CHAPTER 1 Data Mining and Analysis

Data mining is the process of discovering insightful, interesting, and novel patterns, as well as descriptive, understandable, and predictive models from large-scale data. We begin this chapter by looking at basic properties of data modeled as a data matrix. We emphasize the geometric and algebraic views, as well as the probabilistic interpretation of data. We then discuss the main data mining tasks, which span exploratory data analysis, frequent pattern mining, clustering, and classification, laying out the roadmap for the book.

1.1 DATA MATRIX

Data can often be represented or abstracted as an $n \times d$ *data matrix*, with n rows and d columns, where rows correspond to entities in the dataset, and columns represent attributes or properties of interest. Each row in the data matrix records the observed attribute values for a given entity. The $n \times d$ data matrix is given as

$$\mathbf{D} = \begin{pmatrix} & X_1 & X_2 & \cdots & X_d \\ \mathbf{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \mathbf{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

where $\mathbf{x}_i$ denotes the ith row, which is a d-tuple given as

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$$

and X_j denotes the jth column, which is an n-tuple given as

$$X_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$$

Depending on the application domain, rows may also be referred to as *entities*, *instances*, *examples*, *records*, *transactions*, *objects*, *points*, *feature-vectors*, *tuples*, and so on. Likewise, columns may also be called *attributes*, *properties*, *features*, *dimensions*, *variables*, *fields*, and so on. The number of instances n is referred to as the *size* of

Table 1.1. Extract from the Iris dataset

	Sepal length X_1	Sepal width X_2	Petal length X_3	Petal width X_4	Class X_5
$\mathbf{x}_1$	5.9	3.0	4.2	1.5	Iris-versicolor
$\mathbf{x}_2$	6.9	3.1	4.9	1.5	Iris-versicolor
$\mathbf{x}_3$	6.6	2.9	4.6	1.3	Iris-versicolor
$\mathbf{x}_4$	4.6	3.2	1.4	0.2	Iris-setosa
$\mathbf{x}_5$	6.0	2.2	4.0	1.0	Iris-versicolor
$\mathbf{x}_6$	4.7	3.2	1.3	0.2	Iris-setosa
$\mathbf{x}_7$	6.5	3.0	5.8	2.2	Iris-virginica
$\mathbf{x}_8$	5.8	2.7	5.1	1.9	Iris-virginica
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{x}_{149}$	7.7	3.8	6.7	2.2	Iris-virginica
$\mathbf{x}_{150}$	5.1	3.4	1.5	0.2	Iris-setosa

the data, whereas the number of attributes d is called the *dimensionality* of the data. The analysis of a single attribute is referred to as *univariate analysis*, whereas the simultaneous analysis of two attributes is called *bivariate analysis* and the simultaneous analysis of more than two attributes is called *multivariate analysis*.

Example 1.1. Table 1.1 shows an extract of the Iris dataset; the complete data forms a 150×5 data matrix. Each entity is an Iris flower, and the attributes include `sepal length`, `sepal width`, `petal length`, and `petal width` in centimeters, and the type or `class` of the Iris flower. The first row is given as the 5-tuple

$$\mathbf{x}_1 = (5.9, 3.0, 4.2, 1.5, \texttt{Iris-versicolor})$$

Not all datasets are in the form of a data matrix. For instance, more complex datasets can be in the form of sequences (e.g., DNA and protein sequences), text, time-series, images, audio, video, and so on, which may need special techniques for analysis. However, in many cases even if the raw data is not a data matrix it can usually be transformed into that form via feature extraction. For example, given a database of images, we can create a data matrix in which rows represent images and columns correspond to image features such as color, texture, and so on. Sometimes, certain attributes may have special semantics associated with them requiring special treatment. For instance, temporal or spatial attributes are often treated differently. It is also worth noting that traditional data analysis assumes that each entity or instance is independent. However, given the interconnected nature of the world we live in, this assumption may not always hold. Instances may be connected to other instances via various kinds of relationships, giving rise to a *data graph*, where a node represents an entity and an edge represents the relationship between two entities.

1.2 ATTRIBUTES

Attributes may be classified into two main types depending on their domain, that is, depending on the types of values they take on.

Numeric Attributes

A *numeric* attribute is one that has a real-valued or integer-valued domain. For example, Age with $domain(\texttt{Age}) = \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers (non-negative integers), is numeric, and so is petal length in Table 1.1, with $domain(\texttt{petal length}) = \mathbb{R}^+$ (the set of all positive real numbers). Numeric attributes that take on a finite or countably infinite set of values are called *discrete*, whereas those that can take on any real value are called *continuous*. As a special case of discrete, if an attribute has as its domain the set $\{0, 1\}$, it is called a *binary* attribute. Numeric attributes can be classified further into two types:

- *Interval-scaled*: For these kinds of attributes only differences (addition or subtraction) make sense. For example, attribute temperature measured in °C or °F is interval-scaled. If it is 20 °C on one day and 10 °C on the following day, it is meaningful to talk about a temperature drop of 10 °C, but it is not meaningful to say that it is twice as cold as the previous day.
- *Ratio-scaled*: Here one can compute both differences as well as ratios between values. For example, for attribute Age, we can say that someone who is 20 years old is twice as old as someone who is 10 years old.

Categorical Attributes

A *categorical* attribute is one that has a set-valued domain composed of a set of symbols. For example, Sex and Education could be categorical attributes with their domains given as

$$domain(\texttt{Sex}) = \{\texttt{M}, \texttt{F}\}$$
$$domain(\texttt{Education}) = \{\texttt{HighSchool}, \texttt{BS}, \texttt{MS}, \texttt{PhD}\}$$

Categorical attributes may be of two types:

- *Nominal*: The attribute values in the domain are unordered, and thus only equality comparisons are meaningful. That is, we can check only whether the value of the attribute for two given instances is the same or not. For example, Sex is a nominal attribute. Also class in Table 1.1 is a nominal attribute with $domain(\texttt{class}) = \{\texttt{iris-setosa}, \texttt{iris-versicolor}, \texttt{iris-virginica}\}$.
- *Ordinal*: The attribute values are ordered, and thus both equality comparisons (is one value equal to another?) and inequality comparisons (is one value less than or greater than another?) are allowed, though it may not be possible to quantify the difference between values. For example, Education is an ordinal attribute because its domain values are ordered by increasing educational qualification.

1.3 DATA: ALGEBRAIC AND GEOMETRIC VIEW

If the d attributes or dimensions in the data matrix $\mathbf{D}$ are all numeric, then each row can be considered as a d-dimensional point:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in \mathbb{R}^d$$

or equivalently, each row may be considered as a d-dimensional column vector (all vectors are assumed to be column vectors by default):

$$\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{pmatrix} = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{id} \end{pmatrix}^T \in \mathbb{R}^d$$

where T is the *matrix transpose* operator.

The d-dimensional Cartesian coordinate space is specified via the d unit vectors, called the standard basis vectors, along each of the axes. The jth *standard basis vector* $\mathbf{e}_j$ is the d-dimensional unit vector whose jth component is 1 and the rest of the components are 0

$$\mathbf{e}_j = (0, \ldots, 1_j, \ldots, 0)^T$$

Any other vector in $\mathbb{R}^d$ can be written as a *linear combination* of the standard basis vectors. For example, each of the points $\mathbf{x}_i$ can be written as the linear combination

$$\mathbf{x}_i = x_{i1}\mathbf{e}_1 + x_{i2}\mathbf{e}_2 + \cdots + x_{id}\mathbf{e}_d = \sum_{j=1}^{d} x_{ij}\mathbf{e}_j$$

where the scalar value x_{ij} is the coordinate value along the jth axis or attribute.

Example 1.2. Consider the Iris data in Table 1.1. If we *project* the entire data onto the first two attributes, then each row can be considered as a point or a vector in 2-dimensional space. For example, the projection of the 5-tuple $\mathbf{x}_1 = (5.9, 3.0, 4.2, 1.5, \texttt{Iris-versicolor})$ on the first two attributes is shown in Figure 1.1a. Figure 1.2 shows the scatterplot of all the $n = 150$ points in the 2-dimensional space spanned by the first two attributes. Likewise, Figure 1.1b shows $\mathbf{x}_1$ as a point and vector in 3-dimensional space, by projecting the data onto the first three attributes. The point $(5.9, 3.0, 4.2)$ can be seen as specifying the coefficients in the linear combination of the standard basis vectors in $\mathbb{R}^3$:

$$\mathbf{x}_1 = 5.9\mathbf{e}_1 + 3.0\mathbf{e}_2 + 4.2\mathbf{e}_3 = 5.9\begin{pmatrix}1\\0\\0\end{pmatrix} + 3.0\begin{pmatrix}0\\1\\0\end{pmatrix} + 4.2\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}5.9\\3.0\\4.2\end{pmatrix}$$

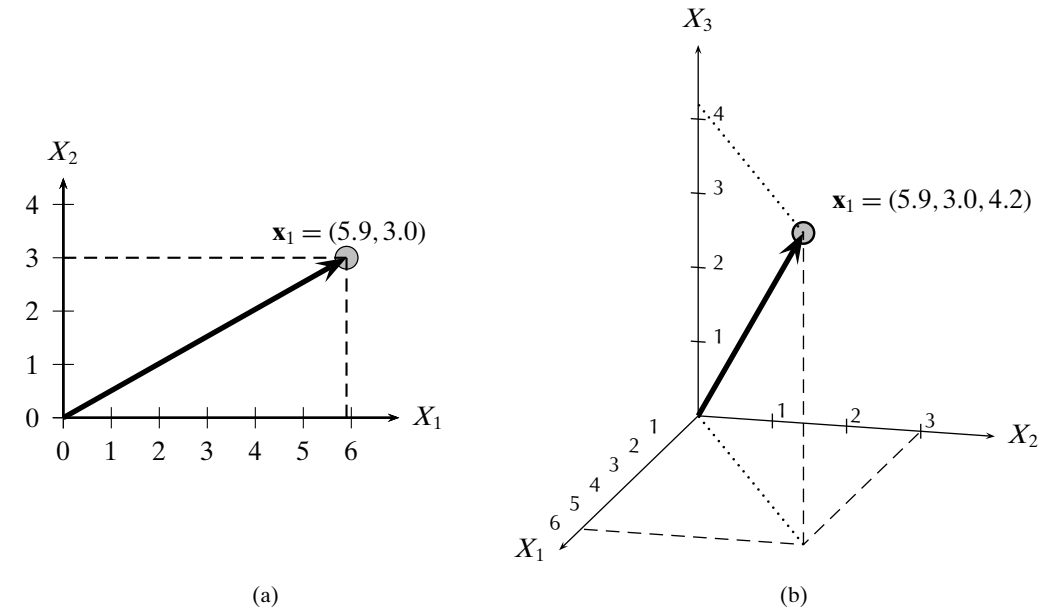

Figure 1.1. Row $\mathbf{x}_1$ as a point and vector in (a) $\mathbb{R}^2$ and (b) $\mathbb{R}^3$.

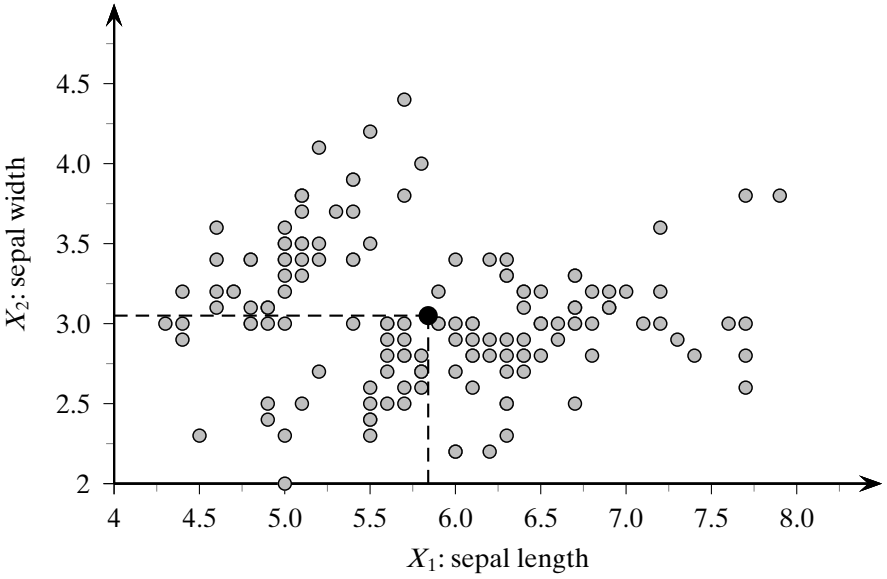

Figure 1.2. Scatterplot: `sepal length` versus `sepal width`. The solid circle shows the mean point.

Each numeric column or attribute can also be treated as a vector in an n-dimensional space $\mathbb{R}^n$:

$$X_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

If all attributes are numeric, then the data matrix $\mathbf{D}$ is in fact an $n \times d$ matrix, also written as $\mathbf{D} \in \mathbb{R}^{n \times d}$, given as

$$\mathbf{D} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix} = \begin{pmatrix} — \mathbf{x}_1^T — \\ — \mathbf{x}_2^T — \\ \vdots \\ — \mathbf{x}_n^T — \end{pmatrix} = \begin{pmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{pmatrix}$$

As we can see, we can consider the entire dataset as an $n \times d$ matrix, or equivalently as a set of n row vectors $\mathbf{x}_i^T \in \mathbb{R}^d$ or as a set of d column vectors $X_j \in \mathbb{R}^n$.

1.3.1 Distance and Angle

Treating data instances and attributes as vectors, and the entire dataset as a matrix, enables one to apply both geometric and algebraic methods to aid in the data mining and analysis tasks.

Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ be two m-dimensional vectors given as

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Dot Product

The *dot product* between $\mathbf{a}$ and $\mathbf{b}$ is defined as the scalar value

$$\begin{aligned} \mathbf{a}^T\mathbf{b} &= \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \\ &= a_1 b_1 + a_2 b_2 + \cdots + a_m b_m \\ &= \sum_{i=1}^{m} a_i b_i \end{aligned}$$

Length

The *Euclidean norm* or *length* of a vector $\mathbf{a} \in \mathbb{R}^m$ is defined as

$$\|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}} = \sqrt{a_1^2 + a_2^2 + \cdots + a_m^2} = \sqrt{\sum_{i=1}^{m} a_i^2}$$

The *unit vector* in the direction of $\mathbf{a}$ is given as

$$\mathbf{u} = \frac{\mathbf{a}}{\|\mathbf{a}\|} = \left(\frac{1}{\|\mathbf{a}\|}\right)\mathbf{a}$$

By definition $\mathbf{u}$ has length $\|\mathbf{u}\| = 1$, and it is also called a *normalized* vector, which can be used in lieu of $\mathbf{a}$ in some analysis tasks.

The Euclidean norm is a special case of a general class of norms, known as L_p*-norm*, defined as

$$\|\mathbf{a}\|_p = \left(|a_1|^p + |a_2|^p + \cdots + |a_m|^p\right)^{\frac{1}{p}} = \left(\sum_{i=1}^{m} |a_i|^p\right)^{\frac{1}{p}}$$

for any $p \neq 0$. Thus, the Euclidean norm corresponds to the case when $p = 2$.

Distance

From the Euclidean norm we can define the *Euclidean distance* between $\mathbf{a}$ and $\mathbf{b}$, as follows

$$\delta(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\| = \sqrt{(\mathbf{a} - \mathbf{b})^T(\mathbf{a} - \mathbf{b})} = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2} \tag{1.1}$$

Thus, the length of a vector is simply its distance from the zero vector $\mathbf{0}$, all of whose elements are 0, that is, $\|\mathbf{a}\| = \|\mathbf{a} - \mathbf{0}\| = \delta(\mathbf{a}, \mathbf{0})$.

From the general L_p-norm we can define the corresponding L_p-distance function, given as follows

$$\delta_p(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_p \tag{1.2}$$

If p is unspecified, as in Eq. (1.1), it is assumed to be $p = 2$ by default.

Angle

The cosine of the smallest angle between vectors $\mathbf{a}$ and $\mathbf{b}$, also called the *cosine similarity*, is given as

$$\cos\theta = \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|\,\|\mathbf{b}\|} = \left(\frac{\mathbf{a}}{\|\mathbf{a}\|}\right)^T \left(\frac{\mathbf{b}}{\|\mathbf{b}\|}\right) \tag{1.3}$$

Thus, the cosine of the angle between $\mathbf{a}$ and $\mathbf{b}$ is given as the dot product of the unit vectors $\frac{\mathbf{a}}{\|\mathbf{a}\|}$ and $\frac{\mathbf{b}}{\|\mathbf{b}\|}$.

The *Cauchy–Schwartz* inequality states that for any vectors $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{R}^m$

$$|\mathbf{a}^T\mathbf{b}| \leq \|\mathbf{a}\| \cdot \|\mathbf{b}\|$$

It follows immediately from the Cauchy–Schwartz inequality that

$$-1 \leq \cos\theta \leq 1$$

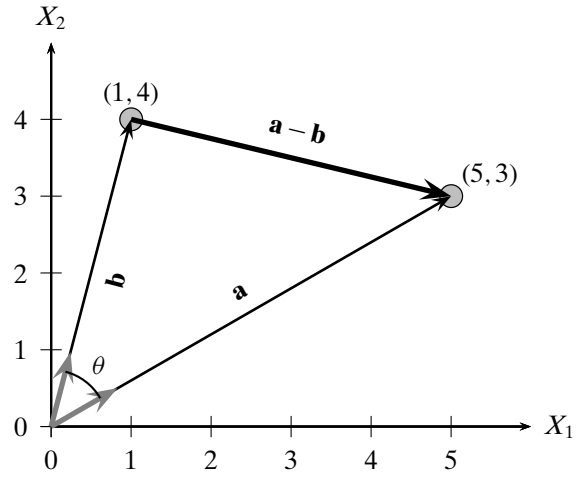

Figure 1.3. Distance and angle. Unit vectors are shown in gray.

Because the smallest angle $\theta \in [0^\circ, 180^\circ]$ and because $\cos\theta \in [-1, 1]$, the cosine similarity value ranges from $+1$, corresponding to an angle of 0°, to -1, corresponding to an angle of 180° (or π radians).

Orthogonality

Two vectors $\mathbf{a}$ and $\mathbf{b}$ are said to be *orthogonal* if and only if $\mathbf{a}^T\mathbf{b} = 0$, which in turn implies that $\cos\theta = 0$, that is, the angle between them is 90° or $\frac{\pi}{2}$ radians. In this case, we say that they have no similarity.

Example 1.3 (Distance and Angle). Figure 1.3 shows the two vectors

$$\mathbf{a} = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

Using Eq. (1.1), the Euclidean distance between them is given as

$$\delta(\mathbf{a}, \mathbf{b}) = \sqrt{(5-1)^2 + (3-4)^2} = \sqrt{16+1} = \sqrt{17} = 4.12$$

The distance can also be computed as the magnitude of the vector:

$$\mathbf{a} - \mathbf{b} = \begin{pmatrix} 5 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$$

because $\|\mathbf{a} - \mathbf{b}\| = \sqrt{4^2 + (-1)^2} = \sqrt{17} = 4.12$.

The unit vector in the direction of $\mathbf{a}$ is given as

$$\mathbf{u}_a = \frac{\mathbf{a}}{\|\mathbf{a}\|} = \frac{1}{\sqrt{5^2+3^2}} \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \frac{1}{\sqrt{34}} \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \begin{pmatrix} 0.86 \\ 0.51 \end{pmatrix}$$

The unit vector in the direction of **b** can be computed similarly:

$$\mathbf{u}_b = \begin{pmatrix} 0.24 \\ 0.97 \end{pmatrix}$$

These unit vectors are also shown in gray in Figure 1.3.

By Eq. (1.3) the cosine of the angle between **a** and **b** is given as

$$\cos\theta = \frac{\begin{pmatrix} 5 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix}}{\sqrt{5^2+3^2}\sqrt{1^2+4^2}} = \frac{17}{\sqrt{34 \times 17}} = \frac{1}{\sqrt{2}}$$

We can get the angle by computing the inverse of the cosine:

$$\theta = \cos^{-1}\left(1/\sqrt{2}\right) = 45^\circ$$

Let us consider the L_p-norm for **a** with $p = 3$; we get

$$\|\mathbf{a}\|_3 = \left(5^3 + 3^3\right)^{1/3} = (152)^{1/3} = 5.34$$

The distance between **a** and **b** using Eq. (1.2) for the L_p-norm with $p = 3$ is given as

$$\|\mathbf{a} - \mathbf{b}\|_3 = \left\|(4,-1)^T\right\|_3 = \left(4^3 + |-1|^3\right)^{1/3} = (65)^{1/3} = 4.02$$

1.3.2 Mean and Total Variance

Mean

The *mean* of the data matrix **D** is the vector obtained as the average of all the points:

$$mean(\mathbf{D}) = \boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

Total Variance

The *total variance* of the data matrix **D** is the average squared distance of each point from the mean:

$$var(\mathbf{D}) = \frac{1}{n}\sum_{i=1}^{n}\delta(\mathbf{x}_i, \boldsymbol{\mu})^2 = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 \tag{1.4}$$

Simplifying Eq. (1.4) we obtain

$$\begin{aligned} var(\mathbf{D}) &= \frac{1}{n}\sum_{i=1}^{n}\left(\|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T\boldsymbol{\mu} + \|\boldsymbol{\mu}\|^2\right) \\ &= \frac{1}{n}\left(\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 - 2n\boldsymbol{\mu}^T\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right) + n\|\boldsymbol{\mu}\|^2\right) \end{aligned}$$

$$= \frac{1}{n}\left(\sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - 2n\boldsymbol{\mu}^T\boldsymbol{\mu} + n\|\boldsymbol{\mu}\|^2\right)$$
$$= \frac{1}{n}\left(\sum_{i=1}^{n} \|\mathbf{x}_i\|^2\right) - \|\boldsymbol{\mu}\|^2$$

The total variance is thus the difference between the average of the squared magnitude of the data points and the squared magnitude of the mean (average of the points).

Centered Data Matrix

Often we need to center the data matrix by making the mean coincide with the origin of the data space. The *centered data matrix* is obtained by subtracting the mean from all the points:

$$\mathbf{Z} = \mathbf{D} - \mathbf{1}\cdot\boldsymbol{\mu}^T = \begin{pmatrix}\mathbf{x}_1^T\\ \mathbf{x}_2^T\\ \vdots\\ \mathbf{x}_n^T\end{pmatrix} - \begin{pmatrix}\boldsymbol{\mu}^T\\ \boldsymbol{\mu}^T\\ \vdots\\ \boldsymbol{\mu}^T\end{pmatrix} = \begin{pmatrix}\mathbf{x}_1^T - \boldsymbol{\mu}^T\\ \mathbf{x}_2^T - \boldsymbol{\mu}^T\\ \vdots\\ \mathbf{x}_n^T - \boldsymbol{\mu}^T\end{pmatrix} = \begin{pmatrix}\mathbf{z}_1^T\\ \mathbf{z}_2^T\\ \vdots\\ \mathbf{z}_n^T\end{pmatrix} \tag{1.5}$$

where $\mathbf{z}_i = \mathbf{x}_i - \boldsymbol{\mu}$ represents the centered point corresponding to $\mathbf{x}_i$, and $\mathbf{1} \in \mathbb{R}^n$ is the n-dimensional vector all of whose elements have value 1. The mean of the centered data matrix $\mathbf{Z}$ is $\mathbf{0} \in \mathbb{R}^d$, because we have subtracted the mean $\boldsymbol{\mu}$ from all the points $\mathbf{x}_i$.

1.3.3 Orthogonal Projection

Often in data mining we need to project a point or vector onto another vector, for example, to obtain a new point after a change of the basis vectors. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ be two m-dimensional vectors. An *orthogonal decomposition* of the vector $\mathbf{b}$ in the direction

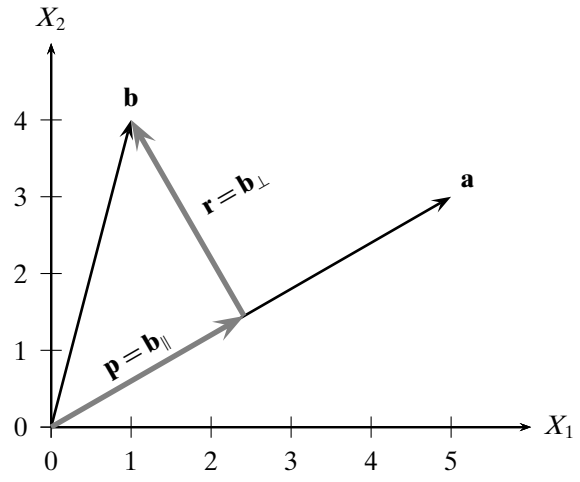

Figure 1.4. Orthogonal projection.

of another vector $\mathbf{a}$, illustrated in Figure 1.4, is given as

$$\mathbf{b} = \mathbf{b}_{\parallel} + \mathbf{b}_{\perp} = \mathbf{p} + \mathbf{r} \tag{1.6}$$

where $\mathbf{p} = \mathbf{b}_{\parallel}$ is parallel to $\mathbf{a}$, and $\mathbf{r} = \mathbf{b}_{\perp}$ is perpendicular or orthogonal to $\mathbf{a}$. The vector $\mathbf{p}$ is called the *orthogonal projection* or simply projection of $\mathbf{b}$ on the vector $\mathbf{a}$. Note that the point $\mathbf{p} \in \mathbb{R}^m$ is the point closest to $\mathbf{b}$ on the line passing through $\mathbf{a}$. Thus, the magnitude of the vector $\mathbf{r} = \mathbf{b} - \mathbf{p}$ gives the *perpendicular distance* between $\mathbf{b}$ and $\mathbf{a}$, which is often interpreted as the residual or error vector between the points $\mathbf{b}$ and $\mathbf{p}$.

We can derive an expression for $\mathbf{p}$ by noting that $\mathbf{p} = c\mathbf{a}$ for some scalar c, as $\mathbf{p}$ is parallel to $\mathbf{a}$. Thus, $\mathbf{r} = \mathbf{b} - \mathbf{p} = \mathbf{b} - c\mathbf{a}$. Because $\mathbf{p}$ and $\mathbf{r}$ are orthogonal, we have

$$\mathbf{p}^T\mathbf{r} = (c\mathbf{a})^T(\mathbf{b} - c\mathbf{a}) = c\mathbf{a}^T\mathbf{b} - c^2\mathbf{a}^T\mathbf{a} = 0$$

which implies that

$$c = \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}$$

Therefore, the projection of $\mathbf{b}$ on $\mathbf{a}$ is given as

$$\mathbf{p} = \mathbf{b}_{\parallel} = c\mathbf{a} = \left(\frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}\right)\mathbf{a} \tag{1.7}$$

Example 1.4. Restricting the Iris dataset to the first two dimensions, `sepal length` and `sepal width`, the mean point is given as

$$mean(\mathbf{D}) = \begin{pmatrix} 5.843 \\ 3.054 \end{pmatrix}$$

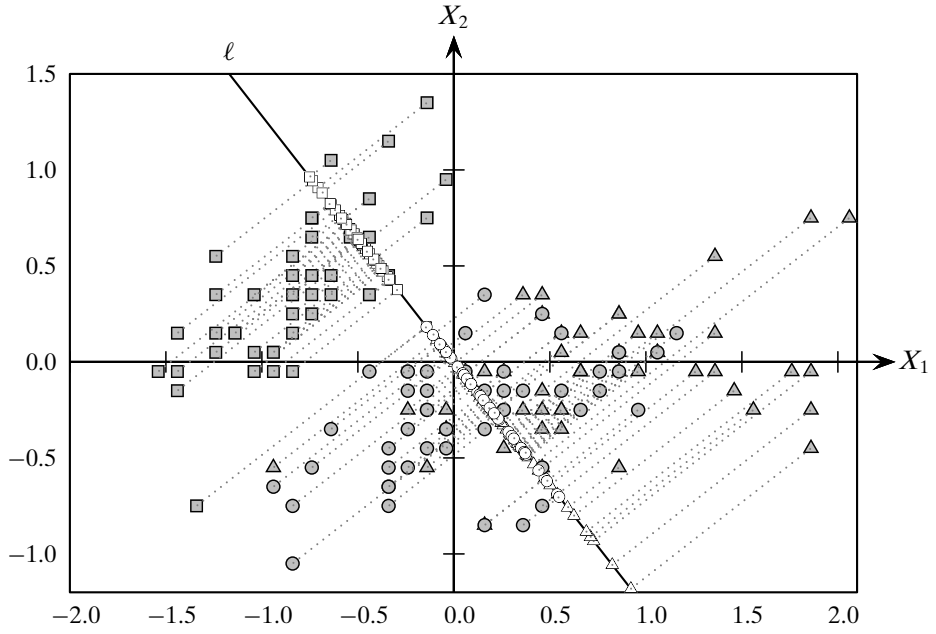

Figure 1.5. Projecting the centered data onto the line ℓ.

which is shown as the black circle in Figure 1.2. The corresponding centered data is shown in Figure 1.5, and the total variance is $var(\mathbf{D}) = 0.868$ (centering does not change this value).

Figure 1.5 shows the projection of each point onto the line ℓ, which is the line that maximizes the separation between the class `iris-setosa` (squares) from the other two classes, namely `iris-versicolor` (circles) and `iris-virginica` (triangles). The line ℓ is given as the set of all the points $(x_1, x_2)^T$ satisfying the constraint $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = c \begin{pmatrix} -2.15 \\ 2.75 \end{pmatrix}$ for all scalars $c \in \mathbb{R}$.

1.3.4 Linear Independence and Dimensionality

Given the data matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{pmatrix}^T = \begin{pmatrix} X_1 & X_2 & \cdots & X_d \end{pmatrix}$$

we are often interested in the linear combinations of the rows (points) or the columns (attributes). For instance, different linear combinations of the original d attributes yield new derived attributes, which play a key role in feature extraction and dimensionality reduction.

Given any set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ in an m-dimensional vector space $\mathbb{R}^m$, their *linear combination* is given as

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_k\mathbf{v}_k$$

where $c_i \in \mathbb{R}$ are scalar values. The set of all possible linear combinations of the k vectors is called the *span*, denoted as $span(\mathbf{v}_1, \ldots, \mathbf{v}_k)$, which is itself a vector space being a *subspace* of $\mathbb{R}^m$. If $span(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \mathbb{R}^m$, then we say that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ is a *spanning set* for $\mathbb{R}^m$.

Row and Column Space

There are several interesting vector spaces associated with the data matrix $\mathbf{D}$, two of which are the column space and row space of $\mathbf{D}$. The *column space* of $\mathbf{D}$, denoted $col(\mathbf{D})$, is the set of all linear combinations of the d attributes $X_j \in \mathbb{R}^n$, that is,

$$col(\mathbf{D}) = span(X_1, X_2, \ldots, X_d)$$

By definition $col(\mathbf{D})$ is a subspace of $\mathbb{R}^n$. The *row space* of $\mathbf{D}$, denoted $row(\mathbf{D})$, is the set of all linear combinations of the n points $\mathbf{x}_i \in \mathbb{R}^d$, that is,

$$row(\mathbf{D}) = span(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$$

By definition $row(\mathbf{D})$ is a subspace of $\mathbb{R}^d$. Note also that the row space of $\mathbf{D}$ is the column space of $\mathbf{D}^T$:

$$row(\mathbf{D}) = col(\mathbf{D}^T)$$

Linear Independence

We say that the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are *linearly dependent* if at least one vector can be written as a linear combination of the others. Alternatively, the k vectors are linearly dependent if there are scalars $c_1, c_2, \ldots, c_k$, at least one of which is not zero, such that

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_k\mathbf{v}_k = \mathbf{0}$$

On the other hand, $\mathbf{v}_1, \cdots, \mathbf{v}_k$ are *linearly independent* if and only if

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_k\mathbf{v}_k = \mathbf{0} \text{ implies } c_1 = c_2 = \cdots = c_k = 0$$

Simply put, a set of vectors is linearly independent if none of them can be written as a linear combination of the other vectors in the set.

Dimension and Rank

Let S be a subspace of $\mathbb{R}^m$. A *basis* for S is a set of vectors in S, say $\mathbf{v}_1, \ldots, \mathbf{v}_k$, that are linearly independent and they span S, that is, $span(\mathbf{v}_1, \ldots, \mathbf{v}_k) = S$. In fact, a basis is a minimal spanning set. If the vectors in the basis are pairwise orthogonal, they are said to form an *orthogonal basis* for S. If, in addition, they are also normalized to be unit vectors, then they make up an *orthonormal basis* for S. For instance, the *standard basis* for $\mathbb{R}^m$ is an orthonormal basis consisting of the vectors

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \qquad \cdots \qquad \mathbf{e}_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Any two bases for S must have the same number of vectors, and the number of vectors in a basis for S is called the *dimension* of S, denoted as $dim(S)$. Because S is a subspace of $\mathbb{R}^m$, we must have $dim(S) \leq m$.

It is a remarkable fact that, for any matrix, the dimension of its row and column space is the same, and this dimension is also called the *rank* of the matrix. For the data matrix $\mathbf{D} \in \mathbb{R}^{n \times d}$, we have $rank(\mathbf{D}) \leq \min(n, d)$, which follows from the fact that the column space can have dimension at most d, and the row space can have dimension at most n. Thus, even though the data points are ostensibly in a d dimensional attribute space (the *extrinsic dimensionality*), if $rank(\mathbf{D}) < d$, then the data points reside in a lower dimensional subspace of $\mathbb{R}^d$, and in this case $rank(\mathbf{D})$ gives an indication about the *intrinsic* dimensionality of the data. In fact, with dimensionality reduction methods it is often possible to approximate $\mathbf{D} \in \mathbb{R}^{n \times d}$ with a derived data matrix $\mathbf{D}' \in \mathbb{R}^{n \times k}$, which has much lower dimensionality, that is, $k \ll d$. In this case k may reflect the "true" intrinsic dimensionality of the data.

Example 1.5. The line ℓ in Figure 1.5 is given as $\ell = span\left(\begin{pmatrix} -2.15 & 2.75 \end{pmatrix}^T\right)$, with $dim(\ell) = 1$. After normalization, we obtain the orthonormal basis for ℓ as the unit vector

$$\frac{1}{\sqrt{12.19}} \begin{pmatrix} -2.15 \\ 2.75 \end{pmatrix} = \begin{pmatrix} -0.615 \\ 0.788 \end{pmatrix}$$

Table 1.2. Iris dataset: `sepal length` (in centimeters).

5.9	6.9	6.6	4.6	6.0	4.7	6.5	5.8	6.7	6.7	5.1	5.1	5.7	6.1	4.9
5.0	5.0	5.7	5.0	7.2	5.9	6.5	5.7	5.5	4.9	5.0	5.5	4.6	7.2	6.8
5.4	5.0	5.7	5.8	5.1	5.6	5.8	5.1	6.3	6.3	5.6	6.1	6.8	7.3	5.6
4.8	7.1	5.7	5.3	5.7	5.7	5.6	4.4	6.3	5.4	6.3	6.9	7.7	6.1	5.6
6.1	6.4	5.0	5.1	5.6	5.4	5.8	4.9	4.6	5.2	7.9	7.7	6.1	5.5	4.6
4.7	4.4	6.2	4.8	6.0	6.2	5.0	6.4	6.3	6.7	5.0	5.9	6.7	5.4	6.3
4.8	4.4	6.4	6.2	6.0	7.4	4.9	7.0	5.5	6.3	6.8	6.1	6.5	6.7	6.7
4.8	4.9	6.9	4.5	4.3	5.2	5.0	6.4	5.2	5.8	5.5	7.6	6.3	6.4	6.3
5.8	5.0	6.7	6.0	5.1	4.8	5.7	5.1	6.6	6.4	5.2	6.4	7.7	5.8	4.9
5.4	5.1	6.0	6.5	5.5	7.2	6.9	6.2	6.5	6.0	5.4	5.5	6.7	7.7	5.1

1.4 DATA: PROBABILISTIC VIEW

The probabilistic view of the data assumes that each numeric attribute X is a *random variable*, defined as a function that assigns a real number to each outcome of an experiment (i.e., some process of observation or measurement). Formally, X is a function $X: \mathcal{O} \to \mathbb{R}$, where $\mathcal{O}$, the domain of X, is the set of all possible outcomes of the experiment, also called the *sample space*, and $\mathbb{R}$, the *range* of X, is the set of real numbers. If the outcomes are numeric, and represent the observed values of the random variable, then $X: \mathcal{O} \to \mathcal{O}$ is simply the identity function: $X(v) = v$ for all $v \in \mathcal{O}$. The distinction between the outcomes and the value of the random variable is important, as we may want to treat the observed values differently depending on the context, as seen in Example 1.6.

A random variable X is called a *discrete random variable* if it takes on only a finite or countably infinite number of values in its range, whereas X is called a *continuous random variable* if it can take on any value in its range.

Example 1.6. Consider the `sepal length` attribute (X_1) for the Iris dataset in Table 1.1. All $n = 150$ values of this attribute are shown in Table 1.2, which lie in the range $[4.3, 7.9]$, with centimeters as the unit of measurement. Let us assume that these constitute the set of all possible outcomes $\mathcal{O}$.

By default, we can consider the attribute X_1 to be a continuous random variable, given as the identity function $X_1(v) = v$, because the outcomes (sepal length values) are all numeric.

On the other hand, if we want to distinguish between Iris flowers with short and long sepal lengths, with long being, say, a length of 7 cm or more, we can define a discrete random variable A as follows:

$$A(v) = \begin{cases} 0 & \text{if } v < 7 \\ 1 & \text{if } v \geq 7 \end{cases}$$

In this case the domain of A is $[4.3, 7.9]$, and its range is $\{0, 1\}$. Thus, A assumes nonzero probability only at the discrete values 0 and 1.

Probability Mass Function

If X is discrete, the *probability mass function* of X is defined as

$$f(x) = P(X = x) \qquad \text{for all } x \in \mathbb{R}$$

In other words, the function f gives the probability $P(X = x)$ that the random variable X has the exact value x. The name "probability mass function" intuitively conveys the fact that the probability is concentrated or massed at only discrete values in the range of X, and is zero for all other values. f must also obey the basic rules of probability. That is, f must be non-negative:

$$f(x) \geq 0$$

and the sum of all probabilities should add to 1:

$$\sum_x f(x) = 1$$

Example 1.7 (Bernoulli and Binomial Distribution). In Example 1.6, A was defined as a discrete random variable representing long sepal length. From the sepal length data in Table 1.2 we find that only 13 Irises have sepal length of at least 7 cm. We can thus estimate the probability mass function of A as follows:

$$f(1) = P(A = 1) = \frac{13}{150} = 0.087 = p$$

and

$$f(0) = P(A = 0) = \frac{137}{150} = 0.913 = 1 - p$$

In this case we say that A has a *Bernoulli distribution* with parameter $p \in [0, 1]$, which denotes the probability of a *success*, that is, the probability of picking an Iris with a long sepal length at random from the set of all points. On the other hand, $1 - p$ is the probability of a *failure*, that is, of not picking an Iris with long sepal length.

Let us consider another discrete random variable B, denoting the number of Irises with long sepal length in m independent Bernoulli trials with probability of success p. In this case, B takes on the discrete values $[0, m]$, and its probability mass function is given by the *Binomial distribution*

$$f(k) = P(B = k) = \binom{m}{k} p^k (1 - p)^{m-k}$$

The formula can be understood as follows. There are $\binom{m}{k}$ ways of picking k long sepal length Irises out of the m trials. For each selection of k long sepal length Irises, the total probability of the k successes is p^k, and the total probability of $m - k$ failures is $(1 - p)^{m-k}$. For example, because $p = 0.087$ from above, the probability of observing exactly $k = 2$ Irises with long sepal length in $m = 10$ trials is given as

$$f(2) = P(B = 2) = \binom{10}{2} (0.087)^2 (0.913)^8 = 0.164$$

Figure 1.6 shows the full probability mass function for different values of k for $m = 10$. Because p is quite small, the probability of k successes in so few a trials falls off rapidly as k increases, becoming practically zero for values of $k \geq 6$.

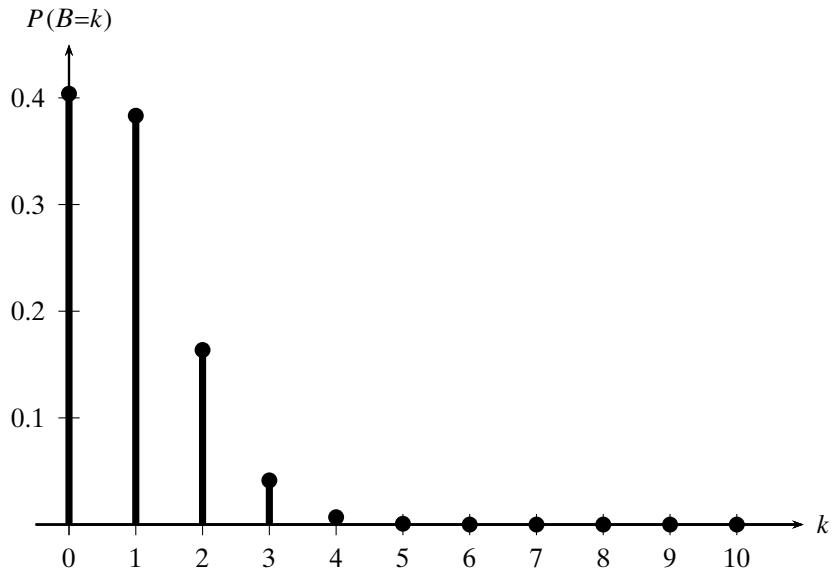

Figure 1.6. Binomial distribution: probability mass function ($m = 10$, $p = 0.087$).

Probability Density Function

If X is continuous, its range is the entire set of real numbers $\mathbb{R}$. The probability of any specific value x is only one out of the infinitely many possible values in the range of X, which means that $P(X = x) = 0$ for all $x \in \mathbb{R}$. However, this does not mean that the value x is impossible, because in that case we would conclude that all values are impossible! What it means is that the probability mass is spread so thinly over the range of values that it can be measured only over intervals $[a, b] \subset \mathbb{R}$, rather than at specific points. Thus, instead of the probability mass function, we define the *probability density function*, which specifies the probability that the variable X takes on values in any interval $[a, b] \subset \mathbb{R}$:

$$P\big(X \in [a, b]\big) = \int_a^b f(x)\, dx$$

As before, the density function f must satisfy the basic laws of probability:

$$f(x) \geq 0, \qquad \text{for all } x \in \mathbb{R}$$

and

$$\int_{-\infty}^{\infty} f(x)\, dx = 1$$

We can get an intuitive understanding of the density function f by considering the probability density over a small interval of width $2\epsilon > 0$, centered at x, namely

$[x-\epsilon, x+\epsilon]$:

$$P\big(X \in [x-\epsilon, x+\epsilon]\big) = \int_{x-\epsilon}^{x+\epsilon} f(x)\, dx \simeq 2\epsilon \cdot f(x)$$

$$f(x) \simeq \frac{P\big(X \in [x-\epsilon, x+\epsilon]\big)}{2\epsilon} \tag{1.8}$$

$f(x)$ thus gives the probability density at x, given as the ratio of the probability mass to the width of the interval, that is, the probability mass per unit distance. Thus, it is important to note that $P(X=x) \neq f(x)$.

Even though the probability density function $f(x)$ does not specify the probability $P(X=x)$, it can be used to obtain the relative probability of one value x_1 over another x_2 because for a given $\epsilon > 0$, by Eq. (1.8), we have

$$\frac{P(X \in [x_1-\epsilon, x_1+\epsilon])}{P(X \in [x_2-\epsilon, x_2+\epsilon])} \simeq \frac{2\epsilon \cdot f(x_1)}{2\epsilon \cdot f(x_2)} = \frac{f(x_1)}{f(x_2)} \tag{1.9}$$

Thus, if $f(x_1)$ is larger than $f(x_2)$, then values of X close to x_1 are more probable than values close to x_2, and vice versa.

Example 1.8 (Normal Distribution). Consider again the `sepal length` values from the Iris dataset, as shown in Table 1.2. Let us assume that these values follow a *Gaussian* or *normal* density function, given as

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

There are two parameters of the normal density distribution, namely, μ, which represents the mean value, and σ^2, which represents the variance of the values (these parameters are discussed in Chapter 2). Figure 1.7 shows the characteristic "bell" shape plot of the normal distribution. The parameters, $\mu = 5.84$ and $\sigma^2 = 0.681$, were estimated directly from the data for `sepal length` in Table 1.2.

Whereas $f(x=\mu) = f(5.84) = \frac{1}{\sqrt{2\pi \cdot 0.681}} \exp\{0\} = 0.483$, we emphasize that the probability of observing $X = \mu$ is zero, that is, $P(X=\mu) = 0$. Thus, $P(X=x)$ is not given by $f(x)$, rather, $P(X=x)$ is given as the area under the curve for an infinitesimally small interval $[x-\epsilon, x+\epsilon]$ centered at x, with $\epsilon > 0$. Figure 1.7 illustrates this with the shaded region centered at $\mu = 5.84$. From Eq. (1.8), we have

$$P(X=\mu) \simeq 2\epsilon \cdot f(\mu) = 2\epsilon \cdot 0.483 = 0.967\epsilon$$

As $\epsilon \to 0$, we get $P(X=\mu) \to 0$. However, based on Eq. (1.9) we can claim that the probability of observing values close to the mean value $\mu = 5.84$ is 2.69 times the probability of observing values close to $x = 7$, as

$$\frac{f(5.84)}{f(7)} = \frac{0.483}{0.18} = 2.69$$

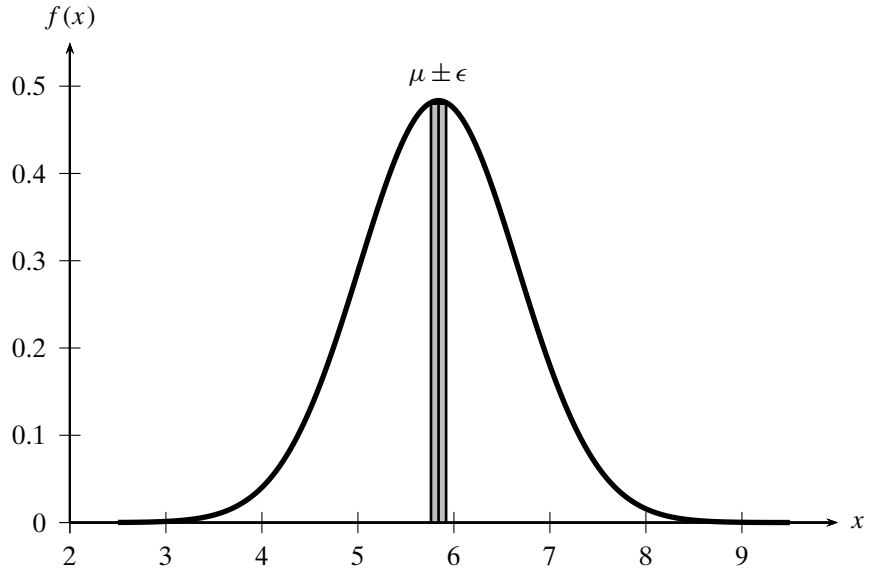

Figure 1.7. Normal distribution: probability density function ($\mu = 5.84$, $\sigma^2 = 0.681$).

Cumulative Distribution Function

For any random variable X, whether discrete or continuous, we can define the *cumulative distribution function (CDF)* $F : \mathbb{R} \to [0, 1]$, which gives the probability of observing a value at most some given value x:

$$F(x) = P(X \leq x) \qquad \text{for all } -\infty < x < \infty$$

When X is discrete, F is given as

$$F(x) = P(X \leq x) = \sum_{u \leq x} f(u)$$

and when X is continuous, F is given as

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)\, du$$

Example 1.9 (Cumulative Distribution Function). Figure 1.8 shows the cumulative distribution function for the binomial distribution in Figure 1.6. It has the characteristic step shape (right continuous, non-decreasing), as expected for a discrete random variable. $F(x)$ has the same value $F(k)$ for all $x \in [k, k+1)$ with $0 \leq k < m$, where m is the number of trials and k is the number of successes. The closed (filled) and open circles demarcate the corresponding closed and open interval $[k, k+1)$. For instance, $F(x) = 0.404 = F(0)$ for all $x \in [0, 1)$.

Figure 1.9 shows the cumulative distribution function for the normal density function shown in Figure 1.7. As expected, for a continuous random variable, the CDF is also continuous, and non-decreasing. Because the normal distribution is symmetric about the mean, we have $F(\mu) = P(X \leq \mu) = 0.5$.

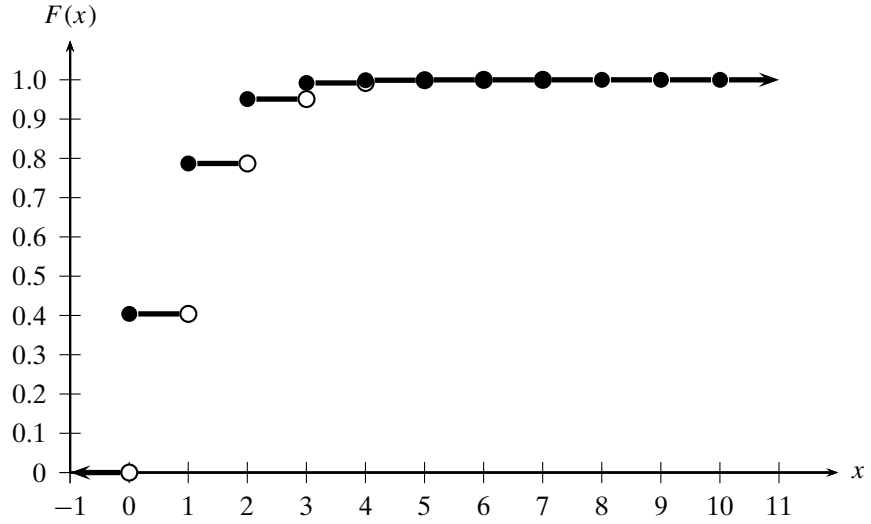

Figure 1.8. Cumulative distribution function for the binomial distribution.

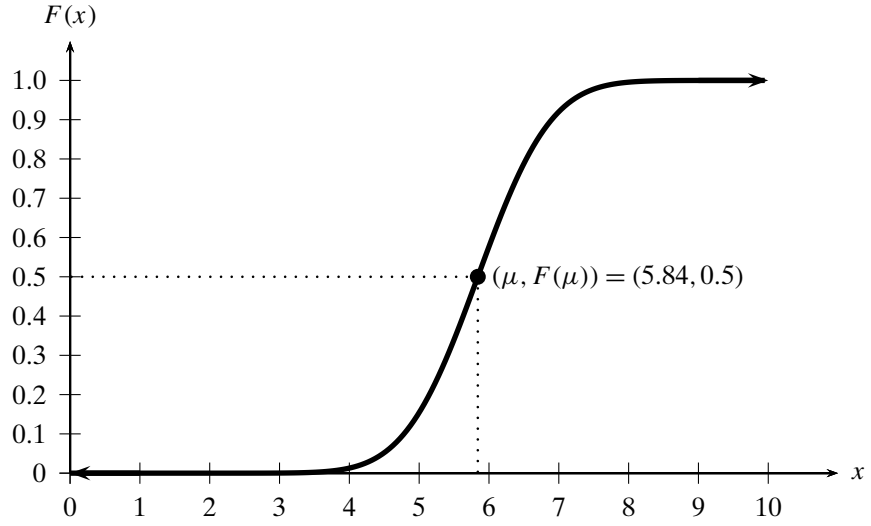

Figure 1.9. Cumulative distribution function for the normal distribution.

1.4.1 Bivariate Random Variables

Instead of considering each attribute as a random variable, we can also perform pair-wise analysis by considering a pair of attributes, X_1 and X_2, as a *bivariate random variable*:

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

$\mathbf{X} : \mathcal{O} \rightarrow \mathbb{R}^2$ is a function that assigns to each outcome in the sample space, a pair of real numbers, that is, a 2-dimensional vector $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$. As in the univariate case,

if the outcomes are numeric, then the default is to assume $\mathbf{X}$ to be the identity function.

Joint Probability Mass Function

If X_1 and X_2 are both discrete random variables then $\mathbf{X}$ has a *joint probability mass function* given as follows:

$$f(\mathbf{x}) = f(x_1, x_2) = P(X_1 = x_1, X_2 = x_2) = P(\mathbf{X} = \mathbf{x})$$

f must satisfy the following two conditions:

$$f(\mathbf{x}) = f(x_1, x_2) \geq 0 \qquad \text{for all } -\infty < x_1, x_2 < \infty$$

$$\sum_{\mathbf{x}} f(\mathbf{x}) = \sum_{x_1} \sum_{x_2} f(x_1, x_2) = 1$$

Joint Probability Density Function

If X_1 and X_2 are both continuous random variables then $\mathbf{X}$ has a *joint probability density function* f given as follows:

$$P(\mathbf{X} \in W) = \int_{\mathbf{x} \in W} \int f(\mathbf{x})\, d\mathbf{x} = \int_{(x_1, x_2)^T \in W} \int f(x_1, x_2)\, dx_1\, dx_2$$

where $W \subset \mathbb{R}^2$ is some subset of the 2-dimensional space of reals. f must also satisfy the following two conditions:

$$f(\mathbf{x}) = f(x_1, x_2) \geq 0 \qquad \text{for all } -\infty < x_1, x_2 < \infty$$

$$\int_{\mathbb{R}^2} f(\mathbf{x})\, d\mathbf{x} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2)\, dx_1\, dx_2 = 1$$

As in the univariate case, the probability mass $P(\mathbf{x}) = P\big((x_1, x_2)^T\big) = 0$ for any particular point $\mathbf{x}$. However, we can use f to compute the probability density at $\mathbf{x}$. Consider the square region $W = \big([x_1 - \epsilon, x_1 + \epsilon], [x_2 - \epsilon, x_2 + \epsilon]\big)$, that is, a 2-dimensional window of width 2ϵ centered at $\mathbf{x} = (x_1, x_2)^T$. The probability density at $\mathbf{x}$ can be approximated as

$$\begin{aligned} P(\mathbf{X} \in W) &= P\Big(\mathbf{X} \in \big([x_1 - \epsilon, x_1 + \epsilon], [x_2 - \epsilon, x_2 + \epsilon]\big)\Big) \\ &= \int_{x_1 - \epsilon}^{x_1 + \epsilon} \int_{x_2 - \epsilon}^{x_2 + \epsilon} f(x_1, x_2)\, dx_1\, dx_2 \\ &\simeq 2\epsilon \cdot 2\epsilon \cdot f(x_1, x_2) \end{aligned}$$

which implies that

$$f(x_1, x_2) = \frac{P(\mathbf{X} \in W)}{(2\epsilon)^2}$$

The relative probability of one value (a_1, a_2) versus another (b_1, b_2) can therefore be computed via the probability density function:

$$\frac{P(\mathbf{X} \in \big([a_1 - \epsilon, a_1 + \epsilon], [a_2 - \epsilon, a_2 + \epsilon]\big))}{P(\mathbf{X} \in \big([b_1 - \epsilon, b_1 + \epsilon], [b_2 - \epsilon, b_2 + \epsilon]\big))} \simeq \frac{(2\epsilon)^2 \cdot f(a_1, a_2)}{(2\epsilon)^2 \cdot f(b_1, b_2)} = \frac{f(a_1, a_2)}{f(b_1, b_2)}$$

Example 1.10 (Bivariate Distributions). Consider the `sepal length` and `sepal width` attributes in the Iris dataset, plotted in Figure 1.2. Let A denote the Bernoulli random variable corresponding to long sepal length (at least 7 cm), as defined in Example 1.7.

Define another Bernoulli random variable B corresponding to long sepal width, say, at least 3.5 cm. Let $\mathbf{X} = \begin{pmatrix} A \\ B \end{pmatrix}$ be a discrete bivariate random variable; then the joint probability mass function of $\mathbf{X}$ can be estimated from the data as follows:

$$f(0,0) = P(A=0, B=0) = \frac{116}{150} = 0.773$$
$$f(0,1) = P(A=0, B=1) = \frac{21}{150} = 0.140$$
$$f(1,0) = P(A=1, B=0) = \frac{10}{150} = 0.067$$
$$f(1,1) = P(A=1, B=1) = \frac{3}{150} = 0.020$$

Figure 1.10 shows a plot of this probability mass function.

Treating attributes X_1 and X_2 in the Iris dataset (see Table 1.1) as continuous random variables, we can define a continuous bivariate random variable $\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$. Assuming that $\mathbf{X}$ follows a *bivariate normal distribution*, its joint probability density function is given as

$$f(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} \exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}{2}\right\}$$

Here $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the parameters of the bivariate normal distribution, representing the 2-dimensional mean vector and covariance matrix, which are discussed in detail

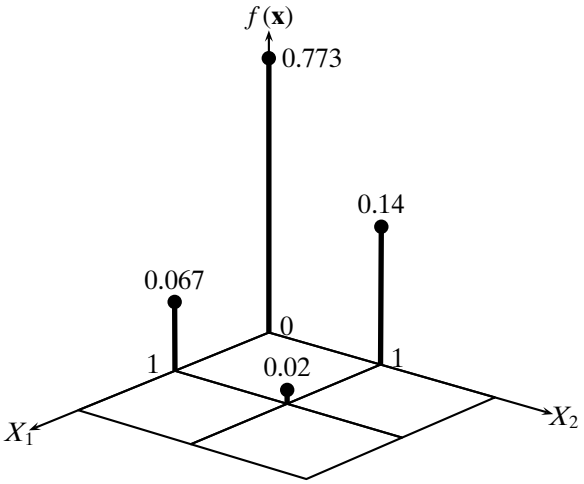

Figure 1.10. Joint probability mass function: X_1 (long sepal length), X_2 (long sepal width).

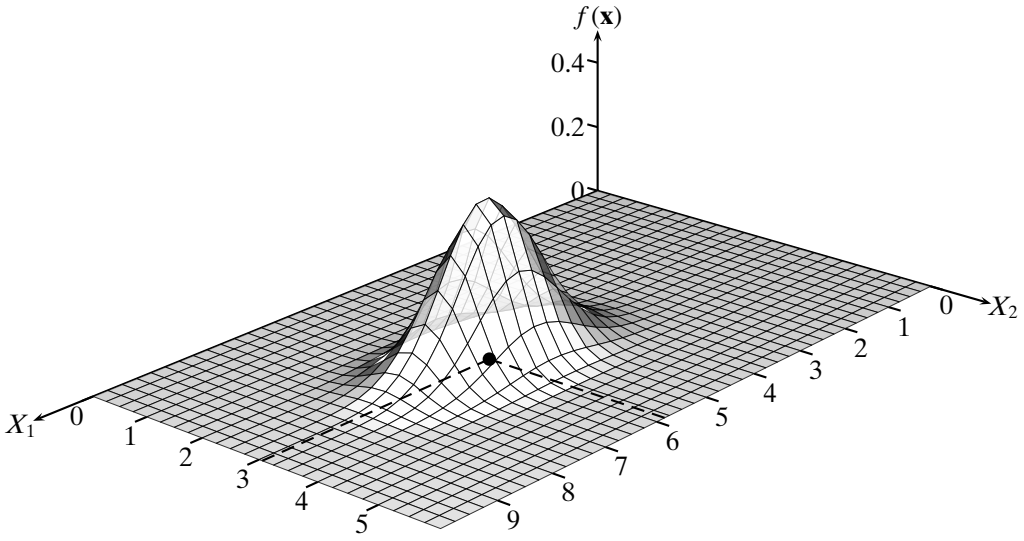

Figure 1.11. Bivariate normal density: $\boldsymbol{\mu} = (5.843, 3.054)^T$ (solid circle).

in Chapter 2. Further, $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$. The plot of the bivariate normal density is given in Figure 1.11, with mean

$$\boldsymbol{\mu} = (5.843, 3.054)^T$$

and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.681 & -0.039 \\ -0.039 & 0.187 \end{pmatrix}$$

It is important to emphasize that the function $f(\mathbf{x})$ specifies only the probability density at $\mathbf{x}$, and $f(\mathbf{x}) \neq P(\mathbf{X} = \mathbf{x})$. As before, we have $P(\mathbf{X} = \mathbf{x}) = 0$.

Joint Cumulative Distribution Function

The *joint cumulative distribution function* for two random variables X_1 and X_2 is defined as the function F, such that for all values $x_1, x_2 \in (-\infty, \infty)$,

$$F(\mathbf{x}) = F(x_1, x_2) = P(X_1 \leq x_1 \text{ and } X_2 \leq x_2) = P(\mathbf{X} \leq \mathbf{x})$$

Statistical Independence

Two random variables X_1 and X_2 are said to be (statistically) *independent* if, for every $W_1 \subset \mathbb{R}$ and $W_2 \subset \mathbb{R}$, we have

$$P(X_1 \in W_1 \text{ and } X_2 \in W_2) = P(X_1 \in W_1) \cdot P(X_2 \in W_2)$$

Furthermore, if X_1 and X_2 are independent, then the following two conditions are also satisfied:

$$F(\mathbf{x}) = F(x_1, x_2) = F_1(x_1) \cdot F_2(x_2)$$

$$f(\mathbf{x}) = f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$$

where F_i is the cumulative distribution function, and f_i is the probability mass or density function for random variable X_i.

1.4.2 Multivariate Random Variable

A d-dimensional *multivariate random variable* $\mathbf{X} = (X_1, X_2, \ldots, X_d)^T$, also called a *vector random variable*, is defined as a function that assigns a vector of real numbers to each outcome in the sample space, that is, $\mathbf{X} : \mathcal{O} \to \mathbb{R}^d$. The range of $\mathbf{X}$ can be denoted as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$. In case all X_j are numeric, then $\mathbf{X}$ is by default assumed to be the identity function. In other words, if all attributes are numeric, we can treat each outcome in the sample space (i.e., each point in the data matrix) as a vector random variable. On the other hand, if the attributes are not all numeric, then $\mathbf{X}$ maps the outcomes to numeric vectors in its range.

If all X_j are discrete, then $\mathbf{X}$ is jointly discrete and its joint probability mass function f is given as

$$f(\mathbf{x}) = P(\mathbf{X} = \mathbf{x})$$
$$f(x_1, x_2, \ldots, x_d) = P(X_1 = x_1, X_2 = x_2, \ldots, X_d = x_d)$$

If all X_j are continuous, then $\mathbf{X}$ is jointly continuous and its joint probability density function is given as

$$P(\mathbf{X} \in W) = \int_{\mathbf{x} \in W} \cdots \int f(\mathbf{x})\, d\mathbf{x}$$
$$P\left((X_1, X_2, \ldots, X_d)^T \in W\right) = \int_{(x_1, x_2, \ldots, x_d)^T \in W} \cdots \int f(x_1, x_2, \ldots, x_d)\, dx_1\, dx_2 \ldots dx_d$$

for any d-dimensional region $W \subseteq \mathbb{R}^d$.

The laws of probability must be obeyed as usual, that is, $f(\mathbf{x}) \geq 0$ and sum of f over all $\mathbf{x}$ in the range of $\mathbf{X}$ must be 1. The joint cumulative distribution function of $\mathbf{X} = (X_1, \ldots, X_d)^T$ is given as

$$F(\mathbf{x}) = P(\mathbf{X} \leq \mathbf{x})$$
$$F(x_1, x_2, \ldots, x_d) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_d \leq x_d)$$

for every point $\mathbf{x} \in \mathbb{R}^d$.

We say that $X_1, X_2, \ldots, X_d$ are independent random variables if and only if, for every region $W_i \subset \mathbb{R}$, we have

$$P(X_1 \in W_1 \text{ and } X_2 \in W_2 \cdots \text{ and } X_d \in W_d) = P(X_1 \in W_1) \cdot P(X_2 \in W_2) \cdot \cdots \cdot P(X_d \in W_d) \quad (1.10)$$

If $X_1, X_2, \ldots, X_d$ are independent then the following conditions are also satisfied

$$F(\mathbf{x}) = F(x_1, \ldots, x_d) = F_1(x_1) \cdot F_2(x_2) \cdot \ldots \cdot F_d(x_d)$$
$$f(\mathbf{x}) = f(x_1, \ldots, x_d) = f_1(x_1) \cdot f_2(x_2) \cdot \ldots \cdot f_d(x_d) \quad (1.11)$$

where F_i is the cumulative distribution function, and f_i is the probability mass or density function for random variable X_i.

1.4.3 Random Sample and Statistics

The probability mass or density function of a random variable X may follow some known form, or as is often the case in data analysis, it may be unknown. When the probability function is not known, it may still be convenient to assume that the values follow some known distribution, based on the characteristics of the data. However, even in this case, the parameters of the distribution may still be unknown. Thus, in general, either the parameters, or the entire distribution, may have to be estimated from the data.

In statistics, the word *population* is used to refer to the set or universe of all entities under study. Usually we are interested in certain characteristics or parameters of the entire population (e.g., the mean age of all computer science students in the United States). However, looking at the entire population may not be feasible or may be too expensive. Instead, we try to make inferences about the population parameters by drawing a random sample from the population, and by computing appropriate *statistics* from the sample that give estimates of the corresponding population parameters of interest.

Univariate Sample

Given a random variable X, a *random sample* of size n from X is defined as a set of n *independent and identically distributed (IID)* random variables $S_1, S_2, \ldots, S_n$, that is, all of the S_i's are statistically independent of each other, and follow the same probability mass or density function as X.

If we treat attribute X as a random variable, then each of the observed values of X, namely, x_i $(1 \leq i \leq n)$, are themselves treated as identity random variables, and the observed data is assumed to be a random sample drawn from X. That is, all x_i are considered to be mutually independent and identically distributed as X. By Eq. (1.11) their joint probability function is given as

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_X(x_i)$$

where f_X is the probability mass or density function for X.

Multivariate Sample

For multivariate parameter estimation, the n data points $\mathbf{x}_i$ (with $1 \leq i \leq n$) constitute a d-dimensional multivariate random sample drawn from the vector random variable $\mathbf{X} = (X_1, X_2, \ldots, X_d)$. That is, $\mathbf{x}_i$ are assumed to be independent and identically distributed, and thus their joint distribution is given as

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \prod_{i=1}^{n} f_{\mathbf{X}}(\mathbf{x}_i) \tag{1.12}$$

where $f_{\mathbf{X}}$ is the probability mass or density function for $\mathbf{X}$.

Estimating the parameters of a multivariate joint probability distribution is usually difficult and computationally intensive. One simplifying assumption that is typically made is that the d attributes $X_1, X_2, \ldots, X_d$ are statistically independent. However, we do not assume that they are identically distributed, because that is almost never justified. Under the attribute independence assumption Eq. (1.12) can be rewritten as

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \prod_{i=1}^{n} f(\mathbf{x}_i) = \prod_{i=1}^{n} \prod_{j=1}^{d} f_{X_j}(x_{ij})$$

Statistic

We can estimate a parameter of the population by defining an appropriate sample *statistic*, which is defined as a function of the sample. More precisely, let $\{\mathbf{S}_i\}_{i=1}^{m}$ denote the random sample of size m drawn from a (multivariate) random variable $\mathbf{X}$. A statistic $\hat{\theta}$ is a function $\hat{\theta} : (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m) \to \mathbb{R}$. The statistic is an estimate of the corresponding population parameter θ. As such, the statistic $\hat{\theta}$ is itself a random variable. If we use the value of a statistic to estimate a population parameter, this value is called a *point estimate* of the parameter, and the statistic is called an *estimator* of the parameter. In Chapter 2 we will study different estimators for population parameters that reflect the location (or centrality) and dispersion of values.

Example 1.11 (Sample Mean). Consider attribute `sepal length` (X_1) in the Iris dataset, whose values are shown in Table 1.2. Assume that the mean value of X_1 is not known. Let us assume that the observed values $\{x_i\}_{i=1}^{n}$ constitute a random sample drawn from X_1.

The *sample mean* is a statistic, defined as the average

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Plugging in values from Table 1.2, we obtain

$$\hat{\mu} = \frac{1}{150}(5.9 + 6.9 + \cdots + 7.7 + 5.1) = \frac{876.5}{150} = 5.84$$

The value $\hat{\mu} = 5.84$ is a point estimate for the unknown population parameter μ, the (true) mean value of variable X_1.

1.5 DATA MINING

Data mining comprises the core algorithms that enable one to gain fundamental insights and knowledge from massive data. It is an interdisciplinary field merging concepts from allied areas such as database systems, statistics, machine learning, and pattern recognition. In fact, data mining is part of a larger knowledge discovery process, which includes pre-processing tasks such as data extraction, data cleaning, data fusion, data reduction and feature construction, as well as post-processing steps

such as pattern and model interpretation, hypothesis confirmation and generation, and so on. This knowledge discovery and data mining process tends to be highly iterative and interactive.

The algebraic, geometric, and probabilistic viewpoints of data play a key role in data mining. Given a dataset of n points in a d-dimensional space, the fundamental analysis and mining tasks covered in this book include exploratory data analysis, frequent pattern discovery, data clustering, and classification models, which are described next.

1.5.1 Exploratory Data Analysis

Exploratory data analysis aims to explore the numeric and categorical attributes of the data individually or jointly to extract key characteristics of the data sample via statistics that give information about the centrality, dispersion, and so on. Moving away from the IID assumption among the data points, it is also important to consider the statistics that deal with the data as a graph, where the nodes denote the points and weighted edges denote the connections between points. This enables one to extract important topological attributes that give insights into the structure and models of networks and graphs. Kernel methods provide a fundamental connection between the independent pointwise view of data, and the viewpoint that deals with pairwise similarities between points. Many of the exploratory data analysis and mining tasks can be cast as kernel problems via the *kernel trick*, that is, by showing that the operations involve only dot-products between pairs of points. However, kernel methods also enable us to perform nonlinear analysis by using familiar linear algebraic and statistical methods in high-dimensional spaces comprising "nonlinear" dimensions. They further allow us to mine complex data as long as we have a way to measure the pairwise similarity between two abstract objects. Given that data mining deals with massive datasets with thousands of attributes and millions of points, another goal of exploratory analysis is to reduce the amount of data to be mined. For instance, feature selection and dimensionality reduction methods are used to select the most important dimensions, discretization methods can be used to reduce the number of values of an attribute, data sampling methods can be used to reduce the data size, and so on.

Part I of this book begins with basic statistical analysis of univariate and multivariate numeric data in Chapter 2. We describe measures of central tendency such as mean, median, and mode, and then we consider measures of dispersion such as range, variance, and covariance. We emphasize the dual algebraic and probabilistic views, and highlight the geometric interpretation of the various measures. We especially focus on the multivariate normal distribution, which is widely used as the default parametric model for data in both classification and clustering. In Chapter 3 we show how categorical data can be modeled via the multivariate binomial and the multinomial distributions. We describe the contingency table analysis approach to test for dependence between categorical attributes. Next, in Chapter 4 we show how to analyze graph data in terms of the topological structure, with special focus on various graph centrality measures such as closeness, betweenness, prestige, PageRank, and so on. We also study basic topological properties of real-world networks such as the *small*

world property, which states that real graphs have small average path length between pairs of nodes, the *clustering effect*, which indicates local clustering around nodes, and the *scale-free property*, which manifests itself in a *power-law* degree distribution. We describe models that can explain some of these characteristics of real-world graphs; these include the Erdös–Rényi random graph model, the Watts–Strogatz model, and the Barabási–Albert model. Kernel methods are then introduced in Chapter 5, which provide new insights and connections between linear, nonlinear, graph, and complex data mining tasks. We briefly highlight the theory behind kernel functions, with the key concept being that a positive semidefinite kernel corresponds to a dot product in some high-dimensional feature space, and thus we can use familiar numeric analysis methods for nonlinear or complex object analysis provided we can compute the pairwise kernel matrix of similarities between object instances. We describe various kernels for numeric or vector data, as well as sequence and graph data. In Chapter 6 we consider the peculiarities of high-dimensional space, colorfully referred to as *the curse of dimensionality*. In particular, we study the scattering effect, that is, the fact that data points lie along the surface and corners in high dimensions, with the "center" of the space being virtually empty. We show the proliferation of orthogonal axes and also the behavior of the multivariate normal distribution in high dimensions. Finally, in Chapter 7 we describe the widely used dimensionality reduction methods such as principal component analysis (PCA) and singular value decomposition (SVD). PCA finds the optimal k-dimensional subspace that captures most of the variance in the data. We also show how kernel PCA can be used to find nonlinear directions that capture the most variance. We conclude with the powerful SVD spectral decomposition method, studying its geometry, and its relationship to PCA.

1.5.2 Frequent Pattern Mining

Frequent pattern mining refers to the task of extracting informative and useful patterns in massive and complex datasets. Patterns comprise sets of co-occurring attribute values, called *itemsets*, or more complex patterns, such as sequences, which consider explicit precedence relationships (either positional or temporal), and graphs, which consider arbitrary relationships between points. The key goal is to discover hidden trends and behaviors in the data to understand better the interactions among the points and attributes.

Part II begins by presenting efficient algorithms for frequent itemset mining in Chapter 8. The key methods include the level-wise Apriori algorithm, the "vertical" intersection based Eclat algorithm, and the frequent pattern tree and projection based FPGrowth method. Typically the mining process results in too many frequent patterns that can be hard to interpret. In Chapter 9 we consider approaches to summarize the mined patterns; these include maximal (GenMax algorithm), closed (Charm algorithm), and non-derivable itemsets. We describe effective methods for frequent sequence mining in Chapter 10, which include the level-wise GSP method, the vertical SPADE algorithm, and the projection-based PrefixSpan approach. We also describe how consecutive subsequences, also called substrings, can be mined much more efficiently via Ukkonen's linear time and space suffix tree method. Moving

beyond sequences to arbitrary graphs, we describe the popular and efficient gSpan algorithm for frequent subgraph mining in Chapter 11. Graph mining involves two key steps, namely graph isomorphism checks to eliminate duplicate patterns during pattern enumeration and subgraph isomorphism checks during frequency computation. These operations can be performed in polynomial time for sets and sequences, but for graphs it is known that subgraph isomorphism is NP-hard, and thus there is no polynomial time method possible unless P = NP. The gSpan method proposes a new canonical code and a systematic approach to subgraph extension, which allow it to efficiently detect duplicates and to perform several subgraph isomorphism checks much more efficiently than performing them individually. Given that pattern mining methods generate many output results it is very important to assess the mined patterns. We discuss strategies for assessing both the frequent patterns and rules that can be mined from them in Chapter 12, emphasizing methods for significance testing.

1.5.3 Clustering

Clustering is the task of partitioning the points into *natural groups* called clusters, such that points within a group are very similar, whereas points across clusters are as dissimilar as possible. Depending on the data and desired cluster characteristics, there are different types of clustering paradigms such as representative-based, hierarchical, density-based, graph-based, and spectral clustering.

Part III starts with representative-based clustering methods (Chapter 13), which include the K-means and Expectation-Maximization (EM) algorithms. K-means is a greedy algorithm that minimizes the squared error of points from their respective cluster means, and it performs hard clustering, that is, each point is assigned to only one cluster. We also show how kernel K-means can be used for nonlinear clusters. EM generalizes K-means by modeling the data as a mixture of normal distributions, and it finds the cluster parameters (the mean and covariance matrix) by maximizing the likelihood of the data. It is a soft clustering approach, that is, instead of making a hard assignment, it returns the probability that a point belongs to each cluster. In Chapter 14 we consider various agglomerative hierarchical clustering methods, which start from each point in its own cluster, and successively merge (or agglomerate) pairs of clusters until the desired number of clusters have been found. We consider various cluster proximity measures that distinguish the different hierarchical methods. There are some datasets where the points from different clusters may in fact be closer in distance than points from the same cluster; this usually happens when the clusters are nonconvex in shape. Density-based clustering methods described in Chapter 15 use the density or connectedness properties to find such nonconvex clusters. The two main methods are DBSCAN and its generalization DENCLUE, which is based on kernel density estimation. We consider graph clustering methods in Chapter 16, which are typically based on spectral analysis of graph data. Graph clustering can be considered as an optimization problem over a k-way cut in a graph; different objectives can be cast as spectral decomposition of different graph matrices, such as the (normalized) adjacency matrix, Laplacian matrix, and so on, derived from the original graph data or from the kernel matrix. Finally, given the proliferation of different types of clustering methods,

it is important to assess the mined clusters as to how good they are in capturing the natural groups in data. In Chapter 17, we describe various clustering validation and evaluation strategies, spanning external and internal measures to compare a clustering with the ground-truth if it is available, or to compare two clusterings. We also highlight methods for clustering stability, that is, the sensitivity of the clustering to data perturbation, and clustering tendency, that is, the clusterability of the data. We also consider methods to choose the parameter k, which is the user-specified value for the number of clusters to discover.

1.5.4 Classification

The classification task is to predict the label or class for a given unlabeled point. Formally, a classifier is a model or function M that predicts the class label $\hat{y}$ for a given input example $\mathbf{x}$, that is, $\hat{y} = M(\mathbf{x})$, where $\hat{y} \in \{c_1, c_2, \ldots, c_k\}$ and each c_i is a class label (a categorical attribute value). To build the model we require a set of points with their correct class labels, which is called a *training set*. After learning the model M, we can automatically predict the class for any new point. Many different types of classification models have been proposed such as decision trees, probabilistic classifiers, support vector machines, and so on.

Part IV starts with the powerful Bayes classifier, which is an example of the probabilistic classification approach (Chapter 18). It uses the Bayes theorem to predict the class as the one that maximizes the posterior probability $P(c_i|\mathbf{x})$. The main task is to estimate the joint probability density function $f(\mathbf{x})$ for each class, which is modeled via a multivariate normal distribution. One limitation of the Bayes approach is the number of parameters to be estimated which scales as $O(d^2)$. The naive Bayes classifier makes the simplifying assumption that all attributes are independent, which requires the estimation of only $O(d)$ parameters. It is, however, surprisingly effective for many datasets. In Chapter 19 we consider the popular decision tree classifier, one of whose strengths is that it yields models that are easier to understand compared to other methods. A decision tree recursively partitions the data space into "pure" regions that contain data points from only one class, with relatively few exceptions. Next, in Chapter 20, we consider the task of finding an optimal direction that separates the points from two classes via linear discriminant analysis. It can be considered as a dimensionality reduction method that also takes the class labels into account, unlike PCA, which does not consider the class attribute. We also describe the generalization of linear to kernel discriminant analysis, which allows us to find nonlinear directions via the kernel trick. In Chapter 21 we describe the support vector machine (SVM) approach in detail, which is one of the most effective classifiers for many different problem domains. The goal of SVMs is to find the optimal hyperplane that maximizes the *margin* between the classes. Via the kernel trick, SVMs can be used to find nonlinear boundaries, which nevertheless correspond to some linear hyperplane in some high-dimensional "nonlinear" space. One of the important tasks in classification is to assess how good the models are. We conclude this part with Chapter 22, which presents the various methodologies for assessing classification models. We define various classification performance measures including ROC analysis. We then describe the bootstrap and cross-validation approaches for classifier evaluation. Finally, we

discuss the bias–variance tradeoff in classification, and how ensemble classifiers can help improve the variance or the bias of a classifier.

1.6 FURTHER READING

For a review of the linear algebra concepts see Strang (2006) and Poole (2010), and for the probabilistic view see Evans and Rosenthal (2011). There are several good books on data mining, and machine and statistical learning; these include Hand, Mannila, and Smyth (2001), Han, Kamber, and Pei (2006), Witten, Frank, and Hall (2011), Tan, Steinbach, and Kumar (2013), and Bishop (2006) and Hastie, Tibshirani, and Friedman (2009).

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer Science + Business Media.

Evans, M. and Rosenthal, J. (2011). *Probability and Statistics: The Science of Uncertainty*. 2nd ed. New York: W. H. Freeman.

Han, J., Kamber, M., and Pei, J. (2006). *Data Mining: Concepts and Techniques*. 2nd ed. The Morgan Kaufmann Series in Data Management Systems. Philadelphia: Elsevier Science.

Hand, D., Mannila, H., and Smyth, P. (2001). *Principles of data mining*. Adaptative computation and machine learning series. Cambridge, MA: MIT Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*. 2nd ed. Springer series in statistics. New York: Springer Science + Business Media.

Poole, D. (2010). *Linear Algebra: A Modern Introduction*. 3rd ed. Independence, KY: Cengage Learning.

Strang, G. (2006). *Linear Algebra and Its Applications*. 4th ed. Independence, KY: Thomson Brooks/Cole, Cengage learning.

Tan, P., Steinbach, M., and Kumar, V. (2013). *Introduction to Data Mining*. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

Witten, I., Frank, E., and Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. 3rd ed. The Morgan Kaufmann Series in Data Management Systems. Philadelphia: Elsevier Science.

1.7 EXERCISES

Q1. Show that the mean of the centered data matrix $\mathbf{Z}$ in Eq. (1.5) is $\mathbf{0}$.

Q2. Prove that for the L_p-distance in Eq. (1.2), we have

$$\delta_\infty(\mathbf{x}, \mathbf{y}) = \lim_{p \to \infty} \delta_p(\mathbf{x}, \mathbf{y}) = \max_{i=1}^{d} \{|x_i - y_i|\}$$

for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.