Contents

2 Getting to Know Your Data 3
  2.1 Data Objects and Attribute Types ......................... 4
  2.1.1 What Is an Attribute? .............................. 4
  2.1.2 Nominal Attributes ................................ 5
  2.1.3 Binary Attributes .................................. 6
  2.1.4 Ordinal Attributes .................................. 6
  2.1.5 Numeric Attributes .................................. 7
  2.1.6 Discrete Versus Continuous Attributes ................. 8
  2.2 Basic Statistical Descriptions of Data ..................... 9
  2.2.1 Measuring the Central Tendency: Mean, Median, Mode 9
  2.2.2 Measuring the Dispersion of Data: Range, Quartiles, Vari- 12
       ance, Standard Deviation, and Interquartile Range
  2.2.3 Graphic Displays of Basic Statistical Descriptions of Data 16
  2.3 Data Visualization ..................................... 21
  2.3.1 Pixel-Oriented Visualization Techniques .............. 21
  2.3.2 Geometric Projection Visualization Techniques ....... 23
  2.3.3 Icon-Based Visualization Techniques ................. 25
  2.3.4 Hierarchical Visualization Techniques ............... 26
  2.3.5 Visualizing Complex Data and Relations .............. 28
  2.4 Measuring Data Similarity and Dissimilarity ............. 29
  2.4.1 Data Matrix vs. Dissimilarity Matrix .............. 31
  2.4.2 Proximity Measures for Nominal Attributes .......... 33
  2.4.3 Proximity Measures for Binary Attributes .......... 35
  2.4.4 Dissimilarity on Numeric Data: Minkowski Distance 37
  2.4.5 Proximity Measures for Ordinal Attributes .......... 39
  2.4.6 Dissimilarity for Attributes of Mixed Types ....... 40
  2.4.7 Cosine Similarity ................................ 42
  2.5 Summary .............................................. 43
  2.6 Exercises ............................................. 45
  2.7 Bibliographic Notes ................................... 47
Chapter 2

Getting to Know Your Data

It’s tempting to jump straight into mining, but first, we need to get the data ready. This involves having a closer look at attributes and data values. Real-world data are typically noisy, enormous in volume (often several gigabytes or more), and may originate from a hodgepodge of heterogeneous sources. This chapter is about getting familiar with your data. Knowledge about your data is useful for data preprocessing (Chapter 3), the first major task of the data mining process. You will want to know the following: What are the types of attributes or fields that make up your data? What kind of values does each attribute have? Which attributes are discrete, and which are continuous-valued? What do the data look like? How are the values distributed? Are there ways we can visualize the data to get a better sense of it all? Can we spot any outliers? Can we measure the similarity of some data objects with respect to others? Gaining such insight into the data will help with the subsequent analysis.

“So what can we learn about our data that’s helpful in data preprocessing?”

We begin in Section 2.1 by studying the various attribute types. These include nominal attributes, binary attributes, ordinal attributes, and numeric attributes. Basic statistical descriptions can be used to learn more about each attribute’s values, as described in Section 2.2. Given a temperature attribute, for example, we can determine its mean (“average” value), median (“middle” value), and mode (most common value). These are measures of central tendency, which give us an idea of the “middle” or center of distribution. Knowing such basic statistics regarding each attribute makes it easier to fill in missing values, smooth noisy values, and spot outliers during data preprocessing. Knowledge of the attributes and attribute values can also help in fixing inconsistencies incurred during data integration. Plotting the measures of central tendency shows us if the data are symmetric or skewed. Quantile plots, histograms, and scatter plots are other graphic displays of basic statistical descriptions. These can all be useful during data preprocessing, as well as provide insight into areas for mining.

The field of data visualization provides many additional techniques for viewing data through graphical means. These can help identify relations, trends,
and biases “hidden” in unstructured data sets. Techniques may be as simple as scatterplot matrices (where two attributes are mapped onto a 2D grid) to more sophisticated methods such as tree-map (where a hierarchical partitioning of the screen is displayed based on the attribute values). Data visualization techniques are described in Section 2.3.

Finally, we may want to examine how similar (or dissimilar) data objects are. For example, suppose we have a database where the data objects are patients, described by their symptoms. We may want to find clusters of like patients within the dataset. It may also be used to detect outliers in the data, or to perform nearest-neighbor classification. (Clustering is the topic of Chapters 10 and 11, while nearest-neighbor classification is discussed in Chapter 8.) There are many measures for assessing similarity and dissimilarity. You can think of the proximity of two objects as a function of the distance between their attribute values, although proximity can also be calculated based on probabilities rather than actual distance. Measures of data proximity are described in Section 2.4.

In summary, by the end of this chapter, you will know the different attribute types, and basic statistical measures to describe the central tendency and dispersion (spread) of attribute data. You will also know techniques to visualize attribute distributions, and how to compute the similarity or dissimilarity between objects.

2.1 Data Objects and Attribute Types

Data sets are made up of data objects. A data object represents an entity. In a sales database, the objects could be customers, store items, or sales, for instance. In a medical database, the objects may be patients. In a university database, the objects could be students, professors, and courses. Data objects are typically described by attributes. Data objects can also be referred to as samples, examples, instances, data points, or objects. If the data objects are stored in a database, they are data tuples. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes. In this section, we define attributes and look at the various attribute types.

2.1.1 What Is an Attribute?

An attribute is a data field, representing a characteristic or feature of a data object. The nouns attribute, dimension, feature, and variable are often used interchangeably in literature. The term dimension is commonly used in data warehousing. Machine learning literature tends to use the term feature, while statisticians prefer the term variable. Data mining and database professionals commonly use the term attribute. We use the term attribute here as well. Attributes describing a customer object can include, for example, customer ID, name, and address. Observed values for a given attribute are known as obser-
A set of attributes used to describe a given object is called an *attribute vector* (or *feature vector*). The distribution of data involving one attribute (or variable) is called *univariate*. A *bivariate* distribution involves two attributes, and so on.

The *type* of an attribute is determined by the set of possible values the attribute can have. Attributes can be nominal, binary, ordinal, or numeric. In the following subsections, we introduce each type.

### 2.1.2 Nominal Attributes

Nominal means “relating to names.” The values of a *nominal attribute* are symbols or *names of things*. Each value represents some kind of category, code, or state and so nominal attributes are also referred to as *categorical*. The values do not have any meaningful order about them. In computer science, the values are also known as *enumerations*.

**Example 2.1** Nominal attributes. Suppose that *Hair_color* and *Marital_status* are two attributes describing *person* objects. In our application, possible values for *Hair_color* are *black, brown blond, red, auburn, grey, and white*. *Marital_status* can take on the values *single, married, divorced, and widowed*. Both *Hair_color* and *Marital_status* are nominal attributes. *Occupation* is another example, with the values *teacher, dentist, programmer, farmer*, and so on.

Although we said that the values of a nominal attribute are symbols or “names of things”, it is possible to represent such symbols or “names” with numbers. With *Hair_color*, for instance, we can assign a code of 0 for *black*, 1 for *brown*, and so on. *Customer_ID* is another example of a nominal attribute whose possible values are all numeric. However, in such cases, the numbers are not intended to be used quantitatively. That is, mathematical operations on values of nominal attributes are not meaningful. It makes no sense to subtract one customer ID number from another, unlike, say, subtracting an age value from another. (*Age* is a numeric attribute). Even though a nominal attribute may have integers as values, it is not considered a numeric attribute because the integers are not meant to be used quantitatively. We will say more on numeric attributes in Section 2.1.5.

Because nominal attribute values do not have any meaningful order about them and are not quantitative, it makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects. One thing that is of interest, however, is the attribute’s most commonly occurring value. This value, known as the *mode*, is one of the measures of central tendency. We will say more about how to compute the measures of central tendency in Section 2.2.
2.1.3 Binary Attributes

A binary attribute is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent, and 1 means that it is present. Binary attributes are referred to as Boolean if the two states correspond to true and false.

Example 2.2 Binary attributes. Given the attribute Smoker describing a patient object, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Similarly, suppose the patient undergoes a medical test that has two possible outcomes. The attribute Medical test is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.

A binary attribute is symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1. One such example could be the attribute gender having the states male and female. A binary attribute is asymmetric if the outcomes of the states are not equally important, such as the positive and negative outcomes of a medical test for HIV. By convention, we shall code the most important outcome, which is usually the rarest one, by 1 (e.g., HIV positive) and the other by 0 (e.g., HIV negative).

2.1.4 Ordinal Attributes

An ordinal attribute is an attribute whose possible values have a meaningful order or ranking among them, but the magnitude between successive values is not known.

Example 2.3 Ordinal attributes. Suppose that Drink size corresponds to the size of drinks available at a fast food restaurant. This nominal attribute has three possible values – small, medium, and large. The values have a meaningful sequence (which corresponds to increasing drink size), however, we cannot tell from the values how much bigger, say, a medium is from a large. Other examples of ordinal attributes include Grade (e.g., A+, A, A−, B+, and so on) and Professional rank. Professional ranks can be enumerated in a sequential order, such as assistant, associate, and full for professors, and private, private first class, specialist, corporal, sergeant for army ranks.

Ordinal attributes are useful for registering subjective assessments of qualities that cannot be measured objectively. Hence, ordinal attributes are often used in surveys for ratings. In one survey, participants were asked to rate how satisfied they were as customers. Customer satisfaction had the following ordinal categories: 0: very dissatisfied, 1: somewhat dissatisfied, 2: neutral, 3: satisfied, and 4: very satisfied.

Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories.
as described in Chapter 3 on data reduction.

The central tendency of an ordinal attribute can be represented by its mode and its median (the middle value in an ordered sequence), but the mean cannot be defined.

Note that nominal, binary, and ordinal attributes are qualitative. That is, they describe a feature of an object, without giving an actual size or quantity. The values of such qualitative attributes are typically words representing categories. If integers are used, they represent computer codes for the categories, as opposed to measurable quantities (e.g., 0 for small drink size, 1 for medium, and 2 for large). In the following subsection we look at numeric attributes, which provide quantitative measurements of an object.

2.1.5 Numeric Attributes

A numeric attribute is quantitative, that is, it is a measurable quantity, represented in integer or real values. Numeric attributes can be interval-scaled or ratio-scaled.

Interval-Scaled Attributes

Interval-scaled attributes are measured on a scale of equal-sized units. The values of interval-scaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.

Example 2.4 Interval-scaled attributes. Temperature is an interval-scaled attribute. Suppose that we have the outdoor temperature value for a number of different days, where each day is an object. By ordering the values, we obtain a ranking of the objects with respect to temperature. In addition, we can quantify the difference between values. For example, a temperature of 20°C is 5 degrees higher than a temperature of 15°C. Calendar dates are another example. For instance, the years 2002 and 2010 are 8 years apart.

Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is, neither 0°C nor 0°F indicates “no temperature.” (On the Celsius scale, for example, the unit of measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure.) Although we can compute the difference between temperature values, we cannot talk of one temperature value as being a multiple of another. Without a true zero, we cannot say, for instance, that 10°C is twice as warm as 5°C. That is, we cannot speak of the values in terms of ratios. Similarly, there is no true zero-point for calendar dates. (The year 0 does not correspond to the beginning of time.) This brings us to the next type of attribute – ratio-scaled attributes, for which a true zero-point exits.

Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.
Ratio-Scaled Attributes

A ratio-scaled attribute is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value. In addition, the values are ordered, and we can also compute the difference between values. The mean, median, and mode can be computed as well.

Example 2.5 Ratio-scaled attributes. Unlike temperatures in Celsius and Fahrenheit, the Kelvin (K) temperature scale has what is considered a true zero-point (0 degrees K = −273.15°C): it is the point at which the particles that comprise matter have zero kinetic energy. Other examples of ratio-scaled attributes include Count attributes such as Years of experience (where the objects are employees, for example) and Number of words (where the objects are documents). Additional examples include attributes to measure weight, height, latitude and longitude coordinates (e.g., when clustering houses), and monetary quantities (e.g., you are 100 times richer with $100 than with $1).

2.1.6 Discrete Versus Continuous Attributes

In our presentation, we have organized attributes into nominal, binary, ordinal, and numeric types. There are many ways to organize attribute types. The types are not mutually exclusive.

Classification algorithms developed from the field of machine learning often talk of attributes as being either discrete or continuous. Each type may be processed differently. A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers. The attributes Hair_color, Smoker, Medical_test, and Drink_size each have a finite number of values, and so are discrete. Note that discrete attributes may have numeric values, such as 0 and 1 for binary attributes, or, the values 0 to 110 for the attribute Age. An attribute is countably infinite if the set of possible values is infinite, but the values can be put in a one-to-one correspondence with natural numbers. For example, the attribute customer_ID is countably infinite. The number of customers can grow to infinity, but in reality, the actual set of values is countable (where the values can be put in one-to-one correspondence with the set of integers). Zip codes are another example.

If an attribute is not discrete, it is continuous. The terms “numeric attribute” and “continuous attribute” are often used interchangeably in the literature. (This can be confusing because, in the classical sense, continuous values are real numbers, whereas numeric values can be either integers or real numbers.) In practice, real values are represented using a finite number of digits. Continuous attributes are typically represented as floating-point variables.
2.2 Basic Statistical Descriptions of Data

For data preprocessing to be successful, it is essential to have an overall picture of your data. Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

This section discusses three areas of basic statistical descriptions. We start with measures of central tendency (Section 2.2.1), which measure the location of the middle or center of a data distribution. Intuitively speaking, given an attribute, where do most of its values fall? In particular, we discuss the mean, median, mode, and midrange.

In addition to assessing the central tendency of our data set, we also would like to have an idea of the dispersion of the data. That is, how are the data spread out? The most common measures of data dispersion are the range, quartiles, and interquartile range, the five-number summary and boxplots, and the variance and standard deviation of the data. These measures are useful for identifying outliers. These are described in Section 2.2.2.

Finally, we can use many graphic displays of basic statistical descriptions to visually inspect our data (Section 2.2.3). Most statistical or graphical data presentation software packages include bar charts, pie charts, and line graphs. Other popular displays of data summaries and distributions include quantile plots, quantile-quantile plots, histograms, and scatter plots.

2.2.1 Measuring the Central Tendency: Mean, Median, Mode

In this section, we look at various ways to measure the central tendency of data. Suppose that we have some attribute \( X \), like salary, which has been recorded for a set of objects. Let \( x_1, x_2, \ldots, x_N \) be the set of \( N \) observed values or observations for \( X \). These values may also be referred to as “the data set” (for \( X \)) in the remainder of this section. If we were to plot the observations for salary, where would most of the values fall? This gives us an idea of the central tendency of the data. Measures of central tendency include the mean, median, mode, and midrange.

The most common and effective numeric measure of the “center” of a set of data is the (arithmetic) mean. Let \( x_1, x_2, \ldots, x_N \) be a set of \( N \) values or observations, such as for some numeric attribute \( X \), like salary. The mean of this set of values is

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i = \frac{x_1 + x_2 + \cdots + x_N}{N}.
\]  

(2.1)

This corresponds to the built-in aggregate function, `average (avg()) in SQL`, provided in relational database systems.

Example 2.6 Mean. Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 31, 47, 50, 52, 52, 56, 60, 63, 70, 70,
1. Using Equation (2.1), we have

\[
\bar{x} = \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12}
\]

\[
= \frac{696}{12} = 58.
\]

Thus, the mean salary is $58K.

Sometimes, each value \(x_i\) in a set may be associated with a weight \(w_i\), for \(i = 1, \ldots, N\). The weights reflect the significance, importance, or occurrence frequency attached to their respective values. In this case, we can compute

\[
\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}.
\]

(2.2)

This is called the \textbf{weighted arithmetic mean} or the \textbf{weighted average}.

Although the mean is the single most useful quantity for describing a data set, it is not always the best way of measuring the center of the data. A major problem with the \textit{mean} is its sensitivity to extreme (e.g., outlier) values. Even a small number of extreme values can corrupt the mean. For example, the mean salary at a company may be substantially pushed up by that of a few highly paid managers. Similarly, the mean score of a class in an exam could be pulled down quite a bit by a few very low scores. To offset the effect caused by a small number of extreme values, we can instead use the \textbf{trimmed mean}, which is the mean obtained after chopping off values at the high and low extremes. For example, we can sort the values observed for \(salary\) and remove the top and bottom 2% before computing the mean. We should avoid trimming too large a portion (such as 20%) at both ends as this can result in the loss of valuable information.

For skewed (asymmetric) data, a better measure of the center of data is the \textbf{median}. The median is the middle value in a set of ordered data values. It is the value that separates the higher half of a data set from the lower half.

In probability and statistics, the median generally applies to numeric data, however, we may extend the concept to ordinal data. Suppose that a given data set of \(N\) values for an attribute \(X\) is sorted in increasing order. If \(N\) is odd, then the median is the \textit{middle value} of the ordered set. If \(N\) is even then the median is not unique; it is the two middlemost values and any value in between. If \(X\) is a numeric attribute in this case, by convention, the median is taken as the average of the two middlemost values.

**Example 2.7 Median.** Let’s find the median of the data from Example 2.2.1. The data are already sorted in increasing order. There is an even number of observations (i.e., 12), therefore, the median is not unique. It can be any value within the
two middlemost values of 52 and 56 (that is, within the 5th and 6th values in the list). By convention, we assign the average of the two middlemost values as the median. That is, \[ \frac{52 + 56}{2} = \frac{108}{2} = 54 \] Thus, the median is $54K.

Suppose that we had only the first eleven values in the list. Given an odd number of values, the median is the middlemost value. This is the 5th value in this list, which has a value of $52K.

The median is expensive to compute when we have a large number of observations. For numeric attributes, however, we can easily approximate the value. Assume that data are grouped in intervals according to their \( x_i \) data values and that the frequency (i.e., number of data values) of each interval is known. For example, employees may be grouped according to their annual salary in intervals such as 10–20K, 20–30K, and so on. Let the interval that contains the median frequency be the median interval. We can approximate the median of the entire data set (e.g., the median salary) by interpolation using the formula:

\[
\text{median} = L_1 + \left( \frac{N/2 - (\sum freq)_l}{freq_{\text{median}}} \right) \text{width},
\]

where \( L_1 \) is the lower boundary of the median interval, \( N \) is the number of values in the entire data set, \((\sum freq)_l\) is the sum of the frequencies of all of the intervals that are lower than the median interval, \(freq_{\text{median}}\) is the frequency of the median interval, and \(\text{width}\) is the width of the median interval.

The mode is another measure of central tendency. The mode for a set of data is the value that occurs most frequently in the set. Therefore, it can be determined for qualitative and quantitative attributes. It is possible for the greatest frequency to correspond to several different values, which results in more than one mode. Data sets with one, two, or three modes are respectively called unimodal, bimodal, and trimodal. In general, a data set with two or more modes is multimodal. At the other extreme, if each data value occurs only once, then there is no mode.

Example 2.8 Mode. The data of Example 2.2.1 are bimodal. The two modes are $52K and $70K.

For unimodal numeric data that are moderately skewed (asymmetrical), we have the following empirical relation:

\[
\text{mean} - \text{mode} \approx 3 \times (\text{mean} - \text{median}).
\]

This implies that the mode for unimodal frequency curves that are moderately skewed can easily be approximated if the mean and median values are known.

The midrange can also be used to assess the central tendency of a numeric data set. It is the average of the largest and smallest values in the set. This measure is easy to compute using the SQL aggregate functions, \( \text{max()} \) and \( \text{min()} \).

Example 2.9 Midrange. The midrange of the data of Example 2.2.1 is \[ \frac{10 + 110}{2} = \frac{120}{2} = 70K. \]
In a unimodal frequency curve with perfect symmetric data distribution, the mean, median, and mode are all at the same center value, as shown in Figure 2.1(a).

Data in most real applications are not symmetric. They may instead be either positively skewed, where the mode occurs at a value that is smaller than the median (Figure 2.1(b)), or negatively skewed, where the mode occurs at a value greater than the median (Figure 2.1(c)).

2.2.2 Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range

We now look at measures to assess the dispersion or spread of numeric data. The measures include range, quantiles, quartiles, percentiles, and the interquartile range. The five-number summary, which can be displayed as a boxplot, is useful in identifying outliers. Variance and standard deviation also indicate the spread of a data distribution.

Range, Quartiles, and the Interquartile Range (IQR)

To start off, let’s study the range, quantiles, quartiles, percentiles, and the interquartile range as measures of data dispersion.

Let $x_1, x_2, \ldots, x_N$ be a set of observations for some numeric attribute, $X$. The range of the set is the difference between the largest ($\max()$) and smallest ($\min()$) values.

Suppose that the data for attribute $X$ are sorted in increasing numeric order. Imagine that we can pick certain data points so as to split the data distribution into equal-sized consecutive sets, as in Figure 2.2. These data points are called quantiles. Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equal-sized consecutive sets. (We say “essentially” because there may not be data values of $X$ that divide the data into exactly
Here will be a graph: not drawn yet. Any volunteer to draw such a graph?! I have a nice drawing software called SmarDraw. –JH

Figure 2.2: A plot of the data distribution for some attribute \( X \). Here, the quantiles plotted are quartiles. The three quartiles divide the distribution into four equal-sized consecutive subsets. The second quartile corresponds to the median.

equal-sized subsets. For readability, we will refer to them hereafter as equal.)

The \( k^{th} \) \( q \)-quantile for a given data distribution is the value \( x \) such that at most \( k/q \) of the data values are less than \( x \) and at most \( (q - k)/q \) of the data values are more than \( x \), where \( k \) is an integer such that \( 0 < k < q \). There are \( q - 1 \) \( q \)-quantiles.

The 2-quantile is the data point dividing the lower and upper halves of the data distribution. It corresponds to the median. The 4-quantiles are the three data points that split the data distribution into four equal parts so that each part represents one fourth of the data distribution. They are more commonly referred to as quartiles. The 100-quantiles are more commonly referred to as percentiles; they divide the data distribution into 100 equal-sized consecutive sets. The median, quartiles, and percentiles are the most widely used forms of quantiles.

The quartiles give an indication of the center, spread, and shape of a distribution. The first quartile, denoted by \( Q_1 \), is the 25th percentile. It cuts off the lowest 25% of the data. The third quartile, denoted by \( Q_3 \), is the 75th percentile. It cuts off the lowest 75% (or highest 25%) of the data. The second quartile is the 50th percentile. As the median, it gives the center of the data distribution.

The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the interquartile range \( (IQR) \) and is defined as

\[
IQR = Q_3 - Q_1.
\]

(2.5)

Example 2.10 Interquartile range. The quartiles are the three values that split the sorted data set into four equal parts. The data of Example 2.2.1 contain 12 observations, already sorted in increasing order. Thus, the quartiles for this data are the 3rd, 6th, and 9th values, respectively, in the sorted list. Therefore, \( Q_1 = $47K \) and \( Q_3 = $63K \). Thus, the interquartile range is \( IQR = 63 - 47 = $16K \).
(Note that the 6th value is a median, $52K, although this data set has two medians since the number of data values is even.)

Five-Number Summary, Boxplots, and Outliers

No single numeric measure of spread, such as \( IQR \), is very useful for describing skewed distributions. Have a look at the symmetric and skewed data distributions
of Figure 2.1. In the symmetric distribution, the median (and other measures of central tendency) splits the data into equal-size halves. This does not occur for skewed distributions. Therefore, it is more informative to also provide the two quartiles $Q_1$ and $Q_3$, along with the median. A common rule of thumb for identifying suspected outliers is to single out values falling at least $1.5 \times IQR$ above the third quartile or below the first quartile.

Because $Q_1$, the median, and $Q_3$ together contain no information about the endpoints (e.g., tails) of the data, a fuller summary of the shape of a distribution can be obtained by providing the lowest and highest data values as well. This is known as the five-number summary. The five-number summary of a distribution consists of the median, the quartiles $Q_1$ and $Q_3$, and the smallest and largest individual observations, written in the order of Minimum, $Q_1$, Median, $Q_3$, Maximum.

**Boxplots** are a popular way of visualizing a distribution. A boxplot incorporates the five-number summary as follows:

- Typically, the ends of the box are at the quartiles, so that the box length is the interquartile range, $IQR$.
- The median is marked by a line within the box.
- Two lines (called whiskers) outside the box extend to the smallest (Minimum) and largest (Maximum) observations.

When dealing with a moderate number of observations, it is worthwhile to plot potential outliers individually. To do this in a boxplot, the whiskers are extended to the extreme low and high observations only if these values are less than $1.5 \times IQR$ beyond the quartiles. Otherwise, the whiskers terminate at the most extreme observations occurring within $1.5 \times IQR$ of the quartiles. The remaining cases are plotted individually. Boxplots can be used in the comparisons of several sets of compatible data.

**Example 2.11** **Boxplot.** Figure 2.3 shows boxplots for unit price data for items sold at four branches of AllElectronics during a given time period. For branch 1, we see that the median price of items sold is $80$, $Q_1$ is $60$, $Q_3$ is $100$. Notice that two outlying observations for this branch were plotted individually, as their values of 175 and 202 are more than 1.5 times the IQR here of 40.

Boxplots can be computed in $O(n \log n)$ time. Approximate boxplots can be computed in linear or sublinear time depending on the quality guarantee required.

**Variance and Standard Deviation**

Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is. A low standard deviation means that the data observations tend to be very close to the mean, while high standard deviation indicates that the data are spread out over a large range of values.
2.2. BASIC STATISTICAL DESCRIPTIONS OF DATA

Figure 2.3: Boxplot for the unit price data for items sold at four branches of AllElectronics during a given time period.

The variance of $N$ observations, $x_1, x_2, \ldots, x_N$, for a numeric attribute $X$ is

$$
\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 = \left( \frac{1}{N} \sum_{i=1}^{N} x_i^2 \right) - \bar{x}^2,
$$

(2.6)

where $\bar{x}$ is the mean value of the observations, as defined in Equation (2.1). The standard deviation, $\sigma$, of the observations is the square root of the variance, $\sigma^2$.

Example 2.12 Variance and standard deviation. In Example 2.2.1, we found $\bar{x} = $58K using Equation (2.1) for the mean. To determine the variance and standard deviation of the data from that example, we set $N = 12$ and use Equation (2.6) to obtain

$$
\sigma^2 = \frac{1}{12} (30^2 + 36^2 + 47^2 \ldots + 110^2) - 58^2
$$

$$
\approx 379.17
$$

$$
\sigma \approx \sqrt{379.17} \approx 19.47.
$$

The basic properties of the standard deviation, $\sigma$, as a measure of spread are
• $\sigma$ measures spread about the mean and should be considered only when the mean is chosen as the measure of center.

• $\sigma = 0$ only when there is no spread, that is, when all observations have the same value. Otherwise $\sigma > 0$.

Importantly, an observation is unlikely more than several standard deviations away from the mean. Mathematically, using Chebyshev’s inequality, it can be shown that at least $(1 - \frac{1}{k^2}) \times 100\%$ of the observations are no more than $k$ standard deviations from the mean. Therefore, the standard deviation is a good indicator of the spread of a data set.

The computation of the variance and standard deviation is scalable in large databases.

### 2.2.3 Graphic Displays of Basic Statistical Descriptions of Data

In this section, we study graphic displays of basic statistical descriptions. These include quantile plots, quantile-quantile plots, histograms, and scatter plots. Such graphs are helpful for the visual inspection of data, which is useful for data preprocessing. The first three of these show univariate distributions (i.e., data for one attribute), while scatter plots show bivariate distributions (that is, involving two attributes).

#### Quantile Plot

In this and the following subsections, we cover common graphic displays of data distributions. A quantile plot is a simple and effective way to have a first look at a univariate data distribution. First, it displays all of the data for the given attribute (allowing the user to assess both the overall behavior and unusual occurrences). Second, it plots quantile information. Quantiles were described in Section 2.2.2. Let $x_i$, for $i = 1$ to $N$, be the data sorted in increasing order so that $x_1$ is the smallest observation and $x_N$ is the largest for some ordinal or numeric attribute $X$. Each observation, $x_i$, is paired with a percentage, $f_i$, which indicates that approximately $f_i \times 100\%$ of the data are below the value, $x_i$. We say “approximately” because there may not be a value with exactly a fraction, $f_i$, of the data below $x_i$. Note that the 0.25 percentage corresponds to quartile $Q_1$, the 0.50 percentage is the median, and the 0.75 percentage is $Q_3$.

Let

$$f_i = \frac{i - 0.5}{N}. \quad (2.7)$$

These numbers increase in equal steps of $1/N$, ranging from $\frac{1}{2N}$ (which is slightly above zero) to $1 - \frac{1}{2N}$ (which is slightly below one). On a quantile plot, $x_i$ is graphed against $f_i$. This allows us to compare different distributions based on their quantiles. For example, given the quantile plots of sales data for two different time periods, we can compare their $Q_1$, median, $Q_3$, and other $f_i$ values at a glance.
Table 2.1: A set of unit price data for items sold at a branch of AllElectronics.

<table>
<thead>
<tr>
<th>Unit price ($)</th>
<th>Count of items sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>275</td>
</tr>
<tr>
<td>43</td>
<td>300</td>
</tr>
<tr>
<td>47</td>
<td>250</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>74</td>
<td>360</td>
</tr>
<tr>
<td>75</td>
<td>515</td>
</tr>
<tr>
<td>78</td>
<td>540</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>115</td>
<td>320</td>
</tr>
<tr>
<td>117</td>
<td>270</td>
</tr>
<tr>
<td>120</td>
<td>350</td>
</tr>
</tbody>
</table>

Figure 2.4: A quantile plot for the unit price data of Table 2.1. 

NOTE: This graph must be changed so that Q1, Median, and Q3 denote every 9th point, respectively. Every 9th point is a quantile. Also, add labels Q1, Median, Q3 as darkened points in the graph.

Example 2.13 Quantile plot. Figure 2.4 shows a quantile plot for the unit price data of Table 2.1.

Quantile-Quantile (Q-Q) Plot

A quantile-quantile plot, or q-q plot, graphs the quantiles of one univariate distribution against the corresponding quantiles of another. It is a powerful visualization tool in that it allows the user to view whether there is a shift in going from one distribution to another.

Suppose that we have two sets of observations for the attribute or variable unit price, taken from two different branch locations. Let \( x_1, \ldots, x_N \) be the data from the first branch, and \( y_1, \ldots, y_M \) be the data from the second, where each
Example 2.14 Quantile-quantile plot. Figure 2.5 shows a quantile-quantile plot for unit price data of items sold at two different branches of AllElectronics during a given time period. Each point corresponds to the same quantile for each data set and shows the unit price of items sold at branch 1 versus branch 2 for that quantile. (To aid in comparison, a straight line is shown that represents the case where, for each given quantile, the unit price at each branch is the same. The darker points correspond to the data for $Q_1$, the median, and $Q_3$, respectively.) We see, for example, that at $Q_1$, the unit price of items sold at branch 1 was slightly less than that at branch 2. In other words, 25% of items sold at branch 1 were less than or equal to $60, while 25% of items at branch 2 were less than or equal to $64. At the 50th percentile (marked by the median, which is also $Q_2$), we see that 50% of items sold at branch 1 were less than $80, while 50% of items at branch 2 were less than $90. In general, we note that there is a shift in the distribution of branch 1 with respect to branch 2 in that the unit prices of items sold at branch 1 tend to be lower than those at branch 2.
2.2. BASIC STATISTICAL DESCRIPTIONS OF DATA

Histograms

Histograms (or frequency histograms) are at least a century old and are widely used. “Histog” means pole or mast, and “gram” means chart, so a histogram is a chart of poles. Plotting histograms is a graphical method for summarizing the distribution of a given attribute, X. If X is nominal, such as automobile model or item type, then a pole or vertical bar is drawn for each known value of X. The height of the bar indicates the frequency (i.e., count) of that X value. The resulting graph is more commonly known as a bar chart.

If X is numeric, the term histogram is preferred. The range of values for X is partitioned into disjoint consecutive subranges. The subranges, referred to as buckets, are disjoint subsets of the data distribution for X. The range of a bucket is known as the width. Typically, the buckets are equi-width. For example, a price attribute with a value range of $1 to $200 (rounded up to the nearest dollar) can be partitioned into subranges 1 to 20, 21 to 40, 41 to 60, and so on. For each subrange, a bar is drawn whose height represents the total count of items observed within the subrange. Histograms and partitioning rules are further discussed in Chapter 3 on data reduction.

Example 2.15 Histogram. Figure 2.6 shows a histogram for the data set of Table 2.1, where buckets are defined by equal-width ranges representing $20 increments and the frequency is the count of items sold.

Although histograms are widely used, they may not be as effective as the quantile plot, q-q plot, and boxplot methods in comparing groups of univariate observations.
Scatter Plots and Data Correlation

A scatter plot is one of the most effective graphical methods for determining if there appears to be a relationship, pattern, or trend between two numeric attributes. To construct a scatter plot, each pair of values is treated as a pair of coordinates in an algebraic sense and plotted as points in the plane. Figure 2.7 shows a scatter plot for the set of data in Table 2.1.

The scatter plot is a useful method for providing a first look at bivariate data to see clusters of points and outliers, or to explore the possibility of correlation relationships. Two attributes, $X$, and $Y$, are correlated if one attribute implies the other. Correlations can be positive, negative, or null (uncorrelated). Figure 2.8 shows examples of positive and negative correlations between two attributes. If the pattern of plotted points slopes from lower left to upper right, this means that the values of $X$ increase as the values of $Y$ increase, which suggests a positive correlation (Figure 2.8a). If the pattern of plotted points slopes from upper left to lower right, then the values of $X$ increase as the values of $Y$ decrease, suggesting a negative correlation (Figure 2.8b). A line of best fit can be drawn in order to study the correlation between the variables. Statistical
2.3. DATA VISUALIZATION

How can we convey data to users effectively? **Data visualization** aims to communicate data clearly and effectively through graphical representation. Data visualization has been used extensively in many applications. For example, we can use data visualization at work for reporting, managing business operations, and tracking progress of tasks. More popularly, we may take advantage of visualization techniques to discover data relationships that are otherwise not easily observable by looking at the raw data. Nowadays, people also use data visualization to create fun and interesting graphics.

In this section, we briefly introduce the basic concepts of data visualization. We start with multidimensional data such as those stored in relational databases. We discuss several representative approaches, including pixel-oriented techniques, geometric projection techniques, icon-based techniques, and hierarchical and graph-based techniques. We then discuss the visualization of complex data and relations.

2.3.1 Pixel-Oriented Visualization Techniques

A simple way to visualize the value of a dimension is to use a pixel where the color of the pixel reflects the dimension’s value. For a data set of $m$ dimensions, **pixel-oriented techniques** create $m$ windows on the screen, one for each dimension. The $m$ dimension values of a record are mapped to $m$ pixels at the corresponding positions.
positions in the windows. The colors of the pixels reflect the corresponding values.

Inside a window, the data values are arranged in some global order shared by all windows. The global order may be obtained by sorting all data records in a way that’s meaningful for the task at hand.

Example 2.16 **Pixel-oriented visualization.** *AllElectronics* maintains a customer information table, which consists of four dimensions: *income*, *credit limit*, *transaction volume*, and *age*. Can we analyze the correlation between *income* and the other attributes by visualization?

We can sort all customers in *income* ascending order, and use this order to lay out the customer data in the four visualization windows, as shown in Figure 2.10. The pixel colors are chosen so that the smaller the value is, the lighter the color is. Using pixel-based visualization, we can easily observe the following: *credit limit* increases as *income* increases; customers whose income is in the middle range are more likely to purchase more from *AllElectronics*; there is no clear correlation between *income* and *age*.

In pixel-oriented techniques, data records can also be ordered in a query-dependent way. For example, given a point query, we can sort all records in descending order of similarity to the point query.

Filling a window by laying out the data records in a linear way may not work well for a wide window. The first pixel in a row is far away from the last pixel in the previous row, though they are next to each other in the global order. Moreover, a pixel is next to the one above it in the window, even though the two are not next to each other in the global order. To solve this problem, we can lay out the data records in a space-filling curve to fill the windows. A *space-filling curve* is a curve whose range covers the entire $n$-dimensional unit hypercube. Since the visualization windows are 2-dimensional, we can use any 2-dimensional space-filling curve. Figure 2.11 shows some frequently used 2-dimensional space-filling-curves.
2.3. DATA VISUALIZATION

(a) Hilbert curve  
(b) Gray code  
(c) Z-curve

Figure 2.11: Some frequently used 2-dimensional space-filling curves.

(a)  
(b)  
(c)  

Figure 2.12: The circle segment technique. (a) Representing a data record in circle segments. (b) Laying out pixels in circle segments.

Note that the windows do not have to be rectangular. For example, the circle segment technique uses windows in the shape of segments of a circle, as illustrated in Figure 2.12. This technique can ease the comparison of dimensions because the dimension windows are located side by side and form a circle.

2.3.2 Geometric Projection Visualization Techniques

A drawback of pixel-oriented visualization techniques is that they cannot help us much in understanding the distribution of data in a multidimensional space. For example, they do not show whether there is a dense area in a multidimensional subspace. Geometric projection techniques help users find interesting projections of multidimensional data sets. The central challenge the geometric projection techniques try to address is how to visualize a high dimensional space on a 2-dimensional display.

A scatter plot displays 2-dimensional data points using Cartesian coordinates. A third dimension can be added using different colors or shapes to represent different data points. Figure 2.13 shows an example, where X and Y are
two spatial attributes and the third dimension is represented by different shapes. Through this visualization, we can see that points of type “+” and “×” tend to be co-located.

A 3-dimensional scatter plot uses three axes in a Cartesian coordinate system. If it also uses color, it can display up to 4-dimensional data points (Figure 2.14).

For data sets whose dimensionality is over four, scatter plots are usually ineffective. The scatter plot matrix technique is a useful extension to the scatter plot. For an n-dimensional data set, a scatter plot matrix is an $n \times n$ grid of 2-dimensional scatter plots that provides a visualization of each dimension with every other dimension. Figure 2.15 shows an example, which visualizes the Iris data set. The data set consists of 450 samples from each of three species of Iris flowers. There are five dimensions in the data set: the length and the width of sepal and petal, and the species.

The scatter-plot matrix becomes less effective as the dimensionality increases. Another popular technique, called parallel coordinates, can handle higher dimensionality. To visualize n-dimensional data points, the parallel coordinates technique draws n equally spaced axes, one for each dimension, parallel to one of the display axes. A data record is represented by a polygonal line that intersects each axis at the point corresponding to the associated dimension value (Figure 2.16).

A major limitation of the parallel coordinates technique is that it cannot
2.3. DATA VISUALIZATION

Figure 2.14: Visualization of a 3-dimensional data set using a scatter plot. Source: http://upload.wikimedia.org/wikipedia/commons/c/c4/Scatter_plot.jpg.

effectively show a data set of many records. Even for a data set of several thousand records, visual clutter and overlap often reduce the readability of the visualization and make the patterns hard to find.

2.3.3 Icon-Based Visualization Techniques

Icon-based visualization techniques use small icons to represent multidimensional data values. We look at two popular icon-based techniques – Chernoff faces and stick figures.

Chernoff faces were introduced in 1973 by statistician Herman Chernoff. They display multidimensional data of up to eighteen variables (or dimensions) as a cartoon human face (Figure 2.17). Chernoff faces help reveal trends in the data. Components of the face, such as the eyes, ears, mouth, and nose, represent values of the dimensions by their shape, size, placement, and orientation. For example, dimensions can be mapped to the following facial characteristics: eye size, eye spacing, nose length, nose width, mouth curvature, mouth width, mouth openness, pupil size, eyebrow slow, eye eccentricity, and head eccentricity.

Chernoff faces make use of the ability of the human mind to recognize small differences in facial characteristics and to assimilate many facial characteristics at once. Viewing large tables of data can be tedious. By condensing the data, Chernoff faces make the data easier for users to digest. In this way, they facilitate the visualization of regularities and irregularities present in the data although their power in relating multiple relationships is limited. Another limitation is that specific data values are not shown. Furthermore, the features of the face vary in perceived importance. This means that the similarity of two faces (representing two multidimensional data points) can vary depending on the order in which dimensions are assigned to facial characteristics. Therefore, this mapping
should be carefully chosen. Eye size and eyebrow slant have been found to be important.

Asymmetrical Chernoff faces were proposed as an extension to the original technique. Since a face has vertical symmetry (along the $y$-axis), the left and right side of a face are identical, which wastes space. Asymmetrical Chernoff faces double the number of facial characteristics, thus allowing up to 36 dimensions to be displayed.

The stick figure visualization technique maps multidimensional data to 5-piece stick figures, where each figure has four limbs and a body. Two dimensions are mapped to the display ($x$ and $y$) axes and the remaining dimensions are mapped to the angle and/or length of the limbs. Figure 2.18 shows census data, where age and income are mapped to the display axes, and the remaining dimensions (gender, education, and so on) are mapped to stick figures. If the data items are relatively dense with respect to the two display dimensions, the resulting visualization shows texture patterns, reflecting data trends.

### 2.3.4 Hierarchical Visualization Techniques

The visualization techniques discussed so far focus on visualizing multiple dimensions simultaneously. However, for a large data set of high dimensionality, it would be difficult to visualize all dimensions at the same time. Hierar-
2.3. DATA VISUALIZATION

Figure 2.16: Visualization using parallel coordinates. Source: http://www.stat.columbia.edu/cook/movabletype/mlm/mdg1.png.

Figure 2.17: Chernoff faces. Each face represents an $n$-dimensional data point ($n \leq 18$).

**Chical visualization techniques** partition all dimensions into subsets (that is, subspaces). The subspaces are visualized in a hierarchical manner.

“**Worlds-within-Worlds**”, also known as $n$-Vision, is a representative hierarchical visualization method. Suppose we want to visualize a 6-dimensional data set, where the dimensions are $F, X_1, \ldots, X_5$. We want to observe how dimension $F$ changes with respect to the other dimensions. We can first fix the values of dimensions $X_3, X_4, X_5$ to some selected values, say, $c_3, c_4, c_5$. We can then visualize $F, X_1, X_2$ using a 3-dimensional plot, called a *world*, as shown in Figure 2.19. The position of the origin of the inner world is located at the point $(c_3, c_4, c_5)$ in the outer world, which is another 3-dimensional plot using dimensions $X_3, X_4, X_5$. A user can interactively change, in the outer world, the location of the origin of the inner world. The user then views the resulting changes of the inner world. Moreover, a user can vary the dimensions used in the inner world and the outer world. Given more dimensions, more levels of worlds
can be used, which is why the method is called “worlds-within-worlds”.

As another example of hierarchical visualization methods, treemaps display hierarchical data as a set of nested rectangles. For example, Figure 2.20 shows a treemap visualizing Google news stories. All news stories are organized into seven categories, each shown in a large rectangle of a unique color. Within each category (that is, each rectangle at the top level), the news stories are further partitioned into smaller subcategories.

### 2.3.5 Visualizing Complex Data and Relations

In early days, visualization techniques were mainly for numeric data. Recently, more and more non-numeric data, such as text and social networks, have become available. Visualizing and analyzing such data attracts a lot of interest.

There are many new visualization techniques dedicated to these kinds of data. For example, many people on the Web tag various objects such as pictures, blog entries, and product reviews. A tag cloud is a visualization of statistics of user-generated tags. Often, in a tag cloud, tags are listed alphabetically or in a user-preferred order. The importance of a tag is indicated by the font size or color. Figure 2.21 shows a tag cloud for visualizing the popular tags used in a Web site.

Tag clouds are often used in two ways. First, in a tag cloud for a single item, we can use the size of a tag to represent the number of times that the tag is applied to this item by different users. Second, when visualizing the tag statistics on multiple items, we can use the size of a tag to represent the number of items that the tag has been applied to, that is, the popularity of the tag.

In addition to complex data, complex relations among data entries also raise challenges for visualization. For example, Figure 2.22 uses a disease influence
graph to visualize the correlations between diseases. The nodes in the graph are diseases, and the size of each node is proportional to the prevalence of the corresponding disease. Two nodes are linked by an edge if the corresponding diseases have a strong correlation. The width of an edge is proportional to the strength of the correlation pattern of the two corresponding diseases.

In summary, visualization provides effective tools to explore data. We have introduced some popular methods and the essential ideas behind them. There are many existing tools and methods. Moreover, visualization can be used in data mining in various aspects. In addition to visualizing data, visualization can be used to represent the data mining process, the patterns obtained from a mining method, and user interaction with the data. Visual data mining is an important research and development direction.

2.4 Measuring Data Similarity and Dissimilarity

In data mining applications such as clustering, outlier analysis, and nearest-neighbor classification, we need ways of assessing how alike objects are, or how unlike they are in comparison to one another. For example, a store may like to search for clusters of customer objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age). Such information can then be used for marketing. A cluster is a collection of data objects such that the objects within a cluster are similar to one another and dissimilar to the objects in other clusters. Outlier analysis also employs clustering-based techniques to identify potential outliers as objects that are highly dissimilar to
others. Knowledge of object similarities can also be used in nearest-neighbor classification schemes where a given object (such as a patient) is assigned a class label (relating to, say, a diagnosis) based on its similarity towards other objects in the model.

This section presents measures of similarity and dissimilarity. Similarity and dissimilarity measures are referred to as measures of proximity. Similarity and dissimilarity are related. A similarity measure for two objects, \( i \) and \( j \), will typically return the value 0 if the objects are unalike. The higher the similarity value, the greater the similarity between objects. (Typically, a value of 1 indicates complete similarity, that is, that the objects are identical.) A dissimilarity measure works the opposite way. It returns a value of 0 if the objects are the same (and therefore, far from being dissimilar). The higher the dissimilarity value, the more dissimilar the two objects are.

We begin in Section 2.4.1 by presenting two data structures that are commonly used in the above types of applications. These are the data matrix (used to store the data objects) and the dissimilarity matrix (used to store dissimilarity
values for pairs of objects. We also switch to a different notation for data objects than previously used in this chapter since now we are dealing with objects described by more than one attribute. We then discuss how object dissimilarity can be computed for objects described by nominal attributes (Section 2.4.2); by binary attributes (Section 2.4.3); by numeric attributes (Section 2.4.4); by ordinal attributes (Section 2.4.5); or by combinations of these attribute types (Section 2.4.6). Section 2.4.7 provides similarity measures for very long and sparse data vectors, such as term-frequency vectors representing documents in information retrieval. Knowing how to compute dissimilarity is useful in studying attributes and will also be referenced in later topics on clustering (Chapters 10 and 11), outlier analysis (Chapter 12), and nearest-neighbor classification (Chapter 8).

2.4.1 Data Matrix vs. Dissimilarity Matrix

In Section 2.2, we looked at ways of studying the central tendency, dispersion, and spread of observed values for some attribute $X$. Our objects there were 1-dimensional, that is, described by a single attribute. In this section, we talk about objects described by multiple attributes. Therefore, we need a change in notation. Suppose that we have $n$ objects (such as persons, items, or courses) described by $p$ attributes (also called measurements or features), such as age, height, weight, or gender. The objects are $x_1 = (x_{11}, x_{12}, \ldots, x_{1p})$, $x_2 = (x_{21}, x_{22}, \ldots, x_{2p})$, and so on, where $x_{ij}$ is the value for object $x_i$ of the $j^{th}$...
attribute. For brevity, we hereafter refer to object $x_i$ as object $i$. The objects may be tuples in a relational database, and are also referred to as data samples or feature vectors.

Main memory-based clustering and nearest-neighbor algorithms typically operate on either of the following two data structures.

- **Data matrix** (or object-by-attribute structure): This structure stores the $n$ data objects in the form of a relational table, or $n$-by-$p$ matrix ($n$ objects $x_p$ attributes):

$$
\begin{bmatrix}
    x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\
    \vdots & \ddots & \vdots & \ddots & \vdots \\
    x_{n1} & \cdots & x_{nf} & \cdots & x_{np}
\end{bmatrix}
\quad (2.8)
$$

Each row corresponds to an object. As part of our notation, we may use $f$ to index through the $p$ attributes.

- **Dissimilarity matrix** (or object-by-object structure): This stores a collection of proximities that are available for all pairs of $n$ objects. It is often represented by an $n$-by-$n$ table:

$$
\begin{bmatrix}
    0 & d(2, 1) & 0 \\
    d(3, 1) & d(3, 2) & 0 \\
    \vdots & \vdots & \vdots \\
    d(n, 1) & d(n, 2) & \cdots & 0
\end{bmatrix}
\quad (2.9)
$$

Figure 2.22: The disease influence graph of people of age at least 20 in the NHANES data set.
2.4. MEASURING DATA SIMILARITY AND DISSIMILARITY

where \( d(i, j) \) is the measured **dissimilarity** or “difference” between objects \( i \) and \( j \). In general, \( d(i, j) \) is a nonnegative number that is close to 0 when objects \( i \) and \( j \) are highly similar or “near” each other, and becomes larger the more they differ. Note that \( d(i, i) = 0 \), that is, the difference between an object and itself is 0. Furthermore, \( d(i, j) = d(j, i) \). (For readability, we do not show the \( d(j, i) \) entries; the matrix is symmetric.) Measures of dissimilarity are discussed throughout the remainder of this chapter.

Measures of similarity can often be expressed as a function of measures of dissimilarity. For example, for nominal data,

\[
\text{sim}(i, j) = 1 - d(i, j)
\]

(2.10)

where \( \text{sim}(i, j) \) is the similarity between objects \( i \) and \( j \). Throughout the rest of this chapter, we will also comment on measures of similarity.

A data matrix is made up of two entities or “things”, namely rows (for objects) and columns (for attributes). Therefore, the data matrix is often called a **two-mode** matrix. The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a **one-mode** matrix. Many clustering and nearest-neighbor algorithms operate on a dissimilarity matrix. Data in the form of a data matrix can be transformed into a dissimilarity matrix before applying such algorithms.

### 2.4.2 Proximity Measures for Nominal Attributes

A nominal attribute can take on two or more states (Section 2.1.2). For example, \( \text{map-color} \) is a nominal attribute that may have, say, five states: \( \text{red, yellow, green, pink, blue} \).

Let the number of states of a nominal attribute be \( M \). The states can be denoted by letters, symbols, or a set of integers, such as \( 1, 2, \ldots, M \). Notice that such integers are used just for data handling and do not represent any specific ordering.

“How is dissimilarity computed between objects described by nominal attributes?” The dissimilarity between two objects \( i \) and \( j \) can be computed based on the ratio of mismatches:

\[
d(i, j) = \frac{p - m}{p},
\]

(2.11)

where \( m \) is the number of matches (i.e., the number of attributes for which \( i \) and \( j \) are in the same state), and \( p \) is the total number of attributes describing the objects. Weights can be assigned to increase the effect of \( m \) or to assign greater weight to the matches in attributes having a larger number of states.

**Example 2.17** **Dissimilarity between nominal attributes.** Suppose that we have the sample data of Table 2.2, except that only the **object-identifier** and the attribute
test-1 are available, where test-1 is nominal. (We will use test-2 and test-3 in later examples.) Let’s compute the dissimilarity matrix (2.9), that is,

\[
\begin{bmatrix}
0 & d(2, 1) & 0 \\
 d(3, 1) & d(3, 2) & 0 \\
 d(4, 1) & d(4, 2) & d(4, 3) & 0
\end{bmatrix}
\]

Since here we have one nominal attribute, test-1, we set \( p = 1 \) in Equation (2.11) so that \( d(i, j) \) evaluates to 0 if objects \( i \) and \( j \) match, and 1 if the objects differ. Thus, we get

\[
\begin{bmatrix}
0 & 1 & 0 \\
 1 & 1 & 0 \\
 0 & 1 & 1
\end{bmatrix}
\]

From this, we see that all objects are dissimilar to one another except objects 1 and 4 (that is, \( d(4, 1) = 0 \)).

Alternatively, similarity can be computed as

\[
sim(i, j) = 1 - d(i, j) = \frac{m}{p}.
\]

Proximity between objects described by nominal attributes can be computed using an alternative encoding scheme. Nominal attributes can be encoded by asymmetric binary attributes by creating a new binary attribute for each of the \( M \) states. For an object with a given state value, the binary attribute representing that state is set to 1, while the remaining binary attributes are set to 0. For example, to encode the nominal attribute \( \text{map}_\text{color} \), a binary attribute can be created for each of the five colors listed above. For an object having the color yellow, the \( \text{yellow} \) attribute is set to 1, while the remaining four attributes are set to 0. Proximity measures for this form of encoding can be calculated using the methods discussed in the following section.

<table>
<thead>
<tr>
<th>object identifier</th>
<th>test-1 (nominal)</th>
<th>test-2 (ordinal)</th>
<th>test-3 (numeric)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>code-A</td>
<td>excellent</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>code-B</td>
<td>fair</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>code-C</td>
<td>good</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>code-A</td>
<td>excellent</td>
<td>28</td>
</tr>
</tbody>
</table>
### 2.4.3 Proximity Measures for Binary Attributes

Let’s look at dissimilarity and similarity measures for objects described by either symmetric or asymmetric binary attributes.

Recall that a binary attribute has only two states: 0 or 1, where 0 means that the attribute is absent, and 1 means that it is present (Section 2.1.3). Given the attribute smoker describing a patient, for instance, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Treating binary attributes as if they are numeric can be misleading. Therefore, methods specific to binary data are necessary for computing dissimilarities.

“**So, how can we compute the dissimilarity between two binary attributes?**”

One approach involves computing a dissimilarity matrix from the given binary data. If all binary attributes are thought of as having the same weight, we have the 2-by-2 contingency table of Table 2.3, where $q$ is the number of attributes that equal 1 for both objects $i$ and $j$, $r$ is the number of attributes that equal 1 for object $i$ but that are 0 for object $j$, $s$ is the number of attributes that equal 0 for object $i$ but equal 1 for object $j$, and $t$ is the number of attributes that equal 0 for both objects $i$ and $j$. The total number of attributes is $p$, where $p = q + r + s + t$.

Recall that for symmetric binary attributes, each state is equally valuable. Dissimilarity that is based on symmetric binary attributes is called symmetric binary dissimilarity. If objects $i$ and $j$ are described by symmetric binary attributes, then the dissimilarity between $i$ and $j$ is

$$d(i, j) = \frac{r + s}{q + r + s + t}. \quad (2.13)$$

For asymmetric binary attributes, the two states are not equally important, such as the positive (1) and negative (0) outcomes of a disease test. Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match). Therefore, such binary attributes are often considered “monary” (as if having one state). The dissimilarity based on such attributes is called asymmetric binary dissimilarity, where the number of negative matches, $t$, is considered unimportant and thus is ignored in the computation, as shown in Equation (2.14).

$$d(i, j) = \frac{r + s}{q + r + s}. \quad (2.14)$$

---

**Table 2.3: A contingency table for binary attributes.**

<table>
<thead>
<tr>
<th>object $j$</th>
<th>1</th>
<th>0</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>object $i$</td>
<td>$q$</td>
<td>$r$</td>
<td>$q+r$</td>
</tr>
<tr>
<td>0</td>
<td>$s$</td>
<td>$t$</td>
<td>$s+t$</td>
</tr>
<tr>
<td>sum</td>
<td>$q+s$</td>
<td>$r+t$</td>
<td>$p$</td>
</tr>
</tbody>
</table>
Complementarily, we can measure the difference between two binary attributes based on the notion of similarity instead of dissimilarity. For example, the asymmetric binary similarity between the objects $i$ and $j$ can be computed as,

$$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j).$$

(2.15)

The coefficient $sim(i, j)$ of Equation (2.15) is called the Jaccard coefficient and is popularly referenced in the literature.

When both symmetric and asymmetric binary attributes occur in the same data set, the mixed attributes approach described in Section 2.4.6 can be applied.

Example 2.18 **Dissimilarity between binary attributes.** Suppose that a patient record table (Table 2.4) contains the attributes name, gender, fever, cough, test-1, test-2, test-3, and test-4, where name is an object identifier, gender is a symmetric attribute, and the remaining attributes are asymmetric binary.

For asymmetric attribute values, let the values $Y$ (yes) and $P$ (positive) be set to 1, and the value $N$ (no or negative) be set to 0. Suppose that the distance between objects (patients) is computed based only on the asymmetric attributes. According to Equation (2.14), the distance between each pair of the three patients, Jack, Mary, and Jim, is

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(Jim, Mary) = \frac{1+2}{1+1+2} = 0.75$$

These measurements suggest that Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs. Of the three patients, Jack and Mary are the most likely to have a similar disease.

<table>
<thead>
<tr>
<th>name</th>
<th>gender</th>
<th>fever</th>
<th>cough</th>
<th>test-1</th>
<th>test-2</th>
<th>test-3</th>
<th>test-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>M</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Jim</td>
<td>M</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Mary</td>
<td>F</td>
<td>Y</td>
<td>N</td>
<td>P</td>
<td>N</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: A relational table where patients are described by binary attributes.
2.4.4 Dissimilarity on Numeric Data: Minkowski Distance

In this section, we describe distance measures that are commonly used for computing the dissimilarity of objects described by numeric attributes. These measures include the **Euclidean**, **Manhattan** and **Minkowski** distances.

In some cases, the data are normalized before applying distance calculations. This involves transforming the data to fall within a smaller or common range, such as $[-1, 1]$ or $[0.0, 1.0]$. Consider a **height** attribute, for example, which could be measured in either meters or inches. In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such attributes greater effect or “weight”. Normalizing the data attempts to give all attributes an equal weight. It may or may not be useful in a particular application. Methods for normalizing data are discussed in detail in Chapter 3 on data preprocessing.

The most popular distance measure is the **Euclidean distance** (i.e., straight line or “as the crow flies” distance). Let $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ be two objects described by $p$ numeric attributes. The Euclidean distance between objects $i$ and $j$ is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}. \tag{2.16}$$

Another well-known measure is the **Manhattan (or city block) distance**, named so because it is the distance in blocks between any two points in a city (such as 2 blocks down and 3 blocks over for a total of 5 blocks). It is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|. \tag{2.17}$$

Both the Euclidean distance and Manhattan distance satisfy the following mathematical properties:

**Non-negativity:** $d(i, j) \geq 0$: Distance is a nonnegative number.

**Identity of indiscernibles:** $d(i, i) = 0$: The distance of an object to itself is 0.

**Symmetry:** $d(i, j) = d(j, i)$: Distance is a symmetric function.

**Triangle inequality:** $d(i, j) \leq d(i, k) + d(k, j)$: Going directly from object $i$ to object $j$ in space is no more than making a detour over any other object $k$ (triangular inequality).

A measure that satisfies these conditions is known as a **metric**. Please note that the non-negativity property is implied by the other three properties.

**Example 2.19** Euclidean distance and Manhattan distance. Let $x_1 = (1, 2)$ and $x_2 = (3, 5)$ represent two objects as in Figure 2.23. The Euclidean distance between the two is $\sqrt{2^2 + 3^2} = 3.61$. The Manhattan distance between the two is $2 + 3 = 5$. 
Minkowski distance is a generalization of both Euclidean distance and Manhattan distance. It is defined as

\[ d(i, j) = \sqrt[|h|]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}, \]  

(2.18)

The original before taking the student comment. where \( h \) is a real number such that \( h \geq 1 \). (Such a distance is also called \( L_p \) norm in some literature, where the symbol \( p \) refers to our notation of \( h \). We have kept \( p \) as the number of attributes to be consistent with the rest of this chapter.) It represents the Manhattan distance when \( h = 1 \) (i.e., \( L_1 \) norm) and Euclidean distance when \( h = 2 \) (i.e., \( L_2 \) norm).

The supremum distance (also referred to as \( L_{\text{max}} \), \( L_\infty \) norm, and the Chebyshev distance) is a generalization of Minkowski distance for \( h \to \infty \). To compute it, we find the attribute \( f \) that gives the maximum difference in values between the two objects. This difference is the supremum distance, defined more formally as:

\[ d(i, j) = \lim_{h \to \infty} \left( \sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f} |x_{if} - x_{jf}|. \]  

(2.19)

The \( L_\infty \) norm is also known as the uniform norm.

Example 2.20 Supremum distance. Let’s use the same two objects, \( x_1 = (1, 2) \) and \( x_2 = (3, 5) \), as in Figure 2.23. The second attribute gives the greatest difference in values for the objects, which is \( 5 - 2 = 3 \). This is the supremum distance between both objects.

If each attribute is assigned a weight according to its perceived importance, the weighted Euclidean distance can be computed as

\[ d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \cdots + w_m|x_{ip} - x_{jp}|^2}. \]  

(2.20)

Figure 2.23: Euclidean and Manhattan distances between two objects. NOTE: Add supremum distance = 5 - 2 = 3.
2.4. MEASURING DATA SIMILARITY AND DISSIMILARITY

Weighting can also be applied to other distance measures as well.

2.4.5 Proximity Measures for Ordinal Attributes

The values of an ordinal attribute have a meaningful order or ranking about them, yet the magnitude between successive values is unknown (Section 2.1.4). An example includes the sequence small, medium, large for a size attribute. Ordinal attributes may also be obtained from the discretization of numeric attributes by splitting the value range into a finite number of categories. These categories are organized into ranks. That is, the range of a numeric attribute can be mapped to an ordinal attribute \( f \) having \( M_f \) states. For example, the range of interval-scaled attribute \( \text{temperature} \) (in Celsius) can be organized into the following states: \([-30, -10), [-10, 10), [10, 30)\), representing the categories cold temperature, moderate temperature, and warm temperature, respectively. Let the number of possible states that an ordinal attribute can have be \( M \). These ordered states define the ranking \( 1, \ldots, M_f \).

“How are ordinal attributes handled?” The treatment of ordinal attributes is quite similar to that of numeric attributes when computing the dissimilarity between objects. Suppose that \( f \) is an attribute from a set of ordinal attributes describing \( n \) objects. The dissimilarity computation with respect to \( f \) involves the following steps:

1. The value of \( f \) for the \( i \)th object is \( x_{if} \), and \( f \) has \( M_f \) ordered states, representing the ranking \( 1, \ldots, M_f \). Replace each \( x_{if} \) by its corresponding rank, \( r_{if} \in \{1, \ldots, M_f\} \).

2. Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto \([0.0, 1.0]\) so that each attribute has equal weight. We perform such data normalization by replacing the rank \( r_{if} \) of the \( i \)th object in the \( f \)th attribute by

\[
    z_{if} = \frac{r_{if} - 1}{M_f - 1}.
\]  

(2.21)

3. Dissimilarity can then be computed using any of the distance measures described in Section 2.4.4 for numeric attributes, using \( z_{if} \) to represent the \( f \) value for the \( i \)th object.

Example 2.21 Dissimilarity between ordinal attributes. Suppose that we have the sample data of Table 2.2, except that this time only the object-identifier and the continuous ordinal attribute, test-2, are available. There are three states for test-2, namely fair, good, and excellent, that is \( M_f = 3 \). For step 1, if we replace each value for test-2 by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively. Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0. For step 3, we can use,
say, the Euclidean distance (Equation (2.16)), which results in the following dissimilarity matrix:

\[
\begin{bmatrix}
0 & 1.0 & 0 & 0.5 & 0.5 & 0 \\
1.0 & 0 & 0.5 & 0.5 & 0 & 1.0 \\
0 & 0.5 & 0 & 1.0 & 0.5 & 0 \\
0 & 0 & 0.5 & 0 & 1.0 & 0.5 \\
\end{bmatrix}
\]

Therefore, objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (that is, \(d(2, 1) = 1.0\) and \(d(4, 2) = 1.0\)). This makes intuitive sense since objects 1 and 4 are both excellent. Object 2 is fair, which is at the opposite end of the range of values for test-2.

Similarity values for ordinal attributes can be interpreted from dissimilarity as \(sim(i, j) = 1 - d(i, j)\).

### 2.4.6 Dissimilarity for Attributes of Mixed Types

Sections 2.4.2 to 2.4.5 discussed how to compute the dissimilarity between objects described by attributes of the same type, where these types may be either nominal, symmetric binary, asymmetric binary, numeric, or ordinal. However, in many real databases, objects are described by a mixture of attribute types. In general, a database can contain all of the attribute types listed above.

“So, how can we compute the dissimilarity between objects of mixed attribute types?” One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.

A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval \([0.0, 1.0]\).

Suppose that the data set contains \(p\) attributes of mixed type. The dissimilarity \(d(i, j)\) between objects \(i\) and \(j\) is defined as

\[
d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},
\]

where the indicator \(\delta_{ij}^{(f)} = 0\) if either (1) \(x_{if}\) or \(x_{jf}\) is missing (i.e., there is no measurement of attribute \(f\) for object \(i\) or object \(j\)), or (2) \(x_{if} = x_{jf} = 0\) and attribute \(f\) is asymmetric binary; otherwise, \(\delta_{ij}^{(f)} = 1\). The contribution of attribute \(f\) to the dissimilarity between \(i\) and \(j\), that is, \(d_{ij}^{(f)}\), is computed dependent on its type:
If \( f \) is numeric: \[
d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\text{max}_h x_{ih} - \text{min}_h x_{ih}},
\]
where \( h \) runs over all nonmissing objects for attribute \( f \).

If \( f \) is nominal or binary: \[
d_{ij}^{(f)} = 0 \text{ if } x_{if} = x_{jf}; \text{ otherwise } d_{ij}^{(f)} = 1.
\]

If \( f \) is ordinal: compute the ranks \( r_{if} \) and \( z_{if} = \frac{r_{if} - 1}{M_f - 1} \), and treat \( z_{if} \) as numeric.

The above steps are identical to what we have already seen for each of the individual attribute types. The only difference is for numeric attributes, where we normalize so that the values map to the interval \([0.0, 1.0] \). Thus, the dissimilarity between objects can be computed even when the attributes describing the objects are of different types.

**Example 2.22** Dissimilarity between attributes of mixed type. Let’s compute a dissimilarity matrix for the objects of Table 2.2. Now we will consider all of the attributes, which are of different types. In Examples 2.4.2 and 2.4.5, we worked out the dissimilarity matrices for each of the individual attributes. The procedures we followed for \( \text{test-1} \) (which is nominal) and \( \text{test-2} \) (which is ordinal) are the same as outlined above for processing attributes of mixed types. Therefore, we can use the dissimilarity matrices obtained for \( \text{test-1} \) and \( \text{test-2} \) later when we compute Equation (2.22). First, however, we need to compute the dissimilarity matrix for the third attribute, \( \text{test-3} \) (which is numeric). That is, we must compute \( d_{ij}^{(3)} \). Following the case for numeric attributes, we let \( \text{max}_h x_h = 64 \) and \( \text{min}_h x_h = 22 \). The difference between the two is used in Equation (2.22) to normalize the values of the dissimilarity matrix. The resulting dissimilarity matrix for \( \text{test-3} \) is:

\[
\begin{bmatrix}
0 & 0.55 & 0.45 & 0.40 \\
0.55 & 0 & 1.00 & 0.14 \\
0.45 & 1.00 & 0 & 0.86 \\
0.40 & 0.14 & 0.86 & 0
\end{bmatrix}
\]

We can now use the dissimilarity matrices for the three attributes in our computation of Equation (2.22). The indicator \( \delta_{ij}^{(f)} = 1 \) for each of the three attributes, \( f \). We get, for example, \( d(3, 1) = \frac{1(1)+1(0.50)+1(0.45)}{3} = 0.65 \). The resulting dissimilarity matrix obtained for the data described by the three attributes of mixed types is:

\[
\begin{bmatrix}
0 & 0.85 & 0.65 & 0.13 \\
0.85 & 0 & 0.83 & 0.71 \\
0.65 & 0.83 & 0 & 0.79 \\
0.13 & 0.71 & 0.79 & 0
\end{bmatrix}
\]

If we go back and look at Table 2.2, we can intuitively guess that objects 1 and 4 are the most similar, based on their values for \( \text{test-1} \) and \( \text{test-2} \). This is
confirmed by the dissimilarity matrix, where $d(4, 1)$ is the lowest value for any pair of different objects. Similarly, the matrix indicates that objects 1 and 2 are the least similar.

2.4.7 Cosine Similarity

A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as keywords) or phrase in the document. Thus, each document is an object represented by a what is called a term-frequency vector. For example, in Table 2.5, we see that Document1 contains five instances of the word team, while hockey occurs three times. The word coach is absent from the entire document, as indicated by a count value of 0. Such data can be highly asymmetric. Term-frequency vectors are typically very long and sparse (that is, having many 0 values). Applications using such structures include information retrieval, text document clustering, biological taxonomy, and gene feature mapping. The traditional distance measures that we have studied in this chapter do not work well for such sparse numeric data. For example, two term-frequency vectors may have many 0 values in common, meaning that the corresponding documents do not share many words, but this does not make them similar. We need a measure that will focus on the words that the two document do have in common, and the occurrence frequency of such words. In other words, we need a measure for numeric data that ignores 0-matches.

**Cosine similarity** is a measure of similarity that can be used to compare documents, or, say, give a ranking of documents with respect to a given vector of query words. Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors for comparison. Using the cosine measure as a similarity function, we have:

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| ||\mathbf{y}||}, \quad (2.23)$$

where $||\mathbf{x}||$ is the Euclidean norm of vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, defined as $\sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$. Conceptually, it is the length of the vector. Similarly, $||\mathbf{y}||$ is the Euclidean norm of vector $\mathbf{y}$. The measure computes the cosine of the angle between vectors $\mathbf{x}$ and $\mathbf{y}$. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match.
The closer the cosine value to 1, the smaller the angle and the greater the match between vectors. Note that because the cosine similarity measure does not obey all of the properties of Section 2.4.4 defining metric measures, it is therefore referred to as a nonmetric measure.

Example 2.23 Cosine similarity between two term-frequency vectors. Suppose that \( x \) and \( y \) are the first two term-frequency vectors of Table 2.5. That is, \( x = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0) \) and \( y = (3, 0, 2, 0, 1, 0, 1, 0, 1, 1) \). How similar are \( x \) and \( y \)? Using Equation (2.23) to compute the cosine similarity between the two vectors, we get:

\[
x^t \cdot y = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25
\]
\[
||x|| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48
\]
\[
||y|| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12
\]
\[
sim(x, y) = 0.94
\]

Therefore, if we were using the cosine similarity measure to compare these documents, they would be considered quite similar.

When attributes are binary-valued, the cosine similarity function can be interpreted in terms of shared features or attributes. Suppose an object \( x \) possesses the \( i^{th} \) attribute if \( x_i = 1 \). Then \( x^t \cdot y \) is the number of attributes possessed (that is, shared) by both \( x \) and \( y \), and \( ||x|| \) is the geometric mean of the number of attributes possessed by \( x \) and the number possessed by \( y \). Thus \( sim(x, y) \) is a measure of relative possession of common attributes.

A simple variation of cosine similarity for the above scenario is

\[
sim(x, y) = \frac{x \cdot y}{x \cdot x + y \cdot y - x \cdot y} \quad (2.24)
\]

which is the ratio of the number of attributes shared by \( x \) and \( y \) to the number of attributes possessed by \( x \) or \( y \). This function, known as the Tanimoto coefficient or Tanimoto distance, is frequently used in information retrieval and biology taxonomy.

2.5 Summary

- Data sets are made up of data objects. A data object represents an entity. Data objects are described by attributes. Attributes can be nominal, binary, ordinal, or numeric.

- The values of a nominal (or categorical) attribute are symbols or names of things, where each value represents some kind of category, code, or state.
Binary attributes are nominal attributes with only two possible states (such as 1 and 0, or true and false). If the two states are equally important, the attribute is a symmetric binary attribute, otherwise it is an asymmetric binary attribute.

- An ordinal attribute is an attribute whose possible values have a meaningful order or ranking among them, but the magnitude between successive values is not known.

- A numeric attribute is quantitative, that is, it is a measurable quantity, represented in integer or real values. Numeric attribute types can be interval-scaled or ratio-scaled. The values of an interval-scaled attribute are measured in fixed and equal units. Ratio-scaled attributes are numeric attributes with an inherent zero-point. Measurements are ratio-scaled in that we can speak of values as being an order of magnitude larger than the unit of measurement.

- Basic statistical descriptions provide the analytical foundation for data preprocessing. The basic statistical measures for data summarization include mean, weighted mean, median, and mode for measuring the central tendency of data; and range, quantiles, quartiles, interquartile range, variance, and standard deviation for measuring the dispersion of data. Graphical representations, such as boxplots, quantile plots, quantile-quantile plots, histograms, and scatter plots facilitate visual inspection of the data and are thus useful for data preprocessing and mining.

- Data visualization techniques may be pixel-oriented, geometric-based, icon-based, or hierarchical. These methods apply to multidimensional relational data. Additional techniques have been proposed for the visualization of complex data, such as text and social networks.

- Measures of object similarity and dissimilarity are used in data mining applications such as clustering, outlier analysis, and nearest-neighbor classification. Such measures of proximity can be computed for each of the attribute types studied in this chapter, or combinations of such attributes. Examples include the Jaccard coefficient for asymmetric binary attributes; and Euclidean, Manhattan, Minkowski, and supremum distances for numeric attributes. For applications involving sparse numeric data vectors, such as term-frequency vectors, the cosine measure and the Tanimoto coefficient are often used in the assessment of similarity.
2.6 Exercises

1. Give three additional commonly used statistical measures (i.e., not illustrated in this chapter) for the characterization of data dispersion, and discuss how they can be computed efficiently in large databases.

2. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?
(b) What is the mode of the data? Comment on the data’s modality (i.e., bimodal, trimodal, etc.).
(c) What is the midrange of the data?
(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?
(e) Give the five-number summary of the data.
(f) Show a boxplot of the data.
(g) How is a quantile-quantile plot different from a quantile plot?

3. Suppose that the values for a given set of data are grouped into intervals. The intervals and corresponding frequencies are as follows.

\[
\begin{array}{|c|c|}
\hline
\text{age} & \text{frequency} \\
\hline
1-5 & 200 \\
6-15 & 450 \\
16-20 & 300 \\
21-50 & 1500 \\
51-80 & 700 \\
81-110 & 44 \\
\hline
\end{array}
\]

Compute an approximate median value for the data.

4. Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\text{age} & 23 & 23 & 27 & 27 & 39 & 41 & 47 & 49 & 50 \\
\text{%fat} & 9.5 & 26.5 & 7.8 & 17.8 & 31.4 & 25.9 & 27.4 & 27.2 & 31.2 \\
\hline
\text{age} & 52 & 54 & 54 & 56 & 57 & 58 & 58 & 60 & 61 \\
\text{%fat} & 34.6 & 42.5 & 28.8 & 33.4 & 30.2 & 34.1 & 32.9 & 41.2 & 35.7 \\
\hline
\end{array}
\]

(a) Calculate the mean, median and standard deviation of age and %fat.
(b) Draw the boxplots for age and %fat.
(c) Draw a scatter plot and a q-q plot based on these two variables.
5. Briefly outline how to compute the dissimilarity between objects described by the following:

(a) Nominal attributes
(b) Asymmetric binary attributes
(c) Numeric attributes
(d) Term-frequency vectors

6. Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the Euclidean distance between the two objects.
(b) Compute the Manhattan distance between the two objects.
(c) Compute the Minkowski distance between the two objects, using $q = 3$.
(d) Compute the supremum distance between the two objects.

7. The median is one of the most important holistic measures in data analysis. Propose several methods for median approximation. Analyze their respective complexity under different parameter settings and decide to what extent the real value can be approximated. Moreover, suggest a heuristic strategy to balance between accuracy and complexity and then apply it to all methods you have given.

8. It is important to define or select similarity measures in data analysis. However, there is no commonly-accepted subjective similarity measure. Results can vary depending on the similarity measures used. Nonetheless, seemingly different similarity measures may be equivalent after some transformation.

Suppose we have the following two-dimensional data set:

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1.5</td>
<td>1.7</td>
</tr>
<tr>
<td>$x_2$</td>
<td>2</td>
<td>1.9</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1.2</td>
<td>1.5</td>
</tr>
<tr>
<td>$x_5$</td>
<td>1.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

(a) Consider the data as two-dimensional data points. Given a new data point, $x = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using Euclidean distance, Manhattan distance, supremum distance, and cosine similarity.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.
2.7 Bibliographic Notes

Methods for descriptive data summarization have been studied in the statistics literature long before the onset of computers. Good summaries of statistical descriptive data mining methods include Freedman, Pisani and Purves [FPP07], and Devore [Dev95]. For statistics-based visualization of data using boxplots, quantile plots, quantile-quantile plots, scatter plots, and loess curves, see Cleveland [Cle93].


There have been many graphical user interfaces and visualization tools developed for data mining, which can be found in various data mining products. Several books on data mining, such as Data Mining Solutions by Westphal and Blaxton [WB98], present many good examples and visual snapshots. For a survey of visualization techniques, see “Visual techniques for exploring databases” by Keim [Kei97].

Similarity and distance measures among various variables have been introduced in many textbooks that study cluster analysis, including Hartigan [Har75], Jain and Dubes [JD88], Kaufman and Rousseeuw [KR90], and Arabie, Hubert, and De Sorte [AHS96]. Methods for combining attributes of different types into a single dissimilarity matrix were introduced by Kaufman and Rousseeuw [KR90].
Bibliography


