous random variable X is similarly defined as

$$H(X) = \mathbb{E}\big[-\log_2 p(x)\big]$$

$$= -\int_x p(x)\log_2 p(x)dx,$$

where $p(x)$ denotes the p.d.f. of X.

Example 2.3.1 Calculate the entropy for a binary random variable X that takes $x = 1$ in the probability of p and $x = 0$ in the probability of $1 - p$, where $p \in [0, 1]$.

x	1	0
$p(x)$	p	$1-p$

$$H(X) = -\sum_{x=0,1} p(x)\log_2 p(x) = -p\log_2 p - (1-p)\log_2(1-p).$$

We define $0 \times \log_2 0 = 0$.

Figure 2.12 shows $H(X)$ as a function of p and shows that $H(X) = 0$ when $p = 1$ or $p = 0$. In these cases, the entropy $H(X)$ equals to 0 because X surely takes the value of 1 (or 0) when $p = 1$ (or $p = 0$). On the other hand, X achieves the maximum entropy value when $p = 0.5$. In this case, X contains the highest level of uncertainty because it may take either value equiprobably. ♦

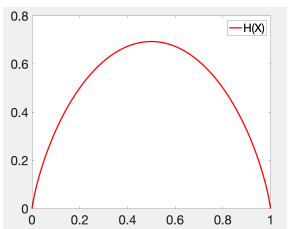


Figure 2.12: An illustration of entropy $H(X)$ as a function of p for a binary random variable.

Example 2.3.2 Calculate the entropy for a continuous random variable X that follows a Gaussian distribution $\mathcal{N}(x \mid \mu_0, \sigma_0^2)$.

$$\begin{aligned} H(X) &= -\int_x \mathcal{N}(x \mid \mu_0, \sigma_0^2)\log_2 \mathcal{N}(x \mid \mu_0, \sigma_0^2)\,dx \\ &= \frac{\log_2(e)}{2}\int_x \Big[\log(2\pi\sigma_0^2) + \frac{(x-\mu_0)^2}{\sigma_0^2}\Big]\mathcal{N}(x \mid \mu_0, \sigma_0^2)\,dx \\ &= \frac{1}{2}\big[\log_2(2\pi\sigma_0^2) + \log_2(e)\big] = \frac{1}{2}\log_2(2\pi e\sigma_0^2). \end{aligned}$$

Refer to the identities of the univariate Gaussian distribution for how to solve this integral.

And we have

$$\log \mathcal{N}(\mu_0, \sigma_0^2) = \frac{\log_2 \mathcal{N}(\mu_0, \sigma_0^2)}{\log_2(e)}.$$

The entropy of a Gaussian variable solely depends on its variance. A larger variance indicates a higher entropy because the random variable scatters more widely. Note that the entropy of a Gaussian variable may become negative when its variance is very small (i.e., $\sigma_0^2 < 1/2\pi e$). ♦

The concept of entropy can be further extended to multiple random variables based on their joint distribution. For example, assuming the joint

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

distribution of two discrete random variables, X and Y, is given as $p(x, y)$, we can define the *joint entropy* for them as follows:

$$H(X,Y) = \mathbb{E}_{X,Y}\big[-\log_2 \Pr(X = x, Y = y)\big] = -\sum_x \sum_y p(x,y) \log_2 p(x,y).$$

If X and Y are continuous random variables, we calculate their joint entropy by replacing the summations with integrals:

$$H(X,Y) = -\int\int p(x,y) \log_2 p(x,y)\, dxdy.$$

Intuitively speaking, the joint entropy represents the total amount of uncertainty associated with these two random variables, namely, the total amount of information we need to resolve both of them.

Furthermore, we can define the so-called *conditional entropy* for two random variables, X and Y, based on their conditional distribution $p(y|x)$ as follows:

$$H(Y|X) = \mathbb{E}_{X,Y}\big[-\log_2 \Pr(Y = y|X = x)\big] = -\sum_x \sum_y p(x,y) \log_2 p(y|x).$$

If X and Y are continuous, we have

$$H(Y|X) = -\int\int p(x,y) \log_2 p(y|x)\, dxdy$$

and

$$H(X|Y) = -\int\int p(x,y) \log_2 p(x|y)\, dxdy.$$

Intuitively speaking, the conditional entropy $H(Y|X)$ indicates the amount of uncertainty associated with Y after X is known, namely, the amount of information we still need to resolve Y even after X is known. Similarly, we can define the conditional entropy $H(X|Y)$ based on the conditional distribution $p(x|y)$ as follows:

$$H(X|Y) = \mathbb{E}_{X,Y}\big[-\log_2 \Pr(X = x|Y = y)\big] = -\sum_x \sum_y p(x,y) \log_2 p(x|y).$$

In the same way, $H(X|Y)$ indicates the amount of uncertainty associated with X after Y is known, namely, the amount of information we still need to resolve X after Y is known.

If two random variables X and Y are independent, we have

$$H(X,Y) = H(X) + H(Y)$$

$$H(X|Y) = H(X) \quad \text{and} \quad H(Y|X) = H(Y).$$

See Exercise Q2.10.

2.3.2 Mutual Information

As we have learned, the entropy $H(X)$ represents the amount of uncertainty related to the random variable X, and the conditional entropy $H(X|Y)$ represents the amount of uncertainty related to the same variable X after another random variable Y is known. Therefore, the difference $H(X) - H(X|Y)$ represents the uncertainty reduction about X before and after Y is known. In other words, it indicates the amount of information another random variable Y can provide for X. We often define this uncertainty reduction as the *mutual information* between these two random

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

If X and Y are continuous,

$$I(X,Y) = \int\int p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\,dxdy.$$

See Exercise Q2.11.

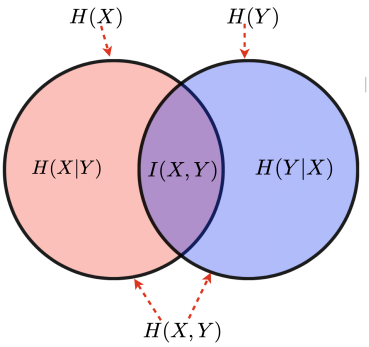


Figure 2.13: An illustration of how mutual information is related to entropy, joint entropy, and conditional entropy.

variables:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= \sum_x\sum_y p(x,y)\log_2\Big(\frac{p(x,y)}{p(x)p(y)}\Big). \end{aligned}$$

Of course, we have several different ways to measure the uncertainty reduction between two random variables, and they all lead to the same mutual information as defined previously:

$$\begin{aligned} I(X,Y) &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned}$$

In general, we can conceptually describe the relations among all of these quantities as shown in Figure 2.13. This diagram is useful to visualize all the aforementioned equations related to mutual information.

In the following discussion, we summarize several important properties of mutual information:

- Mutual information is symmetrical (i.e., $I(X,Y) = I(Y,X)$).
- Mutual information is always nonnegative (i.e., $I(X,Y) \geq 0$).
- $I(X,Y) = 0$ if and only if X and Y are independent.

We can easily verify the first property of symmetry from the definition of mutual information. We will prove the other two properties in the next section. From these, we can see that mutual information is guaranteed to be nonnegative for any random variables. In contrast, entropy is nonnegative only for discrete random variables, and it may become negative for continuous random variables (see Example 2.3.2).

Finally, the next example explains how to use mutual information for feature selection in machine learning.

Example 2.3.3 Mutual Information for Keyword Selection

In many real-world text-classification tasks, we often need to filter out noninformative words in text documents before we build classification models. Mutual information is often used as a popular data-driven criterion to select keywords that are informative.

Assume we want to build a text classifier to automatically classify a news article into one of the predefined topics, such as *sports*, *politics*, *business*, or *science*. We first need to collect some news articles from each of these categories. However, these news articles often contain a large number of distinct words. It will definitely complicate the model learning process if we keep all words used in the text documents. Moreover, in natural

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

languages, there are many common words that are used everywhere, so they do not provide much information in terms of distinguishing news topics. In natural language processing, it is a common practice to filter out all noninformative words in an initial preprocessing stage. Mutual information serves as a popular criterion to calculate the correlation between each word and a news topic for this purpose.

As we have learned, mutual information is defined for random variables. We need to specify random variables before we can actually compute mutual information. Let us first pick up a word (e.g., *score*) and a topic (e.g., "sports"); we define two binary random variables:

$X \in \{0,1\}$: whether a document's topic is "sports" or not,
$Y \in \{0,1\}$: whether a document contains the word *score* or not.

We can go over the entire text corpus to compute a joint distribution for X and Y, as shown in the margin. The probabilities in the table are computed based on the counts for each case. For example, we can do the following counts:

$p(x,y)$	$y=0$	$y=1$	$p(x)$
$x=0$	0.80	0.02	0.82
$x=1$	0.11	0.07	0.18
$p(y)$	0.91	0.09	

$$p(X=1,Y=1) = \frac{\#\text{ of docs with topic "sports" and containing } score}{\text{total \# of docs in the corpus}}$$

$$p(X=1,Y=0) = \frac{\#\text{ of docs with topic "sports" but not containing } score}{\text{total \# of docs in the corpus}}$$

$$p(X=0,Y=0) = \frac{\#\text{ of docs without topic "sports" and not containing } score}{\text{total \# of docs in the corpus}}$$

$$p(X=0,Y=1) = \frac{\#\text{ of docs without topic "sports" but containing } score}{\text{total \# of docs in the corpus}}.$$

Once all probabilities are computed for the joint distribution, the mutual information $I(X,Y)$ can be computed as

$$I(X,Y) = \sum_{x\in\{0,1\}} \sum_{y\in\{0,1\}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} = 0.126.$$

$$\begin{aligned} I(X,Y) = & \\ & 0.80 \times \log_2 \frac{0.80}{0.82 \times 0.91} \\ & +0.02 \times \log_2 \frac{0.02}{0.82 \times 0.09} \\ & +0.11 \times \log_2 \frac{0.11}{0.18 \times 0.91} \\ & +0.07 \times \log_2 \frac{0.07}{0.18 \times 0.09} \\ & = 0.126 \end{aligned}$$

The mutual information $I(X,Y)$ reflects the correlation between the word *score* and the topic "sports." If we repeat this procedure for another word, *what*, and the topic "sports," we may obtain the corresponding $I(X,Y) = 0.00007$. From these two cases, we can tell that the word *score* is much more informative than *what* in relation to the topic "sports." Finally, we just need to repeat these steps of mutual information computation for all combinations of words and topics, then filter out all words that yield low mutual-information values with respect to all topics. ♦

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

2.3.3 KL Divergence

Kullback–Leibler (KL) divergence is a criterion to measure the difference between two probability distributions that have the same support. Given any two distributions (e.g., $p(x)$ and $q(x)$), if the domains of their underlying random variables are the same, the KL divergence is defined as the expectation of the logarithm difference between two distributions over the entire domain:

$$\mathrm{KL}\Big(p(x)\,\Big\|\,q(x)\Big) \triangleq \mathbb{E}_{x\sim p(x)}\left[\log\left(\frac{p(x)}{q(x)}\right)\right].$$

Note that the expectation is computed with respect to the first distribution in the KL divergence. As a result, the KL divergence is not symmetric; that is, $\mathrm{KL}(q(x)\,||\,p(x)) \neq \mathrm{KL}(p(x)\,||\,q(x))$.

For discrete random variables, we can calculate the KL divergence as follows:

$$\mathrm{KL}\Big(p(x)\,\Big\|\,q(x)\Big) = \sum_x p(x)\,\log\left(\frac{p(x)}{q(x)}\right).$$

On the other hand, if the random variables are continuous, the KL divergence is computed with the integral as follows:

$$\mathrm{KL}\Big(p(x)\,\Big\|\,q(x)\Big) = \int p(x)\,\log\left(\frac{p(x)}{q(x)}\right)dx.$$

Regarding the property of the KL divergence, we have the following result from mathematical statistics:

Theorem 2.3.1 *The KL divergence is always nonnegative:*

$$\mathrm{KL}\Big(p(x)\,\Big\|\,q(x)\Big) \geq 0.$$

Furthermore, $\mathrm{KL}\Big(p(x)\,\Big\|\,q(x)\Big) = 0$ *if and only if* $p(x) = q(x)$ *holds almost everywhere in the domain.*

Proof:

Step 1: Reviewing Jensen's inequality

Let's first review Jensen's inequality [114] because this theorem can be derived as a corollary from Jensen's inequality. As shown in Figure 2.14, a real-valued function is called *convex* if the line segment between any two points on the graph of the function lies above or on the graph. If $f(x)$ is convex, for any two points x_1 and x_2, we have

$$f\big(\varepsilon x_1 + (1-\varepsilon)x_2\big) \leq \varepsilon f(x_1) + (1-\varepsilon)f(x_2).$$

$$\log\left(\frac{p(x)}{q(x)}\right) = \log p(x) - \log q(x).$$

By definition, we have

$$\mathrm{KL}\Big(q(x)\,\Big\|\,p(x)\Big)$$

$$\triangleq \mathbb{E}_{x\sim q(x)}\left[\log\left(\frac{q(x)}{p(x)}\right)\right].$$

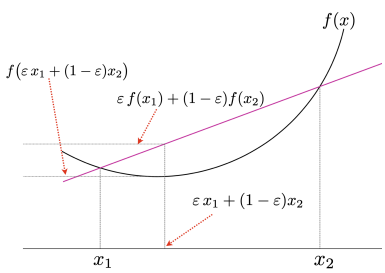


Figure 2.14: An illustration of Jensen's inequality for two points of a convex function. (Image credit: Eli Osherovich/CC-BY-SA-3.0.)

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

for any $\varepsilon \in [0, 1]$. Jensen's inequality generalizes the statement that the secant line of a convex function lies above the graph of the function from two points to any number of points. In the context of probability theory, Jensen's inequality states that if X is a random variable and $f(\cdot)$ is a convex function, then we have

$$f\big(\mathbb{E}[X]\big) \leq \mathbb{E}\big[f(X)\big]$$

The complete proof of Jensen's inequality is not shown here because its complexity is beyond the scope of this book.

Step 2: Showing the function $-\log(x)$ is strictly convex

We know that any twice-differentiable function is convex if it has positive second-order derivatives everywhere. It is easy to show that $-\log(x)$ has positive second-order derivatives (see margin note).

We have

$$\frac{d^2}{dx^2}\Big(-\log(x)\Big) = \frac{1}{x^2} > 0$$

for all $x > 0$.

Step 3: Applying Jensen's inequality to $-\log(x)$

$$\begin{aligned}
\mathrm{KL}\Big(p(x) \,\|\, q(x)\Big) &= \mathbb{E}_{x\sim p(x)}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] = \mathbb{E}_{x\sim p(x)}\left[-\log\left(\frac{q(x)}{p(x)}\right)\right] \\
&\geq -\log\left(\mathbb{E}_{x\sim p(x)}\left[\frac{q(x)}{p(x)}\right]\right) = -\log\left(\int \cancel{p(x)}\,\frac{q(x)}{\cancel{p(x)}}\,dx\right) \\
&= -\log\int q(x)\,dx = -\log(1) = 0.
\end{aligned}$$

$q(x)$ satisfies the sum-to-1 constraint because it is a probability distribution.

According to Jensen's inequality, equality holds if and only if $\log\big(p(x)/q(x)\big)$ is a constant. Because both $p(x)$ and $q(x)$ satisfy the sum-to-1 condition, this leads to $p(x)/q(x) = 1$ almost everywhere in the domain. ■

Because of the property stated in the theorem, the KL divergence is often used as a measure of how one probability distribution differs from another reference probability distribution, analogous to the Euclidean distance for two points in a space. Intuitively speaking, the KL divergence $\mathrm{KL}\big(q(x) \,||\, p(x)\big)$ represents the amount of information lost when we replace one probability distribution $p(x)$ with another distribution $q(x)$. However, we have to note that the KL divergence does not qualify as a formal statistical metric because the KL divergence is not symmetric, and it does not satisfy the triangle inequality.

In the context of machine learning, the KL divergence is often used as a loss measure when we use a simple statistical model $q(x)$ to approximate a complicated model $p(x)$. In this case, the best-fit simple model, denoted as $q^*(x)$, can be derived by minimizing the KL divergence between them:

$$q^*(x) = \arg\min_{q(x)} \mathrm{KL}\big(q(x) \,||\, p(x)\big).$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

The best-fit model $q^*(x)$ found here is optimal because the minimum amount of information is lost when a complicated model is approximated by a simple one. We will come back to discuss this idea further in Chapter 13 and Chapter 14.

Finally, we can also see that mutual information $I(X,Y)$ can be cast as the KL divergence of the following form:

$$I(X,Y) = \mathrm{KL}\Big(p(x,y) \,\Big\|\, p(x)p(y)\Big).$$

This formulation first proves that mutual information $I(X,Y)$ is always nonnegative because it is a special kind of the KL divergence. We can also see that $I(X,Y) = 0$ if and only if $p(x,y) = p(x)p(y)$ holds for all x and y, which implies that X and Y are independent. Therefore, mutual information can be viewed as an information gain from the assumption that random variables are independent.

Example 2.3.4 Compute the KL divergence between two univariate Gaussian distributions with a common variance σ^2: $\mathcal{N}(x \mid \mu_1, \sigma^2)$ and $\mathcal{N}(x \mid \mu_2, \sigma^2)$.

Based on the definition of the KL divergence, we have

$$\begin{aligned}
& \mathrm{KL}\Big(\mathcal{N}(x \mid \mu_1, \sigma^2) \,\Big\|\, \mathcal{N}(x \mid \mu_2, \sigma^2)\Big) \\
= & \int \mathcal{N}(x \mid \mu_1, \sigma^2) \log \frac{\mathcal{N}(x \mid \mu_1, \sigma^2)}{\mathcal{N}(x \mid \mu_2, \sigma^2)}\, dx \\
= & -\frac{1}{2\sigma^2} \int \mathcal{N}(x \mid \mu_1, \sigma^2) \Big[(\mu_1^2 - \mu_2^2) - 2x(\mu_1 - \mu_2)\Big]\, dx \\
= & -\frac{1}{2\sigma^2}\Big[(\mu_1^2 - \mu_2^2) - 2\mu_1(\mu_1 - \mu_2)\Big] = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}.
\end{aligned}$$

Note that

$$\int x\mathcal{N}(x \mid \mu_1, \sigma^2)\, dx = \mu_1$$

We can see that the KL divergence is nonnegative and equals to 0 only when these two Gaussian distributions are identical (i.e., $\mu_1 = \mu_2$). ♦

2.4 Mathematical Optimization

Many real-world problems in engineering and science require us to find the best-fit candidate among a family of feasible choices in that it satisfies certain design criteria in an optimal way. These problems can be cast as a universal problem in mathematics, called *mathematical optimization* (or *optimization* for short). In an optimization problem, we always start with a criterion and formulate an objective function that can quantitatively measure the underlying criterion as a function of all available choices. The optimization problem is solved by just finding the variables that maximize

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

or minimize the objective function among all feasible choices. The feasibility of a choice is usually specified by some constraints in the optimization problem. The following discussion first introduces a general formulation for all mathematical optimization problems, along with some related concepts and terminologies. Next, some analytic results regarding optimality conditions for this general optimization problem under several typical scenarios are presented. As we will see, for many simple optimization problems, we can handily derive closed-form solutions based on these optimality conditions. However, for other sophisticated optimization problems arising from practical applications, we will have to rely on numerical methods to derive a satisfactory solution in an iterative manner. Finally, some popular numerical optimization methods that play an important role in machine learning, such as a variety of gradient descent methods, will be introduced.

2.4.1 General Formulation

We first assume each candidate in an optimization problem can be specified by a bunch of free variables that are collectively represented as a vector $\mathbf{x} \in \mathbb{R}^n$, and the underlying objective function is given as $f(\mathbf{x})$. All kinds of constraints on a feasible choice can always be described by another set of functions of $\mathbf{x}$, among which we may have equality and/or inequality constraints. Without losing generality, any mathematical optimization problem can be formulated as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ f(\mathbf{x}), \tag{2.4}$$

subject to

$$h_i(\mathbf{x}) = 0 \quad (i = 1, 2, \cdots, m), \tag{2.5}$$

$$g_j(\mathbf{x}) \leq 0 \quad (j = 1, 2, \cdots, n). \tag{2.6}$$

The m equality constraints in Eq. (2.5) and the n inequality constraints in Eq. (2.6) collectively define a feasible set Ω for the free variable $\mathbf{x}$. We assume that these constraints are specified in a meaningful way so that the resultant feasible set Ω is nonempty. An optimization problem essentially requires us to search for all values of $\mathbf{x}$ in Ω so as to yield the best value $\mathbf{x}^*$ that minimizes the objective function in Ω.

This formulation is general enough to accommodate almost all optimization problems. However, without further assumptions on the amenability of the objective function $f(\mathbf{x})$ and all constraint functions $\{h_i(\mathbf{x}), g_j(\mathbf{x})\}$, the optimization problem is in general unsolvable [172]. In the history of mathematical optimization, *linear programming* is the first category of

If we need to maximize $f(\mathbf{x})$, we convert it into minimization as follows:

$$\arg\max_{\mathbf{x}} f(\mathbf{x}) \Leftrightarrow \arg\min_{\mathbf{x}} -f(\mathbf{x}).$$

Similarly, we have

$$g_j(\mathbf{x}) \geq 0 \Leftrightarrow -g_j(\mathbf{x}) \leq 0.$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

optimization problems that has been extensively studied. The optimization is said to be a linear programming problem if all functions in the formulation, including $f(\mathbf{x})$ and $\{h_i(\mathbf{x}), g_j(\mathbf{x})\}$, are linear or affine. Linear programming problems are considered to be easy to solve, and plenty of efficient numerical methods have been developed to solve all sorts of linear programming problems with reasonable theoretical guarantees.

A linear function takes the form of

$$y = \mathbf{a}^\top \mathbf{x},$$

and an affine function takes the form:

$$y = \mathbf{a}^\top \mathbf{x} + b.$$

During the past decades, the research in mathematical optimization has mainly focused on a more general group of optimization problems called *convex optimization* [28, 172]. The optimization is a convex optimization problem if the objective function $f(\mathbf{x})$ is a convex function (see Figure 2.14) and the feasible set Ω defined by all constraints is a convex set (see margin note). All convex optimization problems have the nice property that a locally optimal solution is guaranteed to be globally optimal. Because of this, convex optimization problems can be efficiently solved by many local search algorithms, which are theoretically guaranteed to converge at a reasonable speed. Compared with linear programming, convex optimization represents a much wider range of optimization problems, including linear programming, quadratic programming, second-order cone programming, and semidefinite programming. Many real-world problems can be formulated or approximated as a convex optimization problem. As we will see, convex optimization also plays an important role in machine learning. The learning problems of many useful models are actually convex optimization. The nice properties of convex optimization ensure that these models can be efficiently learned in practice.

A set is said to be convex if for any two points in the set, the line segment joining them lies entirely within the set.

2.4.2 Optimality Conditions

Here, we will first review the necessary and/or sufficient conditions for any $\mathbf{x}^*$ that is an optimal solution to the optimization problem in Eq. (2.4). These optimality conditions will not only provide us with a good understanding of optimization problems in theory but also help to derive a closed-form solution for some relatively simple problems. We will discuss the optimality conditions for three different scenarios of the optimization problem in Eq. (2.4), namely, without any constraint, under only equality constraints, and under both equality and inequality constraints.

Unconstrained Optimization

Let's start with the cases where we aim to minimize an objective function without any constraint. In general, an unconstrained optimization problem can be represented as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}). \tag{2.7}$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

For any function $f(\mathbf{x})$, we can define the following concepts relevant to the optimality conditions of Eq. (2.7):

- **Global minimum (maximum)**
  A point $\hat{\mathbf{x}}$ is said to be a global minimum (or maximum) of $f(\mathbf{x})$ if $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ (or $f(\hat{\mathbf{x}}) \geq f(\mathbf{x})$) holds for any $\mathbf{x}$ in the domain of the function $f(\mathbf{x})$; see Figure 2.15.
- **Local minimum (maximum)**
  A point $\hat{\mathbf{x}}$ is said to be a local minimum (or maximum) of $f(\mathbf{x})$ if $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ (or $f(\hat{\mathbf{x}}) \geq f(\mathbf{x})$) holds for all $\mathbf{x}$ within a local neighborhood of $\hat{\mathbf{x}}$; that is, $\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon$ for some $\epsilon > 0$, as shown in Figure 2.15. All local minimum and maximum points are also called *local extreme points*.
- **Stationary point**
  If a function $f(\mathbf{x})$ is differentiable, we can compute partial derivatives with respect to each element in $\mathbf{x}$. These partial derivatives are often arranged as the so-called *gradient* vector, denoted as follows:

$$\nabla f(\mathbf{x}) \triangleq \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

  The gradient can be computed for any $\mathbf{x}$ in the function domain. If the gradient $\nabla f(\mathbf{x})$ is nonzero, it points to the direction of the fastest increase of the function value at $\mathbf{x}$. On the other hand, a point $\hat{\mathbf{x}}$ is said to be a stationary point of $f(\mathbf{x})$ if all partial derivatives are 0 at $\hat{\mathbf{x}}$; that is, the gradient vanishes at $\hat{\mathbf{x}}$:

$$\nabla f(\hat{\mathbf{x}}) \triangleq \nabla f(\mathbf{x})\Big|_{\mathbf{x}=\hat{\mathbf{x}}} = 0.$$

- **Critical point**
  A point $\hat{\mathbf{x}}$ is a critical point of a function if it is either a stationary point or a point where the gradient is undefined. For a general function, critical points include all stationary points and all singular points where the function is not differentiable. On the other hand, if the function is differentiable everywhere, every critical point is also a stationary point.
- **Saddle point**
  If a point $\hat{\mathbf{x}}$ is a critical point but it is not a local extreme point of the function $f(\mathbf{x})$, it is called a *saddle point*. There are usually a large number of saddle points on the high-dimensional surface of a multivariate function, as shown in Figure 2.16.

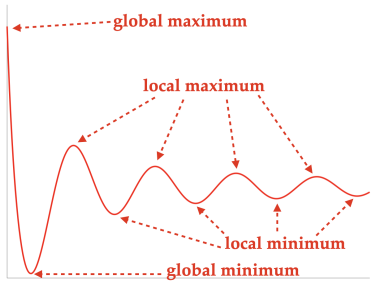


Figure 2.15: An illustration of global minimum (maximum) points versus local minimum (maximum) points.

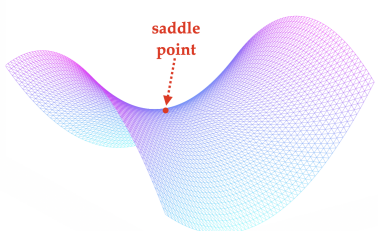


Figure 2.16: An illustration of a saddle point at $x = 0, y = 0$ on the surface of $f(x, y) = x^2 - y^2$. It is not an extreme point, but we can verify that the gradient vanishes there.

Figure 2.17 summarizes the relationship between all of the previously

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

discussed concepts for a differentiable function.

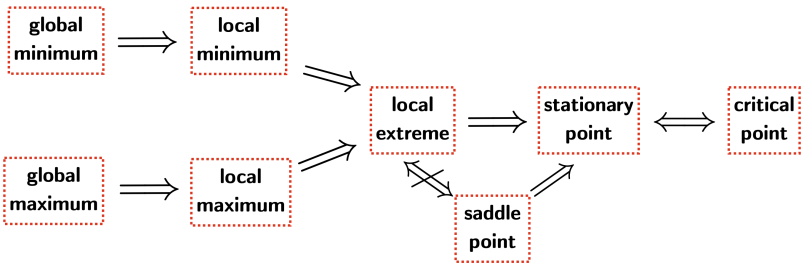


Figure 2.17: A diagram to illustrate all concepts related to stationary points for a differentiable function:
1. $A \implies B$ means A is B.
2. $A \not\Longleftrightarrow B$ means A and B are not the same (disjoint).
3. $A \iff B$ means A and B are equivalent.

Strictly speaking, only a global minimum point constitutes an optimal solution to the optimization problem in Eq. (2.7). *Global optimization* methods aim to find a global minimum for the optimization problem in Eq. (2.7). However, for most objective functions, finding a global optimal point is an extremely challenging task, in which the computational complexity often exponentially grows with the number of free variables. As a result, we often have to relax to resort to a *local optimization* strategy, where a local optimization algorithm can only find a local minimum for the optimization problem in Eq. (2.7).

For any differentiable objective function, we have the following necessary condition for any locally optimal solution:

Theorem 2.4.1 (necessary condition for unconstrained optimization) *Assume the objective function $f(\mathbf{x})$ is differentiable everywhere. If $\mathbf{x}^*$ is a local minimum of Eq. (2.7), then $\mathbf{x}^*$ must be a stationary point; that is, the gradient vanishes at $\mathbf{x}^*$ as $\nabla f(\mathbf{x}^*) = 0$.*

This theorem suggests a simple strategy to solve any unconstrained optimization problem in Eq. (2.7). If we can compute the gradient of the objective function $\nabla f(\mathbf{x})$, we can vanish it by solving the following:

$$\nabla f(\mathbf{x}) = 0,$$

which results in a group of n equations. If we can solve these equations explicitly, their solution may be a locally optimal solution to the original unconstrained optimization problem. Because the previous theorem only states a necessary condition, the found solution may also be a local maximum or a saddle point of the original problem. In practice, we will have to verify whether the solution found by vanishing the gradient is indeed a true local minimum to the original problem.

If the objective function is twice differentiable, we can establish stronger optimality conditions based on the second-order derivatives. In particular, we can compute all second-order partial derivatives for the objective

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

function $f(\mathbf{x})$ in the following $n \times n$ matrix:

$$\mathbf{H}(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right]_{n \times n} = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix},$$

where $\mathbf{H}(\mathbf{x})$ is often called the *Hessian* matrix. Similar to the gradient, we can compute the Hessian matrix at any point $\mathbf{x}$ for a twice-differentiable function. The Hessian matrix $\mathbf{H}(\mathbf{x})$ describes the local curvature of the function surface $f(\mathbf{x})$ at $\mathbf{x}$.

If we have obtained a stationary point $\mathbf{x}^*$ by vanishing the gradient as $\nabla f(\mathbf{x}^*) = 0$, we can know more about $\mathbf{x}^*$ by examining the Hessian matrix at $\mathbf{x}^*$. If $\mathbf{H}(\mathbf{x}^*)$ contains both positive and negative eigenvalues (neither positive nor negative definite), $\mathbf{x}^*$ must be a saddle point, as in Figure 2.16, where the function value increases along some directions and decreases along other directions. If $\mathbf{H}(\mathbf{x}^*)$ contains all positive eigenvalues (positive definite), $\mathbf{x}^*$ is a strict isolated local minimum, where the function value increases along all directions, as in Figure 2.18. If $\mathbf{H}(\mathbf{x}^*)$ only contains both positive and 0 eigenvalues (positive semidefinite), $\mathbf{x}^*$ is still a local minimum, but it is located at a flat valley in Figure 2.18, where the function value remains constant along some directions related to 0 eigenvalues. Finally, if the Hessian matrix also vanishes (i.e., $\mathbf{H}(\mathbf{x}^*) = 0$), $\mathbf{x}^*$ is located on a plateau of the function surface.

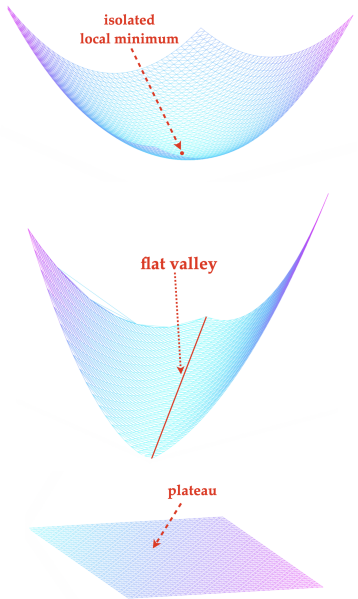


Figure 2.18: An illustration of several different scenarios of a stationary point, where the curvature of the surface is indicated by the Hessian matrix:
1. Isolated minimum: $\mathbf{H}(\mathbf{x}) \succ 0$
2. Flat valley: $\mathbf{H}(\mathbf{x}) \succeq 0$ and $\mathbf{H}(\mathbf{x}) \neq 0$
3. Plateau: $\mathbf{H}(\mathbf{x}) = 0$

Based on the Hessian matrix, we can establish the following second-order necessary or sufficient condition for any local optimal solution of Eq. (2.7) as follows:

Theorem 2.4.2 (second-order necessary condition) *Assume the objective function $f(\mathbf{x})$ is twice differentiable. If $\mathbf{x}^*$ is a local minimum of Eq. (2.7), then*

$$\nabla f(\mathbf{x}^*) = 0 \quad \textit{and} \quad \mathbf{H}(\mathbf{x}^*) \succeq 0.$$

Theorem 2.4.3 (second-order sufficient condition) *Assume the objective function $f(\mathbf{x})$ is twice differentiable. If a point $\mathbf{x}^*$ satisfies*

$$\nabla f(\mathbf{x}^*) = 0 \quad \textit{and} \quad \mathbf{H}(\mathbf{x}^*) \succ 0,$$

then $\mathbf{x}^*$ is an isolated local minimum of Eq. (2.7).

The proofs of Theorems 2.4.1, 2.4.2, and 2.4.3 are straightforward, and they are left for Exercise Q2.14.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Equality Constraints

Let's further discuss the optimality conditions for an optimization problem under only equality constraints, such as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ f(\mathbf{x}), \tag{2.8}$$

subject to

$$h_i(\mathbf{x}) = 0 \quad (i = 1, 2, \cdots, m). \tag{2.9}$$

Generally speaking, stationary points of the objective function $f(\mathbf{x})$ usually are not the optimal solution anymore because these stationary points may not satisfy the constraints in Eq. (2.9). The Lagrange multiplier theorem establishes a first-order necessary condition for the optimization under equality constraints as follows:

Theorem 2.4.4 (Lagrange necessary conditions) *Assume the objective function $f(\mathbf{x})$ and all constraint functions $\{h_i(\mathbf{x})\}$ in Eq. (2.9) are differentiable. If a point $\mathbf{x}^*$ is a local optimal solution to the problem in Eq. (2.8), then the gradients of these functions are linearly dependent at $\mathbf{x}^*$:*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = 0,$$

where $\lambda_i \in \mathbb{R}$ $(i = 1, 2, \cdots, m)$ are called the Lagrange multipliers.

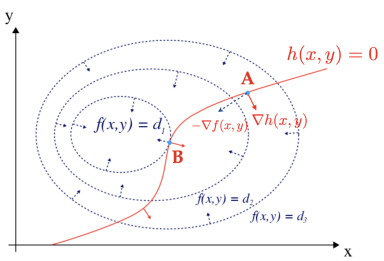


Figure 2.19: An illustration of the Lagrange necessary conditions for an objective function of two free variables $f(x, y)$, which is displayed with the contours, under one equality constraint $h(x, y) = 0$, which is plotted as the red curve.

$$\min_{x,y} \ f(x, y),$$

subject to

$$h(x, y) = 0.$$

We can intuitively explain the Lagrange necessary conditions with the simple example shown in Figure 2.19, where we minimize a function of two variables (i.e., $f(x, y)$) under one equality constraint $h(x, y) = 0$ (plotted as the red curve). Looking at an arbitrary point A on the constraint curve, the negative gradient (i.e., $-\nabla f(x, y)$) points to a direction of the fastest decrease of the function value, and $\nabla h(x, y)$ indicates the norm vector of the curve at A. Imagine we want to move A along the constraint curve to further decrease the function value; we can always project the negative gradient to the tangent plane perpendicular to the norm vector. If we move A slightly along this projected direction, the function value will decrease accordingly. As a result, A cannot be a local optimal point to the original optimization problem. We can continue to move A until it reaches the point B, where the negative gradient is in the same space of the norm vector so that the projection is not possible anymore. This indicates that B may be a local optimal point, and we can verify that the Lagrange condition holds at B.

The Lagrange necessary conditions suggest a convenient treatment to handle equality constraints in any optimization problem in Eq. (2.8) and Eq.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

(2.9). For each equality constraint $h_i(\mathbf{x}) = 0$, we introduce a new free variable λ, called a *Lagrange multiplier*, and construct the so-called *Lagrangian* function:

$$L(\mathbf{x}, \{\lambda_i\}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, h_i(\mathbf{x}).$$

If we can optimize the Lagrangian function with respect to the original variables $\mathbf{x}$ and all Lagrange multipliers, we can derive the solution to the original constrained optimization in Eq. (2.8). We can see that the Lagrangian function is a useful technique to convert a constrained optimization problem into an unconstrained one.

Example 2.4.1 As shown in Figure 2.20, compute the distance from a point $\mathbf{x}_0 \in \mathbb{R}^n$ to a hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ in the space $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are given.

We can formulate this problem as the following constrained optimization problem:

$$d^2 = \min_{\mathbf{x}} \ \|\mathbf{x} - \mathbf{x}_0\|^2,$$

subject to

$$\mathbf{w}^\top \mathbf{x} + b = 0.$$

We introduce a Lagrange multiplier λ for this equality constraint and further construct the Lagrangian function as follows:

$$\begin{aligned} L(\mathbf{x}, \lambda) &= \|\mathbf{x} - \mathbf{x}_0\|^2 + \lambda \left(\mathbf{w}^\top \mathbf{x} + b\right) \\ &= \left(\mathbf{x} - \mathbf{x}_0\right)^\top \left(\mathbf{x} - \mathbf{x}_0\right) + \lambda \left(\mathbf{w}^\top \mathbf{x} + b\right) \end{aligned}$$

$$\begin{aligned} \frac{\partial L(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = 0 \quad &\Longrightarrow \quad 2\left(\mathbf{x} - \mathbf{x}_0\right) + \lambda \mathbf{w} = 0 \\ &\Longrightarrow \quad \mathbf{x}^* = \mathbf{x}_0 - \frac{\lambda^*}{2}\mathbf{w}, \end{aligned}$$

Substituting it into the constraint $\mathbf{w}^\top \mathbf{x}^* + b = 0$, we can solve for λ^* as follows:

$$\lambda^* = \frac{2\left(\mathbf{w}^\top \mathbf{x}_0 + b\right)}{\mathbf{w}^\top \mathbf{w}} = \frac{2\left(\mathbf{w}^\top \mathbf{x}_0 + b\right)}{\|\mathbf{w}\|^2}.$$

Finally, we have

$$d^2 = \|\mathbf{x}^* - \mathbf{x}_0\|^2 = \frac{\lambda^{*2}}{4}\|\mathbf{w}\|^2 = \frac{|\mathbf{w}^\top \mathbf{x}_0 + b|^2}{\|\mathbf{w}\|^2}$$

$$\Longrightarrow \ d = \frac{|\mathbf{w}^\top \mathbf{x}_0 + b|}{\|\mathbf{w}\|}. \qquad \blacklozenge$$

$$\min_{\mathbf{x}, \{\lambda_i\}} \ L(\mathbf{x}, \{\lambda_i\})$$

$$\Longrightarrow \ \frac{\partial L(\mathbf{x}, \{\lambda_i\})}{\partial \mathbf{x}} = 0.$$

This further leads to the same Lagrange conditions in Theorem 2.4.4:

$$\nabla f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, \nabla h_i(\mathbf{x}) = 0.$$

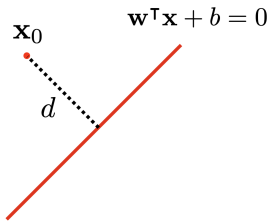


Figure 2.20: An illustration of the distance from any point $\mathbf{x}_0$ to a hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x}),$$

subject to

$$h_i(\mathbf{x}) = 0 \quad (i = 1, 2, \cdots, m)$$

$$g_j(\mathbf{x}) \le 0 \quad (j = 1, 2, \cdots, n).$$

We assume all these constraints define a nonempty feasible set for $\mathbf{x}$, denoted as Ω.

Inequality Constraints

In this section, we will investigate how to establish the optimality conditions for the general optimization problems in Eq. (2.4) (copied at left), which involves both equality and inequality constraints.

First of all, following a similar idea as before, we introduce a Lagrange multiplier λ_i ($\forall i$) for each equality constraint function and a *nonnegative* Lagrange multiplier $\nu_j \ge 0$ ($\forall j$) for each inequality constraint function to construct a Lagrangian function as follows:

$$\begin{aligned} L\Big(\mathbf{x}, \{\lambda_i, \nu_j\}\Big) &= f(\mathbf{x}) + \sum_{i=1}^{m} \overbrace{\lambda_i\, h_i(\mathbf{x})}^{=0} + \sum_{j=1}^{n} \overbrace{\nu_j\, g_j(\mathbf{x})}^{\le 0} \\ &\le f(\mathbf{x}) \qquad (\forall \mathbf{x} \in \Omega). \end{aligned}$$

Because of the constraints $\nu_j \ge 0$ ($\forall j$), this Lagrangian is a lower bound of the original objective function $f(\mathbf{x})$ in the feasible set Ω. We can further minimize out the original variables $\mathbf{x}$ inside Ω so as to derive a function of all Lagrange multipliers:

inf is a generalization of min for open sets.

$$L^*\Big(\{\lambda_i, \nu_j\}\Big) = \inf_{\mathbf{x} \in \Omega} L\Big(\mathbf{x}, \{\lambda_i, \nu_j\}\Big).$$

This function is often called the *Lagrange dual function*. From the above definitions, we can easily show that the dual function is also a lower bound of the original objective function:

$$L^*\Big(\{\lambda_i, \nu_j\}\Big) \le L\Big(\mathbf{x}, \{\lambda_i, \nu_j\}\Big) \le f(\mathbf{x}) \qquad (\mathbf{x} \in \Omega).$$

In other words, the Lagrange dual function is below the original objective function $f(\mathbf{x})$ for all $\mathbf{x}$ in Ω. Assuming $\mathbf{x}^*$ is an optimal solution to the original optimization problem in Eq. (2.4), we still have

$$L^*\Big(\{\lambda_i, \nu_j\}\Big) \le f(\mathbf{x}^*). \tag{2.10}$$

An interesting thing to do is to further optimize all Lagrange multipliers to maximize the dual function in order to close the gap as much as possible, which leads to a new optimization problem, as follows:

$$\{\lambda_i^*, \nu_j^*\} = \arg\max_{\{\lambda_i, \nu_j\}} L^*\Big(\{\lambda_i, \nu_j\}\Big),$$

subject to

$$\nu_j \ge 0 \qquad (j = 1, 2, \cdots, n).$$

This new optimization problem is called the *Lagrange dual problem*. In contrast, the original optimization problem in Eq. (2.4) is called the *primal*

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

problem. From Eq. (2.10), we can immediately derive

$$L^*\Big(\{\lambda_i^*, \nu_j^*\}\Big) \leq f(\mathbf{x}^*).$$

If the original primary problem is convex optimization and some minor qualification conditions (such as Slater's condition [225]) are met, the gap becomes 0, and the equality

$$L^*\Big(\{\lambda_i^*, \nu_j^*\}\Big) = f(\mathbf{x}^*)$$

holds, which is called *strong duality*. When strong duality holds, $\mathbf{x}^*$ and $\{\lambda_i^*, \nu_j^*\}$ form a saddle point of the Lagrangian $L\big(\mathbf{x}, \{\lambda_i, \nu_j\}\big)$, as shown in Figure 2.21, where the Lagrangian increases with respect to $\mathbf{x}$ but decreases with respect to $\{\lambda_i, \nu_j\}$. In this case, both the primary and dual problems are equivalent because they lead to the same optimal solution at the saddle point.

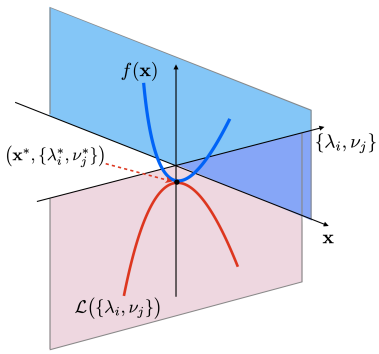


Figure 2.21: An illustration of strong duality occurring in a saddle point of the Lagrangian function.

When strong duality holds, we have

$$\begin{aligned} f(\mathbf{x}^*) &= L^*\Big(\{\lambda_i^*, \nu_j^*\}\Big) \leq L\Big(\mathbf{x}^*, \{\lambda_i^*, \nu_j^*\}\Big) \\ &= f(\mathbf{x}^*) + \sum_{i=1}^{m} \underbrace{\lambda_i^* \, h_i(\mathbf{x}^*)}_{=\,0} + \sum_{j=1}^{n} \nu_j^* \, g_j(\mathbf{x}^*). \end{aligned}$$

From this, we can see that $\sum_{j=1}^{n} \nu_j^* \, g_j(\mathbf{x}^*) \geq 0$. On the other hand, by definition, we have $\nu_j^* \, g_j(\mathbf{x}^*) \leq 0$ for all $j = 1, 2 \cdots, n$. These results further suggest the so-called *complementary slackness* conditions:

$$\nu_j^* \, g_j(\mathbf{x}^*) = 0 \quad (j = 1, 2, \cdots, n).$$

Finally, we summarize all of the previous results, called Karush–Kuhn–Tucker (KKT) conditions [124, 135], in the following theorem:

Theorem 2.4.5 (KKT necessary conditions) *If $\mathbf{x}^*$ and $\{\lambda_i^*, \nu_j^*\}$ form a saddle point of the Lagrangian function $L\big(\mathbf{x}, \{\lambda_i, \nu_j\}\big)$, then $\mathbf{x}^*$ is an optimal solution to the problem in Eq. (2.4). The saddle point satisfies the following conditions:*
1. Stationariness:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \, \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{n} \nu_j^* \, \nabla g_j(\mathbf{x}^*) = 0.$$

2. Primal feasibility:

$$h_i(\mathbf{x}^*) = 0 \quad \textit{and} \quad g_j(\mathbf{x}^*) \leq 0 \quad (\forall i = 1, 2, \cdots, m; j = 1, 2, \cdots, n).$$

Note that the stationariness condition is derived as such because the saddle point is a stationary point, where the gradient vanishes.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

3. Dual feasibility:

$$\nu_j^* \geq 0 \qquad (\forall j = 1, 2, \cdots, n).$$

4. Complementary slackness:

$$\nu_j^* \, g_j(\mathbf{x}^*) = 0 \qquad (\forall j = 1, 2, \cdots, n).$$

Next, we will use an example to show how to apply the KKT conditions to solve an optimization problem under inequality constraints.

Example 2.4.2 Compute the distance from a point $\mathbf{x}_0 \in \mathbb{R}^n$ to a half-space $\mathbf{w}^\intercal \mathbf{x} + b \leq 0$ in the space $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are given.

Similar to Example 2.4.1, we can formulate this problem as the following constrained optimization problem:

$$d^2 = \min_{\mathbf{x}} \ \|\mathbf{x} - \mathbf{x}_0\|^2,$$

subject to

$$\mathbf{w}^\intercal \mathbf{x} + b \leq 0.$$

We introduce a Lagrange multiplier ν for the inequality constraint. As opposed to Example 2.4.1, because this is an inequality constraint, we have the complementary slackness and dual-feasibility conditions:

$$\nu^* \left(\mathbf{w}^\intercal \mathbf{x}^* + b\right) = 0 \qquad \text{and} \qquad \nu^* \geq 0.$$

Accordingly, we can conclude that the optimal solution $\mathbf{x}^*$ and ν^* must be one of the following two cases:

$$(a) \ \mathbf{w}^\intercal \mathbf{x}^* + b = 0 \quad \text{and} \quad \nu^* \geq 0,$$

$$(b) \ \nu^* = 0 \quad \text{and} \quad \mathbf{w}^\intercal \mathbf{x}^* + b \leq 0.$$

For case (a), where $\mathbf{w}^\intercal \mathbf{x}^* + b = 0$ must hold, we can derive from the stationariness condition in the same way as in Example 2.4.1:

$$L(\mathbf{x}, \nu) = \|\mathbf{x} - \mathbf{x}_0\|^2 + \nu \left(\mathbf{w}^\intercal \mathbf{x} + b\right),$$

$$\frac{\partial L(\mathbf{x}, \nu)}{\partial \mathbf{x}} = 0 \implies \nu^* = \frac{2\left(\mathbf{w}^\intercal \mathbf{x}_0 + b\right)}{\|\mathbf{w}\|^2}.$$

If $\mathbf{w}^\intercal \mathbf{x}_0 + b \geq 0$, corresponding to the case where the half-space does not contain $\mathbf{x}_0$ (see the left side in Figure 2.22), we have $\nu^* \geq 0$ for this case. This leads to the same problem as Example 2.4.1. We can finally derive $d = (\mathbf{w}^\intercal \mathbf{x}_0 + b)/\|\mathbf{w}\|$ for this case. However, if $\mathbf{w}^\intercal \mathbf{x}_0 + b < 0$, corresponding to the case where the half-space contains $\mathbf{x}_0$ (see the right side in Figure

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

2.22), then we have $v^* < 0$. This result is invalid because it violates the dual-feasibility condition.

Moreover, let us consider case (b), where $v^* = 0$ and $\mathbf{w}^\top \mathbf{x}^* + b \leq 0$ must hold. After substituting $v^* = 0$ into the stationariness condition, we can derive $\mathbf{x}^* = \mathbf{x}_0$ and $d = 0$ immediately for this case. This is the correct result for the case where the half-space contains $\mathbf{x}_0$ in the right side of Figure 2.22.

Finally, we can summarize these results as follows:

$$d = \begin{cases} \frac{\mathbf{w}^\top \mathbf{x}_0 + b}{\|\mathbf{w}\|} & \text{if} \quad \mathbf{w}^\top \mathbf{x}_0 + b \geq 0 \\ 0 & \text{if} \quad \mathbf{w}^\top \mathbf{x}_0 + b \leq 0. \end{cases}$$

♦

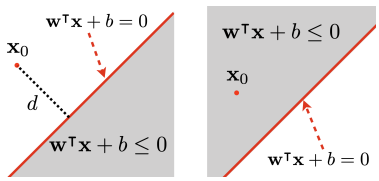


Figure 2.22: An illustration of two cases when computing the distance from a point $\mathbf{x}_0$ to a half-space $\mathbf{w}^\top \mathbf{x} + b \leq 0$:
1. Left: $\mathbf{x}_0$ not in the half-space
2. Right: $\mathbf{x}_0$ in the half-space

2.4.3 Numerical Optimization Methods

For many optimization problems arising from real-world applications, the analytic methods based on the optimality conditions do not always lead to a useful closed-form solution. For these practical problems, we have to rely on numerical methods to derive a reasonable solution in an iterative fashion. Depending on what information is used in each iteration of these numerical methods, they can be roughly classified into several categories, namely, the *zero-order*, *first-order*, and *second-order* methods. In this section, we will briefly review some common methods from each category but focus more on the first-order methods because they are the most popular choices in machine learning. For simplicity, we will use the unconstrained optimization problem

$$\arg\min_{\mathbf{x} \in \mathbb{R}^n} \; f(\mathbf{x})$$

as an example to introduce these numerical methods, but many numerical methods can be easily adapted to deal with constraints.

For example, we can project an unconstrained gradient into the feasible set for all first-order methods, leading to the so-called *projected gradient descent* method.

Zero-Order Methods

The zero-order methods only rely on the zero-order information of the objective function, namely, the function value $f(\mathbf{x})$. We usually need to build a coordinate grid for all free variables in $f(\mathbf{x})$ and then use the grid search strategy to exhaustively check the function value at each point until a satisfactory solution is found. This method is simple, but it suffers from the curse of dimensionality because the number of points in a grid exponentially grows with the number of free parameters. In machine learning, the zero-order methods are mainly used only for the cases where

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

we have a small number of variables (less than 10), such as hyperparameter optimization.

First-Order Methods

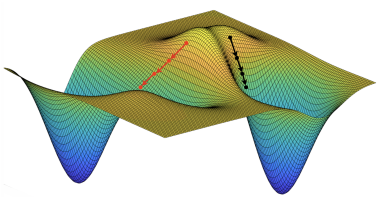


Figure 2.23: An illustration of the gradient descent method, where two trajectories indicate two initial points used by the algorithm.

The first-order methods can access both the zero-order and the first-order information of the objective function, namely, the function value $f(\mathbf{x})$ and the gradient $\nabla f(\mathbf{x})$. As we have learned, the gradient $\nabla f(\mathbf{x})$ points to a direction of the fastest increase of the function value at $\mathbf{x}$. As shown in Figure 2.23, starting from any point on the function surface, if we move a sufficiently small step along the direction of the negative gradient, it is guaranteed that the function value will be more or less decreased. We can repeat this step over and over until it converges to any stationary point. This idea leads to a simple iterative optimization method, called *gradient descent* (a.k.a. *steepest descent*), shown in Algorithm 2.1.

Algorithm 2.1 Gradient Descent Method

randomly choose $\mathbf{x}^{(0)}$, and set η_0
set $n = 0$
while not converged **do**
  update: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \eta_n \nabla f(\mathbf{x}^{(n)})$
  adjust: $\eta_n \to \eta_{n+1}$
  $n = n + 1$
end while

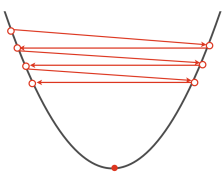


Figure 2.24: An illustration of how a large step size may affect the convergence of the gradient descent method.

Because the gradient cannot tell us how much we should move along the direction, we have to use a manually specified *step size* η_n for each move. The key in the gradient descent method is how to properly choose the step size for each iteration. If the step sizes are too small, the convergence will be slow because it needs to run too many updates to reach any stationary point. On the other hand, if the step sizes are too large, each update may overshoot the target and cause the fluctuation shown in Figure 2.24. As we come close to a stationary point, we usually need to use an even smaller step size to ensure the convergence. As a result, we need to follow a schedule to adjust the step size at the end of each iteration. When we run gradient descent Algorithm 2.1 from any starting point $\mathbf{x}^{(0)}$, it will generate a trajectory $\{\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots\}$ on the function surface, which gradually converges to a stationary point of the objective function. As shown in Figure 2.23, each trajectory heavily depends on the initial point. In other words, if we start from a different initial point $\mathbf{x}^{(0)}$ at the beginning, we may eventually end up with a different solution. Choosing a good initial point is another key factor in ensuring the success of the gradient descent method.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

The gradient descent method is conceptually simple and only needs to use the gradient, which can be easily computed for almost any meaningful objective function. As a result, the gradient descent method becomes a very popular numerical optimization method in practice. If the objective function is smooth and differentiable, we can theoretically prove that the gradient descent algorithm is guaranteed to converge to a stationary point as long as a sufficiently small step size is used at each iteration (see Exercise Q2.16). However, the convergence rate is relatively slow (a *sublinear* rate). If we want to achieve $\|\nabla f(\mathbf{x}^{(n)})\| \leq \epsilon$, we need to run at least $O(1/\epsilon^2)$ iterations. However, if we can make stronger assumptions on the objective function, such as $f(\mathbf{x})$ is convex and at least twice differentiable and its derivatives are sufficiently smooth, we can prove that the gradient descent algorithm is guaranteed to converge to a local minimum $\mathbf{x}^*$ if small enough steps are used. Under these conditions, the gradient descent method can achieve a much faster convergence rate (a *linear* rate). If we want to achieve $\|\mathbf{x}^{(n)} - \mathbf{x}^*\| \leq \epsilon$, we just need to run approximately $O(\ln(1/\epsilon))$ iterations.

In machine learning, we often need to optimize an objective function that can be decomposed as a sum of many homogeneous components:

$$f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}).$$

For example, the objective function $f(\mathbf{x})$ represents the average loss measured on all samples in a training set, and each $f_i(\mathbf{x})$ indicates the loss measure on each training sample. If we use the gradient descent Algorithm 2.1 to minimize this objective function, at each iteration, we need to go through all training samples to compute the gradient as follows:

$$\nabla f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\mathbf{x}).$$

If the training set is large, this step is extremely expensive. To address this issue, we often adopt a stochastic approximation strategy to compute the gradient, where the gradient is approximated as $\nabla f_k(\mathbf{x})$ using a randomly chosen sample k rather than being averaged over all samples. This idea leads to the well-known stochastic gradient descent (SGD) method [24], as shown in Algorithm 2.2.

In Algorithm 2.2, each $\nabla f_k(\mathbf{x})$ can be viewed as a noisy estimate of the true gradient $\nabla f(\mathbf{x})$. Because $\nabla f_k(\mathbf{x})$ can be computed in a relatively cheap way in SGD, we can afford to run much more iterations using a much smaller step size than in the regular gradient descent method. By doing so, the SGD algorithm can converge to a reasonable solution, even in a much shorter training time. Moreover, many empirical results have shown that

If a sequence $\{x_k\}$ converges to the limit x^*: $\lim_{k\to\infty} x_k = x^*$. The rate of convergence is defined as

$$\mu = \lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|}.$$

The convergence is said to be

$$\begin{cases} \text{superlinear} & \text{if } \mu = 0, \\ \text{linear} & \text{if } 0 < \mu < 1, \\ \text{sublinear} & \text{if } \mu = 1. \end{cases}$$

Example 1: For a sequence of exponentially decaying errors:

$$|x_k - x^*| = C\rho^k,$$

where $0 < \rho < 1$. We can verify its convergence rate is linear because

$$\mu = \frac{|x_{k+1} - x^*|}{|x_k - x^*|} = \rho.$$

For any error tolerance $\epsilon > 0$, if we want to achieve $|x_k - x^*| \leq \epsilon$, we have to access x_k with

$$k \geq \frac{\ln C + \ln\frac{1}{\epsilon}}{\ln\frac{1}{\rho}} \approx O(\ln\frac{1}{\epsilon}).$$

Example 2: For another sequence of decaying errors:

$$|x_k - x^*| = \frac{C}{\sqrt{k}},$$

its convergence rate is sublinear as

$$\mu = \lim_{k\to\infty} \frac{\sqrt{k}}{\sqrt{k+1}} = 1.$$

To achieve $|x_k - x^*| \leq \epsilon$, we need

$$k \geq \frac{C^2}{\epsilon^2} \approx O(\frac{1}{\epsilon^2}).$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Algorithm 2.2 Stochastic Gradient Descent (SGD) Method

randomly choose $\mathbf{x}^{(0)}$, and set η_0
set $n = 0$
while not converged **do**
  randomly choose a sample k
  update: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \eta_n \nabla f_k(\mathbf{x}^{(n)})$
  adjust: $\eta_n \to \eta_{n+1}$
  $n = n + 1$
end while

small noises in gradient estimation can even help to converge to a better solution because small noises make the algorithm escape from poor local minimums or saddle points.

Along these lines, an enhanced SGD version is proposed in Algorithm 2.3, where the gradient is estimated at each step, not using only a single sample but from a small subset of randomly chosen samples. Each subset is often called a *mini-batch*. The samples in a mini-batch are randomly chosen every time to ensure all training samples are accessed equally. In Algorithm 2.3, called *mini-batch SGD*, we can choose the size of all mini-batches to control how much noise is injected into each gradient estimation.

Algorithm 2.3 Mini-Batch SGD

randomly choose $\mathbf{x}^{(0)}$, and set η_0
set $n = 0$
while not converged **do**
  randomly shuffle all training samples into mini-batches
  for each mini-batch B **do**
    update: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \frac{\eta_n}{|B|} \sum_{k \in B} \nabla f_k(\mathbf{x})$
    adjust $\eta_n \to \eta_{n+1}$
    $n = n + 1$
  end for
end while

$|B|$ denotes the number of samples in B.

The mini-batch SGD algorithm is a very flexible optimization method because we can always choose several key hyperparameters properly, such as the mini-batch size, the initial learning rate, and the strategy to adjust the learning rate at the end of each step, to make the optimization process converge to a reasonable solution for a large number of practical problems. Therefore, the mini-batch SGD is often regarded as one of the most popular optimization methods in machine learning.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Second-Order Methods

The second-order optimization methods require the use of the zero-order, first-order, and second-order information of the objective function, namely, the function value $f(\mathbf{x})$, the gradient $\nabla f(\mathbf{x})$, and the Hessian matrix $\mathbf{H}(\mathbf{x})$.

For a multivariate function $f(\mathbf{x})$, we can use Taylor's theorem to expand it around any fixed point $\mathbf{x}_0$, as follows:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^\intercal \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\intercal \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + o\big(\|\mathbf{x} - \mathbf{x}_0\|^2\big).$$

If we ignore all high-order terms, we can derive the stationary point by vanishing the gradient $\mathbf{x}^*$, as follows:

$$\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 0 \quad \Longrightarrow \quad \mathbf{x}^* = \mathbf{x}_0 - \mathbf{H}^{-1}(\mathbf{x}_0)\,\nabla f(\mathbf{x}_0).$$

We first compute the gradient as

$$\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \nabla f(\mathbf{x}_0) + \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0).$$

Then we vanish the gradient $\nabla f(\mathbf{x}) = 0$:

$$\nabla f(\mathbf{x}_0) + \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) = 0$$

$$\Longrightarrow \mathbf{x}^* = \mathbf{x}_0 - \mathbf{H}^{-1}(\mathbf{x}_0)\nabla f(\mathbf{x}_0).$$

If $f(\mathbf{x})$ is a quadratic function, no matter where we start, we can use this formula to derive the stationary point in one step. For a general objective function $f(\mathbf{x})$, we can still use the updating rule

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \mathbf{H}^{-1}(\mathbf{x}^{(n)})\,\nabla f(\mathbf{x}^{(n)})$$

in an iterative algorithm, as in Algorithm 2.1, which leads to the *Newton method*. If the objective function is convex and at least twice differentiable and its derivatives are sufficiently smooth, the Newton method is guaranteed to converge to a local minimum $\mathbf{x}^*$, and it can achieve a *superlinear* rate. If we want to achieve $\|\mathbf{x}^{(n)} - \mathbf{x}^*\| \leq \epsilon$, we just need to run approximately $O(\ln\ln(1/\epsilon))$ iterations.

The Newton method is fast in terms of how many iterations are needed to converge. However, each iteration in the Newton method is actually extremely expensive because it involves computing, maintaining, and even inverting a large Hessian matrix. In most machine learning problems, it is impossible to handle the Hessian matrix because we usually have a large number of free variables in $\mathbf{x}$. This is why the Newton method is seldom used in machine learning. Alternatively, there are many approximate second-order methods, called *quasi-Newton* methods, that aim to approximate the Hessian matrix in certain ways (e.g., using some diagonal or block-diagonal matrices to approximate the real Hessian so as to make matrix inversion possible in the updating formula). The popular quasi-Newton methods include the DFP [65], the BFGS [65], Quickprop [60], and the Hessian-free method [175, 160].

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Exercises

Q2.1 Given two matrices, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times n}$, prove that

$$\text{tr}(\mathbf{A}^\top\mathbf{B}) = \text{tr}(\mathbf{A}\mathbf{B}^\top) = \text{tr}(\mathbf{B}\mathbf{A}^\top) = \text{tr}(\mathbf{B}^\top\mathbf{A}) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}b_{ij},$$

where a_{ij} and b_{ij} denote an element in the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively.

Q2.2 For any two square matrices, $\mathbf{X} \in \mathbb{R}^{n\times n}$ and $\mathbf{Y} \in \mathbb{R}^{n\times n}$, show that

a. $\text{tr}(\mathbf{XY}) = \text{tr}(\mathbf{YX})$, and
b. $\text{tr}(\mathbf{X}^{-1}\mathbf{YX}) = \text{tr}(\mathbf{Y})$ if $\mathbf{X}$ is invertible.

Q2.3 Given two sets of m vectors, $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^n$ for all $i = 1, 2, \cdots, m$, verify that the summations $\sum_{i=1}^m \mathbf{x}_i\mathbf{x}_i^\top$ and $\sum_{i=1}^m \mathbf{x}_i\mathbf{y}_i^\top$ can be vectorized as the following matrix multiplications:

$$\sum_{i=1}^{m} \mathbf{x}_i\mathbf{x}_i^\top = \mathbf{X}\mathbf{X}^\top \quad \text{and} \quad \sum_{i=1}^{m} \mathbf{x}_i\mathbf{y}_i^\top = \mathbf{X}\mathbf{Y}^\top,$$

where $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 \, \mathbf{x}_2 \, \cdots \, \mathbf{x}_m\end{bmatrix} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1 \, \mathbf{y}_2 \, \cdots \, \mathbf{y}_m\end{bmatrix} \in \mathbb{R}^{n\times m}$.

Q2.4 Given $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m\times n}$ $(m < n)$,

a. prove that $\mathbf{z}^\top\mathbf{A}\mathbf{x} = \text{tr}\big(\mathbf{x}\,\mathbf{z}^\top\mathbf{A}\big)$,
b. compute the derivative $(\partial/\partial\mathbf{x})\,\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$, and
c. compute the derivative $(\partial/\partial\mathbf{A})\,\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$.

Q2.5 For any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, if we use $\mathbf{a}_i$ $(i = 1, 2, \ldots, n)$ to denote the ith column of the matrix $\mathbf{A}$ and use $g_{ij} = |\cos\theta_{ij}| = |\mathbf{a}_i \cdot \mathbf{a}_j|/(\|\mathbf{a}_i\|\|\mathbf{a}_j\|)$ to denote the absolute cosine of the angle θ_{ij} between any two vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ (for all $1 \le i, j \le n$), show that

$$\frac{\partial}{\partial\mathbf{A}}\left(\sum_{i=1}^{n}\sum_{j=i+1}^{n} g_{ij}\right) = (\mathbf{D} - \mathbf{B})\mathbf{A},$$

where $\mathbf{D}$ is an $n \times n$ matrix with its elements computed as $d_{ij} = \text{sign}(\mathbf{a}_i \cdot \mathbf{a}_j)/(\|\mathbf{a}_i\|\|\mathbf{a}_j\|)$ $(1 \le i, j \le n)$, and $\mathbf{B}$ is an $n \times n$ diagonal matrix with its diagonal elements computed as $b_{ii} = (\sum_{j=1} g_{ij})/\|\mathbf{a}_i\|^2$ $(1 \le i \le n)$.

Q2.6 Consider a multinomial distribution of m discrete random variables as follows:

$$\begin{aligned}\Pr(X_1 = r_1, X_2 = r_2, \ldots, X_m = r_m) &= \text{Mult}\big(r_1, r_2, \ldots, r_m \,\big|\, N, p_1, p_2, \ldots, p_m\big)\\ &= \frac{N!}{r_1!\,r_2!\cdots r_m!}\, p_1^{r_1} p_2^{r_2} \cdots p_m^{r_m}\end{aligned}$$

a. Prove that the multinomial distribution satisfies the sum-to-1 constraint $\sum_{X_1,\cdots,X_m} \Pr(X_1 = r_1, X_2 = r_2, \cdots, X_m = r_m) = 1$.
b. Show the procedure to derive the mean and variance for each X_i $(\forall i = 1, 2, \ldots, m)$ and the covariance for any two X_i and X_j $(\forall i, j = 1, 2, \ldots, m)$.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Q2.7 Assume m continuous random variables $\{X_1, X_2, \ldots, X_m\}$ follow the Dirichlet distribution as follows:

$$\mathrm{Dir}\left(p_1, p_2, \cdots, p_m \,\middle|\, r_1, r_2, \ldots, r_m\right) = \frac{\Gamma(r_1 + \cdots + r_m)}{\Gamma(r_1) \cdots \Gamma(r_m)} p_1^{r_1 - 1} \times p_2^{r_2 - 1} \times \cdots \times p_m^{r_m - 1}.$$

Derive the following results:

$$\mathbb{E}[X_i] = \frac{r_i}{r_0} \qquad \mathrm{var}(X_i) = \frac{r_i(r_0 - r_i)}{r_0^2(r_0 + 1)} \qquad \mathrm{cov}\left(X_i, X_j\right) = -\frac{r_i r_j}{r_0^2(r_0 + 1)},$$

where we denote $r_0 = \sum_{i=1}^{m} r_i$.
Hints: $\Gamma(x + 1) = x \cdot \Gamma(x)$.

Q2.8 Assume n continuous random variables $\{X_1, X_2, \cdots, X_n\}$ jointly follow a multivariate Gaussian distribution $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma)$.

a. For any random variable X_i $(\forall i)$, derive its marginal distribution $p(X_i)$.
b. For any two random variables X_i and X_j $(\forall i, j)$, derive the conditional distribution $p(X_i|X_j)$.
c. For any subset of these random variables S, derive the marginal distribution for S.
d. Split all n random variables into two disjoint subsets S_1 and S_2, and then derive the conditional distribution $p(S_1|S_2)$.

Hints: Some identities for the inversion and determinant of a symmetric block matrix, where $\mathbf{\Sigma}_{11} \in \mathbb{R}^{p \times p}$, $\mathbf{\Sigma}_{12} \in \mathbb{R}^{p \times q}$, $\mathbf{\Sigma}_{22} \in \mathbb{R}^{q \times q}$, are as follows:

$$\begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{12}^{\mathsf{T}} & \mathbf{\Sigma}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{\Sigma}_{11}^{-1} - \mathbf{N}\mathbf{M}^{-1}\mathbf{N}^{\mathsf{T}} & -\mathbf{N}\mathbf{M}^{-1} \\ -\left(\mathbf{N}\mathbf{M}^{-1}\right)^{\mathsf{T}} & \mathbf{M}^{-1} \end{bmatrix}$$

$$\left| \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{12}^{\mathsf{T}} & \mathbf{\Sigma}_{22} \end{bmatrix} \right| = \left|\mathbf{\Sigma}_{11}\right| \left|\mathbf{M}\right|,$$

where $\mathbf{M} = \left(\mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{12}^{\mathsf{T}} \mathbf{\Sigma}_{11}^{-1} \mathbf{\Sigma}_{12}\right)$, and $\mathbf{N} = \mathbf{\Sigma}_{11}^{-1} \mathbf{\Sigma}_{12}$.

Q2.9 Assume a random vector $\mathbf{x} \in \mathbb{R}^n$ follows a multivariate Gaussian distribution (i.e., $p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \middle| \boldsymbol{\mu}, \Sigma\right)$). If we apply an invertible linear transformation to convert $\mathbf{x}$ into another random vector as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ ($\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$), prove that the joint distribution $p(\mathbf{y})$ is also a multivariate Gaussian distribution, and compute its mean vector and covariance matrix.

Q2.10 Show that any two random variables X and Y are independent if and only if any one of the following equations holds:

$$H(X, Y) = H(X) + H(Y)$$
$$H(X|Y) = H(X)$$
$$H(Y|X) = H(Y)$$

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press

Q2.11 Show that mutual information satisfies the following:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned}$$

Q2.12 Assume a random vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ follows a bivariate Gaussian distribution: $\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ is the mean vector and $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$ is the covariance matrix. Derive the formula to compute mutual information between x_1 and x_2 (i.e., $I(x_1, x_2)$).

Q2.13 Given two multivariate Gaussian distributions: $\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the mean vectors, and $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are the covariance matrices, derive the formula to compute the KL divergence between these two Gaussian distributions.

Q2.14 Prove Theorems 2.4.1, 2.4.2, and 2.4.3.

Q2.15 Compute the distance of a point $\mathbf{x}_0 \in \mathbb{R}^n$ to

a. the surface of a unit ball: $\|\mathbf{x}\| = 1$;
b. a unit ball $\|\mathbf{x}\| \leq 1$;
c. an elliptic surface $\mathbf{x}^\intercal \mathbf{A} \mathbf{x} = 1$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} > 0$; and
d. an ovoid $\mathbf{x}^\intercal \mathbf{A} \mathbf{x} \leq 1$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} > 0$.

Hints: Give a numerical procedure if no closed-form solution exists.

Q2.16 Assume a differentiable objective function $f(\mathbf{x})$ is Lipschitz continuous; namely, there exists a real constant $L > 0$, and for any two points $\mathbf{x}_1$ and $\mathbf{x}_2$, $\left|f(\mathbf{x}_1) - f(\mathbf{x}_2)\right| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|$ always holds. Prove that the gradient descent Algorithm 2.1 always converges to a stationary point, namely, $\lim_{n \to \infty} \|\nabla f(\mathbf{x}^{(n)})\| = 0$, as long as all used step sizes are small enough, satisfying $\eta_n < 1/L$.

https://doi.org/10.1017/9781108938051.004 Published online by Cambridge University Press