

## PART ONE

DATA ANALYSIS  
FOUNDATIONS

This part lays the algebraic and probabilistic foundations for data analysis. It 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.

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 Erdős–Rényi random graph model, the Watts–Strogatz model, and the Barabási–Albert model.

Kernel methods are introduced in Chapter 5, which provides 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 boundaries 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 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.

## CHAPTER 1 Data Matrix

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, which play a key role in machine learning and data mining.

### 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  $i$ th row, which is a  $d$ -tuple given as

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

and  $X_j$  denotes the  $j$ th column, which is an  $n$ -tuple given as

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

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 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*.

Table 1.1. Extract from the Iris dataset

	<b>Sepal length</b>	<b>Sepal width</b>	<b>Petal length</b>	<b>Petal width</b>	<b>Class</b>
	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
$\mathbf{x}_1$	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

**Example 1.1.** Table 1.1 shows an extract of the Iris dataset; the complete data forms a  $150 \times 5$  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

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

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  $\text{domain}(\text{Age}) = \mathbb{N}$ , where  $\mathbb{N}$  denotes the set of natural numbers (non-negative integers), is numeric, and so is `petal length` in Table 1.1, with  $\text{domain}(\text{petal length}) = \mathbb{R}^+$  (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

$$\begin{aligned}\text{domain}(\text{Sex}) &= \{\text{M}, \text{F}\} \\ \text{domain}(\text{Education}) &= \{\text{HighSchool}, \text{BS}, \text{MS}, \text{PhD}\}\end{aligned}$$

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  $\text{domain}(\text{class}) = \{\text{iris-setosa}, \text{iris-versicolor}, \text{iris-virginica}\}$ .
- *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:

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

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

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

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  $j$ th *standard basis vector*  $\mathbf{e}_j$  is the  $d$ -dimensional unit vector whose  $j$ th component is 1 and the rest of the components are 0

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

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  $j$ th 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  $\mathbf{x}_1 = (5.9, 3.0, 4.2, 1.5, \text{Iris-versicolor})$  on the first two attributes is shown in Figure 1.1(a). 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 1.1(b) 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  $\mathbb{R}^3$ :

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

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

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

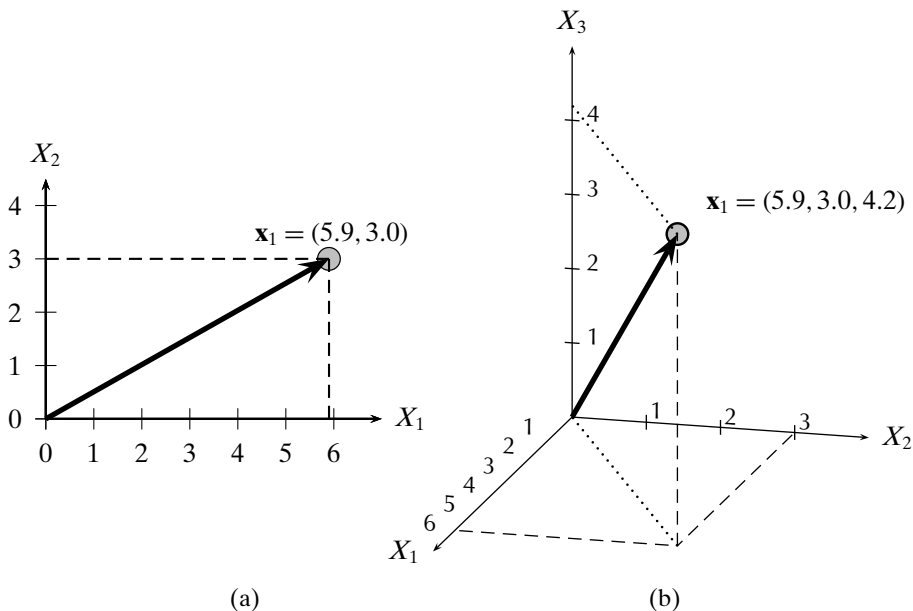


Figure 1.1. Row  $\mathbf{x}_1$  as a point and vector in (a)  $\mathbb{R}^2$  and (b)  $\mathbb{R}^3$ .

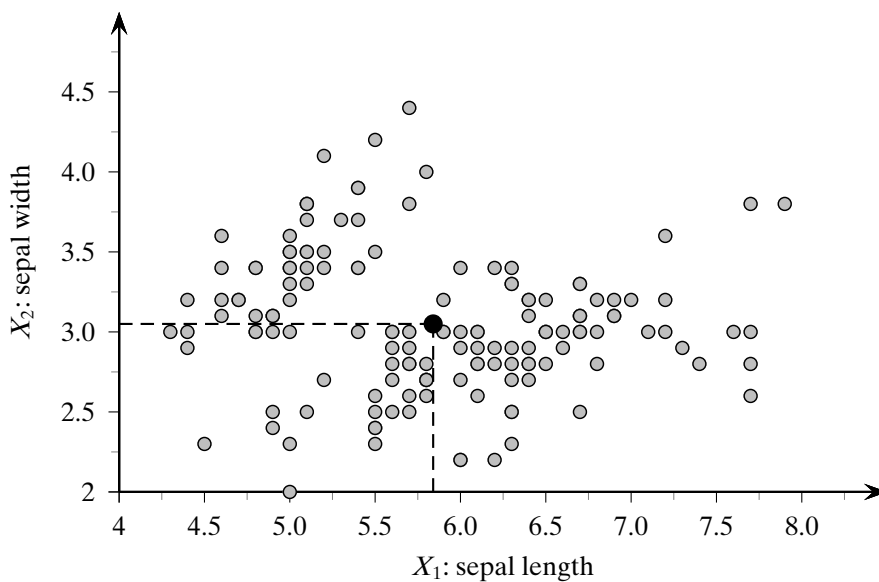


Figure 1.2. Scatterplot: sepal length versus sepal width. The solid circle shows the mean point.

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

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

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

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

#### Dot Product

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

$$\mathbf{a}^T \mathbf{b} = (a_1 \ a_2 \ \cdots \ a_m) \times \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = a_1 b_1 + a_2 b_2 + \cdots + a_m b_m = \sum_{i=1}^m a_i b_i \quad (1.1)$$

#### Length

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

$$\|\mathbf{a}\| = \sqrt{\mathbf{a}^T \mathbf{a}} = \sqrt{a_1^2 + a_2^2 + \cdots + a_m^2} = \sqrt{\sum_{i=1}^m a_i^2} \quad (1.2)$$

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

$$\|\mathbf{a}\|_p = \left( |a_1|^p + |a_2|^p + \cdots + |a_m|^p \right)^{\frac{1}{p}} = \left( \sum_{i=1}^m |a_i|^p \right)^{\frac{1}{p}} \quad (1.3)$$

for any  $p \neq 0$ . Thus, the Euclidean norm corresponds to the case when  $p = 2$ ; it is also called the  $L_2$ -norm.



**Distance**

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

$$\|\mathbf{a} - \mathbf{b}\| = \sqrt{(\mathbf{a} - \mathbf{b})^T(\mathbf{a} - \mathbf{b})} = \sqrt{\sum_{i=1}^m (a_i - b_i)^2} \quad (1.4)$$

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

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

$$\|\mathbf{a} - \mathbf{b}\|_p = \left( \sum_{i=1}^m |a_i - b_i|^p \right)^{\frac{1}{p}} \quad (1.5)$$

If  $p$  is unspecified, as in Eq. (1.4), 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

$$\cos \theta = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \left( \frac{\mathbf{a}}{\|\mathbf{a}\|} \right)^T \left( \frac{\mathbf{b}}{\|\mathbf{b}\|} \right) \quad (1.6)$$

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

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

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  $0^\circ$ , to  $-1$ , corresponding to an angle of  $180^\circ$  (or  $\pi$  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  $90^\circ$  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

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

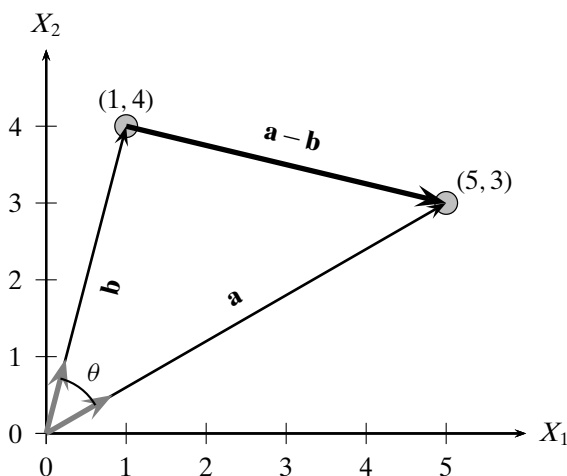


Figure 1.3. Distance and angle. Unit vectors are shown in gray.

Using Eq. (1.4), the Euclidean distance between them is given as

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

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  $\mathbf{b}$  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. (1.6) the cosine of the angle between  $\mathbf{a}$  and  $\mathbf{b}$  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:

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

Let us consider the  $L_p$ -norm for  $\mathbf{a}$  with  $p = 3$ ; we get

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