Chapter 4: Getting the Data—Basic Preparation

Overview

Data preparation requires two different types of activities: first, finding and assembling the data set, and second, manipulating the data to enhance its utility for mining. The first activity involves the miner in many procedural and administrative activities. The second requires appropriately applying automated tools. However, manipulating the data cannot begin until the data to be used is identified and assembled and its basic structure and features are understood. In this chapter we look at the process of finding and assembling the data and assessing the basic characteristics of the data set. This lays the groundwork for understanding how to best manipulate the data for mining.

What does this groundwork consist of? As the ancient Chinese proverb says: "A journey of a thousand miles begins with a single step." Basic data preparation requires three such steps: data discovery, data characterization, and data set assembly.

- *Data discovery* consists of discovering and actually locating the data to be used.

- *Data characterization* describes the data in ways useful to the miner and begins the process of understanding what is in the data—that is, is it reliable and suitable for the purpose?

- *Data set assembly* builds a standard representation for the incoming data so that it can be mined—taking data found to be reliable and suitable and, usually by building a table, preparing it for adjustment and actual mining.

These three stages produce the *data assay*. The first meaning of the word "assay" in the *Oxford English Dictionary* is "the trying in order to test the virtue, fitness, etc. (of a person or thing)." This is the exact intent of the data assay, to try (test or examine) the data to determine its fitness for mining. The assay produces detailed knowledge, and usually a report, of the quality, problems, shortcomings, and suitability of the data for mining. Although simple to state, assaying data is not always easy or straightforward. In practice it is frequently extremely time-consuming. In many real-world projects, this stage is the most difficult and time-consuming of the whole project. At other times, the basic preparation is relatively straightforward, quick, and easy.

As an example, imagine that First National Bank of Anywhere (FNBA) decides to run a credit card marketing campaign to solicit new customers. (This example is based on an actual mining project.) The marketing solicitations are made to "affinity groups," that is, groups of people that share some experience or interest, such as having attended a particular college or belonging to a particular country club. FNBA buys lists of names and addresses of such groups and decides to use data mining to build market segmentation and customer response models to optimize the return from the campaign. As the campaign progresses, the models will have to be updated to reflect changing market conditions and response. Various models of different types will be required, although the details have not yet been pinned down. Figure 4.1 shows an overview of the process.

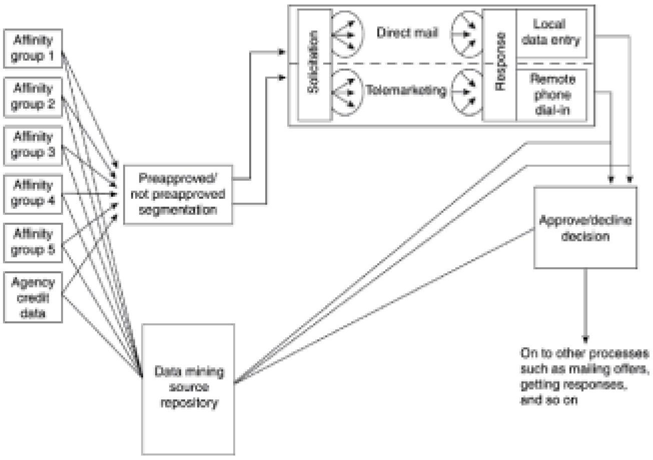


Figure 4.1 Simplified credit card direct-mail solicitation showing six different data feeds. Each data feed arrives from a different source, in a different format, at a different time and stage in the process.

4.1 Data Discovery

Current mining tools almost always require the data set to be assembled in the form of a "flat file," or table. This means that the data is represented entirely in the row and column format described in Chapter 2. Some mining tools represent that they query databases and data warehouses directly, but it is the end result of the query, an extracted table, that is usually mined. This is because data mining operations are column (variable) oriented. Databases and data warehouses are record (instance) oriented. Directly mining a warehouse or database places an unsupportable load on the warehouse query software. This is beginning to change, and some vendors are attempting to build in support for mining operations. These modifications to the underlying structural operation of accessing a data warehouse promise to make mining directly from a warehouse more practical at some future time. Even when this is done, the query load that any mining tool can levy on the warehouse will still present a considerable problem. For present practical purposes, the starting point for all current mining operations has to be regarded as a table, or flat file. "Discovering data" means that the miner needs to determine the original source from which the table will be built.

The search starts by identifying the data source. The originating data source may be a transaction processing system fed by an ATM machine or a POS terminal in a store. It may be some other record-capturing transaction or event. Whatever it is, a record is made of the original measurements. These are the founding “droplets” of data that start the process. From here on, each individual droplet of data adds to other droplets, and trickle adds to trickle until the data forms a stream that flows into a small pool—some sort of data repository. In the case of FNBA, the pools are moderately large when first encountered: they are the affinity group membership records.

The affinity group member information is likely stored in a variety of forms. The groups may well be almost unknown to each other. Some may have membership records stored on PCs, others on Macs. Some will provide their member lists on floppy disk, some on 8mm tape, some on 4mm tape, some on a Jazz drive, and others on 9-track tape. Naturally, the format for each, the field layout and nomenclature, will be equally unique. These are the initial sources of data in the FNBA project. This is not the point of data creation, but as far as the project is concerned it is the point of first contact with the raw data. The first need is to note the contact and source information. The FNBA assay starts literally with names, addresses, contact telephone numbers, media type, transmission mode, and data format for each source.

4.1.1 Data Access Issues

Before the data can be identified and assessed, however, the miner needs to answer two major questions: Is the data accessible? and How do I get it?

There are many reasons why data might not be readily accessible. In many organizations, particularly those without warehouses, data is often not well inventoried or controlled. This can lead to confusion about what data is actually available.

- *Legal issues*. There may well be legal barriers to accessing some data, or some parts of a data set. For example, in the FNBA project it is not legal to have credit information about identifiable people to whom credit is not actually going to be offered. (The law on this point is in constant change and the precise details of what is and is not legally permissible varies from time to time.) In other applications, such as healthcare, there may be some similar legal restriction or confidentiality requirement for any potential data stream.

- *Departmental access*. These restrictions are similar to legal barriers. Particularly in financial trading companies, data from one operation is held behind a “Chinese Wall” of privacy from another operation for ethical reasons. Medical and legal data are often restricted for ethical reasons.

- *Political reasons*. Data, and particularly its ownership, is often regarded as belonging to a particular department, maybe one that does not support the mining initiative for any number of reasons. The proposed data stream, while perhaps physically present, is not practically accessible. Or perhaps it is accessible, but not in a timely or complete fashion.

- *Data format*. For decades, data has been generated and collected in many formats. Even modern computer systems use many different ways of encoding and storing data. There are media format differences (9-track magnetic tape, diskettes, tape, etc.) and format differences (ASCII, EBCDIC, binary packed decimal, etc.) that can complicate assembling data from disparate sources.

- *Connectivity*. Accessing data requires that it be available online and connected to the system that will be used for mining. It is no use having the data available on a high-density 9-track tape if there is no suitable 9-track tape drive available on the mining system.

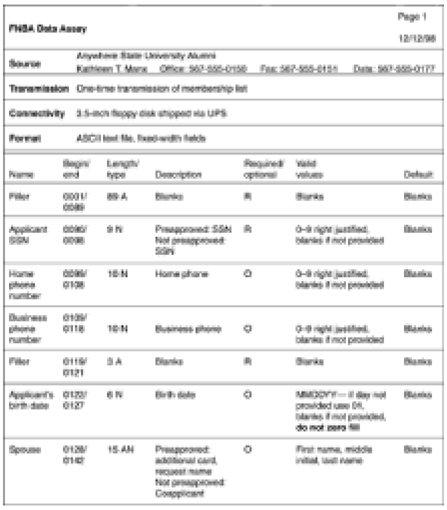
- *Architectural reasons*. If data is sourced from different database architectures, it may be extremely difficult, or unacceptably time-consuming, to translate the formats involved. Date and time information is notoriously difficult to work with. Some architectures simply have no equivalent data types to other architectures, and unifying the data representation can be a sizeable problem.

- *Timing*. The validating event (described in Chapter 2) may not happen at a comparable time for each stream. For example, merging psychographic data from one source with current credit information may not produce a useful data set. The credit information may be accurate as of 30 days ago, whereas the psychographic information is only current as of six months ago. So it is that the various data streams, possibly using different production mechanisms, may not be equally current. If a discrepancy is unavoidable, it needs to at least remain constant—that is, if psychographic information suddenly began to be current as of three months ago rather than six months ago, the relationships within the data set would change.

This is not a comprehensive listing of all possible data access issues. Circumstances differ in each mining application. However, the miner must always identify and note the details of the accessibility of each data stream, including any restrictions or caveats.

Data sources may be usefully characterized also as internal/external. This can be important if there is an actual dollar cost to acquiring outside data, or if internal data is regarded as a confidential asset of the business. It is particularly worth noting that there is *always* at least a time and effort cost to acquiring data for modeling. Identifying and controlling the costs, and getting the maximum economic benefit from each source, can be as important as any other part of a successful mining project.

FNBA has several primary data sources to define. For each source it is important to consider each of the access issues. Figure 4.2 shows part of the data assay documentation for one of the input streams.



FNBA Data Assay

Page 1

12/12/98

Source	Anywhere State University Alumni Kathleen T. Mans Office: 567-555-0150 Fax: 567-555-0151 Data: 567-555-0177
Transmission	One-time transmission of membership list
Connectivity	3.5-inch floppy disk shipped via UPS
Format	ASCII text file, fixed-width fields

Name	Begin/ end	Length/ type	Description	Required/ optional	Valid values	Default
Filler	0001/ 0089	89 A	Blanks	R	Blanks	Blanks
Applicant SSN	0090/ 0098	9 N	Preapproved: SSN Not preapproved: SSN	R	0–9 right justified, blanks if not provided	Blanks
Home phone number	0099/ 0108	10 N	Home phone	O	0–9 right justified, blanks if not provided	Blanks
Business phone number	0109/ 0118	10 N	Business phone	O	0–9 right justified, blanks if not provided	Blanks
Filler	0119/ 0121	3 A	Blanks	R	Blanks	Blanks
Applicant's birth date	0122/ 0127	6 N	Birth date	O	MMDDYY — if day not provided use 01, blanks if not provided, **do not zero fill**	Blanks
Spouse	0128/ 0142	15 AN	Preapproved: additional card, request name Not preapproved: Coapplicant	O	First name, middle initial, last name	Blanks

Figure 4.2 Part of the description of one of the input streams for FNBA.

4.2 Data Characterization

After finding the source for all of the possible data streams, the *nature* of the data streams has to be characterized, that is, the data that each stream can actually deliver. The miner already knows the data format; that is to say, the field names and lengths that comprise the records in the data. That was established when investigating data access. Now each variable needs to be characterized in a number of ways so that they can be assessed according to their usefulness for modeling.

Usually, summary information is available about a data set. This information helps the miner check that the received data actually appears as represented and matches the summary provided. Most of the remainder of characterization is a matter of looking at simple frequency distributions and cross-tabs. The purpose of characterization is to understand the nature of the data, and to avoid the “GI” piece of GIGO.

4.2.1 Detail/Aggregation Level (Granularity)

All variables fall somewhere along a spectrum from detailed (such as transaction records) to aggregated (such as summaries). As a general rule of thumb, detailed data is preferred to aggregated data for mining. But the level of aggregation is a continuum. Even detailed data may actually represent an aggregation. FNBA may be able to obtain outstanding loan balances from the credit information, but not the patterns of payment that led to those balances. Describing what a particular variable measures is important. For example, if a variable is discovered to be highly predictive, during the data modeling process the strategy for using the predictions will depend on the meaning of the variables involved.

The level of detail, or granularity, available in a data set determines the level of detail that is possible for the output. Usually, the level of detail in the input streams needs to be at least one level of aggregation more detailed than the required level of detail in the output. Knowing the granularity available in the data allows the miner to assess the level of inference or prediction that the data could potentially support. It is only potential support because there are many other factors that will influence the quality of a model, but granularity is particularly important as it sets a lower bound on what is possible.

For instance, the marketing manager at FNBA is interested, in part, in the weekly variance of predicted approvals to actual approvals. To support this level of detail, the input stream requires at least daily approval information. With daily approval rates available, the miner will also be able to build inferential models when the manager wants to discover the reason for the changing trends.

There are cases where the rule of thumb does not hold, such as predicting Stock Keeping Units (SKU) sales based on summaries from higher in the hierarchy chain. However, even when these exceptions do occur, the level of granularity still needs to be known.

4.2.2 Consistency

Inconsistent data can defeat any modeling technique until the inconsistency is discovered and corrected. A fundamental problem here is that different things may be represented by the same name in different systems, and the same thing may be represented by different names in different systems. One data assay for a major metropolitan utility revealed that almost 90% of the data volume was in fact duplicate. However, it was highly inconsistent and rationalization itself took a vast effort.

The perspective with which a system of variables (mentioned in Chapter 2) is built has a huge effect on what is intended by the labels attached to the data. Each system is built for a specific purpose, almost certainly different from the purposes of other systems. Variable content, however labeled, is defined by the purpose of the system of which it is a part. The clearest illustration of this type of inconsistency comes from considering the definition of an employee from the perspective of different systems. To a payroll system, an employee is anyone who receives a paycheck. The same company's personnel system regards an employee as anyone who has an employee number. However, are temporary staff, who have employee numbers for identification purposes, employees to the payroll system? Not if their paychecks come from an external temporary agency. So to ask the two systems "How many employees are there?" will produce two different, but potentially completely accurate answers.

Problems with data consistency also exist when data originates from a single application system. Take the experience of an insurance company in California that offers car insurance. A field identifying "auto_type" seems innocent enough, but it turns out that the labels entered into the system—"Merc," "Mercedes," "M-Benz," and "Mrcds," to mention only a few examples—all represent the same manufacturer.

4.2.3 Pollution

Data pollution can occur for a variety of reasons. One of the most common is when users attempt to stretch a system beyond its original intended functionality. In the FNBA data, for instance, the miner might find “B” in the “gender” field. The “B” doesn’t stand for “Boy,” however, but for “Business.” Originally, the system was built to support personal cards, but when corporately held credit cards were issued, there was no place to indicate that the responsible party was a genderless entity.

Pollution can came from other sources. Sometimes fields contain unidentifiable garbage. Perhaps during copying, the format was incorrectly specified and the content from one field was accidentally transposed into another. One such case involved a file specified as a comma-delimited file. Unfortunately, the addresses in the field “address” occasionally contained commas, and the data was imported into offset fields that differed from record to record. Since only a few of the addresses contained embedded commas, visual inspection of parts of many thousands of records revealed no problem. However, it was impossible to attain the totals expected. Tracking down the problem took considerable time and effort.

Human resistance is another source of data pollution. While data fields are often optimistically included to capture what could be very valuable information, they can be blank, incomplete, or just plain inaccurate. One automobile manufacturer had a very promising looking data set. All kinds of demographic information appeared to be captured such as family size, hobbies, and many others. Although this was information of great value to marketing, the dealer at the point of sale saw this data-gathering exercise as a hindrance to the sales process. Usually the sales people discovered some combination of entries that satisfied the system and allowed them to move ahead with the real business at hand. This was fine for the sales process, but did the data that they captured represent the customer base? Hardly.

4.2.4 Objects

Chapter 2 explained that the world can be seen as consisting of objects about which measurements are taken. Those measurements form the data that is being characterized, while the objects are a more or less subjective abstraction. The precise nature of the object being measured needs to be understood. For instance, “consumer spending” and “consumer buying patterns” seem to be very similar. But one may focus on the total dollar spending by consumers, the other on product types that consumers seek. The information captured may or may not be similar, but the miner needs to understand why the information was captured in the first place and for what specific purpose. This perspective may color the data, just as was described for employees above.

It is not necessary for the miner to build entity-relationship diagrams, or use one of the other data modeling methodologies now available. Just understand the data, get whatever insight is possible, and understand the purpose for collecting it.

4.2.5 Relationship

With multiple data input streams, defining the relationship between streams is important. This relationship is easily specified as a common key that defines the correct association between instances in the input streams, thus allowing them to be merged. Because of the problems with possible inconsistency and pollution, merging the streams is not necessarily as easy to do as it is to describe! Because keys may be missing, it is important to check that the summaries for the assembled data set reflect the expected summary statistics for each individual stream. This is really the only way to be sure that the data is assembled as required.

Note that the data streams cannot be regarded as tables because of the potentially huge differences in format, media, and so on. Nonetheless, anyone who knows SQL is familiar with many of the issues in discovering the correct relationships. For instance, what should be done when one stream has keys not found in the other stream? What about duplicate keys in one stream without corresponding duplicates in another—which gets merged with what? Most of the SQL "join"-type problems are present in establishing the relationship between streams—along with a few additional ones thrown in for good measure.

4.2.6 Domain

Each variable consists of a particular domain, or range of permissible values. Summary statistics and frequency counts will reveal any erroneous values outside of the domain. However, some variables only have valid values in some conditional domain. Medical and insurance data typically has many conditional domains in which the values in one field, say, "diagnosis," are conditioned by values in another field, say, "gender." That is to say, there are some diagnoses that are valid only for patients of one particular gender.

Business or procedural rules enforce other conditional domains. For example, fraud investigations may not be conducted for claims of less than $1000. A variable indicating that a fraud investigation was triggered should never be true for claims of less than $1000.

Perhaps the miner doesn't know that such business rules exist. There are automated tools that can examine data and extract business rules and exceptions by examining data. A demonstration version of one such tool, WizRule, is included on the CD-ROM with this book. Such a rule report can be very valuable in determining domain consistency. Example 2 later in this chapter shows the use of this tool.

4.2.7 Defaults

Many data capturing programs include default values for some of the variables. Such default values may or may not cause a problem for the miner, but it is necessary to be aware of the values if possible. A default value may also be conditional, depending on the values of other entries for the actual default entered. Such conditional defaults can create seemingly significant patterns for the miner to discover when, in fact, they simply represent a lack of data rather than a positive presence of data. The patterns may be meaningful for predictive or inferential models, but if generated from the default rules inside the data capture system, they will have to be carefully evaluated since such patterns are often of limited value.

4.2.8 Integrity

Checking integrity evaluates the relationships permitted between the variables. For instance, an employee may have several cars, but is unlikely to be permitted to have multiple employee numbers or multiple spouses. Each field needs to be evaluated to determine the bounds of its integrity and if they are breached.

Thinking of integrity in terms of an acceptable range of values leads to the consideration of outliers, that is, values potentially out of bounds. But outliers need to be treated carefully, particularly in insurance and financial data sets. Modeling insurance data, as an example, frequently involves dealing with what look like outliers, but are in fact perfectly valid values. In fact, the outlier might represent exactly what is most sought, representing a massive claim far from the value of the rest. Fraud too frequently looks like outlying data since the vast majority of transactions are not fraudulent. The relatively few fraudulent transactions may seem like sparsely occurring outlying values.

4.2.9 Concurrency

When merging separate data streams, it may well be that the time of data capture is different from stream to stream. While this is partly a data access issue and is discussed in “Data Access Issues” above, it also needs to be considered and documented when characterizing the data streams.

4.2.10 Duplicate or Redundant Variables

Redundant data can be easily merged from different streams or may be present in one stream. Redundancy occurs when essentially identical information is entered in multiple variables, such as “date_of_birth” and “age.” Another example is “price_per_unit,” “number_purchased,” and “total_price.” If the information is not actually identical, the worst damage is likely to be only that it takes a longer time to build the models. However, most modeling techniques are affected more by the number of variables than by the number of instances. Removing redundant variables, particularly if there are many of them, will increase modeling speed.

If, by accident, two variables should happen to carry identical values, some modeling techniques—specifically, regression-based methods—have extreme problems digesting such data. If they are not suitably protected, they may cause the algorithm to “crash.” Such colinearity can cause major problems for matrix-based methods (implemented by some neural network algorithms, for instance) as well as regression-based methods. On the other hand, if two variables are almost colinear, it is often useful to create a new variable that expresses the difference between the nearly colinear variables.

4.3 Data Set Assembly

At this point, the miner should know a considerable amount about the input streams and the data in them. Before the assay can continue, the data needs to be assembled into the table format of rows and columns that will be used for mining. This may be a simple task or a very considerable undertaking, depending on the content of the streams. One particular type of transformation that the miner often uses, and that can cause many challenges, is a *reverse pivot*.

4.3.1 Reverse Pivoting

Often, what needs to be modeled cannot be derived from the existing transaction data. If the transactions were credit card purchases, for example, the purchasing behavior of the cardholders may need to be modeled. The principal object that needs to be modeled, then, is the cardholder. Each transaction is associated with a particular account number unique to the cardholder. In order to describe the cardholder, all of the transactions for each particular cardholder have to be associated and translated into derived fields (or features) describing cardholder activity. The miner, perhaps advised by a domain expert, has to determine the appropriate derived fields that will contribute to building useful models.

Figure 4.3 shows an example of a reverse pivot. Suppose a bank wants to model customer activity using transaction records. Any customer banking activity is associated with an account number that is recorded in the transaction. In the figure, the individual transaction records, represented by the table on the left, are aggregated into their appropriate feature (Date, Account Number, etc.) in the constructed Customer Record. The Customer Record contains only one entry per customer. All of the transactions that a customer makes in a period are aggregated into that customer’s record. Transactions of different types, such as loan activity, checking activity, and ATM activity are represented. Each of the aggregations represents some selected level of detail. For instance, within ATM activity in a customer record, the activity is recorded by dollar volume and number of transactions within a period. This is represented by the expansion of one of the aggregation areas in the customer record. The “P*n*” represents a selected period, with “#” the number of transactions and “$” the dollar volume for the period. Such reverse pivots can aggregate activity into many hundreds of features.

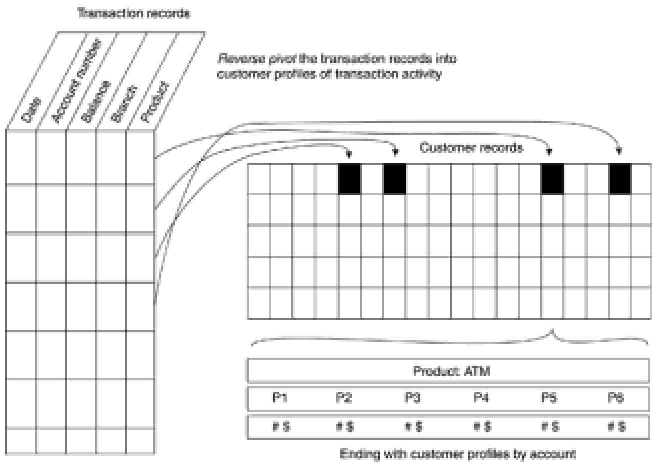

Transaction records
Date
Account number
Balance
Branch
Product
Reverse pivot the transaction records into customer profiles of transaction activity
Customer records
Product: ATM
P1
P2
P3
P4
P5
P6
# $
# $
# $
# $
# $
# $
Ending with customer profiles by account

Figure 4.3 Illustrating the effect of a reverse pivot operation.

One company had many point-of-sale (POS) transactions and wanted to discover the main factors driving catalog orders. The POS transactions recorded date and time, department, dollar amount, and tender type in addition to the account number. These transactions were reverse pivoted to describe customer activity. But what were the appropriate derived features? Did time of day matter? Weekends? Public holidays? If so, how were they best described? In fact, many derived features proved important, such as the time in days to or from particular public holidays (such as Christmas) or from local paydays, the order in which departments were visited, the frequency of visits, the frequency of visits to particular departments, and the total amount spent in particular departments. Other features, such as tender type, returns to particular departments, and total dollar returns, were insignificant.

4.3.2 Feature Extraction

Discussing reverse pivoting leads to the consideration of feature extraction. By choosing to extract particular features, the miner determines how the data is presented to the mining tool. Essentially, the miner must judge what features might be predictive. For this reason, reverse pivoting cannot become a fully automated feature of data preparation. Exactly which features from the multitudinous possibilities are likely to be of use is a judgment call based on circumstance. Once the miner decides which features are potentially useful, then it is possible to automate the process of aggregating their contents from the transaction records.

Feature extraction is not limited to the reverse pivot. Features derived from other combinations of variables may be used to replace the source variables and so reduce the dimensionality of the data set. Even if not used to reduce dimensionality, derived features can add information that speeds the modeling process and reduces susceptibility to noise. Chapter 2 discussed the use of feature extraction as a way of helping expose the information content in a data set.

Physical models frequently require feature extraction. The reason for this is that when physical processes are measured, it is likely that very little changes from one stage to the next. Imagine monitoring the weather measured at hourly intervals. Probably the barometric pressure, wind speed, and direction change little in an hour. Interestingly, when the changes are rapid, they signify changing weather patterns. The feature of interest then is the amount of change in the measurements happening from hour to hour, rather than the absolute level of the measurement alone.

4.3.3 Physical or Behavioral Data Sets

There is a marked difference in the character of a physical data set as opposed to a behavioral data set. *Physical data sets* measure mainly physical characteristics about the world: temperature, pressure, flow rate, rainfall, density, speed, hours run, and so on. Physical systems generally tend to produce data that can be easily characterized according to the range and distribution of measurements. While the interactions between the variables may be complex or nonlinear, they tend to be fairly consistent. *Behavioral data*, on the other hand, is very often inconsistent, frequently with missing or incomplete values. Often a very large sample of behavioral data is needed to ensure a representative sample.

Industrial automation typically produces physical data sets that measure physical processes. But there are many examples of modeling physical data sets for business reasons. Modeling a truck fleet to determine optimum maintenance periods and to predict maintenance requirements also uses a physical data set. The stock market, on the other hand, is a fine example of a behavioral data set. The market reflects the aggregate result of millions of individual decisions, each made from individual motivations for each buyer or seller. A response model for a marketing program or an inferential model for fraud would both be built using behavioral data sets.

4.3.4 Explanatory Structure

Devising useful features to extract requires domain knowledge. Inventing features that might be useful without some underlying idea of why such a feature, or set of features, might be useful is seldom of value. More than that, whenever data is collected and used for a mining project, the miner needs to have some underlying idea, rationale, or theory as to why that particular data set can address the problem area. This idea, rationale, or theory forms the explanatory structure for the data set. It explains how the variables are expected to relate to each other, and how the data set as a whole relates to the problem. It establishes a reason for why the selected data set is appropriate to use.

Such an explanatory structure should be checked against the data, or the data against the explanation, as a form of “sanity check.” The question to ask is, Does the data work in the way proposed? Or does this model make sense in the context of this data?

Checking that the explanatory structure actually holds as expected for the data available is the final stage in the assay process. Many tools can be used for this purpose. Some of the most useful are the wide array of powerful and flexible OLAP (On-Line Analytical Processing) tools that are now available. These make it very easy to interactively examine an assembled data set. While such tools do not build models, they have powerful data manipulation and visualization features.

4.3.5 Data Enhancement or Enrichment

Although the assay ends with validating the explanatory structure, it may turn out that the data set as assembled is not sufficient. FNBA, for instance, might decide that affinity group membership information is not enough to make credit-offering decisions. They could add credit histories to the original information. This additional information actually forms another data stream and enriches the original data. *Enrichment* is the process of adding external data to the data set.

Note that data enhancement is sometimes confused with enrichment. *Enhancement* means embellishing or expanding the existing data set without adding external sources. *Feature extraction* is one way of enhancing data. Another method is introducing bias for a particular purpose. Adding bias introduces a *perspective* to a data set; that is, the information in the data set is more readily perceived from a particular point of view or for a particular purpose. A data set with a perspective may or may not retain its value for other purposes. Bias, as used here, simply means that some effect has distorted the measurements.

Consider how FNBA could enhance the data by adding a perspective to the data set. It is likely that response to a random FNBA mailing would be about 3%, a typical response rate for an unsolicited mailing. Building a response model with this level of response would present a problem for some techniques such as a neural network. Looking at the response data from the perspective of responders would involve increasing the concentration from 3% to, say, 30%. This has to be done carefully to try to avoid introducing any bias other than the desired effect. (Chapter 10 discusses this in more detail.) Increasing the density of responders is an example of enhancing the data. No external data is added, but the existing data is restructured to be more useful in a particular situation.

Another form of data enhancement is *data multiplication*. When modeling events that rarely occur, it may not be possible to increase the density of the rate of occurrence of the event enough to build good models. For example, if modeling catastrophic failure of some physical process, say, a nuclear power plant, or indicators predicting terrorist attacks on commercial aircraft, there is very little data about such events. What data there is cannot be concentrated enough to build a representative training data set. In this case it is possible to multiply the few examples of the phenomena that are available by carefully adding constructed noise to them. (See Chapter 10.)

Proposed enhancement or enrichment strategies are often noted in the assay, although they do not form an integral part of it.

4.3.6 Sampling Bias

Undetected sampling bias can cause the best-laid plans, and the most carefully constructed and tested model, to founder on the rocks of reality. The key word here is “undetected.”

The goal of the U.S. census, for instance, is to produce an unbiased survey of the population by requiring that everyone in the U.S. be counted. No guessing, no estimation, no statistical sampling; just get out and count them. The main problem is that this is not possible. For one thing, the census cannot identify people who have no fixed address: they are hard to find and very easily slip through the census takers’ net. Whatever characteristics these people would contribute to U.S. demographic figures are simply missing. Suppose, simply for the sake of example, that each of these people has an extremely low income. If they were included in the census, the “average” income for the population would be lower than is actually captured.

Telephone opinion polls suffer from the same problem. They can only reach people who have telephones for a start. When reached, only those willing to answer the pollster’s questions actually do so. Are the opinions of people who own telephones different from those who do not? Are the opinions of those willing to give an opinion over the telephone different from those who are not? Who knows? If the answer to either question is “Yes,” then the opinions reflected in the survey do not in fact represent the population as a whole.

Is this bias important? It may be critical. If unknown bias exists, it is a more or less unjustified assumption that the data reflects the real world, and particularly that it has any bearing on the issue in question. Any model built on such assumptions reflects only the distorted data, and when applied to an undistorted world, the results are not likely to be as anticipated.

Sampling bias is in fact impossible to detect using only the data set itself as a reference. There are automated methods of deriving measurements about the data set indicating the possible presence of sampling bias, but such measurements are no more than indicators. These methods are discussed in Chapter 11, which deals with the data survey. The assay cannot use these automated techniques since the data survey requires a fully assembled and prepared data set. This does not exist when the assay is being made.

At this stage, using the explanatory structure for the data, along with whatever domain knowledge is available, the miner needs to discover and explicate any known bias or biases that affected the collection of the data. Biasing the data set is sometimes desirable, even necessary. It is critical to note intentional biases and to seek out other possible sources of bias.

4.4 Example 1: CREDIT

The purpose of the data assay, then, is to check that the data is coherent, sufficient, can be assembled into the needed format, and makes sense within a proposed framework. What does this look like in practice?

For FNBA, much of the data comes in the form of credit histories purchased from credit bureaus. During the solicitation campaign, FNBA contacts the targeted market by mail and telephone. The prospective credit card user either responds to the invitation to take a credit card or does not respond. One of the data input streams is (or includes) a flag indicating if the targeted person responded or not. Therefore, the initial model for the campaign is a predictive model that builds a profile of people who are most likely to respond. This allows the marketing efforts to be focused on only that segment of the population that is most likely to want the FNBA credit card with the offered terms and conditions.

4.4.1 Looking at the Variables

As a result of the campaign, various data streams are assembled into a table format for mining. (The file CREDIT that is used in this example is included on the accompanying CD-ROM. Table 4.1 shows entries for 41 fields. In practice, there will usually be far more data, in both number of fields and number of records, than are shown in this example. There is plenty of data here for a sample assay.)

TABLE 4.1 Status report for the CREDIT file.

FIELD	MAX	MIN	DISTINCT	EMPTY	CONF	REQ	VAR	LIN	VAR-TYPE
AGE _INFERR	57.0	35.0	3	0	0.96	280	0.8	0.9	N
BCBAL	24251.0	0.0	3803	211	0.95	1192	251.5	0.8	N
BCLIMIT	46435.0	0.0	2347	151	0.95	843	424.5	0.9	N

BCOPEN	0.0	0.0	1	59	0.95	59	0.0	0.0	E
BEACON_C	804.0	670.0	124	0	0.95	545	1.6	1.0	N
BUYER	1.0	0.0	2	0	0.95	353	0.1	0.7	N
CHILDREN	1.0	0.0	2	0	0.95	515	0.0	0.8	N
CRITERIA	1.0	1.0	1	0	0.95	60	0.0	0.0	N
DAS_C	513.0	–202.0	604	0	0.95	437	10.3	1.0	N
DOB_MONTH	12.0	0.0	14	8912	0.95	9697	0.3	0.6	N
DOB_YEAR	70.0	0.0	42	285	0.95	879	0.5	1.0	N
EQBAL	67950.0	0.0	80	73	0.95	75	0.0	1.0	E
EQCURBAL	220000.0	0.0	179	66	0.95	67	0.0	0.0	E
EQHIGHBAL	237000.0	0.0	178	66	0.95	67	0.0	0.0	E
EQLIMIT	67950.0	0.0	45	73	0.95	75	0.0	1.0	E
EST_INC_C	87500.0	43000.0	3	0	0.95	262	1514.0	0.9	N
HOME_ED	160.0	0.0	8	0	0.95	853	3.5	0.7	N
HOME_INC	150.0	0.0	91	0	0.95	1298	0.7	0.9	N
HOME_VALUE	531.0	0.0	191	0	0.95	870	2.6	0.9	N
ICURBAL	126424.0	0.0	4322	1075	0.96	2263	397.4	0.9	N
IHIGHBAL	116545.0	0.0	4184	573	0.96	1192	951.3	0.9	N

LST_R_OPEN	99.0	0.0	100	9	0.96	482	3.6	0.9	N
MARRIED	0.0	0.0	2	0	0.95	258	0.2	0.0	C
MOF	976.0	0.0	528	0	0.95	951	3.8	0.9	N
MTCURBAL	578000.0	0.0	3973	433	0.95	919	3801.7	1.0	N
MTHIGHBAL	579000.0	0.0	1742	365	0.95	779	4019.7	0.9	N
OWN_HOME	0.0	0.0	1	0	0.95	60	0.0	0.0	N
PRCNT_PROF	86.0	0.0	66	0	0.95	579	0.8	1.0	N
PRCNT_WHIT	99.0	0.0	58	0	0.95	568	3.3	0.6	N
RBAL	78928.0	0.0	5066	18	0.97	795	600.3	0.8	N
RBALNO	14.0	0.0	14	0	0.95	642	0.1	0.9	N
RBAL_LIMIT	9.0	0.0	10	0	0.95	618	0.1	0.8	N
RLIMIT	113800.0	0.0	6067	11	0.95	553	796.3	0.9	N
ROPEN	17.0	0.0	17	0	0.96	908	0.1	0.9	N
SEX	0.0	0.0	3	0	0.95	351	0.2	0.0	C
TBALNO	370260.0	0.0	7375	9	0.95	852	2383.7	0.7	N
TOPEN	17.0	0.0	18	0	0.95	617	0.1	0.9	N
UNSECBAL	23917.0	0.0	2275	781	0.95	1349	420.1	0.8	N
UNSECLIMIT	39395.0	0.0	1596	906	0.95	1571	387.9	0.9	N
YEARS_RES	15.0	0.0	17	21	0.95	431	0.4	0.9	N

_Q_MVP	0.0	0.0	207	0	0.95	1086	0.4	0.1	C

How is this data assayed? Start looking at the basic statistics for the file. Table 4.1 shows a statistics file produced by the data preparation software on the accompanying CD-ROM for the file CREDIT. How does this file help?

First, the column headings indicate the following measurements about the variables:

- FIELD. The name of the field.

- MAX. The maximum value sampled for numeric variables.

- MIN. The minimum value sampled for numeric variables.

- DISTINCT. The number of distinct values for the variable in the sample. For example, if the field "months" was recorded with standardized three-letter abbreviations, there are a maximum of 12 valid, distinct values that the field can contain. A missing value counts toward the total number of distinct values, so the field "months" can have 13 distinct values including the value "missing." More than 13 values clearly indicates that erroneous entries are polluting the data.

- EMPTY. The number of records with missing values.

- CONF. The confidence level that the variability was captured. (Confidence levels, and how they are discovered and used, are covered in Chapter 5 and are not used in the assay.)

- REQ. The minimum number of records required to establish the confidence level. (See Chapter 5.)

- VAR. A measure of the variability in a variable. (See Chapter 5.)

- LIN. A measure of interstitial linearity (again, discussed in Chapter 5) and used in the assay. Interstitial linearity is one measure used to indicate possible problems with a variable, including monotonicity.

- VARTYPE. The type of variable detected. "N" indicates numeric, "C" indicates character, "E" indicates empty. (The demonstration code will only recognize these three types.)

Now, consider what can be learned about a few of the fields:

- AGE_INFERR. This has three discrete values, and every field has one of the three values. This is a numeric variable.

- BCOPEN. This is a completely empty variable; that is, none of the records has an entry in this field. Thus it has one distinct value (missing) in all of the records.

- BEACON_C. As a rule of thumb, if the linearity of a variable (LIN) is above 0.98, it is worth checking if the variable is monotonic. (As it happens it isn't in this case, but knowing that requires domain knowledge.)

- CRITERIA. This is shown as a numeric variable having one DISTINCT value and no variance (indicated by the 0.0 entry in VAR). This means that while all of the values are populated, they all have the same value. So this is actually a constant, not a variable, and it should be removed.

- EQBAL. What is going on here? It is shown as empty ("E" in VARTYPE) and yet it contains 80 DISTINCT values! This is a feature of the sampling process. As shown in REQ, it required 75 samples to establish the confidence level needed. Out of those 75 sampled, 73 were EMPTY, which was sufficient to establish the required level of confidence that it was indeed empty below the required threshold. From that point on, the variable was no longer sampled. This speeds the sampling process. Other variables required far more samples to establish their required confidence level. At the end of the sampling process, the data preparation software builds a fully populated sample file with prepared data. When the full sample was taken, the full range of what was found in EQBAL was noted. The 80 in DISTINCT indicates that although the variable was populated at too low a level for use at the required confidence level, it still did have some very sparse content and that sparse content did have 80 distinct values. However, since it was too empty to use, it is not included in the prepared data.

- DOB_MONTH. This variable sits almost on the edge of falling below the selected sparsity threshold. It is not quite 95% empty, the level required for rejection in this example, but it is 92% (8912/9697) empty. Because of the emptiness and distortion, the system required 785 (9697 – 8912) nonempty samples to capture its variability. Even if the miner elected to use this field in the final model, there is still the question of why there are 14 months. To discover what is possibly wrong here, another report produced by the demonstration software is needed, the "Complete Content" report. This is a very large report listing, among other things, all of the values discovered in the sample along with their frequencies. Table 4.2 shows the part of the Complete Content report that covers DOB_MONTH, the part of the interest here. From inspection of the CONTENT it seems obvious that "00" serves as a surrogate for a missing value. Adding the 646 "00" with the 8912 that are missing, this takes the variable below the sparsity threshold selected and the variable should be discarded.

TABLE 4.2 Part of the Complete Content report for the CREDIT data.

FIELD	CONTENT	CCOUNT
DOB_MONTH		8912
DOB_MONTH	00	646
DOB_MONTH	01	12
DOB_MONTH	02	7
DOB_MONTH	03	10
DOB_MONTH	04	9
DOB_MONTH	05	15
DOB_MONTH	06	14
DOB_MONTH	07	11
DOB_MONTH	08	10
DOB_MONTH	09	13
DOB_MONTH	10	10
DOB_MONTH	11	15
DOB_MONTH	12	13

- HOME_VALUE. There are no empty values. Nonetheless, it does not seem likely that 0.0, shown in MIN as the minimum value, is a reasonable home valuation! There are 191 DISTINCT values, but how many are “0.0”? The appropriate part of the Complete Content report (Table 4.3) again shows what is happening. Once again it may seem obvious that the value 000 is a surrogate of a missing value. It may be beneficial to replace the 000 with a blank so that the system will treat it as a missing value rather than treating it as if it had a valid value of 000. On the other hand, it may be that a renter, not owning a home, is shown as having a 000 home value. In that case, the value acts as a “rent/own” flag, having a completely different meaning and perhaps a different significance. Only domain knowledge can really answer this question.

TABLE 4.3 Part of the Complete Content report showing the first few values of HOME_VALUE.

FIELD	CONTENT	CCOUNT
HOME_VALUE	000	284
HOME_VALUE	027	3
HOME_VALUE	028	3
HOME_VALUE	029	3
HOME_VALUE	030	3
HOME_VALUE	031	2
HOME_VALUE	032	5

4.4.2 Relationships between Variables

Each field, or variable, raises various questions similar to those just discussed. Is this range of values reasonable? Is the distribution of those values reasonable? Should the variable be kept or removed? Just the basic report of frequencies can point to a number of questions, some of which can only be answered by understanding the domain. Similarly, the relationship between variables also needs to be considered.

In every data mining application, the data set used for mining should have some underlying rationale for its use. Each of the variables used should have some expected relationship with other variables. These expected relationships need to be confirmed during the assay. Before building predictive or inferential models, the miner needs at least some assurance that the data represents an expected reflection of the real world. An excellent tool to use for this exploration and confirmation is a single-variable CHAID analysis. Any of the plethora of OLAP tools may also provide the needed confirmation or