CHAPTER

What's It All About?

1

Human *in vitro* fertilization involves collecting several eggs from a woman's ovaries, which, after fertilization with partner or donor sperm, produce several embryos. Some of these are selected and transferred to the woman's uterus. The problem is to select the "best" embryos to use—the ones that are most likely to survive. Selection is based on around 60 recorded features of the embryos—characterizing their morphology, oocyte, follicle, and the sperm sample. The number of features is sufficiently large that it is difficult for an embryologist to assess them all simultaneously and correlate historical data with the crucial outcome of whether that embryo did or did not result in a live child. In a research project in England, machine learning is being investigated as a technique for making the selection, using as training data historical records of embryos and their outcome.

Every year, dairy farmers in New Zealand have to make a tough business decision: which cows to retain in their herd and which to sell off to an abattoir. Typically, one-fifth of the cows in a dairy herd are culled each year near the end of the milking season as feed reserves dwindle. Each cow's breeding and milk production history influences this decision. Other factors include age (a cow is nearing the end of its productive life at 8 years), health problems, history of difficult calving, undesirable temperament traits (kicking or jumping fences), and not being in calf for the following season. About 700 attributes for each of several million cows have been recorded over the years. Machine learning is being investigated as a way of ascertaining which factors are taken into account by successful farmers—not to automate the decision but to propagate their skills and experience to others.

Life and death. From Europe to the antipodes. Family and business. Machine learning is a burgeoning new technology for mining knowledge from data, a technology that a lot of people are starting to take seriously.

1.1 DATA MINING AND MACHINE LEARNING

We are overwhelmed with data. The amount of data in the world, in our lives, continues to increase—and there's no end in sight. Omnipresent personal computers make it too easy to save things that previously we would have trashed. Inexpensive multigigabyte disks make it too easy to postpone decisions about what to do with all this stuff—we simply buy another disk and keep it all. Ubiquitous electronics record our decisions, our choices in the supermarket, our financial habits, our comings and goings. We swipe our way through the world, every swipe a record in a database. The World Wide Web overwhelms us with information; meanwhile, every choice we make is recorded. And all these are just personal choices: they have countless counterparts in the world of commerce and industry. We would all testify to the growing gap between the *generation* of data and our *understanding* of it. As the volume of data increases, inexorably, the proportion of it that people understand decreases, alarmingly. Lying hidden in all this data is information, potentially useful information, that is rarely made explicit or taken advantage of.

This book is about looking for patterns in data. There is nothing new about this. People have been seeking patterns in data since human life began. Hunters seek patterns in animal migration behavior, farmers seek patterns in crop growth, politicians seek patterns in voter opinion, and lovers seek patterns in their partners' responses. A scientist's job (like a baby's) is to make sense of data, to discover the patterns that govern how the physical world works and encapsulate them in theories that can be used for predicting what will happen in new situations. The entrepreneur's job is to identify opportunities, that is, patterns in behavior that can be turned into a profitable business, and exploit them.

In *data mining,* the data is stored electronically and the search is automated—or at least augmented—by computer. Even this is not particularly new. Economists, statisticians, forecasters, and communication engineers have long worked with the idea that patterns in data can be sought automatically, identified, validated, and used for prediction. What is new is the staggering increase in opportunities for finding patterns in data. The unbridled growth of databases in recent years, databases on such everyday activities as customer choices, brings data mining to the forefront of new business technologies. It has been estimated that the amount of data stored in the world's databases doubles every 20 months, and although it would surely be difficult to justify this figure in any quantitative sense, we can all relate to the pace of growth qualitatively. As the flood of data swells and machines that can undertake the searching become commonplace, the opportunities for data mining increase. As the world grows in complexity, overwhelming us with the data it generates, data mining becomes our only hope for elucidating the patterns that underlie it. Intelligently analyzed data is a valuable resource. It can lead to new insights and, in commercial settings, to competitive advantages.

Data mining is about solving problems by analyzing data already present in databases. Suppose, to take a well-worn example, the problem is fickle customer loyalty in a highly competitive marketplace. A database of customer choices, along with customer profiles, holds the key to this problem. Patterns of behavior of former customers can be analyzed to identify distinguishing characteristics of those likely to switch products and those likely to remain loyal. Once such characteristics are found, they can be put to work to identify present customers who are likely to jump ship. This group can be targeted for special treatment, treatment too costly to apply to the customer base as a whole. More positively, the same techniques can be used to identify customers who might be attracted to another service the enterprise provides, one they are not presently enjoying, to target them for special offers that promote this service. In today's highly competitive, customer-centered, service-oriented economy, data is the raw material that fuels business growth—if only it can be mined.

Data mining is defined as the process of discovering patterns in data. The process must be automatic or (more usually) semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage, usually an economic advantage. The data is invariably present in substantial quantities.

How are the patterns expressed? Useful patterns allow us to make nontrivial predictions on new data. There are two extremes for the expression of a pattern: as a black box whose innards are effectively incomprehensible and as a transparent box whose construction reveals the structure of the pattern. Both, we are assuming, make good predictions. The difference is whether or not the patterns that are mined are represented in terms of a structure that can be examined, reasoned about, and used to inform future decisions. Such patterns we call *structural* because they capture the decision structure in an explicit way. In other words, they help to explain something about the data.

Now, finally, we can say what this book is about. It is about techniques for finding and describing structural patterns in data. Most of the techniques that we cover have developed within a field known as *machine learning*. But first let us look at what structural patterns are.

1.1.1 Describing Structural Patterns

What is meant by *structural patterns?* How do you describe them? And what form does the input take? We will answer these questions by way of illustration rather than by attempting formal, and ultimately sterile, definitions. We will present plenty of examples later in this chapter, but let's examine one right now to get a feeling for what we're talking about.

Look at the contact lens data in Table 1.1. This gives the conditions under which an optician might want to prescribe soft contact lenses, hard contact lenses, or no contact lenses at all; we will say more about what the individual features mean later. Each line of the table is one of the examples. Part of a structural description of this information might be as follows:

```
If tear production rate = reduced then recommendation = none
Otherwise, if age = young and astigmatic = no
        then recommendation = soft
```

Structural descriptions need not necessarily be couched as rules such as these. Decision trees, which specify the sequences of decisions that need to

Table 1.1 The Contact Lens Data

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	Hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

be made and the resulting recommendation, are another popular means of expression.

This example is a simplistic one. First, all combinations of possible values are represented in the table. There are 24 rows, representing three possible values of age and two values each for spectacle prescription, astigmatism, and tear production rate ($3 \times 2 \times 2 \times 2 = 24$). The rules do not really generalize from the data; they merely summarize it. In most learning situations, the set of examples given as input is far from complete, and part of the job is to generalize to other, new examples. You can imagine omitting some of the rows in the table for which tear production rate is *reduced* and still coming up with the rule

```
If tear production rate = reduced then recommendation = none
```

which would generalize to the missing rows and fill them in correctly. Second, values are specified for all the features in all the examples. Real-life datasets invariably contain examples in which the values of some features, for some reason or other, are unknown—for example, measurements were not taken or were lost. Third, the preceding rules classify the examples correctly, whereas often, because of errors or *noise* in the data, misclassifications occur even on the data that is used to train the classifier.

1.1.2 Machine Learning

Now that we have some idea about the inputs and outputs, let's turn to machine learning. What is learning, anyway? What is machine learning? These are philosophic questions, and we will not be much concerned with philosophy in this book; our emphasis is firmly on the practical. However, it is worth spending a few moments at the outset on fundamental issues, just to see how tricky they are, before rolling up our sleeves and looking at machine learning in practice. Our dictionary defines "to learn" as follows:

- To get knowledge of by study, experience, or being taught.
- To become aware by information or from observation.
- To commit to memory.
- To be informed of, ascertain.
- To receive instruction.

These meanings have some shortcomings when it comes to talking about computers. For the first two, it is virtually impossible to test whether learning has been achieved or not. How do you know whether a machine has got knowledge of something? You probably can't just ask it questions; even if you could, you wouldn't be testing its ability to learn but would be testing its ability to answer questions. How do you know whether it has become aware of something? The whole question of whether computers can be aware, or conscious, is a burning philosophic issue. As for the last three meanings, although we can see what they denote in human terms, merely "committing to memory" and "receiving instruction" seem to fall far short of what we might mean by machine learning. They are too passive, and we know that computers find these tasks trivial. Instead, we are interested in improvements in performance, or at least in the potential for performance, in new situations. You can "commit something to memory" or "be informed of something" by rote learning without being able to apply the new knowledge to new situations. You can receive instruction without benefiting from it at all.

Earlier we defined data mining operationally as the process of discovering patterns, automatically or semiautomatically, in large quantities of data—and the patterns must be useful. An operational definition can be formulated in the same way for learning:

> Things learn when they change their behavior in a way that makes them perform better in the future.

This ties learning to performance rather than knowledge. You can test learning by observing the behavior and comparing it with past behavior. This is a much more objective kind of definition and appears to be far more satisfactory.

But there's still a problem. Learning is a rather slippery concept. Lots of things change their behavior in ways that make them perform better in the future, yet we wouldn't want to say that they have actually *learned.* A good example is a comfortable slipper. Has it *learned* the shape of your foot? It has certainly changed its behavior to make it perform better as a slipper! Yet we would hardly want to call this *learning.* In everyday language, we often use the word "training" to denote a mindless kind of learning. We train animals and even plants, although it would be stretching the word a bit to talk of training objects such as slippers that are not in any sense alive. But learning is different. Learning implies thinking. Learning implies purpose. Something that learns has to do so intentionally. That is why we wouldn't say that a vine has learned to grow round a trellis in a vineyard—we'd say it has been *trained.* Learning without purpose is merely training. Or, more to the point, in learning the purpose is the learner's, whereas in training it is the teacher's.

Thus, on closer examination the second definition of learning, in operational, performance-oriented terms, has its own problems when it comes to talking about computers. To decide whether something has actually learned, you need to see whether it intended to or whether there was any purpose involved. That makes the concept moot when applied to machines because whether artifacts can behave purposefully is unclear. Philosophic discussions of what is *really* meant by "learning," like discussions of what is *really* meant by "intention" or "purpose," are fraught with difficulty. Even courts of law find intention hard to grapple with.

1.1.3 Data Mining

Fortunately, the kind of learning techniques explained in this book do not present these conceptual problems—they are called machine learning without really presupposing any particular philosophic stance about what learning actually is. Data mining is a practical topic and involves learning in a practical, not a theoretic, sense. We are interested in techniques for finding and describing structural patterns in data as a tool for helping to explain that data and make predictions from it. The data will take the form of a set of examples—examples of customers who have switched loyalties, for instance, or situations in which certain kinds of contact lenses can be prescribed. The output takes the form of predictions about new examples—a prediction of whether a particular customer will switch or a prediction of what kind of lens will be prescribed under given circumstances. But because this book is about finding *and describing* patterns in data, the output may also include an actual description of a structure that can be used to classify unknown examples to explain the decision. As well as performance, it is helpful to supply an explicit representation of the knowledge that is acquired. In essence, this reflects both definitions of learning considered previously: the acquisition of knowledge and the ability to use it.

Many learning techniques look for structural descriptions of what is learned, descriptions that can become fairly complex and are typically expressed as sets of rules such as the ones described previously or the decision trees described later in this chapter. Because people can understand them, these descriptions explain what has been learned and explain the basis for new predictions. Experience shows that in many applications of machine learning to data mining, the explicit knowledge structures that are acquired, the structural descriptions, are at least as important, and often much more important, than the ability to perform well on new examples. People frequently use data mining to gain knowledge, not just predictions. Gaining knowledge from data certainly sounds like a good idea if you can do it. To find out how, read on!

1.2 SIMPLE EXAMPLES: THE WEATHER PROBLEM AND OTHERS

We use a lot of examples in this book, which seems particularly appropriate considering that the book is all about learning from examples! There are several standard datasets that we will come back to repeatedly. Different datasets tend to expose new issues and challenges, and it is interesting and instructive to have in mind a variety of problems when considering learning methods. In fact, the need to work with different datasets is so important that a corpus containing around 100 example problems has been gathered together so that different algorithms can be tested and compared on the same set of problems.

The illustrations used here are all unrealistically simple. Serious application of data mining involves thousands, hundreds of thousands, or even millions of individual cases. But when explaining what algorithms do and how they work, we need simple examples that capture the essence of the problem but are small enough to be comprehensible in every detail. The illustrations we will be working with are intended to be "academic" in the sense that they will help us to understand what is going on. Some actual fielded applications of learning techniques are discussed in Section 1.3, and many more are covered in the books mentioned in the Further Reading section at the end of the chapter.

Another problem with actual real-life datasets is that they are often proprietary. No corporation is going to share its customer and product choice database with you so that you can understand the details of its data mining application and how it works. Corporate data is a valuable asset, one whose value has increased enormously with the development of data mining techniques such as those described in this book. Yet we are concerned here with understanding how the methods used for data mining work and understanding the details of these methods so that we can trace their operation on actual data. That is why our illustrations are simple ones. But they are not *simplistic:* they exhibit the features of real datasets.

1.2.1 The Weather Problem

The weather problem is a tiny dataset that we will use repeatedly to illustrate machine learning methods. Entirely fictitious, it supposedly concerns the conditions that are suitable for playing some unspecified game. In general, instances in a dataset are characterized by the values of features, or *attributes,* that measure different aspects of the instance. In this case there are four attributes: *outlook, temperature, humidity,* and *windy.* The outcome is whether or not to play.

In its simplest form, shown in Table 1.2, all four attributes have values that are symbolic categories rather than numbers. Outlook can be *sunny, overcast,* or *rainy;* temperature can be *hot, mild,* or *cool;* humidity can be *high* or *normal;* and windy can be *true* or *false.* This creates 36 possible combinations ($3 \times 3 \times 2 \times 2 = 36$), of which 14 are present in the set of input examples.

A set of rules learned from this information—not necessarily a very good one—might look as follows:

```
If outlook = sunny and humidity = high  then play = no
If outlook = rainy and windy = true     then play = no
If outlook = overcast                   then play = yes
If humidity = normal                    then play = yes
If none of the above                    then play = yes
```

These rules are meant to be interpreted in order: the first one; then, if it doesn't apply, the second; and so on.

A set of rules intended to be interpreted in sequence is called a *decision list.* Interpreted as a decision list, the rules correctly classify all of the examples in the table, whereas taken individually, out of context, some of the rules are incorrect. For example, the rule `if humidity = normal, then play = yes` gets one of the examples wrong (check which one). The meaning of a set of rules depends on how it is interpreted—not surprisingly!

In the slightly more complex form shown in Table 1.3, two of the attributes—temperature and humidity—have numeric values. This means that any learning method must create inequalities involving these attributes rather than simple

Table 1.2 The Weather Data

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

equality tests, as in the former case. This is called a *numeric-attribute problem*—in this case, a *mixed-attribute problem* because not all attributes are numeric.

Now the first rule given earlier might take the following form:

```
If outlook = sunny and humidity > 83 then play = no
```

A slightly more complex process is required to come up with rules that involve numeric tests.

The rules we have seen so far are *classification rules:* they predict the classification of the example in terms of whether or not to play. It is equally possible to disregard the classification and just look for any rules that strongly associate different attribute values. These are called *association rules.* Many association rules can be derived from the weather data in Table 1.2. Some good ones are as follows:

```
If temperature = cool                   then humidity = normal
If humidity = normal and windy = false  then play = yes
If outlook = sunny and play = no        then humidity = high
If windy = false and play = no          then outlook = sunny
                                          and humidity = high.
```

Table 1.3 Weather Data with Some Numeric Attribute

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	70	96	False	Yes
Rainy	68	80	False	Yes
Rainy	65	70	True	No
Overcast	64	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	80	False	Yes
Sunny	75	70	True	Yes
Overcast	72	90	True	Yes
Overcast	81	75	False	Yes
Rainy	71	91	True	No

All these rules are 100 percent correct on the given data; they make no false predictions. The first two apply to four examples in the dataset, the third to three examples, and the fourth to two examples. There are many other rules: in fact, nearly 60 association rules can be found that apply to two or more examples of the weather data and are completely correct on this data. If you look for rules that are less than 100 percent correct, then you will find many more. There are so many because unlike classification rules, association rules can "predict" any of the attributes, not just a specified class, and can even predict more than one thing. For example, the fourth rule predicts both that *outlook* will be *sunny* and that *humidity* will be *high.*

1.2.2 Contact Lenses: An Idealized Problem

The contact lens data introduced earlier tells you the kind of contact lens to prescribe, given certain information about a patient. Note that this example is intended for illustration only: it grossly oversimplifies the problem and should certainly not be used for diagnostic purposes!

The first column of Table 1.1 gives the age of the patient. In case you're wondering, *presbyopia* is a form of longsightedness that accompanies the onset of middle age. The second gives the spectacle prescription: *myope* means shortsighted and *hypermetrope* means longsighted. The third shows whether the patient is astigmatic, and the fourth relates to the rate of tear production, which is important in this context because tears lubricate contact lenses. The final column shows which kind of lenses to prescribe: *hard, soft,* or *none.* All possible combinations of the attribute values are represented in the table.

A sample set of rules learned from this information is shown in Figure 1.1. This is a large set of rules, but they do correctly classify all the examples. These rules are complete and deterministic: they give a unique prescription for every conceivable example. Generally, this is not the case. Sometimes there are situations in which no rule applies; other times more than one rule may apply, resulting in conflicting recommendations. Sometimes probabilities or weights may be associated with the rules themselves to indicate that some are more important, or more reliable, than others.

You might be wondering whether there is a smaller rule set that performs as well. If so, would you be better off using the smaller rule set and, if so, why? These are exactly the kinds of questions that will occupy us in this book. Because the examples form a complete set for the problem space, the rules do no more than summarize all the information that is given, expressing it in a different and more concise way. Even though it involves no generalization, this is often a useful

```
If tear production rate = reduced then recommendation = none
If age = young and astigmatic = no and
   tear production rate = normal then recommendation = soft
If age = pre-presbyopic and astigmatic = no and
   tear production rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope and
   astigmatic = no then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no and
   tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes and
   tear production rate = normal then recommendation = hard
If age = young and astigmatic = yes and
   tear production rate = normal then recommendation = hard
If age = pre-presbyopic and
   spectacle prescription = hypermetrope and astigmatic = yes
   then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
```

FIGURE 1.1

Rules for the contact lens data.

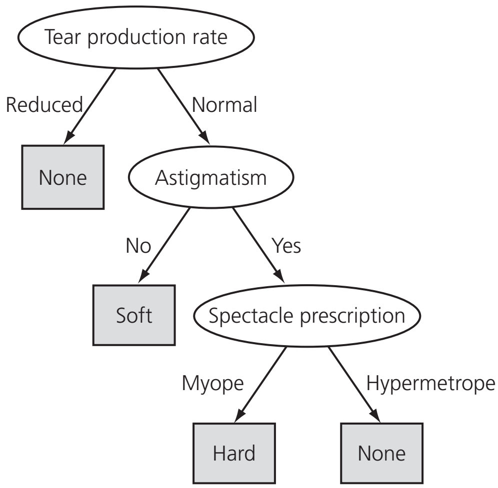

FIGURE 1.2

Decision tree for the contact lens data.

thing to do! People frequently use machine learning techniques to gain insight into the structure of their data rather than to make predictions for new cases. In fact, a prominent and successful line of research in machine learning began as an attempt to compress a huge database of possible chess endgames and their outcomes into a data structure of reasonable size. The data structure chosen for this enterprise was not a set of rules, but a decision tree.

Figure 1.2 presents a structural description for the contact lens data in the form of a decision tree, which for many purposes is a more concise and perspicuous representation of the rules and has the advantage that it can be visualized more easily. (However, this decision tree—in contrast to the rule set given in Figure 1.1—classifies two examples incorrectly.) The tree calls first for a test on *tear production rate,* and the first two branches correspond to the two possible outcomes. If *tear production rate* is *reduced* (the left branch), the outcome is *none.* If it is *normal* (the right branch), a second test is made, this time on *astigmatism.* Eventually, whatever the outcome of the tests, a leaf of the tree is reached that dictates the contact lens recommendation for that case.

1.2.3 Irises: A Classic Numeric Dataset

The iris dataset, which dates back to seminal work by the eminent statistician R. A. Fisher in the mid-1930s and is arguably the most famous dataset used in data mining, contains 50 examples each of three types of plant: *Iris setosa, Iris versicolor,* and *Iris virginica.* It is excerpted in Table 1.4. There are four attributes: *sepal length, sepal width, petal length,* and *petal width* (all measured in centimeters). Unlike previous datasets, all attributes have numeric values.

Table 1.4 The Iris Data

	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Type
1	5.1	3.5	1.4	0.2	*Iris setosa*
2	4.9	3.0	1.4	0.2	*Iris setosa*
3	4.7	3.2	1.3	0.2	*Iris setosa*
4	4.6	3.1	1.5	0.2	*Iris setosa*
5	5.0	3.6	1.4	0.2	*Iris setosa*
. . .					
51	7.0	3.2	4.7	1.4	*Iris versicolor*
52	6.4	3.2	4.5	1.5	*Iris versicolor*
53	6.9	3.1	4.9	1.5	*Iris versicolor*
54	5.5	2.3	4.0	1.3	*Iris versicolor*
55	6.5	2.8	4.6	1.5	*Iris versicolor*
. . .					
101	6.3	3.3	6.0	2.5	*Iris virginica*
102	5.8	2.7	5.1	1.9	*Iris virginica*
103	7.1	3.0	5.9	2.1	*Iris virginica*
104	6.3	2.9	5.6	1.8	*Iris virginica*
105	6.5	3.0	5.8	2.2	*Iris virginica*
. . .					

The following set of rules might be learned from this dataset:

```
If petal length < 2.45 then Iris setosa
If sepal width < 2.10 then Iris versicolor
If sepal width < 2.45 and petal length < 4.55 then Iris versicolor
If sepal width < 2.95 and petal width < 1.35 then Iris versicolor
If petal length ≥ 2.45 and petal length < 4.45 then Iris versicolor
If sepal length ≥ 5.85 and petal length < 4.75 then Iris versicolor
If sepal width < 2.55 and petal length < 4.95 and
   petal width < 1.55 then Iris versicolor
If petal length ≥ 2.45 and petal length < 4.95 and
   petal width < 1.55 then Iris versicolor
If sepal length ≥ 6.55 and petal length < 5.05 then Iris versicolor
```

```
If sepal width < 2.75 and petal width < 1.65 and
   sepal length < 6.05 then Iris versicolor
If sepal length ≥ 5.85 and sepal length < 5.95 and
   petal length < 4.85 then Iris versicolor
If petal length ≥ 5.15 then Iris virginica
If petal width ≥ 1.85 then Iris virginica
If petal width ≥ 1.75 and sepal width < 3.05 then Iris virginica
If petal length ≥ 4.95 and petal width < 1.55 then Iris virginica
```

These rules are very cumbersome; more compact rules can be expressed that convey the same information.

1.2.4 CPU Performance: Introducing Numeric Prediction

Although the iris dataset involves numeric attributes, the outcome—the type of iris—is a category, not a numeric value. Table 1.5 shows some data for which the outcome and the attributes are numeric. It concerns the relative performance of computer processing power on the basis of a number of relevant attributes; each row represents 1 of 209 different computer configurations.

The classic way of dealing with continuous prediction is to write the outcome as a linear sum of the attribute values with appropriate weights, for example:

$$\begin{aligned}\mathrm{PRP} = &-55.9 + 0.0489\ \mathrm{MYCT} + 0.0153\ \mathrm{MMIN} + 0.0056\ \mathrm{MMAX} + 0.6410\ \mathrm{CACH}\\ &- 0.2700\ \mathrm{CHMIN} + 1.480\ \mathrm{CHMAX}\end{aligned}$$

Table 1.5 The CPU Performance Data

	Cycle Time (ns) MYCT	Main Memory (KB) Minimum MMIN	Main Memory (KB) Maximum MMAX	Cache (KB) CACH	Channels Minimum CHMIN	Channels Maximum CHMAX	Performance PRP
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
3	29	8000	32000	32	8	32	220
4	29	8000	32000	32	8	32	172
5	29	8000	16000	32	8	16	132
. . .							
207	125	2000	8000	0	2	14	52
208	480	512	8000	32	0	0	67
209	480	1000	4000	0	0	0	45

(The abbreviated variable names are given in the second row of the table.) This is called a *regression equation,* and the process of determining the weights is called *regression,* a well-known procedure in statistics. However, the basic regression method is incapable of discovering nonlinear relationships (although variants do exist).

In the iris and central processing unit (CPU) performance data, all the attributes have numeric values. Practical situations frequently present a mixture of numeric and nonnumeric attributes.

1.2.5 Labor Negotiations: A More Realistic Example

The labor negotiations dataset in Table 1.6 summarizes the outcome of Canadian contract negotiations in 1987 and 1988. It includes all collective agreements

Table 1.6 The Labor Negotiations Data

Attribute	Type	1	2	3	. . .	40
Duration	Years	1	2	3		2
Wage increase first year	Percentage	2%	4%	4.3%		4.5
Wage increase second year	Percentage	?	5%	4.4%		4.0
Wage increase third year	Percentage	?	?	?		?
Cost of living adjustment	{none, tcf, tc}	None	Tcf	?		None
Working hours per week	Hours	28	35	38		40
Pension	{none, ret-allw, empl-cntr}	None	?	?		?
Standby pay	Percentage	?	13%	?		?
Shift-work supplement	Percentage	?	5%	4%		4
Education allowance	{yes, no}	Yes	?	?		?
Statutory holidays	Days	11	15	12		12
Vacation	{below-avg, avg, gen}	Avg	Gen	Gen		Avg
Long-term disability assistance	{yes, no}	No	?	?		Yes
Dental plan contribution	{none, half, full}	None	?	Full		Full
Bereavement assistance	{yes, no}	No	?	?		Yes
Health plan contribution	{none, half, full}	None	?	Full		Half
Acceptability of contract	{good, bad}	Bad	Good	Good		Good

reached in the business and personal services sector for organizations with at least 500 members (teachers, nurses, university staff, police, etc.). Each case concerns one contract, and the outcome is whether the contract is deemed *acceptable* or *unacceptable.* The acceptable contracts are ones in which agreements were accepted by both labor and management. The unacceptable ones are either known offers that fell through because one party would not accept them or acceptable contracts that had been significantly perturbed to the extent that, in the view of experts, they would not have been accepted.

There are 40 examples in the dataset (plus another 17 that are normally reserved for test purposes). Unlike the other tables here, Table 1.6 presents the examples as columns rather than as rows; otherwise, it would have to be stretched over several pages. Many of the values are unknown or missing, as indicated by question marks.

This is a much more realistic dataset than the others we have seen. It contains many missing values, and it seems unlikely that an exact classification can be obtained.

Figure 1.3 shows two decision trees that represent the dataset. Figure 1.3(a) is simple and approximate: it doesn't represent the data exactly. For example, it will predict *bad* for some contracts that are actually marked *good.* But it does make intuitive sense: a contract is bad (for the employee!) if the wage increase in the first year is too small (less than 2.5 percent). If the first-year wage increase is larger than this, it is good if there are lots of statutory holidays (more than 10 days). Even if there are fewer statutory holidays, it is good if the first-year wage increase is large enough (more than 4 percent).

Figure 1.3(b) is a more complex decision tree that represents the same dataset. In fact, this is a more accurate representation of the actual dataset that was used to create the tree. But it is not necessarily a more accurate representation of the underlying concept of good versus bad contracts. Look down the left branch. It doesn't seem to make sense intuitively that, if the working hours exceed 36, a contract is bad if there is no health-plan contribution or a full health-plan contribution but is good if there is a half health-plan contribution. It is certainly reasonable that the health-plan contribution plays a role in the decision but not if half is good and both full and none are bad. It seems likely that this is an artifact of the particular values used to create the decision tree rather than a genuine feature of the good versus bad distinction.

The tree in Figure 1.3(b) is more accurate on the data that was used to train the classifier but will probably perform less well on an independent set of test data. It is "overfitted" to the training data—it follows it too slavishly. The tree in Figure 1.3(a) is obtained from the one in Figure 1.3(b) by a process of pruning.

1.2.6 Soybean Classification: A Classic Machine Learning Success

An often-quoted early success story in the application of machine learning to practical problems is the identification of rules for diagnosing soybean diseases.

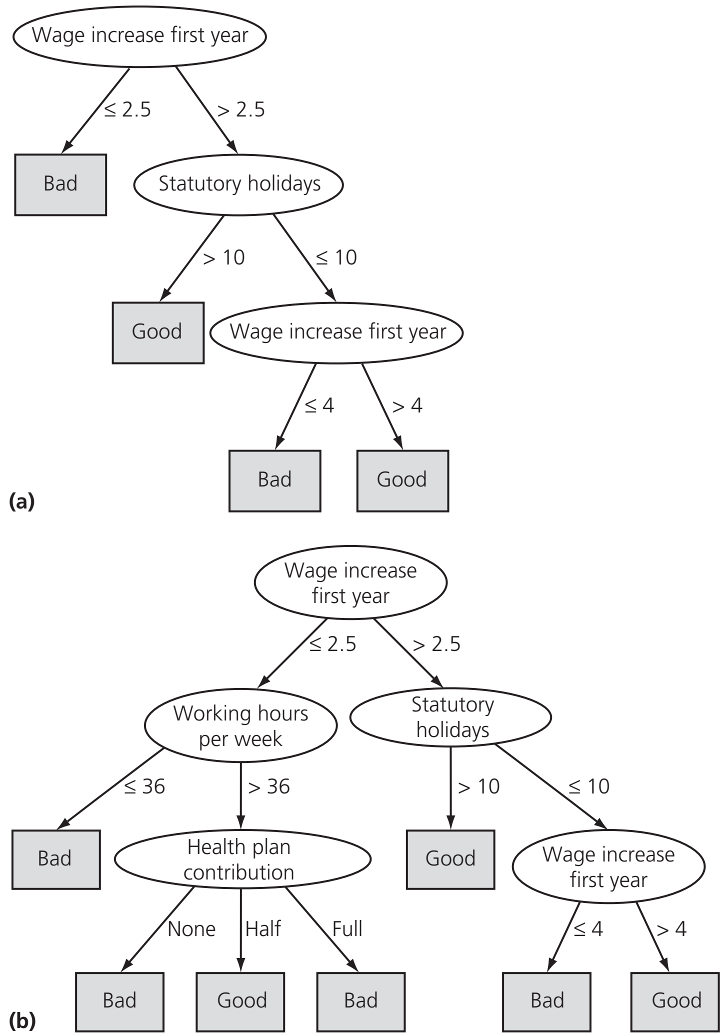

FIGURE 1.3

Decision trees for the labor negotiations data.

The data is taken from questionnaires describing plant diseases. There are about 680 examples, each representing a diseased plant. Plants were measured on 35 attributes, each one having a small set of possible values. Examples are labeled with the diagnosis of an expert in plant biology: there are 19 disease categories altogether—horrible-sounding diseases, such as diaporthe stem canker, rhizoctonia root rot, and bacterial blight, to mention just a few.

Table 1.7 gives the attributes, the number of different values that each can have, and a sample record for one particular plant. The attributes are placed into different categories just to make them easier to read.

Here are two example rules, learned from this data:

```
If   [leaf condition is normal and
     stem condition is abnormal and
     stem cankers is below soil line and
     canker lesion color is brown]
then
     diagnosis is rhizoctonia root rot

If   [leaf malformation is absent and
     stem condition is abnormal and
     stem cankers is below soil line and
     canker lesion color is brown]
then
     diagnosis is rhizoctonia root rot
```

These rules nicely illustrate the potential role of prior knowledge—often called *domain knowledge*—in machine learning, because the only difference between the two descriptions is `leaf condition is` normal versus `leaf malformation is absent`. In this domain, if the leaf condition is normal, then leaf malformation is necessarily absent, so one of these conditions happens to be a special case of the other. Thus, if the first rule is true, the second is necessarily true as well. The only time the second rule comes into play is when leaf malformation is absent

Table 1.7 The Soybean Data

	Attribute	Number of Values	Sample Value
Environment	Time of occurrence	7	July
	Precipitation	3	Above normal
	Temperature	3	Normal
	Cropping history	4	Same as last year
	Hail damage	2	Yes
	Damaged area	4	Scattered
	Severity	3	Severe
	Plant height	2	Normal
	Plant growth	2	Abnormal
	Seed treatment	3	Fungicide
	Germination	3	Less than 80%

Table 1.7 *Continued*

	Attribute	Number of Values	Sample Value
Seed	Condition	2	Normal
	Mold growth	2	Absent
	Discoloration	2	Absent
	Size	2	Normal
	Shriveling	2	Absent
Fruit	Condition of fruit pods	3	Normal
	Fruit spots	5	–
Leaf	Condition	2	Abnormal
	Leaf spot size	3	–
	Yellow leaf spot halo	3	Absent
	Leaf spot margins	3	–
	Shredding	2	Absent
	Leaf malformation	2	Absent
	Leaf mildew growth	3	Absent
Stem	Condition	2	Abnormal
	Stem lodging	2	Yes
	Stem cankers	4	Above soil line
	Canker lesion color	3	–
	Fruiting bodies on stems	2	Present
	External decay of stem	3	Firm and dry
	Mycelium on stem	2	Absent
	Internal discoloration	3	None
	Sclerotia	2	Absent
Root	Condition	3	Normal
Diagnosis			Diaporthe stem
		19	Canker

but leaf condition is *not* normal—that is, when something other than malformation is wrong with the leaf. This is certainly not apparent from a casual reading of the rules.

Research on this problem in the late 1970s found that these diagnostic rules could be generated by a machine learning algorithm, along with rules for every other disease category, from about 300 training examples. The examples were carefully selected from the corpus of cases as being quite different from one another—"far apart" in the example space. At the same time, the plant pathologist who had produced the diagnoses was interviewed, and his expertise was translated into diagnostic rules. Surprisingly, the computer-generated rules outperformed the expert's rules on the remaining test examples. They gave the correct disease top ranking 97.5 percent of the time compared with only 72 percent for the expert-derived rules. Furthermore, not only did the learning algorithm find rules that outperformed those of the expert collaborator, but the same expert was so impressed that he allegedly adopted the discovered rules in place of his own!

1.3 FIELDED APPLICATIONS

The examples that we opened with are speculative research projects, not production systems. And the preceding illustrations are toy problems: they are deliberately chosen to be small so that we can use them to work through algorithms later in the book. Where's the beef? Here are some applications of machine learning that have actually been put into use.

Because they are fielded applications, the illustrations that follow tend to stress the use of learning in performance situations, in which the emphasis is on ability to perform well on new examples. This book also describes the use of learning systems to gain knowledge from decision structures that are inferred from the data. We believe that this is as important—probably even more important in the long run—a use of the technology as merely making high-performance predictions. Still, it will tend to be underrepresented in fielded applications because when learning techniques are used to gain insight, the result is not normally a system that is put to work as an application in its own right. Nevertheless, in three of the examples that follow, the fact that the decision structure is comprehensible is a key feature in the successful adoption of the application.

1.3.1 Decisions Involving Judgment

When you apply for a loan, you have to fill out a questionnaire that asks for relevant financial and personal information. The loan company uses this information as the basis for its decision as to whether to lend you money. Such decisions are typically made in two stages. First, statistical methods are used to determine clear "accept" and "reject" cases. The remaining borderline cases are more difficult and call for human judgment. For example, one loan company uses a statistical decision procedure to calculate a numeric parameter based on the information supplied in the questionnaire. Applicants are accepted if this parameter exceeds a preset threshold and rejected if it falls below a second threshold. This accounts for 90 percent of cases, and the remaining 10 percent are referred to loan officers for a decision. On examining historical data on whether applicants did indeed repay their loans, however, it turned out that half of the borderline applicants who were granted loans actually defaulted. Although it would be tempting simply to deny credit to borderline customers, credit industry professionals pointed out that if only their repayment future could be reliably determined it is precisely these customers whose business should be wooed; they tend to be active customers of a credit institution because their finances remain in a chronically volatile condition. A suitable compromise must be reached between the viewpoint of a company accountant, who dislikes bad debt, and that of a sales executive, who dislikes turning business away.

Enter machine learning. The input was 1000 training examples of borderline cases for which a loan had been made that specified whether the borrower had finally paid off or defaulted. For each training example, about 20 attributes were extracted from the questionnaire, such as age, years with current employer, years at current address, years with the bank, and other credit cards possessed. A machine learning procedure was used to produce a small set of classification rules that made correct predictions on two-thirds of the borderline cases in an independently chosen test set. Not only did these rules improve the success rate of the loan decisions, but the company also found them attractive because they could be used to explain to applicants the reasons behind the decision. Although the project was an exploratory one that took only a small development effort, the loan company was apparently so pleased with the result that the rules were put into use immediately.

1.3.2 Screening Images

Since the early days of satellite technology, environmental scientists have been trying to detect oil slicks from satellite images to give early warning of ecologic disasters and deter illegal dumping. Radar satellites provide an opportunity for monitoring coastal waters day and night, regardless of weather conditions. Oil slicks appear as dark regions in the image whose size and shape evolve depending on weather and sea conditions. However, other look-alike dark regions can be caused by local weather conditions such as high wind. Detecting oil slicks is an expensive manual process requiring highly trained personnel who assess each region in the image.

A hazard detection system has been developed to screen images for subsequent manual processing. Intended to be marketed worldwide to a wide variety of users—government agencies and companies—with different objectives, applications, and geographic areas, it needs to be highly customizable to individual circumstances. Machine learning allows the system to be trained on examples of spills and nonspills supplied by the user and lets the user control the trade-off between undetected spills and false alarms. Unlike other machine learning applications, which generate a classifier that is then deployed in the field, here it is the learning method itself that will be deployed.

The input is a set of raw pixel images from a radar satellite, and the output is a much smaller set of images with putative oil slicks marked by a colored border. First, standard image processing operations are applied to normalize the image. Then, suspicious dark regions are identified. Several dozen attributes are extracted from each region, characterizing its size, shape, area, intensity, sharpness and jaggedness of the boundaries, proximity to other regions, and information about the background in the vicinity of the region. Finally, standard learning techniques are applied to the resulting attribute vectors.

Several interesting problems were encountered. One is the scarcity of training data. Oil slicks are (fortunately) very rare, and manual classification is extremely costly. Another is the unbalanced nature of the problem: of the many dark regions in the training data, only a small fraction are actual oil slicks. A third is that the examples group naturally into batches, with regions drawn from each image forming a single batch, and background characteristics vary from one batch to another. Finally, the performance task is to serve as a filter, and the user must be provided with a convenient means of varying the false-alarm rate.

1.3.3 Load Forecasting

In the electricity supply industry, it is important to determine future demand for power as far in advance as possible. If accurate estimates can be made for the maximum and minimum load for each hour, day, month, season, and year, utility companies can make significant economies in areas such as setting the operating reserve, maintenance scheduling, and fuel inventory management.

An automated load forecasting assistant has been operating at a major utility supplier over the past decade to generate hourly forecasts 2 days in advance. The first step was to use data collected over the previous 15 years to create a sophisticated load model manually. This model had three components: base load for the year, load periodicity over the year, and the effect of holidays. To normalize for the base load, the data for each previous year was standardized by subtracting the average load for that year from each hourly reading and dividing by the standard deviation over the year. Electric load shows periodicity at three fundamental frequencies: diurnal, where usage has an early morning minimum and midday and afternoon maxima; weekly, where demand is lower at weekends; and seasonal, where increased demand during winter and summer for heating and cooling, respectively, creates a yearly cycle. Major holidays such as Thanksgiving, Christmas, and New Year's Day show significant variation from the normal load and are each modeled separately by averaging hourly loads for that day over the past 15 years. Minor official holidays, such as Columbus Day, are lumped together as school holidays and treated as an offset to the normal diurnal pattern. All of these effects are incorporated by reconstructing a year's load as a sequence of typical days, fitting the holidays in their correct position, and denormalizing the load to account for overall growth.

Thus far, the load model is a static one, constructed manually from historical data, and implicitly assumes "normal" climatic conditions over the year. The final step was to take weather conditions into account using a technique that locates the previous day most similar to the current circumstances and uses the historical information from that day as a predictor. In this case the prediction is treated as an additive correction to the static load model. To guard against outliers, the 8 most similar days are located and their additive corrections averaged. A database was constructed of temperature, humidity, wind speed, and cloud cover at three local weather centers for each hour of the 15-year historical record, along with the difference between the actual load and that predicted by the static model. A linear regression analysis was performed to determine the relative effects of these parameters on load, and the coefficients were used to weight the distance function used to locate the most similar days.

The resulting system yielded the same performance as trained human forecasters but was far quicker—taking seconds rather than hours to generate a daily forecast. Human operators can analyze the forecast's sensitivity to simulated changes in weather and bring up for examination the "most similar" days that the system used for weather adjustment.

1.3.4 Diagnosis

Diagnosis is one of the principal application areas of expert systems. Although the handcrafted rules used in expert systems often perform well, machine learning can be useful in situations in which producing rules manually is too labor intensive.

Preventative maintenance of electromechanical devices such as motors and generators can forestall failures that disrupt industrial processes. Technicians regularly inspect each device, measuring vibrations at various points to determine whether the device needs servicing. Typical faults include shaft misalignment, mechanical loosening, faulty bearings, and unbalanced pumps. A particular chemical plant uses more than 1000 different devices, ranging from small pumps to very large turbo-alternators, which until recently were diagnosed by a human expert with 20 years of experience. Faults are identified by measuring vibrations at different places on the device's mounting and using Fourier analysis to check the energy present in three different directions at each harmonic of the basic rotation speed. The expert studies this information, which is noisy because of limitations in the measurement and recording procedure, to arrive at a diagnosis. Although handcrafted expert system rules had been developed for some situations, the elicitation process would have to be repeated several times for different types of machinery; so a learning approach was investigated.

Six hundred faults, each comprising a set of measurements along with the expert's diagnosis, were available, representing 20 years of experience in the field. About half were unsatisfactory for various reasons and had to be discarded; the remainder were used as training examples. The goal was not to determine whether or not a fault existed but to diagnose the kind of fault, given that one was there. Thus, there was no need to include fault-free cases in the training set. The measured attributes were rather low level and had to be augmented by intermediate concepts, that is, functions of basic attributes, which were defined in consultation with the expert and embodied some causal domain knowledge. The derived attributes were run through an induction algorithm to produce a set of diagnostic rules. Initially, the expert was not satisfied with the rules because he could not relate them to his own knowledge and experience. For him, mere statistical evidence was not, by itself, an adequate explanation. Further background knowledge had to be used before satisfactory rules were generated. Although the resulting rules were complex, the expert liked them because he could justify them in light of his mechanical knowledge. He was pleased that a third of the rules coincided with ones he used himself and was delighted to gain new insight from some of the others.

Performance tests indicated that the learned rules were slightly superior to the handcrafted ones that the expert had previously elicited, and subsequent use in the chemical factory confirmed this result. It is interesting to note, however, that the system was put into use not because of its good performance but because the domain expert approved of the rules that had been learned.

1.3.5 Marketing and Sales

Some of the most active applications of data mining have been in the area of marketing and sales. These are domains in which companies possess massive volumes of precisely recorded data, data that—it has only recently been realized—is potentially extremely valuable. In these applications, predictions themselves are the chief interest: the structure of how decisions are made is often completely irrelevant.

We have already mentioned the problem of fickle customer loyalty and the challenge of detecting customers who are likely to defect so that they can be wooed back into the fold by giving them special treatment. Banks were early adopters of data mining technology because of their successes in the use of machine learning for credit assessment. Data mining is now being used to reduce customer attrition by detecting changes in individual banking patterns that may herald a change of bank or even life changes, such as a move to another city, that could result in a different bank being chosen. It may reveal, for example, a group of customers with above-average attrition rate who do most of their banking by phone after hours when telephone response is slow. Data mining may determine groups for whom new services are appropriate, such as a cluster of profitable, reliable customers who rarely get cash advances from their credit card except in November and December, when they are prepared to pay exorbitant interest rates to see them through the holiday season. In another domain, cellular phone companies fight *churn* by detecting patterns of behavior that could benefit from new services and then advertise such services to retain their customer base. Incentives provided specifically to retain existing customers can be expensive, and successful data mining allows them to be precisely targeted to those customers where they are likely to yield maximum benefit.

Market basket analysis is the use of association techniques to find groups of items that tend to occur together in transactions, typically supermarket checkout data. For many retailers, this is the only source of sales information that is available for data mining. For example, automated analysis of checkout data may uncover the fact that customers who buy beer also buy chips, a discovery that could be significant from the supermarket operator's point of view (although rather an obvious one that probably does not need a data mining exercise to discover). Or it may come up with the fact that on Thursdays, customers often purchase diapers and beer together, an initially surprising result that, on reflection, makes some sense as young parents stock up for a weekend at home. Such information could be used for many purposes: planning store layouts, limiting special discounts to just one of a set of items that tend to be purchased together, offering coupons for a matching product when one of them is sold alone, and so on. There is enormous added value in being able to identify individual customer's sales histories. In fact, this value is leading to a proliferation of discount cards or "loyalty" cards that allow retailers to identify individual customers whenever they make a purchase; the personal data that results will be far more valuable than the cash value of the discount. Identification of individual customers not only allows historical analysis of purchasing patterns but also permits precisely targeted special offers to be mailed out to prospective customers.

This brings us to direct marketing, another popular domain for data mining. Promotional offers are expensive and have an extremely low—but highly profitable—response rate. Any technique that allows a promotional mailout to be more tightly focused, achieving the same or nearly the same response from a much smaller sample, is valuable. Commercially available databases containing demographic information based on ZIP codes that characterize the associated neighborhood can be correlated with information on existing customers to find a socioeconomic model that predicts what kind of people will turn out to be actual customers. This model can then be used on information gained in response to an initial mailout, where people send back a response card or call an 800 number for more information, to predict likely future customers. Direct mail companies have the advantage over shopping mall retailers of having complete purchasing histories for each individual customer and can use data mining to determine those likely to respond to special offers. Targeted campaigns are cheaper than mass-marketed campaigns because companies save money by sending offers only to those likely to want the product. Machine learning can help companies to find the targets.

1.3.6 Other Applications

There are countless other applications of machine learning. We briefly mention a few more areas to illustrate the breadth of what has been done.

Sophisticated manufacturing processes often involve tweaking control parameters. Separating crude oil from natural gas is an essential prerequisite to oil refinement, and controlling the separation process is a tricky job. British Petroleum used machine learning to create rules for setting the parameters. This now takes just 10 minutes, whereas previously human experts took more than a day. Westinghouse faced problems in its process for manufacturing nuclear fuel pellets and used machine learning to create rules to control the process. This was reported to save the company more than $10 million per year (in 1984). The Tennessee printing company R. R. Donnelley applied the same idea to control rotogravure printing presses to reduce artifacts caused by inappropriate parameter settings, reducing the number of artifacts from more than 500 each year to fewer than 30.

In the realm of customer support and service, we have already described adjudicating loans and marketing and sales applications. Another example arises when a customer reports a telephone problem and the company must decide what kind of technician to assign to the job. An expert system developed by Bell Atlantic in 1991 to make this decision was replaced in 1999 by a set of rules acquired using machine learning, which saved more than $10 million per year by making fewer incorrect decisions.

There are many scientific applications. In biology, machine learning is used to help identify the thousands of genes within each new genome. In biomedicine, it is used to predict drug activity by analyzing not just the chemical properties of drugs but also their three-dimensional structure. This accelerates drug discovery and reduces its cost. In astronomy, machine learning has been used to develop a fully automatic cataloging system for celestial objects that are too faint to be seen by visual inspection. In chemistry, it has been used to predict the structure of certain organic compounds from magnetic resonance spectra. In all these applications, machine learning techniques have attained levels of performance—or should we say skill?—that rival or surpass human experts.

Automation is especially welcome in situations involving continuous monitoring, a job that is time consuming and exceptionally tedious for humans. Ecologic applications include the oil spill monitoring described earlier. Some other applications are rather less consequential—for example, machine learning is being used to predict preferences for TV programs based on past choices and advise viewers about the available channels. Still others may save lives. Intensive care patients may be monitored to detect changes in variables that cannot be explained by circadian rhythm, medication, and so on, raising an alarm when appropriate.

Finally, in a world that relies on vulnerable networked computer systems and is increasingly concerned about cyber security, machine learning is used to detect intrusion by recognizing unusual patterns of operation.

1.4 MACHINE LEARNING AND STATISTICS

What's the difference between machine learning and statistics? Cynics, looking wryly at the explosion of commercial interest (and hype) in this area, equate data mining to statistics plus marketing. In truth, you should not look for a dividing line between machine learning and statistics because there is a continuum—and a multidimensional one at that—of data analysis techniques. Some derive from the skills taught in standard statistics courses, and others are more closely associated with the kind of machine learning that has arisen out of computer science. Historically, the two sides have had rather different traditions. If forced to point to a single difference of emphasis, it might be that statistics has been more concerned with testing hypotheses, whereas machine learning has been more concerned with formulating the process of generalization as a search through possible hypotheses. But this is a gross oversimplification: statistics is far more than hypothesis testing, and many machine learning techniques do not involve any searching at all.

In the past, similar methods have developed in parallel in machine learning and statistics. One is decision tree induction. Four statisticians (Breiman et al. 1984) published a book, *Classification and Regression Trees,* in the mid-1980s, and throughout the 1970s and early 1980s a prominent machine learning researcher, J. Ross Quinlan, was developing a system for inferring classification trees from examples. These two independent projects produced similar methods for generating trees from examples, and the researchers only became aware of one another's work much later. A second area in which similar methods have arisen involves the use of nearest-neighbor methods for classification. These are standard statistical techniques that have been extensively adapted by machine learning researchers, both to improve classification performance and to make the procedure more efficient computationally.

But now the two perspectives have converged. The techniques we will examine in this book incorporate a great deal of statistical thinking. From the beginning, when constructing and refining the initial example set, standard statistical methods apply: visualization of data, selection of attributes, discarding outliers, and so on. Most learning algorithms use statistical tests when constructing rules or trees and for correcting models that are "overfitted," in that they depend too strongly on the details of the particular examples used to produce them (we have already seen an example of this in the two decision trees of Figure 1.3 for the labor negotiations problem). Statistical tests are used to validate machine learning models and to evaluate machine learning algorithms. In our study of practical techniques for data mining, we will learn a great deal about statistics.

1.5 GENERALIZATION AS SEARCH

One way of visualizing the problem of learning—and one that distinguishes it from statistical approaches—is to imagine a search through a space of possible concept descriptions for one that fits the data. Although the idea of generalization as search is a powerful conceptual tool for thinking about machine learning, it is not essential for understanding the practical methods described here. That is why this section is considered optional.

Suppose, for definiteness, that *concepts*—the result of learning—are expressed as rules such as the ones given for the weather problem in Section 1.2 (although other concept description languages would do just as well). Suppose that we list all possible sets of rules and then look for ones that satisfy a given set of examples. A big job? Yes. An *infinite* job? At first glance it seems so because there is no limit to the number of rules there might be. But actually the number of possible rule sets is finite. Note first that each individual rule is no greater than a fixed maximum size, with at most one term for each attribute: for the weather data of Table 1.2 this involves four terms in all. Because the number of possible rules is finite, the number of possible rule *sets* is finite, too, although extremely large. However, we'd hardly be interested in sets that contained a very large number of rules. In fact, we'd hardly be interested in sets that had more rules than there are examples because it is difficult to imagine needing more than one rule for each example. So if we were to restrict consideration to rule sets smaller than that, the problem would be substantially reduced, although still very large.

The threat of an infinite number of possible concept descriptions seems more serious for the second version of the weather problem in Table 1.3 because these rules contain numbers. If they are real numbers, you can't enumerate them, even in principle. However, on reflection, the problem again disappears because the numbers really just represent breakpoints in the numeric values that appear in the examples. For instance, consider the *temperature* attribute in Table 1.3. It involves the numbers 64, 65, 68, 69, 70, 71, 72, 75, 80, 81, 83, and 85—12 different numbers. There are 13 possible places in which we might want to put a breakpoint for a rule involving temperature. The problem isn't infinite after all.

So the process of generalization can be regarded as a search through an enormous, but finite, search space. In principle, the problem can be solved by enumerating descriptions and striking out those that do not fit the examples presented. A positive example eliminates all descriptions that it does not match, and a negative one eliminates those it does match. With each example, the set of remaining descriptions shrinks (or stays the same). If only one is left, it is the target description—the target concept.

If several descriptions are left, they may still be used to classify unknown objects. An unknown object that matches all remaining descriptions should be classified as matching the target; if it fails to match any description, it should be classified as being outside the target concept. Only when it matches some descriptions, but not others, is there ambiguity. In this case, if the classification of the unknown object were revealed, it would cause the set of remaining descriptions to shrink because rule sets that classified the object the wrong way would be rejected.

1.5.1 Enumerating the Concept Space

Regarding it as search is a good way of looking at the learning process. However, the search space, although finite, is extremely big, and it is generally impractical to enumerate all possible descriptions and then see which ones fit. In the weather problem there are $4 \times 4 \times 3 \times 3 \times 2 = 288$ possibilities for each rule. There are four possibilities for the *outlook* attribute: *sunny, overcast, rainy,* or it may not participate in the rule at all. Similarly, there are four for *temperature,* three for *weather* and *humidity,* and two for the class. If we restrict the rule set to contain no more than 14 rules (because there are 14 examples in the training set), there are around 2.7×10^{34} possible different rule sets. That's a lot to enumerate, especially for such a patently trivial problem.

Although there are ways of making the enumeration procedure more feasible, a serious problem remains: in practice, it is rare for the process to converge on a unique acceptable description. Either many descriptions are still in the running after the examples are processed or the descriptors are all eliminated. The first case arises when the examples are not sufficiently comprehensive to eliminate all possible descriptions except for the "correct" one. In practice, people often want a single "best" description, and it is necessary to apply some other criteria to select the best one from the set of remaining descriptions. The second problem arises either because the description language is not expressive enough to capture the actual concept or because of noise in the examples. If an example comes in with the "wrong" classification because of an error in some of the attribute values or in the class that is assigned to it, this will likely eliminate the correct description from the space. The result is that the set of remaining descriptions becomes empty. This situation is very likely to happen if the examples contain any noise at all, which inevitably they do except in artificial situations.

Another way of looking at generalization as search is to imagine it, not as a process of enumerating descriptions and striking out those that don't apply, but as a kind of hill-climbing in description space to find the description that best matches the set of examples according to some prespecified matching criterion. This is the way that most practical machine learning methods work. However, except in the most trivial cases, it is impractical to search the whole space exhaustively; most practical algorithms involve heuristic search and cannot guarantee to find the optimal description.

1.5.2 Bias

Viewing generalization as a search in a space of possible concepts makes it clear that the following are most important decisions in a machine learning system.

- The concept description language.
- The order in which the space is searched.
- The way that overfitting to the particular training data is avoided.

These three properties are generally referred to as the *bias* of the search and are called *language bias, search bias,* and *overfitting-avoidance bias.* You bias the learning scheme by choosing a language in which to express concepts, by searching in a particular way for an acceptable description, and by deciding when the concept has become so complex that it needs to be simplified.

Language Bias

The most important question for language bias is whether the concept description language is universal, or whether it imposes constraints on what concepts can be learned. If you consider the set of all possible examples, a concept is really just a division of it into subsets. In the weather example, if you were to enumerate all possible weather conditions, the *play* concept is a subset of possible weather conditions. A "universal" language is one that is capable of expressing every possible subset of examples. In practice, the set of possible examples is generally huge, and in this respect our perspective is a theoretic, not a practical, one.

If the concept description language permits statements involving logical *or,* that is, *disjunctions,* then any subset can be represented. If the description language is rule based, disjunction can be achieved by using separate rules. For example, one possible concept representation is just to enumerate the examples:

```
If outlook = overcast and temperature = hot and humidity = high
   and windy = false then play = yes
If outlook = rainy and temperature = mild and humidity = high
   and windy = false then play = yes
If outlook = rainy and temperature = cool and humidity = normal
   and windy = false then play = yes
If outlook = overcast and temperature = cool and humidity = normal
   and windy = true then play = yes
. . .
If none of the above then play = no
```

This is not a particularly enlightening concept description; it simply records the positive examples that have been observed and assumes that all the rest are negative. Each positive example is given its own rule, and the concept is the disjunction of the rules. Alternatively, you could imagine having individual rules for each of the negative examples, too—an equally uninteresting concept. In either case, the concept description does not perform any generalization; it simply records the original data.

On the other hand, if disjunction is *not* allowed, some possible concepts—sets of examples—may not be able to be represented at all. In that case, a machine learning scheme may simply be unable to achieve good performance.

Another kind of language bias is that obtained from knowledge of the particular domain being used. For example, it may be that some combinations of attribute values can never happen. This would be the case if one attribute implied another. We saw an example of this when considering the rules for the soybean problem described earlier. Then, it would be pointless to even consider concepts that involved redundant or impossible combinations of attribute values. Domain knowledge can be used to cut down the search space. Knowledge is power: a little goes a long way, and even a small hint can reduce the search space dramatically.

Search Bias

In realistic data mining problems, there are many alternative concept descriptions that fit the data, and the problem is to find the "best" one according to some criterion—usually simplicity. We use the term *fit* in a statistical sense; we seek the best description that fits the data reasonably well. Moreover, it is often computationally infeasible to search the whole space and guarantee that the description found really is the best. Consequently, the search procedure is heuristic, and no guarantees can be made about the optimality of the final result. This leaves plenty of room for bias: different search heuristics bias the search in different ways.

For example, a learning algorithm might adopt a "greedy" search for rules by trying to find the best rule at each stage and adding it in to the rule set. However, it may be that the best *pair* of rules is not just the two rules that are individually found to be the best. Or when building a decision tree, a commitment to split early on using a particular attribute might turn out later to be ill considered in light of how the tree develops below that node. To get around these problems, a *beam search* could be used in which irrevocable commitments are not made but instead a set of several active alternatives—whose number is the *beam width*—are pursued in parallel. This will complicate the learning algorithm considerably but has the potential to avoid the myopia associated with a greedy search. Of course, if the beam width is not large enough, myopia may still occur. There are more complex search strategies that help to overcome this problem.

A more general and higher-level kind of search bias concerns whether the search is done by starting with a general description and refining it or by starting with a specific example and generalizing it. The former is called a *general-to-specific* search bias, the latter a *specific-to-general* one. Many learning algorithms adopt the former policy, starting with an empty decision tree, or a very general rule, and specializing it to fit the examples. However, it is perfectly possible to work in the other direction. Instance-based methods start with a particular example and see how it can be generalized to cover nearby examples in the same class.

Overfitting-Avoidance Bias

Overfitting-avoidance bias is often just another kind of search bias. But because it addresses a rather special problem, we treat it separately. Recall the disjunction problem described previously. The problem is that if disjunction is allowed, useless concept descriptions that merely summarize the data become possible, whereas if it is prohibited, some concepts are unlearnable. To get around this problem, it is common to search the concept space starting with the simplest concept descriptions and proceeding to more complex ones: simplest-first ordering. This biases the search toward simple concept descriptions.

Using a simplest-first search and stopping when a sufficiently complex concept description is found is a good way of avoiding overfitting. It is sometimes called *forward pruning* or *prepruning* because complex descriptions are pruned away before they are reached. The alternative, *backward pruning* or *postpruning,* is also viable. Here, we first find a description that fits the data well and then prune it back to a simpler description that also fits the data. This is not as redundant as it sounds: often the only way to arrive at a simple theory is to find a complex one and then simplify it. Forward and backward pruning are both a kind of overfitting-avoidance bias.

In summary, although generalization as search is a nice way to think about the learning problem, bias is the only way to make it feasible in practice. Different learning algorithms correspond to different concept description spaces searched with different biases. This is what makes it interesting: different description languages and biases serve some problems well and other problems badly. There is no universal "best" learning method—as every teacher knows!

1.6 DATA MINING AND ETHICS

The use of data—particularly data about people—for data mining has serious ethical implications, and practitioners of data mining techniques must act responsibly by making themselves aware of the ethical issues that surround their particular application.

When applied to people, data mining is frequently used to discriminate—who gets the loan, who gets the special offer, and so on. Certain kinds of discrimination—racial, sexual, religious, and so on—are not only unethical but also illegal. However, the situation is complex: everything depends on the application. Using sexual and racial information for medical diagnosis is certainly ethical, but using the same information when mining loan payment behavior is not. Even when sensitive information is discarded, there is a risk that models will be built that rely on variables that can be shown to substitute for racial or sexual characteristics. For example, people frequently live in areas that are associated with particular ethnic identities, so using an area code in a data mining study runs the risk of building models that are based on race—even though racial information has been explicitly excluded from the data.

It is widely accepted that before people make a decision to provide personal information they need to know how it will be used and what it will be used for, what steps will be taken to protect its confidentiality and integrity, what the consequences of supplying or withholding the information are, and any rights of redress they may have. Whenever such information is collected, individuals should be told these things—not in legalistic small print but straightforwardly in plain language they can understand.

The potential use of data mining techniques means that the ways in which a repository of data can be used may stretch far beyond what was conceived when the data was originally collected. This creates a serious problem: it is necessary to determine the conditions under which the data was collected and for what purposes it may be used. Does the ownership of data bestow the right to use it in ways other than those purported when it was originally recorded? Clearly in the case of explicitly collected personal data it does not. But in general the situation is complex.

Surprising results emerge from data mining. For example, it has been reported that one of the leading consumer groups in France has found that people with red cars are more likely to default on their car loans. What is the status of such a "discovery"? What information is it based on? Under what conditions was that information collected? In what ways is it ethical to use it? Clearly, insurance companies are in the business of discriminating among people based on stereotypes—young males pay heavily for automobile insurance—but such stereotypes are not based solely on statistical correlations; they also involve commonsense knowledge about the world. Whether the preceding finding says something about the kind of person who chooses a red car, or whether it should be discarded as an irrelevancy is a matter for human judgment based on knowledge of the world, rather than on purely statistical criteria.

When presented with data, you need to ask who is permitted to have access to it, for what purpose it was collected, and what kind of conclusions is it legitimate to draw from it. The ethical dimension raises tough questions for those involved in practical data mining. It is necessary to consider the norms of the community that is used to dealing with the kind of data involved, standards that may have evolved over decades or centuries but ones that the information specialist may not know. For example, did you know that in the library community, it is taken for granted that the privacy of readers is a right that is jealously protected? If you call your university library and ask who has such-and-such a textbook out on loan, they will not tell you. This prevents a student from being subjected to pressure from an irate professor to yield access to a book that she desperately needs for her latest grant application. It also prohibits inquiry into the dubious recreational reading tastes of the university ethics committee chairperson. Those who build, say, digital libraries may not be aware of these sensitivities and might incorporate data mining systems that analyze and compare individuals' reading habits to recommend new books—perhaps even selling the results to publishers!

In addition to community standards for the use of data, logical and scientific standards must be adhered to when drawing conclusions from it. If you do come up with conclusions (such as red car owners being greater credit risks), you need to attach caveats to them and back them up with arguments other than purely statistical ones. The point is that data mining is just a tool in the whole process: It is people who take the results, along with other knowledge, and decide what action to apply.

Data mining prompts another question, which is really a political one: To what use are society's resources being put? We mentioned previously the application of data mining to basket analysis, where supermarket checkout records are analyzed to detect associations among items that people purchase. What use should be made of the resulting information? Should the supermarket manager place the beer and chips together, to make it easier for shoppers, or farther apart, making it less convenient for them, maximizing their time in the store, and therefore increasing their likelihood of being drawn into unplanned further purchases? Should the manager move the most expensive, most profitable diapers near the beer, increasing sales to harried fathers of a high-margin item and add further luxury baby products nearby?

Of course, those who use advanced technologies should consider the wisdom of what they are doing. If *data* is characterized as recorded facts, then *information* is the set of patterns, or expectations, that underlie the data. You could go on to define *knowledge* as the accumulation of your set of expectations and *wisdom* as the value attached to knowledge. Although we will not pursue it further here, this issue is worth pondering.

As we saw at the very beginning of this chapter, the techniques described in this book may be called on to help make some of the most profound and intimate decisions that life presents. Data mining is a technology that we need to take seriously.

1.7 RESOURCES

This section describes papers, books, and other resources relevant to the material covered in this chapter. The human *in vitro* fertilization research mentioned in the opening of this chapter was undertaken by the Oxford University Computing Laboratory, and the research on cow culling was performed in the Computer Science Department at the University of Waikato, New Zealand.

The example of the weather problem is from Quinlan (1986) and has been widely used to explain machine learning schemes. The corpus of example problems mentioned in the introduction to Section 1.2 is available from Blake et al. (1998). The contact lens example is from Cendrowska (1998), who introduced the PRISM rule-learning algorithm. The iris dataset was described in a classic early paper on statistical inference (Fisher 1936). The labor negotiations data is from the *Collective Bargaining Review*, a publication of Labour Canada issued by the Industrial Relations Information Service (BLI 1988), and the soybean problem was first described by Michalski and Chilausky (1980).

Some of the applications in Section 1.3 are covered in an excellent paper that gives plenty of other applications of machine learning and rule induction (Langley

& Simon 1995); another source of fielded applications is a special issue of the *Machine Learning Journal* (Kohavi & Provost 1998). The loan company application is described in more detail by Michie (1989), the oil slick detector is from Kubat et al. (1998), the electric load forecasting work is by Jabbour et al. (1988), and the application to preventative maintenance of electromechanical devices is from Saitta and Neri (1998). Fuller descriptions of some of the other projects mentioned in Section 1.3 (including the figures of dollars saved and related literature references) appear at the websites of the Alberta Ingenuity Centre for Machine Learning and MLnet, a European network for machine learning.

The book *Classification and Regression Trees* mentioned in Section 1.4 is by Breiman et al. (1984), and the independently derived but similar scheme of Quinlan was described in a series of papers that eventually led to a book (Quinlan 1993).

The first book on data mining appeared in 1991 (Piatetsky-Shapiro & Frawley 1991), a collection of papers presented at a workshop on knowledge discovery in databases in the late 1980s. Another book from the same stable has appeared since (Fayyad et al. 1996) from a 1994 workshop. There followed a rash of business-oriented books on data mining, focusing mainly on practical aspects of how it can be put into practice with only superficial descriptions of the technology that underlies the methods used. They are valuable sources of applications and inspiration. For example, Adriaans and Zantige (1996) from Syllogic, a European systems and database consultancy, provide an early introduction to data mining. Berry and Linoff (1997), from a Pennsylvania-based company specializing in data warehousing and data mining, give an excellent and example-studded review of data mining techniques for marketing, sales, and customer support. The work of Cabena et al. (1998), written by people from five international IBM laboratories, presents an overview the data mining process with many examples of real-world applications. Dhar and Stein (1997) give a business perspective on data mining and include broad-brush, popularized reviews of many of the technologies involved. Groth (1998), working for a provider of data mining software, gives a brief introduction to data mining and then a fairly extensive review of data mining software products; the book includes a CD containing a demo version of his company's product. Weiss and Indurkhya (1998) look at a wide variety of statistical techniques for making predictions from what they call "big data." Han and Kamber (2001) cover data mining from a database perspective, focusing on the discovery of knowledge in large corporate databases. Finally, Hand et al. (2001) produced an interdisciplinary book on data mining from an international group of authors who are well respected in the field.

Books on machine learning, on the other hand, tend to be academic texts suited for use in university courses rather than practical guides. Mitchell (1997) wrote an excellent book that covers many techniques of machine learning, including some—notably genetic algorithms and reinforcement learning—that are not covered here. Langley (1996) offers another good text. Although the previously mentioned book by Quinlan (1993) concentrates on a particular learning algorithm, C4.5, it is a good introduction to some of the problems and techniques of machine learning. An excellent book on machine learning from a statistical perspective is from Hastie et al. (2001). This is a theoretically oriented work and is beautifully produced with apt and telling illustrations.

Pattern recognition is a topic that is closely related to machine learning, and many of the same techniques apply. Duda et al. (2001) offer the second edition of a classic and successful book on pattern recognition (Duda & Hart 1973). Ripley (1996) and Bishop (1995) describe the use of neural networks for pattern recognition. Data mining with neural networks is the subject of a book by Bigus (1996) of IBM, which features the IBM Neural Network Utility Product that he developed.

There is a great deal of current interest in support vector machines. Cristianini and Shawe-Taylor (2000) give a nice introduction, and a follow-up work generalizes this to cover additional algorithms, kernels, and solutions with applications to pattern discovery problems in fields such as bioinformatics, text analysis, and image analysis (Shawe-Taylor & Cristianini 2004). Schölkopf and Smola (2002) provide a comprehensive introduction to support vector machines and related kernel methods by two young researchers who did their PhD research in this rapidly developing area.