Data for data mining comes in many forms: from computer files typed in by human operators, business information in SQL or some other standard database format, information recorded automatically by equipment such as fault logging devices, to streams of binary data transmitted from satellites. For purposes of data mining (and for the remainder of this book) we will assume that the data takes a particular standard form which is described in the next section. We will look at some of the practical problems of data preparation in Section 2.3.

2.1 Standard Formulation

We will assume that for any data mining application we have a universe of objects that are of interest. This rather grandiose term often refers to a collection of people, perhaps all human beings alive or dead, or possibly all the patients at a hospital, but may also be applied to, say, all dogs in England, or to inanimate objects such as all train journeys from London to Birmingham, all the rocks on the moon or all the pages stored in the World Wide Web.

The universe of objects is normally very large and we have only a small part of it. Usually we want to extract information from the data available to us that we hope is applicable to the large volume of data that we have not yet seen.

Each object is described by a number of variables that correspond to its properties. In data mining variables are often called attributes. We will use both terms in this book.
The set of variable values corresponding to each of the objects is called a *record* or (more commonly) an *instance*. The complete set of data available to us for an application is called a *dataset*. A dataset is often depicted as a table, with each row representing an instance. Each column contains the value of one of the variables (attributes) for each of the instances. A typical example of a dataset is the ‘degrees’ data given in the Introduction (Figure 2.1).

<table>
<thead>
<tr>
<th>SoftEng</th>
<th>ARIN</th>
<th>HCI</th>
<th>CSA</th>
<th>Project</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>Second</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>Second</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>Second</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>First</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>First</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>Second</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>First</td>
</tr>
</tbody>
</table>

**Figure 2.1** The Degrees Dataset

This dataset is an example of *labelled* data, where one attribute is given special significance and the aim is to predict its value. In this book we will give this attribute the standard name ‘class’. When there is no such significant attribute we call the data *unlabelled*.

### 2.2 Types of Variable

In general there are many types of variable that can be used to measure the properties of an object. A lack of understanding of the differences between the various types can lead to problems with any form of data analysis. At least six main types of variable can be distinguished.

**Nominal Variables**

A variable used to put objects into categories, e.g. the name or colour of an object. A nominal variable may be numerical in form, but the numerical values have no mathematical interpretation. For example we might label 10 people as numbers 1, 2, 3, …, 10, but any arithmetic with such values, e.g. $1 + 2 = 3$
would be meaningless. They are simply labels. A **classification** can be viewed as a nominal variable which has been designated as of particular importance.

**Binary Variables**

A binary variable is a special case of a nominal variable that takes only two possible values: true or false, 1 or 0 etc.

**Ordinal Variables**

Ordinal variables are similar to nominal variables, except that an ordinal variable has values that can be arranged in a meaningful order, e.g. small, medium, large.

**Integer Variables**

Integer variables are ones that take values that are genuine integers, for example ‘number of children’. Unlike nominal variables that are numerical in form, arithmetic with integer variables is meaningful (1 child + 2 children = 3 children etc.).

**Interval-scaled Variables**

Interval-scaled variables are variables that take numerical values which are measured at equal intervals from a zero point or origin. However the origin does not imply a true absence of the measured characteristic. Two well-known examples of interval-scaled variables are the Fahrenheit and Celsius temperature scales. To say that one temperature measured in degrees Celsius is greater than another or greater than a constant value such as 25 is clearly meaningful, but to say that one temperature measured in degrees Celsius is twice another is meaningless. It is true that a temperature of 20 degrees is twice as far from the zero value as 10 degrees, but the zero value has been selected arbitrarily and does not imply ‘absence of temperature’. If the temperatures are converted to an equivalent scale, say degrees Fahrenheit, the ‘twice’ relationship will no longer apply.
**Ratio-scaled Variables**

Ratio-scaled variables are similar to interval-scaled variables except that the zero point does reflect the absence of the measured characteristic, for example Kelvin temperature and molecular weight. In the former case the zero value corresponds to the lowest possible temperature ‘absolute zero’, so a temperature of 20 degrees Kelvin is twice one of 10 degrees Kelvin. A weight of 10 kg is twice one of 5 kg, a price of 100 dollars is twice a price of 50 dollars etc.

**2.2.1 Categorical and Continuous Attributes**

Although the distinction between different categories of variable can be important in some cases, many practical data mining systems divide attributes into just two types:

- **categorical** corresponding to nominal, binary and ordinal variables
- **continuous** corresponding to integer, interval-scaled and ratio-scaled variables.

This convention will be followed in this book. For many applications it is helpful to have a third category of attribute, the ‘ignore’ attribute, corresponding to variables that are of no significance for the application, for example the name of a patient in a hospital or the serial number of an instance, but which we do not wish to (or are unable to) delete from the dataset.

It is important to choose methods that are appropriate to the types of variable stored for a particular application. The methods described in this book are applicable to categorical and continuous attributes as defined above. There are other types of variable to which they would not be applicable without modification, for example any variable that is measured on a logarithmic scale. Two examples of logarithmic scales are the Richter scale for measuring earthquakes (an earthquake of magnitude 6 is 10 times more severe than one of magnitude 5, 100 times more severe than one of magnitude 4 etc.) and the Stellar Magnitude Scale for measuring the brightness of stars viewed by an observer on Earth.

**2.3 Data Preparation**

Although this book is about data mining not data preparation, some general comments about the latter may be helpful.
For many applications the data can simply be extracted from a database in the form described in Section 2.1, perhaps using a standard access method such as ODBC. However, for some applications the hardest task may be to get the data into a standard form in which it can be analysed. For example data values may have to be extracted from textual output generated by a fault logging system or (in a crime analysis application) extracted from transcripts of interviews with witnesses. The amount of effort required to do this may be considerable.

2.3.1 Data Cleaning

Even when the data is in the standard form it cannot be assumed that it is error free. In real-world datasets erroneous values can be recorded for a variety of reasons, including measurement errors, subjective judgements and malfunctioning or misuse of automatic recording equipment.

Erroneous values can be divided into those which are possible values of the attribute and those which are not. Although usage of the term noise varies, in this book we will take a noisy value to mean one that is valid for the dataset, but is incorrectly recorded. For example the number 69.72 may accidentally be entered as 6.972, or a categorical attribute value such as brown may accidentally be recorded as another of the possible values, such as blue. Noise of this kind is a perpetual problem with real-world data.

A far smaller problem arises with noisy values that are invalid for the dataset, such as 69.7X for 6.972 or bbrown for brown. We will consider these to be invalid values, not noise. An invalid value can easily be detected and either corrected or rejected.

It is hard to see even very ‘obvious’ errors in the values of a variable when they are ‘buried’ amongst say 100,000 other values. In attempting to ‘clean up’ data it is helpful to have a range of software tools available, especially to give an overall visual impression of the data, when some anomalous values or unexpected concentrations of values may stand out. However, in the absence of special software, even some very basic analysis of the values of variables may be helpful. Simply sorting the values into ascending order (which for fairly small datasets can be accomplished using just a standard spreadsheet) may reveal unexpected results. For example:

- A numerical variable may only take six different values, all widely separated. It would probably be best to treat this as a categorical variable rather than a continuous one.
- All the values of a variable may be identical. The variable should be treated as an ‘ignore’ attribute.
− All the values of a variable except one may be identical. It is then necessary to decide whether the one different value is an error or a significantly different value. In the latter case the variable should be treated as a categorical attribute with just two values.

− There may be some values that are outside the normal range of the variable. For example, the values of a continuous attribute may all be in the range 200 to 5000 except for the highest three values which are 22654.8, 38597 and 44625.7. If the data values were entered by hand a reasonable guess is that the first and third of these abnormal values resulted from pressing the initial key twice by accident and the second one is the result of leaving out the decimal point. If the data were recorded automatically it may be that the equipment malfunctioned. This may not be the case but the values should certainly be investigated.

− We may observe that some values occur an abnormally large number of times. For example if we were analysing data about users who registered for a web-based service by filling in an online form we might notice that the ‘country’ part of their addresses took the value ‘Albania’ in 10% of cases. It may be that we have found a service that is particularly attractive to inhabitants of that country. Another possibility is that users who registered either failed to choose from the choices in the country field, causing a (not very sensible) default value to be taken, or did not wish to supply their country details and simply selected the first value in a list of options. In either case it seems likely that the rest of the address data provided for those users may be suspect too.

− If we are analysing the results of an online survey collected in 2002, we may notice that the age recorded for a high proportion of the respondents was 72. This seems unlikely, especially if the survey was of student satisfaction, say. A possible interpretation for this is that the survey had a ‘date of birth’ field, with subfields for day, month and year and that many of the respondents did not bother to override the default values of 01 (day), 01 (month) and 1930 (year). A poorly designed program then converted the date of birth to an age of 72 before storing it in the database.

It is important to issue a word of caution at this point. Care is needed when dealing with anomalous values such as 22654.8, 38597 and 44625.7 in one of the examples above. They may simply be errors as suggested. Alternatively they may be outliers, i.e. genuine values that are significantly different from the others. The recognition of outliers and their significance may be the key to major discoveries, especially in fields such as medicine and physics, so we need
to be careful before simply discarding them or adjusting them back to ‘normal’ values.

2.4 Missing Values

In many real-world datasets data values are not recorded for all attributes. This can happen simply because there are some attributes that are not applicable for some instances (e.g. certain medical data may only be meaningful for female patients or patients over a certain age). The best approach here may be to divide the dataset into two (or more) parts, e.g. treating male and female patients separately.

It can also happen that there are attribute values that should be recorded that are missing. This can occur for several reasons, for example:

- a malfunction of the equipment used to record the data
- a data collection form to which additional fields were added after some data had been collected
- information that could not be obtained, e.g. about a hospital patient.

There are several possible strategies for dealing with missing values. Two of the most commonly used are as follows.

2.4.1 Discard Instances

This is the simplest strategy: delete all instances where there is at least one missing value and use the remainder.

This strategy is a very conservative one, which has the advantage of avoiding introducing any data errors. Its disadvantage is that discarding data may damage the reliability of the results derived from the data. Although it may be worth trying when the proportion of missing values is small, it is not recommended in general. It is clearly not usable when all or a high proportion of all the instances have missing values.

2.4.2 Replace by Most Frequent/Average Value

A less cautious strategy is to estimate each of the missing values using the values that are present in the dataset.
A straightforward but effective way of doing this for a categorical attribute is to use its most frequently occurring (non-missing) value. This is easy to justify if the attribute values are very unbalanced. For example if attribute $X$ has possible values $a$, $b$ and $c$ which occur in proportions 80%, 15% and 5% respectively, it seems reasonable to estimate any missing values of attribute $X$ by the value $a$. If the values are more evenly distributed, say in proportions 40%, 30% and 30%, the validity of this approach is much less clear.

In the case of continuous attributes it is likely that no specific numerical value will occur more than a small number of times. In this case the estimate used is generally the *average* value.

Replacing a missing value by an estimate of its true value may of course introduce noise into the data, but if the proportion of missing values for a variable is small, this is not likely to have more than a small effect on the results derived from the data. However, it is important to stress that if a variable value is not meaningful for a given instance or set of instances any attempt to replace the ‘missing’ values by an estimate is likely to lead to invalid results. Like many of the methods in this book the ‘replace by most frequent/average value’ strategy has to be used with care.

There are other approaches to dealing with missing values, for example using the ‘association rule’ methods described in Chapter 16 to make a more reliable estimate of each missing value. However, as is generally the case in this field, there is no one method that is more reliable than all the others for all possible datasets and in practice there is little alternative to experimenting with a range of alternative strategies to find the one that gives the best results for a dataset under consideration.

### 2.5 Reducing the Number of Attributes

In some data mining application areas the availability of ever-larger storage capacity at a steadily reducing unit price has led to large numbers of attribute values being stored for every instance, e.g. information about all the purchases made by a supermarket customer for three months or a large amount of detailed information about every patient in a hospital. For some datasets there can be substantially more attributes than there are instances, perhaps as many as 10 or even 100 to one.

Although it is tempting to store more and more information about each instance (especially as it avoids making hard decisions about what information is really needed) it risks being self-defeating. Suppose we have 10,000 pieces of information about each supermarket customer and want to predict which
customers will buy a new brand of dog food. The number of attributes of any relevance to this is probably very small. At best the many irrelevant attributes will place an unnecessary computational overhead on any data mining algorithm. At worst, they may cause the algorithm to give poor results.

Of course, supermarkets, hospitals and other data collectors will reply that they do not necessarily know what is relevant or will come to be recognised as relevant in the future. It is safer for them to record everything than risk throwing away important information.

Although faster processing speeds and larger memories may make it possible to process ever larger numbers of attributes, this is inevitably a losing struggle in the long term. Even if it were not, when the number of attributes becomes large, there is always a risk that the results obtained will have only superficial accuracy and will actually be less reliable than if only a small proportion of the attributes were used—a case of ‘more means less’.

There are several ways in which the number of attributes (or ‘features’) can be reduced before a dataset is processed. The term feature reduction or dimension reduction is generally used for this process. We will return to this topic in Chapter 10.

2.6 The UCI Repository of Datasets

Most of the commercial datasets used by companies for data mining are—unsurprisingly—not available for others to use. However there are a number of ‘libraries’ of datasets that are readily available for downloading from the World Wide Web free of charge by anyone.

The best known of these is the ‘Repository’ of datasets maintained by the University of California at Irvine, generally known as the ‘UCI Repository’ [1]. The URL for the Repository is http://www.ics.uci.edu/~mlearn/MLRepository.html. It contains over 350 datasets on topics as diverse as predicting the age of abalone from physical measurements, predicting good and bad credit risks, classifying patients with a variety of medical conditions and learning concepts from the sensor data of a mobile robot. Some datasets are complete, i.e. include all possible instances, but most are relatively small samples from a much larger number of possible instances. Datasets with missing values and noise are included.

The UCI site also has links to other repositories of both datasets and programs, maintained by a variety of organisations such as the (US) National Space Science Center, the US Bureau of Census and the University of Toronto.
The datasets in the UCI Repository were collected principally to enable data mining algorithms to be compared on a standard range of datasets. There are many new algorithms published each year and it is standard practice to state their performance on some of the better-known datasets in the UCI Repository. Several of these datasets will be described later in this book.

The availability of standard datasets is also very helpful for new users of data mining packages who can gain familiarisation using datasets with published performance results before applying the facilities to their own datasets.

In recent years a potential weakness of establishing such a widely used set of standard datasets has become apparent. In the great majority of cases the datasets in the UCI Repository give good results when processed by standard algorithms of the kind described in this book. Datasets that lead to poor results tend to be associated with unsuccessful projects and so may not be added to the Repository. The achievement of good results with selected datasets from the Repository is no guarantee of the success of a method with new data, but experimentation with such datasets can be a valuable step in the development of new methods.

A welcome relatively recent development is the creation of the UCI ‘Knowledge Discovery in Databases Archive’ at http://kdd.ics.uci.edu. This contains a range of large and complex datasets as a challenge to the data mining research community to scale up its algorithms as the size of stored datasets, especially commercial ones, inexorably rises.

2.7 Chapter Summary

This chapter introduces the standard formulation for the data input to data mining algorithms that will be assumed throughout this book. It goes on to distinguish between different types of variable and to consider issues relating to the preparation of data prior to use, particularly the presence of missing data values and noise. The UCI Repository of datasets is introduced.

2.8 Self-assessment Exercises for Chapter 2

Specimen solutions to self-assessment exercises are given in Appendix E.

1. What is the difference between labelled and unlabelled data?

2. The following information is held in an employee database.
Name, Date of Birth, Sex, Weight, Height, Marital Status, Number of Children

What is the type of each variable?

3. Give two ways of dealing with missing data values.

Reference

Principles of Data Mining
Bramer, M.
2016, XV, 526 p. 123 illus., Softcover