Chapter 3: Data Preparation as a Process

Overview

Data preparation has been placed in the context of data exploration, in which the problem to be solved, rather than the technology, is paramount. Without identifying the problem to solve, it is hard to define how to extract value from the data mining activities that follow. Equally important is specifying the form of a solution. Without a firm idea of what success looks like, it is hard to determine if indeed the result found, and the form that it is delivered in, have actually succeeded. Having specified what a suitable solution looks like, and collected or discovered appropriate data, you can begin the process of data mining.

Data mining is about working with data, which to a greater or lesser degree reflects some real-world activity, event, or object. In this discussion of data preparation for mining, there is a close focus on exploring more exactly what data represents, how and why it is transformed, and what can be done with and said about it. Much more will be said about data as the techniques for manipulating it are introduced. However, before examining how and why data is manipulated, a missing piece still remains to be addressed. Data needs to be prepared so that the information enfolded within it is most easily accessed by the mining tools. The missing piece, the bridge to understanding, is the explanation of what the overall process looks like. The overview of the process as a whole provides a framework and a reference to understand where each component fits into the overall design. This chapter provides the overview. Most detail is deliberately left out so that the process may be seen holistically. The questions that must arise from such a quick dash across the landscape of data preparation are answered in later chapters when each area is revisited in more detail.

Preparation of data is not a process that can be carried out blindly. There is no automatic tool that can be pointed at a data set and told to just “fix” the data. Maybe one day, when artificial intelligence techniques are a good bit more intelligent than they are today, fully automatic data preparation will become more feasible. Until that day there will remain as much art as science in good data preparation. However, just because there is art involved in data preparation does not mean that powerful techniques are not available or useful.

Because data preparation techniques cannot be completely automated, it is necessary to apply them with knowledge of their effect on the data being prepared. Understanding their function and applicability may be more important than understanding how the tools actually work. The functionality of each tool can be captured in computer code and regarded as a “black box.” So long as the tools perform reliably and as intended, knowledge of how the transformations are actually performed is far less important than understanding the appropriate use and limitations of each of the encapsulated techniques.

Art there may be, but successful practice of the art is based on understanding the overall issues and objectives, and how all the pieces relate together. Gaining that understanding of the broad picture is the purpose of this chapter. It connects the description of the data exploration process, data, data sets, and mining tools with data preparation into a whole. Later chapters discuss the detail of what needs to be done to prepare data, and how to do it. This chapter draws together these themes and discusses when and why particular techniques need to be applied and how to decide which technique, from the variety available, needs to be used.

3.1 Data Preparation: Inputs, Outputs, Models, and Decisions

The process takes inputs and yields outputs. The inputs consist of raw data and the miner's decisions (selecting the problem, possible solution, modeling tools, confidence limits, etc.). The outputs are two data sets and the Prepared Information Environment (PIE) modules. Figure 3.1 illustrates this. The decisions that have to be made concern the data, the tools to be used for mining, and those required by the solution.

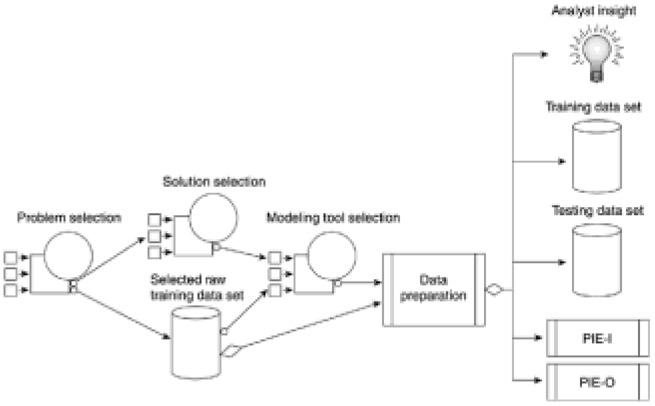


Figure 3.1 The data preparation process illustrating the major decisions, data, and process inputs and outputs.

This section explains

- What the inputs are, what the outputs are, what they do, and why they're needed

- How modeling tools affect what is done

- The stages of data preparation and what needs to be decided at each stage

The fundamental purpose of data preparation is to manipulate and transform raw data so that the information content enfolded in the data set can be exposed, or made more easily accessible. The best way to actually make the changes depends on two key decisions: what the solution requires and what the mining tool requires. While these decisions affect how the data is prepared, the inputs to and outputs from the process are not affected.

During this overview of data preparation, the actual inner workings of the preparation process will be regarded as a black box. The focus here is in what goes into and what comes out of the preparation process. By ignoring the details of the actual preparation process at this stage, it is easier to see why each of the inputs is needed, and the use of each of the output pieces. The purpose here is to try to understand the relationships between all of the pieces, and the role of each piece. With that in place, it is easier to understand the necessity of each step of the preparation process and how it fits into the whole picture.

At the very highest level, mining takes place in three steps:

1. Prepare the data

2. Survey the data

3. Model the data

Each of these steps has different requirements in the data preparation process. Each step takes place separately from the others, and each has to be completed before the next can begin. (Which doesn’t mean that the cycle does not repeat when results of using the model are discovered. Getting the model results might easily mean that the problem or solution needs to be redefined, or at least that more/different/better data is found, which starts off the cycle afresh.)

3.1.1 Step 1: Prepare the Data

Figure 3.1 shows the major steps in the data preparation process. Problem selection is a decision-and-selection process affecting both solution selection and data selection. This has been extensively discussed in Chapter 1 and will not be reiterated here. Modeling tool selection is driven by the nature of the specified solution and by the data available, which is discussed later in this chapter in “Modeling Tools and Data Preparation.” Chapter 12 discusses tool use and the effect of using prepared data with different techniques.

Some initial decisions have to be made about how the data is to be prepared. In part, the nature of the problem determines tool selection. If rules are needed, for example, it is necessary to select a tool that can produce them. In turn, tool selection may influence how the data is prepared. Inspection of the data may require reformatting or creating some additional features. Looking at the preliminary decisions that need to be made before applying the appropriate techniques is covered in part in this chapter and also in the next.

The miner must determine how the data is to be appropriately prepared. This is based on the nature of the problem, the tools to be used, and the types of variables in the data set. With this determined, preparation begins. Preparation has to provide at least four separate components as outputs:

- A training data set

- A testing data set

- A PIE-I (Prepared Information Environment Input module)

- A PIE-O (Prepared Information Environment Output module)

Each of these is a necessary output and has a specific function, purpose, and use. Each is needed because of the nature of data sets extracted from the real world. These four components are the absolute minimum required for mining, and it is likely that additional data sets will be needed. For example, a validation data set may also be considered essential. It is not included in the list of four essential components since valid models can be created without actually validating them at the time the miner creates them. If there is insufficient data on hand for three representative data sets, for instance, the model could be validated later when more data is available. But in some sense, each of these four components is indispensable. Why these four?

The training data set is required to build a model. A testing data set is required for the modeling tool to detect overtraining. The PIE-I is what allows the model to be applied to other data sets. The PIE-O translates the model's answers into applicable measured values. Since these are the critical output components of the data preparation process, we must look at each of these four components more closely.

A mining tool's purpose is to learn the relationships that exist between the variables in the data set. Preparation of the training data set is designed to make the information enfolded in the data set as accessible and available as possible to the modeling tool. So what's the purpose of the test data set?

Data sets are not perfect reflections of the world. Far from it. Even if they were, the nature of the measuring process necessarily captures uncertainty, distortion, and noise. This noise is integral to the nature of the world, not just the result of mistakes or poor procedures. There are a huge variety of errors that can infect data. Many of these errors have already been discussed in Chapter 2—for instance, measurement error. Some of these errors are an inextricable part of the data and cannot be removed or "cleaned." The accumulated errors, and other forms of distortion of "true" values, are called *noise*. The term "noise" comes from telephony, where the added error to the true signal is actually heard as the noise of a hiss in a telephone earpiece. AM radio also suffers from noise in the transmitted signal, especially if lightning is nearby. In general, noise simply means distortion of the original signal. Somehow a modeling tool must deal with the noise in the data.

Each modeling tool has a different way of expressing the nature of the relationships that it finds between variables. But however it is expressed, some of the relationship between variables exists because of the “true” measurement and some part is made up of the relationship caused by the noise. It is very hard, if not impossible, to precisely determine which part is made up from the underlying measurement and which from the noise. However, in order to discover the “true” underlying relationship between the variables, it is vital to find some way of estimating which is relationship and which is noise.

One problem with noise is that there is no consistent detectable pattern to it. If there were, it could be easily detected and removed. So there is an unavoidable component in the training set that should not be characterized by the modeling tool. There are ways to minimize the impact of noise that are discussed later, but there always remains some irreducible minimum. In fact, as discussed later, there are even circumstances when it is advantageous to add noise to some portion of the training set, although this deliberately added noise is very carefully constructed.

Ideally, a modeling tool will learn to characterize the underlying relationships inside the data set without learning the noise. If, for example, the tool is learning to make predictions of the value of some variable, it should learn to predict the true value rather than some distorted value. During training there comes a point at which the model has learned the underlying relationships as well as is possible. Anything further learned from this point will be the noise. Learning noise will make predictions from data inside the training set better. In any two subsets of data drawn from an identical source, the underlying relationship will be the same. The noise, on the other hand, not representing the underlying relationship, has a very high chance of being different in the two data sets. In practice, the chance of the noise patterns being different is so high as to amount to a practical certainty. This means that predictions from any data set other than the training data set will very likely be worse as noise is learned, not better. It is this relationship between the noise in two data sets that creates the need for another data set, the test data set.

To illustrate why the test data set is needed, look at Figure 3.2. The figure illustrates measurement values of two variables; these are shown in two dimensions. Each data point is represented by an X. Although an X is shown for convenience, each X actually represents a fuzzy patch on the graph. The X represents the actual measured value that may or may not be at the center of the patch. Suppose the curved line on the graph represents the underlying relationship between the two variables. The Xs cluster about the line to a greater or lesser degree, displaced from it by the noise in the relationship. The data points in the left-hand graph represent the training data set. The right-hand graph represents the test data set. The underlying relationship is identical in both data sets. The difference between the two data sets is only the noise added to the measurements. The noise means that the actual measured data points are not identically positioned in the two data sets. However, although different in values, note that by using the appropriate data preparation techniques discussed later in the book (see, for example, Chapter 11), it can be known that both data sets do adequately represent the underlying relationship even though the relationship itself is not known.

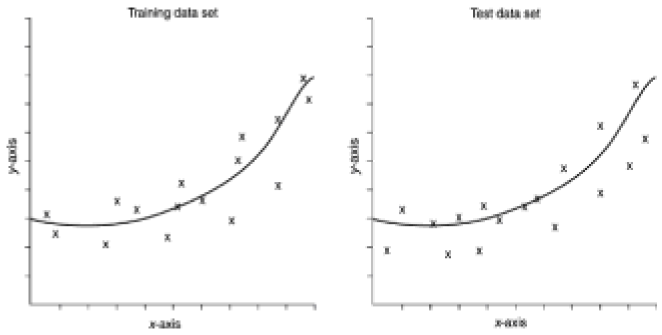


Figure 3.2 The data points in the training and test data sets with the underlying relationship illustrated by the continuous curved lines.

Suppose that some modeling tool trains and tests on the two data sets. After each attempt to learn the underlying relationship, some metric is used to measure the accuracy of the prediction in both the training and test data sets. Figure 3.3 shows four stages of training, and also the fit of the relationship proposed by the tool at a particular stage. The graphs on the left represent the training data set; the graphs on the right represent the test data set.

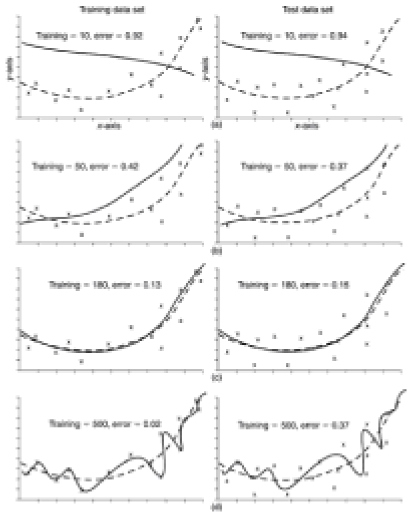


Figure 3.3 The four stages of training with training data sets (left) and test data sets (right): poor fit (a), slightly improved fit due to continued training (b), near-perfect fit (c), and noise as a result of continued training beyond best fit point (d).

In Figure 3.3(a), the relationship is not well learned, and it fits both data sets about equally poorly. After more training, Figure 3.3(b) shows that some improvement has occurred in learning the relationship, and again the error is now lower in both data sets, and about equal. In Figure 3.3(c), the relationship has been learned about as well as is possible from the data available, and the error is low, and about equal in both data sets. In Figure 3.3(d), learning has continued in the training (left) data set, and an almost perfect relationship has been extracted between the two variables. The problem is that the modeling tool has learned noise. When the relationship is tried in the test (right) data set, it does not fit the data there well at all, and the error measure has increased.

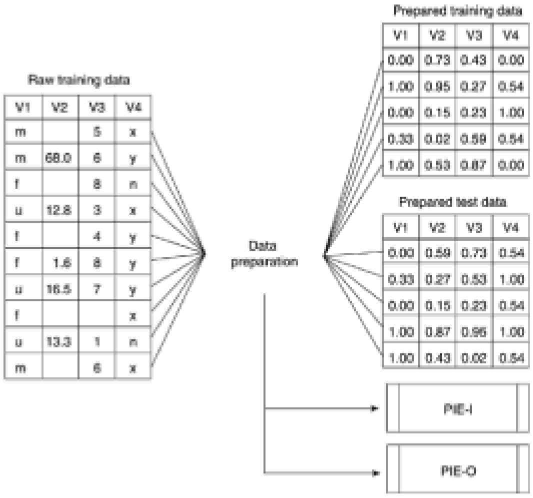
As is illustrated here, the test data set has the same underlying “true” relationships as the training data set, but the two data sets contain noise relationships that are different. During training, if the predictions are tested in both the training and test data sets, at first the predictions will improve in both. So the tool is improving its real predictive power as it learns the underlying relationships and improves its performance based on those relationships. In the example shown in Figure 3.3, real-world improvement continues until the stage shown in Figure 3.3(c). At that point the tool will have learned the underlying relationships as well as the training data set allows. Any further improvement in prediction will then be caused by learning noise. Since the noise differs between the training set and the test set, this is the point at which predictive performance will degrade in the test set. This degradation begins if training continues after the stage shown in Figure 3.3(c), and ends up with the situation shown in Figure 3.3(d). The time to stop learning is at the stage in Figure 3.3(c).

As shown, the relationships are learned in the training data set. The test data set is used as a check to try to avoid learning noise. Here is a very important distinction: the training data set is used for discovering relationships, while the test data set is used for discovering noise. The instances in the test data set are *not valid* for independently testing any predictions. This is because the test data has in fact been used by the modeling tool as part of the training, albeit for noise. In order to independently test the model for predictive or inferential power, yet another data set is needed that does not include any of the instances in either the training or test data sets.

So far, the need for two learning sets, training and test, has been established. It may be that the miner will need another data set for assessing predictive or inferential power. The chances are that all of these will be built from the same source data set, and at the same time. But whatever modifications are made to one data set to prepare it for modeling must also be made to any other data set. This is because the mining tool has learned the relationships in prepared data. The tool has to have data prepared in all data sets in an identical way. Everything done in one has to be done in all. But what do these prepared data sets look like? How does the preparation process alter the data?

Figure 3.4 shows the data view of what is happening during the data preparation process.

The raw training data in this example has a couple of categorical values and a couple of numeric values. Some of the values are missing. This raw data set has to be converted into a format useful for making predictions. The result is that the training and test sets will be turned into all numeric values (if that is what is needed) and normalized in range and distribution, with missing values appropriately replaced. These transformations are illustrated on the right side of Figure 3.4. It is obvious that all of the variables are present and normalized. (Figure 3.4 also shows the PIE-I and PIE-O. These are needed for later use.)



Raw training data

V1	V2	V3	V4
m		5	x
m	68.0	6	y
f		8	n
u	12.8	3	x
f		4	y
f	1.6	8	y
u	16.5	7	y
f			x
u	13.3	1	n
m		6	x

Prepared training data

V1	V2	V3	V4
0.00	0.73	0.43	0.00
1.00	0.95	0.27	0.54
0.00	0.15	0.23	1.00
0.33	0.02	0.59	0.54
1.00	0.53	0.87	0.00

Prepared test data

V1	V2	V3	V4
0.00	0.59	0.73	0.54
0.33	0.27	0.53	1.00
0.00	0.15	0.23	0.54
1.00	0.87	0.96	1.00
1.00	0.43	0.02	0.54

Figure 3.4 Data preparation process transforms raw data into prepared training and test sets, together with the PIE-I and PIE-O modules.

3.1.2 Step 2: Survey the Data

Mining includes surveying the data, that is, taking a high-level overview to discover what is contained in the data set. Here the miner gains enormous and powerful insight into the nature of the data. Although this is an essential, critical, and vitally important part of the data mining process, we will pass quickly over it here to continue the focus on the process of data preparation.

3.1.3 Step 3: Model the Data

In this stage, the miner applies the selected modeling tool to the training and test data sets to produce the desired predictive, inferential, or other model desired. (See Figure 3.5.) Since this book focuses on data preparation, a discussion of modeling issues, methods, and techniques is beyond the present scope. For the purposes here it will be assumed that the model is built.

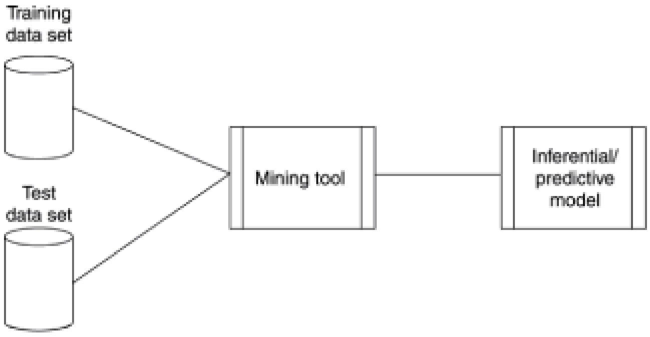
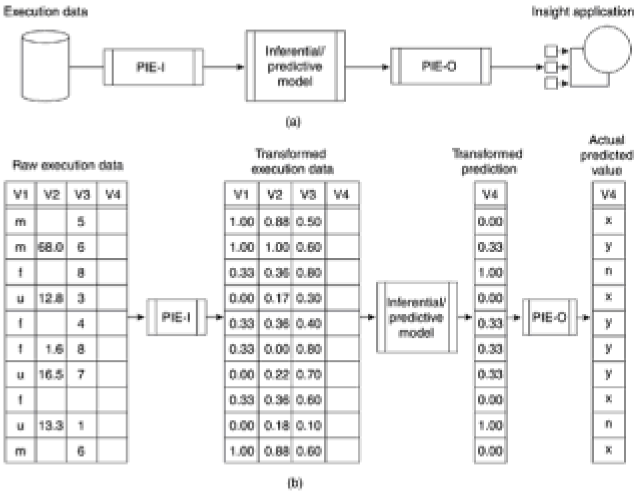


Figure 3.5 Mining the inferential or predictive model.

3.1.4 Use the Model

Having created a satisfactory model, in order to be of practical use it must be applied to “live” data, also called the *execution data*. Presumably, it is very similar in character to the training and test data. It should, after all, be drawn from the same population (discussed in Chapter 5), or the model is not likely to be applicable. Because the execution data is in its “raw” form, and the model works only with prepared data, it is necessary to transform the execution data in the same way that the training and test data were transformed. That is the job of the PIE-I: it takes execution data and transforms it as shown in Figure 3.6(a). Figure 3.6(b) shows what the actual data might look like. In the example it is variable V4 that is missing and needs to be predicted.



Raw execution data

V1	V2	V3	V4
m		5	
m	68.0	6	
f		8	
u	12.8	3	
f		4	
f	1.6	8	
u	16.5	7	
f			
u	13.3	1	
m		6	

Transformed execution data

V1	V2	V3	V4
1.00	0.88	0.50	
1.00	1.00	0.60	
0.33	0.36	0.80	
0.00	0.17	0.30	
0.33	0.36	0.40	
0.33	0.00	0.80	
0.00	0.22	0.70	
0.33	0.36	0.60	
0.00	0.18	0.10	
1.00	0.88	0.60	

Transformed prediction

V4
0.00
0.33
1.00
0.00
0.33
0.33
0.33
0.00
1.00
0.00

Actual predicted value

V4
x
y
n
x
y
y
y
x
n
x

Figure 3.6 Run-time prediction or inferencing with execution data set (a). Stages that the data goes through during actual inference/prediction process (b).

Variable V4 is a categorical variable in this example. The data preparation, however, transformed all of the variables into scaled *numeric* values. The mined model will therefore predict the result in the form of scaled numeric values. However, the prediction must be given as a categorical value. This is the purpose of the PIE-O. It "undoes" the effect of the PIE-I. In this case, it converts the mined model outputs into the desired categorical values.

The whole purpose of the two parts of the PIE is to sit between the real-world data, cleaning and preparing the incoming data stream identically with the way the training and test sets were prepared, and converting predicted, transformed values back into real-world values. While the input execution data is shown as an assembled file, it is quite possible that the real-world application has to be applied to real-time transaction data. In this case, the PIE dynamically prepares each instance value in real time, taking the instance values from whatever source supplies them.

3.2 Modeling Tools and Data Preparation

As always, different tools are valuable for different jobs. So too it is with the modeling tools available. Prior to building any model, the first two questions asked should be: What do we need to find out? and Where is the data? Deciding what to find out leads to the next two questions: Exactly what do we want to know? and In what form do we want to know it? (These are issues discussed in Chapter 1.) A large number of modeling tools are currently available, and each has different features, strengths, and weaknesses. This is certainly true today and is likely to be even more true tomorrow. The reason for the greater differences tomorrow lies in the way the tools are developing.

For a while the focus of data mining has been on algorithms. This is perhaps natural since various machine-learning algorithms have competed with each other during the early, formative stage of data exploration development. More and more, however, makers of data exploration tools realize that the users are more concerned with business problems than algorithms. The focus on business problems means that the newer tools are being packaged to meet specific business needs much more than the early, general-purpose data exploration tools. There are specific tools for market segmentation in database marketing, fraud detection in credit transactions, churn management for telephone companies, and stock market analysis and prediction, to mention only four. However, these so-called "vertical market" applications that focus on specific business needs do have drawbacks. In becoming more capable in specific areas, usually by incorporating specific domain knowledge, they are constrained to produce less general-purpose output. As with most things in life, the exact mix is a compromise.

What this means is that the miner must take even more care now than before to understand the requirements of the modeling tool in terms of data preparation, especially if the data is to be prepared "automatically," without much user interaction. Consider, for example, a futures-trading automation system. It may be intended to predict the movement, trend, and probability of profit for particular spreads for a specific futures market. Some sort of hybrid model works well in such a scenario. If past and present market prices are to be included, they are best regarded as continuous variables and are probably well modeled using a neural-network-based approach. The overall system may also use input from categorized news stories taken off a news wire. News stories are read, categorized, and ranked according to some criteria. Such categorical data is better modeled using one of the rule extraction tools. The output from both of these tools will itself need preparation before being fed into some next stage. The user sees none of the underlying technicality, but the builder of the system will have to make a large number of choices, including those about the optimal data preparation techniques to meet each objective. Categorical data and numeric data may well, and normally do, require different preparation techniques.

At the project design stage, or when directly using general-purpose modeling tools, it is important to be aware of the needs, strengths, and weaknesses of each of the tools employed. Each tool has a slightly different output. It is harder to produce humanly comprehensible rules from any neural network product than from one of the rule extraction variety, for example. Almost certainly it is possible to transform one type of output to another use—to modify selection rules, for instance, into providing a score—but it is frequently easier to use a tool that provides the type of output required.

3.2.1 How Modeling Tools Drive Data Preparation

Modeling tools come in a wide variety of flavors and types. Each tool has its strengths and weaknesses. It is important to understand which particular features of each tool affect how data is prepared.

One main factor by which mining tools affect data preparation is the sensitivity of the tool to the numeric/categorical distinction. A second is sensitivity to missing values, although this sensitivity is largely misunderstood. To understand why these distinctions are important, it is worth looking at what modeling tools try to do.

The way in which modeling tools characterize the relationships between variables is to partition the data such that data in particular partitions associates with particular outcomes. Just as some variables are discrete and some variables are continuous, so some tools partition the data continuously and some partition it discretely. In the examples shown in Figures 3.2 and 3.3 the learning was described as finding some “best-fit” line characterizing the data. This actually describes a continuous partitioning in which you can imagine the partitions are indefinitely small. In such a partitioning, there is a particular mathematical relationship that allows prediction of output value(s) depending on how far distant, and in exactly what direction (in state space), the instance value lies from the optimum. Other mining tools actually create discrete partitions, literally defining areas of state space such that if the predicting values fall into that area, a particular output is predicted. In order to examine what this looks like, the exact mechanism by which the partitions are created will be regarded as a black box.

We have already discussed in Chapter 2 how each variable can be represented as a dimension in state space. For ease of description, we'll use a two-dimensional state space and only two different types of instances. In any more realistic model there will almost certainly be more, maybe many more, than two dimensions and two types of instances. Figure 3.7 shows just such a two-dimensional space as a graph. The Xs and Os in Figure 3.7(a) show the positions of instances of two different instance types. It is the job of the modeling tool to find optimal ways of separating the instances.

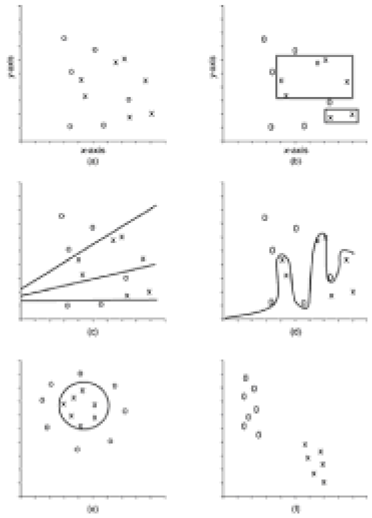


Figure 3.7 Modeling a data set: separating similar data points (a), straight lines parallel to axes of state space (b), straight lines not parallel to axes of state space (c), curves (d), closed area (e), and ideal arrangement (f).

Various "cutting" methods are directly analogous to the ways in which modeling tools separate data. Figure 3.7(b) shows how the space might be cut using straight lines parallel to the axes of the graph. Figure 3.7(c) also shows cuts using straight lines, but in this figure they are not constrained to be parallel to the axes. Figure 3.7(d) shows cuts with lines, but they are no longer constrained to be straight. Figure 3.7(e) shows how separation may be made using areas rather than lines, the areas being outlined.

Whichever method or tool is used, it is generally true that the cuts get more complex traveling from Figure 3.7(b) to 3.7(e). The more complex the type of cut, the more computation it takes to find exactly where to make the cut. More computation translates into "longer." Longer can be very long, too. In large and complex data sets, finding the optimal places to cut can take days, weeks, or months. It can be a very difficult problem to decide when, or even if, some methods have found optimal ways to divide data. For this reason, it is always beneficial to make the task easier by attempting to restructure the data so that it is most easily separated. There are a number of "rules of thumb" that work to make the data more tractable for modeling tools. Figure 3.7(f) shows how easy a time the modeling tool would have if the data could be rearranged as shown during preparation! Maybe automated preparation cannot actually go as far as this, but it can go at least some of the way, and as far as it can go is very useful.

In fact, the illustrations in Figure 3.7 do roughly correspond with the ways in which different tools separate the data. They are not precisely accurate because each vendor modifies “pure” algorithms in order to gain some particular advantage in performance. It is still worthwhile considering where each sits, since the underlying method will greatly affect what can be expected to be learned from each tool.

3.2.2 Decision Trees

Decision trees use a method of logical conjunctions to define regions of state space. These logical conjunctions can be represented in the form of “If . . . then” rules. Generally a decision tree considers variables individually, one at a time. It starts by finding the variable that best divides state space and creating a “rule” to specify the split. The decision tree algorithm finds for each subset of the instances another splitting rule. This continues until the triggering of some stopping criterion. Figure 3.8 illustrates a small portion of this process.

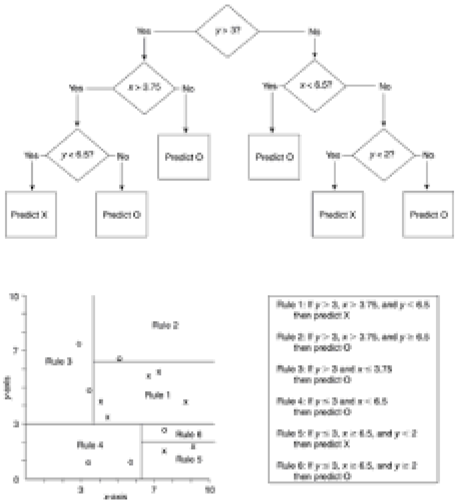


Figure 3.8 A decision tree cutting state space.

Due to the nature of the splitting rules, it can easily be seen that the splits have to be parallel to one of the axes of state space. The rules can cut out smaller and smaller pieces of state space, but always parallel to the axes.

3.2.3 Decision Lists

Decision lists also generate “If . . . then” rules, and graphically appear similar to decision trees. However, decision trees consider the subpopulation of the “left” and “right” splits separately and further split them. Decision lists typically find a rule to well characterize some small portion of the population that is then removed from further consideration. At that point it seeks another rule for some portion of the remaining instances. Figure 3.9 shows how this might be done.

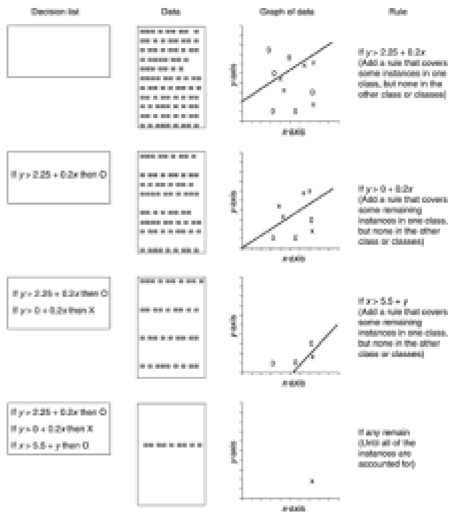
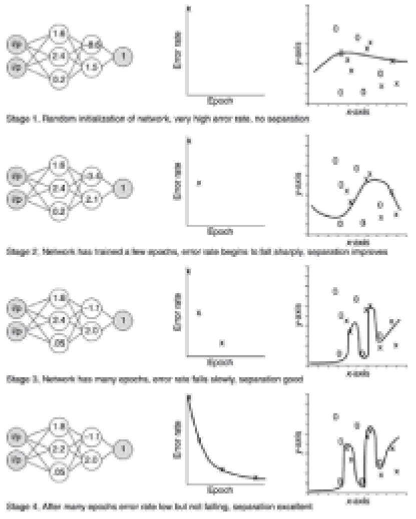

Decision list
Data
Graph of data
Rule
y-axis
x-axis
If y > 2.25 + 0.2x
(Add a rule that covers some instances in one class, but none in the other class or classes)
If y > 2.25 + 0.2x then O
If y > 0 + 0.2x
(Add a rule that covers some remaining instances in one class, but none in the other class or classes)
If y > 2.25 + 0.2x then O
If y > 0 + 0.2x then X
If x > 5.5 + y
(Add a rule that covers some remaining instances in one class, but none in the other class or classes)
If y > 2.25 + 0.2x then O
If y > 0 + 0.2x then X
If x > 5.5 + y then O
If any remain
(Until all of the instances are accounted for)

Figure 3.9 A decision list inducing rules that cover portions of the remaining data until all instances are accounted for.

(Although this is only the most cursory look at basic algorithms, it must be noted that many practical tree and list algorithms at least incorporate techniques for allowing the cuts to be other than parallel to the axes.)

3.2.4 Neural Networks

Neural networks allow state space to be cut into segments with cuts that are not parallel to the axes. This is done by having the network learn a series of “weights” at each of the “nodes.” The result of this learning is that the network produces gradients, or sloping lines, to segment state space. In fact, more complex forms of neural networks can learn to fit curved lines through state space, as shown in Figure 3.10. This allows remarkable flexibility in finding ways to build optimum segmentation. Far from requiring the cuts to be parallel to the axes, they don’t even have to be straight.



Stage 1. Random initialization of network, very high error rate, no separation

Stage 2. Network has trained a few epochs, error rate begins to fall sharply, separation improves

Stage 3. Network has many epochs, error rate falls slowly, separation good

Stage 4. After many epochs error rate low but not falling, separation excellent

Figure 3.10 Neural network training.

As the cuts become less linear, and not parallel to the axes, it becomes more and more difficult to express the rules in the form of logical conjunctions—the “If . . . then” rules. The expression of the relationships becomes more like fairly complex mathematical equations. A statistician might say they resemble “regression” equations, and indeed they do.

(Chapter 10 takes a considerably more detailed look at neural networks, although not for the purposes of predictive or inferential modeling.)

3.2.5 Evolution Programs

In fact, using a technique called *evolution programming*, it is possible to perform a type of regression known as *symbolic regression*. It has little in common with the process of finding regression equations that is used in statistical analysis, but it does allow for the discovery of particularly difficult relationships. It is possible to use this technique to discover the equation that would be needed to draw the curve in Figure 3.7(e).

3.2.6 Modeling Data with the Tools

There are more techniques available than those listed here; however, these are fairly representative of the techniques used in data mining tools available today. Demonstration versions of commercial tools based on some of these ideas are available on the CD-ROM accompanying this book. They all extend the basic ideas in ways the vendor feels enhances performance of the basic algorithm. These tools are included as they generally will benefit from having the data prepared in different ways.

Considered at a high level, modeling tools separate data using one of two approaches. The first way that tools use is to make a number of cuts in the data set, separating the total data set into pieces. This cutting continues until some stopping criterion is met. The second way is to fit a flexible surface, or at least a higher-dimensional extension of one (a manifold), between the data points so as to separate them. It is important to note that in practice it is probably impossible, with the information contained in the data set, to separate all of the points perfectly. Often, perfect separation is not really wanted anyway. Because of noise, the positioning of many of the points may not be truly representative of where they would be if it were possible to measure them without error. To find a perfect fit would be to learn this noise. As discussed earlier, the objective is for the tool to discover the underlying structure in the data without learning the noise.

The key difference to note between tools is that the discrete tools—those that cut the data set into discrete areas—are sensitive to differences in the rank, or order, of the values in the variables. The quantitative differences are not influential. Such tools have advantages and disadvantages. You will recall from Chapter 2 that a rank listing of the joint distances between American cities carries enough information to recover their geographical layout very accurately. So the rank differences do carry a very high information content. Also, discrete tools are not particularly troubled by outliers since it is the positioning in rank that is significant to them. An outlier that is in the 1000th-rank position is in that position whatever its value. On the other hand, discrete tools, not seeing the quantitative difference between values, cannot examine the fine structure embedded there. If there is high information content in the quantitative differences between values, a tool able to model continuous values is needed. Continuous tools can extract both quantitative and qualitative (or rank) information, but are very sensitive to various kinds of distortion in the data set, such as outliers. The choice of tool depends very much on the nature of the data coupled with the requirements of the problem.

The simplified examples shown in Figure 3.7 assume that the data is to be used to predict an output that is in one of two states—O or X. Typically, tools that use linear cuts do have to divide the data into such binary predictions. If a continuous variable needs to be predicted, the range of the variable has to be divided into discrete pieces, and a separate model built for predicting if the range is within a particular subrange. Tools that can produce nonlinear cuts can also produce the equations to make continuous predictions. This means that the output range does not have to be chopped up in the way that the linear cutting tools require.

These issues will be discussed again more fully later. It is also important to reiterate that, in practice, mining tool manufacturers have made various modifications so that the precise compromises made for each tool have to be individually considered.

3.2.7 Predictions and Rules

Tool selection has an important impact on exactly which techniques are applied to the unprepared data. All of the techniques described here produce output in one of two forms—predictions or rules. Data modeling tools end up expressing their learning either as a predicted number, a predicted categorical, or as a set of rules that can be used to separate the data in useful ways.

For instance, suppose that it is required as part of a solution to model the most likely value of the mortgage rate. The mortgage rate is probably best regarded as a continuous variable. Since a prediction of a continuous variable is needed, it indicates that the most appropriate tool to use for the model would be one that is capable of continuous predictions. Such a tool would probably produce some sort of equation to express the relationship of input values to the predicted value. Since a continuous value is required as output, it is advantageous, and works best, when the input values are also continuous. Thus, indications about the type of tools and some of the data preparation decisions are already made when the solution is selected.

Having decided on a predicted mortgage rate, perhaps it is required to make a model to determine if a particular prospective customer is or is not likely to respond to a solicitation with this rate. For this solution it might be most appropriate to use a model with a binary, yes/no output. The most appropriate tool is some sort of classifier that will classify records into the yes/no dichotomy required. Preparing data for a yes/no dichotomy may benefit from techniques such as binning that enhance the ability of many tools to separate the data. *Binning* is a technique of lumping small ranges of values together into categories, or "bins," for the purpose of reducing the variability (removing some of the fine structure) in a data set. For instance, customer information response cards typically ask for household income using "from-to" ranges in which household income falls. Those categories are "bins" that group ranges of income. There are circumstances in mining in which this can be useful.

Continuous and dichotomous modeling methods can be used for more than just making predictions. When building models to understand what is "driving" certain effects in the data set, the models are often used to answer questions about what features are important in particular areas of state space. Such modeling techniques are used to answer questions like "What are the underlying factors associated with fraudulent transactions in the branch offices?" Since the affecting factors may possibly be different from area to area of state space, it is important to use preparation techniques that retain as much of the fine structure—that is, the detailed fluctuations in the data set—over the full range of variability of the variables.

Looking for affecting factors is a form of inferential modeling. Examination of what is common to sets of rules is one way to discover the common themes present in particular situations, such as the branch office fraud alluded to above. The ability to give clear reasons for action are particularly important in several situations, such as credit approval or denial, where there is a legal requirement for explanation to be available. Generation of such rules can also be expressed, say, as SQL statements, if it is needed to extract parts of a data set. Perhaps a mailing list is required for all people meeting particular criteria. What is important here is to focus on how the required output affects the preparation of the input data, rather than the use to which the solution will be put.

3.2.8 Choosing Techniques

In summary, the effect that the choice of modeling tools has on data preparation is determined by the tool's characteristics. Is the tool better able to model continuous or categorical data? Since actual tools are all modifications of "pure" algorithms, this is a question that is hard to answer in its general form. Each tool has to be evaluated individually. Practically speaking, it is probably best to try several preparation techniques, preparing the data as continuous and also using several binning options to create categorized data. However, it is also important to use a mining tool that produces output appropriate to the needs of the solution. If the solution required calls for a categorical prediction, the tool needs to be able to produce such a solution and will probably benefit from categorical training, test, and execution data. The data preparation techniques discussed in this book are designed to allow preparation of the data set in a variety of ways. They allow the data to be manipulated as needed, so the miner can focus attention on deciding which are the appropriate techniques and tools to use in a particular situation.

3.2.9 Missing Data and Modeling Tools

Missing values form a very important issue in preparing data and were discussed in Chapter 2. Whenever there are missing values, it is vital that something be done about them. There are several methods for determining a suitable replacement value, but under no circumstances should the missing values be ignored or discarded. Some tools, particularly those that handle categorical values well, such as decision trees, are said to handle missing values too. Some really can; others can't. Some discrete-type modeling tools can actually elegantly ignore missing values, while others regard a missing value as just another categorical value, which is not really a satisfactory approach. Other tools, such as neural networks, require that each input be given a numeric value and any record that has a missing value has to be either completely ignored, or some default for the missing value must be created.

There are going to be problems with whatever default replacement approach is taken—very often major problems. At the very least, left untreated except by the default solution, missing values cause considerable distortion to the fabric of the data set. Not all missing values, for instance, can be assumed to represent the same value. Yet that is what a decision tree does if it assigns missing values to a separate category—assumes that they all have the same measured value. Similar distortions occur if some default numerical value is assigned. Clearly, a better solution needs to be found. Several choices are available, and the pros and cons of each method are discussed in detail in Chapter 8. For the time being, note that this is one of the issues that *must* be dealt with effectively for the best models to be built.

3.3 Stages of Data Preparation

Data preparation involves two sets of preparatory activities. The first are *nonautomated* activities that are procedural, or activities that result in a decision about the approach that the miner decides to take. There are many activities and decisions to be made in this stage that can be described as "basic preparation," and they are discussed in detail in the next chapter. The second set of activities are *automated* preparation activities. Detailed descriptions of the techniques used in the automated preparation stage, the demonstration code, and the process and decision points that go into data preparation round out the remaining chapters. What follows is a brief overview of the eight stages:

1. Accessing the data
2. Auditing the data
3. Enhancing and enriching the data
4. Looking for sampling bias
5. Determining data structure
6. Building the PIE
7. Surveying the data
8. Modeling the data

3.3.1 Stage 1: Accessing the Data

The starting point for any data preparation project is to *locate the data*. This is sometimes easier said than done! There are a considerable variety of issues that may hinder access to the nominated data, ranging from legal to connectivity. Some of these commonly encountered issues are reviewed later, but a comprehensive review of all issues is almost impossible, simply because every project provides unique circumstances. Nonetheless, locating and securing the source of data supply and ensuring adequate access is not only the first step, it is absolutely essential.

You might say, "Well, I have part of the problem licked because I have access to a data warehouse." It is a fact that data warehouses are becoming repositories of choice. More and more it is a warehouse that is to be mined. However, a warehouse is by no means essential in order to mine data. In fact, a warehouse can be positively detrimental to the mining effort, depending on how the data was loaded. Warehouses also have other drawbacks, a significant one being that they are often created with a particular structure to reflect some specific view of the enterprise. This imposed structure can color all modeling results if care is not taken to avoid bias.

3.3.2 Stage 2: Auditing the Data

Assuming that suitable data is available, the first set of basic issues that have to be addressed concern

- The source of supply

- The quantity of data

- The quality of the data

Building robust models requires data that is sufficient in quantity, and of high enough quality to create the needed model. A data audit provides a methodology for determining the status of the data set and estimates its adequacy for building the model. The reality is that the data audit does not so much assure that the model will be able to be built, but at least assures that the minimum requirements have been met.

Auditing requires examining small samples of the data and assessing the fields for a variety of features, such as number of fields, content of each field, source of each field, maximum and minimum values, number of discrete values, and many other basic metrics. When the data has been assessed for quantity and quality, a key question to ask is, Is there a justifiable reason to suppose that this data has the potential to provide the required solution to the problem? *Here is a critical place to remove the expectation of magic*. Wishful thinking and unsupported hopes that the data set that happens to be available will actually hold something of value seldom results in a satisfactory model. The answer to whether the hopes for a solution are in fact justified lies not in the data, but in the hopes! An important part of the audit, a nontechnical part, is to determine the true feasibility of delivering value with the resources available. Are there, in fact, good reasons for thinking that the actual data available can meet the challenge?

3.3.3 Stage 3: Enhancing and Enriching the Data

With a completed audit in hand, there is at least some firm idea of the adequacy of the data. If the audit revealed that the data does not really support the hopes founded on it, it may be possible to supplement the data set in various ways. Adding data is a common way to increase the information content. Many credit card issuers, for instance, will purchase information from outside agencies. Using this purchased data allows them to better assess the creditworthiness of their existing customers, or of prospects who are not yet their customers.

There are several ways in which the existing data can be manipulated to extend its usefulness. Such manipulation, for example, is to calculate price/earnings (P/E) ratios for modeling the value of share prices. So-called “fundamentalist” investors feel that this ratio has predictive value. They may be right. If they are, you may ask, “Since the price and the earnings are present in the source data, how would providing information about the P/E ratio help?” First, the P/E ratio represents an insight into the domain about what is important. This insight adds information to the modeling tool’s input. Second, presenting this precalculated information saves the modeling tool from having to learn division! Modeling tools can and do learn multiplicative relationships. Indeed, they can learn relationships considerably more complicated than that. However, it takes time and system resources to discover any relationship. Adding enough domain knowledge and learning assistance about important features can boost performance and cut development time dramatically. In some cases, it turns the inability to make any model into the ability to make useful models.

3.3.4 Stage 4: Looking for Sampling Bias

Sampling bias presents some particularly thorny problems. There are some automated methods for helping to detect sampling bias, but no automated method can match reasoned thought. There are many methods of sampling, and sampling is always necessary for reasons discussed in Chapter 5. *Sampling* is the process of taking a small piece of a larger data set in such a way that the small piece accurately reflects the relationships in the larger data set. The problem is that the true relationships that exist in the fullest possible data set (called the *population*) may, for a variety of reasons, be unknowable. That means that it is impossible to actually check to see if the sample is representative of the population in fact. It is critical to bend every effort to making sure that the data captured is as representative of the true state of affairs as possible.

While sampling is discussed in many statistical texts, miners face problems not addressed in such texts. It is generally assumed that the analyst (statistician/modeler) has some control over how the data is generated and collected. If not the analyst, at least the creator or collector of the data may be assumed to have exercised suitable control to avoid sampling bias. Miners, however, sometimes face collections of data that were almost certainly gathered for purposes unknown, by processes unsure, but that are now expected to assist in delivering answers to questions unthought of at the time. With the provenance of the data unknown, it is very difficult to assess what biases are present in the data, and that, if uncorrected, will produce erroneous and inapplicable models.

3.3.5 Stage 5: Determining Data Structure (Super-, Macro-, and Micro-)

Structure refers to the way in which the variables in a data set relate to each other. It is this structure that mining sets out to explore. Bias, mentioned above, stresses the natural structure of a data set so that the distorted data is less representative of the real world than unbiased data. But structure itself has various forms: super, macro, and micro.

Superstructure refers to the scaffolding erected to capture the data and form a data set. The superstructure is consciously and deliberately created and is easy to see. When the data set was created, decisions had to be made as to exactly which measurements were to be captured, measured in which ways, and stored in which formats. Point-of-sale (POS) data, for instance, captures information about a purchasing event at the point that the sale takes place. A vast wealth of possible information could be captured at this point, but capturing it all would swamp the system. Thus, POS information typically does not include any information about the weather, the length of the checkout line, local traffic information affecting access to the store, or the sort of bag the consumer chose for carrying away purchases. This kind of information may be useful and informative, but the structure created to capture data has no place to put it.

Macrostructure concerns the formatting of the variables. For example, granularity is a macro structural feature. *Granularity* refers to the amount of detail captured in any measurement—time to the nearest minute, the nearest hour, or simply differentiating morning, afternoon, and night, for instance. Decisions about macro structure have an important impact on the amount of information that a data set carries, which, in turn, has a very significant effect on the resolution of any model built using that data set. However, macro structure is not part of the scaffolding consciously erected to hold data, but is inherent in the nature of the measurements.

Microstructure, also referred to as *fine structure*, describes the ways in which the variables that have been captured relate to each other. It is this structure that modeling explores. A basic assessment of the state of the micro structure can form a useful part of the data audit (Stage 2 above). This brief examination is a simple assessment of the complexity of the variables' interrelationships. Lack of complexity does not prevent building successful predictive models. However, if complex and unexpected results are desired, additional data will probably be needed.

3.3.6 Stage 6: Building the PIE

The first five steps very largely require assessing and understanding the data that is available. Detailed scrutiny of the data does several things:

- It helps determine the possibility, or necessity, of adjusting or transforming the data.
- It establishes reasonable expectations of achieving a solution.
- It determines the general quality, or validity, of the data.
- It reveals the relevance of the data to the task at hand.

Many of these activities require the application of thought and insight rather than of automated tools. Of course, much of the assessment is supported by information gained by application of data preparation and other discovery tools, but the result is information that affects decisions about how to prepare and use the data.

By this stage, the data's limitations are known, at least insofar as they can be. Decisions have been made based on the information discovered. Fully automated techniques for preparing the data (such as those on the CD-ROM accompanying this book) can now be used.

The decisions made so far determine the sequence of operations. In a production environment, the data set may be in any machine-accessible form. For ease of discussion and explanation, it will be assumed that the data is in the form of a flat file. Also, for ease of illustration, each operation is discussed sequentially. In practice the techniques are not likely to be applied exactly as described. It is far easier to aggregate information that will be used by several subsequent stages during one pass through the file. This description is intended as thematic, to provide an overview and introduction to preparation activities.

Data Issue: Representative Samples

A perennial problem is determining how much data is needed for modeling. One tenet of data mining is "all of the data, all of the time." That is a fine principle, and if it can be achieved, a worthwhile objective. However, for various reasons it is not a practical solution. Even if as much data as possible is to be examined, survey and modeling still require at least three data sets—a training set, a test set, and an execution set. Each data set needs to be representative. Feature enhancement, discussed in Chapters 4 and 10, may require a concentration of instances exhibiting some particular feature. Such a concentration can only be made if a subset of data is extracted from the main data set. So there is always a need to decide how large a data set is required to be an accurate reflection of the data's fine structure.

In this case, when building the PIE, it is critical that it is representative of the fine structure. Every effort must be made to ensure that the PIE itself does not introduce bias! Without checking the whole population of instances, which may be an impossibility, there is no way to be 100% certain that any particular sample is, in fact, representative. However, it is possible to be some specified amount less than 100% certain, say, 99% or 95% certain. It is these certainty measures that allow samples to be taken. Selecting a suitable level of certainty is an arbitrary decision.

Data Issue: Categorical Values

Categoricals are "numerated," or assigned appropriate numbers. Even if, in the final prepared data, the categoricals are to be modeled as categorical values, they are still numerated for estimating missing values.

Chapter 2 contains an example showing that categoricals have a natural ordering that needs to be preserved. It is an ordering that actually exists in the world and is reflected in the categorical measurements. When building predictive or inferential models, it is critical that the natural order of the categorical values be preserved insofar as that is possible. Changing this natural ordering is imposing a structure. Even imposing a random structure loses information carried by the categorical measurement. If it is not random, the situation is worse because it introduces a pattern not present in the world.

The exact method of numeration depends on the structure of the data set. In a mixed numeric/categorical data set, the numeric values are used to reflect their order into the categoricals. This is by far the most successful method, as the numeric values have an order and magnitude spacing. In comprehensive data sets, this allows a fair recovery of the appropriate ordering. In fact, it is interesting to convert a variable that is actually numeric into a categorical value and see the correct ordering and separation recovered.

Data sets that consist entirely of categorical measurements are slightly more problematic. It is certainly possible to recover appropriate orderings of the categoricals. The problem is that without numeric variables in the data set, the recovered values are not anchored to real-world phenomena. The numeration is fine for modeling and has in practice produced useful models. It is, however, a dangerous practice to use the numerated orderings to infer anything absolute about the meaning of the magnitudes. The relationships of the variables, one to another, hold true, but are not anchored back to the real world in the way that numerical values are.

It is important to note that no automated method of recovering order is likely to be as accurate as that provided by domain knowledge. Any data set is but a pale reflection of the real world. A domain expert draws on a vastly broader range of knowledge of the world than can be captured in any data set. So, wherever possible, ordered categorical values should be placed in their appropriate ordering as ordinal values. However, as it is often the case when modeling data that there is no domain expert available, or that no ordinal ranking is apparent, the techniques used here have been effective.

Data Issue: Normalization

Several types of normalization are very useful when modeling. The normalization discussed throughout this book has nothing in common with the sort of normalization used in a database. Recall that the assumption for this discussion is that the data is present as a single table. Putting data into its various normal forms in a database requires use of multiple tables. The form of normalization discussed here requires changing the instance values in specific and clearly defined ways to expose information content within the data and the data set. Although only introduced here, the exact normalization methods are discussed in detail in Chapter 7.

Some tools, such as neural networks, require range normalization. Other tools do not *require* normalization, but do benefit from having normalized data. Once again, as with other issues, it is preferable for the miner to take control of the normalization process.