25

Mining Customer Relationship Management (CRM) Data

Robert Cooley
KXEN Inc.

INTRODUCTION

Retail enterprises collect data about their customers and their interactions with customers from a number of different sources. These data are a natural target for data mining, and indeed one of the first applications of "modern" data mining was *market-basket analysis* (Agrawal

& Srikant, 1994)—discovering which products are typically purchased together. The data comes from a variety of sources and in a myriad of formats, but ultimately it can be broken down to two basic types—*static* data and *event* data. Static data are attributes about customers such as demographics, financial information, or interest profiles. Events are interactions with customers, such as e-mails, phone calls, transactions, or Web clicks. When it comes to data mining these sources, the goal is to optimize the ratio between profitable and unprofitable events for a given set of customers. Certain events, such as product purchases, generate revenue and hopefully profit, whereas others do not, such as customer service phone calls. Ideally, data mining and other forms of analysis show a company how to increase revenues and/or decrease expenses. The most straightforward example of this is response modeling for a direct mail campaign. By predicting who will respond to a campaign, a company can decrease the cost of a campaign while hopefully maintaining a similar overall response. Even a task such as maximizing customer satisfaction can be viewed in terms of event ratios. Satisfied customers are less likely to generate unprofitable events such as complaints to a call center and more likely to reward the company with future profitable events. This chapter explores how data mining of various customer data sources can be used to maximize the profitable event ratio, with a particular emphasis on E-commerce (click-stream) data.

Although the rise and fall of E-commerce has been faster and larger than other new technologies, it has essentially followed the classic hype curve, in which the initial expectations far exceed what the technology can deliver, followed by a precipitous drop when the prevailing opinion is pessimistic, and finally a convergence of reality and hype into realistic expectations. The promise of E-commerce quickly outstripped reality, and for a period in the late 1990s "traditional" sales channels such as physical stores or printed catalogs were deemed to be obsolete. On the tail end of the Internet bubble, many consider E-commerce to have been a fad, improperly equating all of E-commerce with a few of the more visible commercial failures. Although the business model of many of the so-called dot-coms was indeed faulty, the value of E-commerce as a customer touchpoint is indisputable. More important, from a data mining perspective events collected from E-commerce sites are unique with respect to the types of customer behaviors that are recorded. *Channels* or *touchpoints* such as retail stores and catalog sales only record purchasing behavior–what is actually bought. E-commerce allows an organization to record a much broader range of behaviors, such as browsing or selecting items for purchase, regardless of whether the items are eventually bought. No other touchpoint provides such a complete record of both positive and negative customer behavior. Does this make E-commerce data inherently superior to data collected from other touchpoints? Not necessarily, because the additional information comes at a price. A common misperception is that data collected from an E-commerce site is "free," and that any data mining enterprise automatically will be more cost-effective than an equivalent undertaking with traditional data sources. In reality, due to the sheer volume of data and the level of noise, E-commerce mining can be far more costly and complex. Whether the results justify the increased costs depends completely on the context of the business questions being asked and how the results are applied.

Web usage mining is the general application of data mining techniques to Web clickstream data to extract usage patterns (Cooley, 2000). E-commerce mining is a subset of Web usage mining with the goal of maximizing the profit from customer interactions. For enterprises with several customer touchpoints, E-commerce mining can be just one facet of an overall customer relationship management (CRM) mining system. Figure 25.1 shows the high-level steps involved in knowledge discovery for CRM mining. The raw data is gathered from a number of sources and imported into a CRM data warehouse through a set of extraction, translation, and loading (ETL) processes. *Data preparation* pulls data from the warehouse and gets it into a format suitable for data mining. Then *pattern discovery* uses data mining

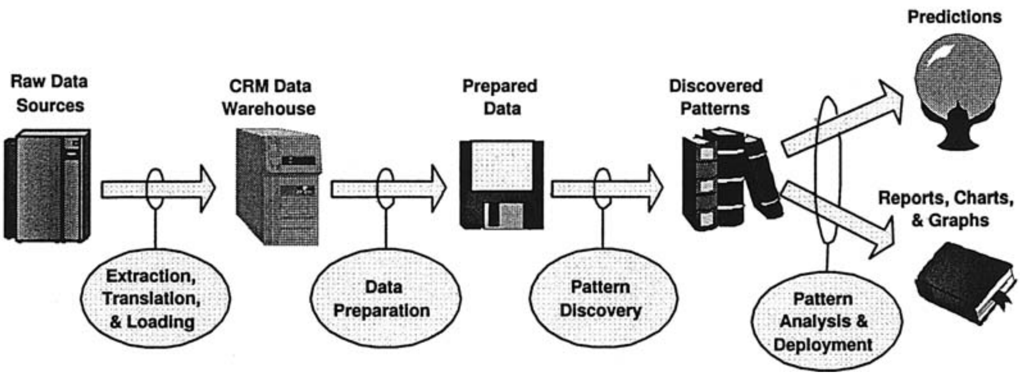

FIG. 25.1. CRM data mining process.

or machine learning algorithms to build models or discover relationships in the data, and *pattern analysis* transforms the patterns into usable knowledge. Finally, patterns that have been deemed to be actionable are deployed as models in an operational system. It is this final phase of deployment that directly affects the event ratio and creates a return on investment.

The rest of this chapter is organized as follows. The second section introduces the potential sources of data, the third section presents the major data preparation tasks that are necessary for CRM mining, the fourth section briefly discusses various pattern discovery techniques that can be used to discover knowledge, the fifth section explores methods for analyzing the discovered patterns, and the sixth section gives some CRM motivated examples. The final section provides a summary.

DATA SOURCES

The launch point for CRM mining is a data warehouse. A CRM data warehouse is populated from a number of sources, including point-of-sale (POS) databases, Web sites, call centers, e-mail responses, product databases, customer databases, and third-party databases. Obviously, the ETL effort to combine all of these data sources into a single clean and consistent schema is immense. For the purposes of this chapter, it will be assumed that these ETL challenges have been overcome, and the task at hand is now purely one of data mining. The rest of this section discusses the types of data that are typically encountered in a CRM data warehouse. The one exception is the discussion of E-commerce data, which starts from the raw logs themselves. The reason for this is that there are specific challenges involved with the ETL process for E-commerce data that often are overlooked.

Data Types

The two basic types of data associated with CRM mining have already been introduced—static and event data. In a data warehousing context these types often are referred to as fact and dimension tables, respectively. The division between these data types is not always a clean one, and a given attribute may act as static data in one context and event data in another. For example, an organization may choose to keep track of its customers' stated preference for a customer service channel. Although a given customer can have only one preferred channel, this channel may change over time, and a preference history can be recorded. When building

TABLE 25.1
Common CRM Data Sources

Static	*Event*
Demographic	Purchase
Financial	Web click
Profile	Phone call
	E-mail
	Store visit
	Physical mailing

a campaign response model, the current preference may be used as a static input to the data mining process, but for predicting customer churn a history of preference change events might be warranted. Table 25.1 lists some of the more common examples of static and event data found in a CRM data mining setting.

An important distinction between the two types of data is that static data describes a customer, and event data describes a customer's behavior. Ultimately, it is the behavior that is of interest, because, as stated earlier, the goal of CRM mining is to modify the ratio of positive and negative behaviors. Most static information, such as age or postal code, are merely surrogates for the actual information of importance—customer behavior. In studies that compare behavior with static information (Ehrenberg, 1988), past customer behavior was consistently shown to be superior for predicting future behavior. Why, then, is static information such as demographics often seen as the data of most value? One of the main reasons is the relative volume of data that is necessary to perform reliable data mining. A single attribute such as postal code can be a reliable predictor of many different customer behaviors. The number of events necessary to get the same prediction rate is often orders of magnitude higher. However, as more events are added to a data set, prediction error will generally decrease to the point at which it is better than with the static data. Also, for many types of analysis, event data must be aggregated before it can be used. The common types of event aggregation used for CRM mining will be addressed in a later section and a full discussion can be found in chapter 14. The use of event data, therefore, is more expensive due to the quantity and preprocessing requirements and also results in a higher-dimensional problem that must be solved. Finally, event data is not necessarily compatible across different domains. If a company wants to run a television advertising campaign, a description of its potential customers based on their buying behaviors is not useful for choosing which stations should run the ad. Different television stations and programs typically describe audiences in terms of demographic attributes. The company would need a demographic description of its customers to determine the right advertising strategy. Based on concerns about resources, scalability, and high dimensionality, static data has become the "default" for many CRM mining systems. However, as hardware costs decrease, processing power grows, and new methods for handling high-dimensional problems are created, the value of mining patterns based on true behaviors is increasing to the point at which it is preferable to static data for many situations.

To illustrate many of the points made in this chapter, a running example of a retail enterprise is used. This example enterprise, "MyWidgets, Inc.," has a physical retail store, Web site (www.mywidgets.com), and a call center that handles both sales and support. Figure 25.2 show the data warehouse schema for MyWidgets. The notation for the relationships between tables uses combinations of lines, circles, and "crows feet" to indicate cardinality. A line and circle indicate zero or one rows, a double line indicates one row, a crow foot and circle indicate zero to many rows, and a crow foot and line indicate one to many rows in the relationship. For example, every customer has one or more purchases (otherwise he or she would not be

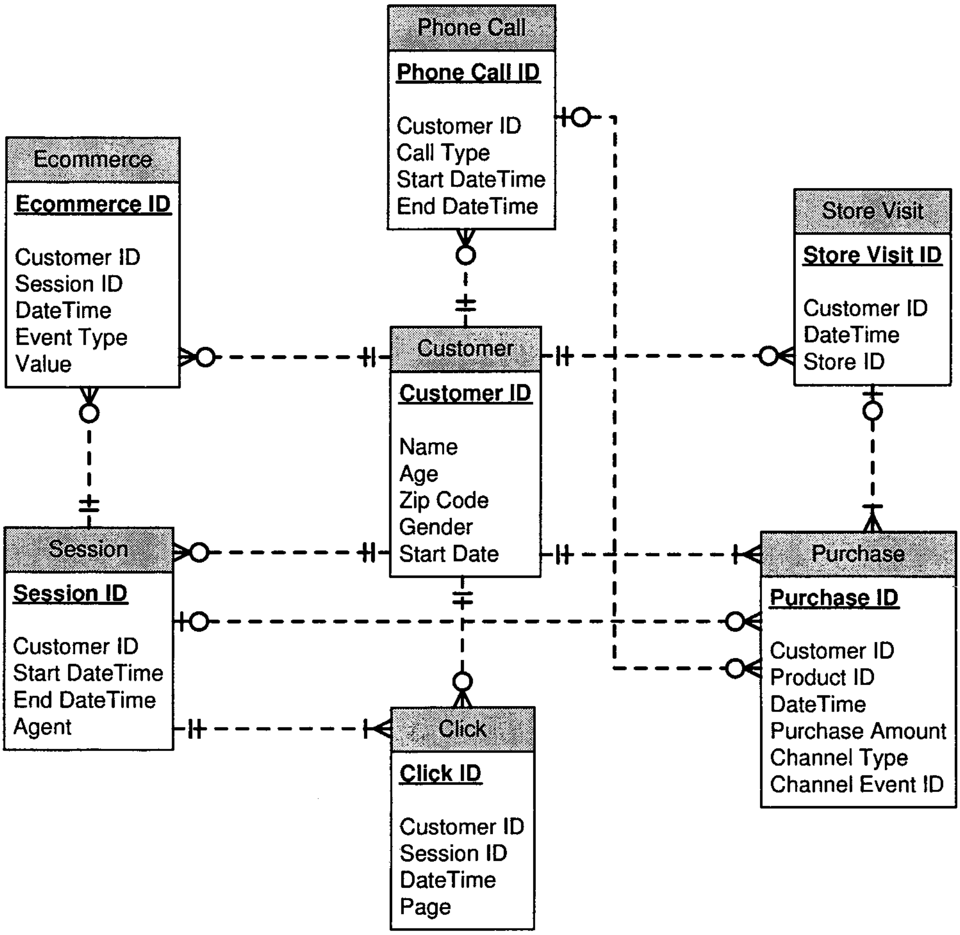

FIG. 25.2. MyWidgets data warehouse schema.

a customer), but a purchase must be associated with one and only one customer. The central table is the "Customer" table and there are six associated event tables. Each customer visit to the MyWidgets.com Web site is captured in the"Session," "E-commerce Event," and "Click" tables. Any calls to the phone center, either for customer service or sales, are captured in the "Phone Call" table. Purchases from the Web site, phone center, or a physical store are captured in the "Purchase" table. The "Store Visit" table has a record of every visit to a physical store that resulted in a sale. Again, unlike the Web site and call center, there is no automated method of tracking visits to a store that do not result in a purchase.

E-Commerce Data

The data collected from E-commerce systems is predominantly clickstream, or Web usage data. A clickstream is defined as a sequence of page views that are accessed by a user. A *page view* consists of all of the files that contribute to the browser presentation seen as the result of a single mouse "click" or action of a user. In addition to clickstream data, a Web site generates Web content and Web structure data. Web content data is the actual information that is contained in the "pages" that make up a Web site. The content data is usually made up of text and graphics, but can also include multimedia content. Web structure data comes from the hypertext links between the various pieces of content. As it turns out, both Web content and Web structure data are critical for E-commerce ETL, as discussed later. Although profile information about customers can be collected through a Web site, the ETL requirements for this type of data are not fundamentally different from the same data collected through a different channel such as a call center.

E-Commerce Events. Although a clickstream gives an accounting of each action taken by potential customers on a Web site, this level of events is not necessarily the most useful for CRM mining. Tracking Web site usage in terms of interactions with *products* can be far more beneficial. These types of events are sometimes referred to as *E-commerce events* to distinguish them from clickstream events. There are two important differences between E-commerce and clickstream events:

- **Cardinality.** There is not necessarily a one-to-one relationship between clickstream and E-commerce events. One clickstream event may be responsible for any number of E-commerce events. A single click can lead to several products being displayed on a page, resulting in more than one "product-view" event. By the same token, a click can lead to a page without any product information, yielding no E-commerce events.
- **Standardized definitions.** The definition of a clickstream event and the associated terminology is fairly standardized. A single user action results in a single event. However, the definition of an E-commerce event is set by each individual enterprise. What it means to view a product and whether it is an event that should be tracked at all is something that must be decided by the organization responsible for a given E-commerce Web site.

Examples of common E-commerce events are *product views, product click-throughs, product adds,* and *product purchases*. A product view is typically considered to be an image or text description displayed on a Web page. A user may or may not have explicitly requested a product view. For instance, products displayed on a home page or in a side frame are usually chosen by the application server, not the user. As will be seen in a later section, personalizing the displays of these "nonrequested" products to increase up-sells and cross-sells is a natural target for CRM mining. A product click-through is an explicit request for more information. An image of a product or text link often leads to a Web page with detailed information about a product. A product click-through can be seen as a sign of interest in a particular product by a customer. A product add is an indication of an intent to purchase by placing a product into a virtual shopping basket. If an E-commerce site is analogous to a physical store, product views are all of the products visible in the aisles a customer walks down, and click-throughs are any product a customer picks up but does not necessarily put in a shopping cart. In addition to the E-commerce events mentioned above, a Web site may define any number of other events, such as bids for auction sites or orders for brokerage sites. There also can be multiple levels of each type of event. For example, a Web site may have a page that gives a quick summary of a product and a second page with the full product details, leading to two types of product click-through events. As another example, purchasing a product may require three separate "checkout" pages. Ideally, a CRM data warehouse will contain both E-commerce and clickstream events, but if forced to choose between the two, E-commerce events are far more valuable for CRM mining than a full clickstream.

E-Commerce ETL. Figure 25.3 depicts a simplified view of the Internet as it pertains to the World Wide Web. As shown, clickstream data can come from a client-level log, proxy-level log, or server-level log. The most common source for clickstream data are Web

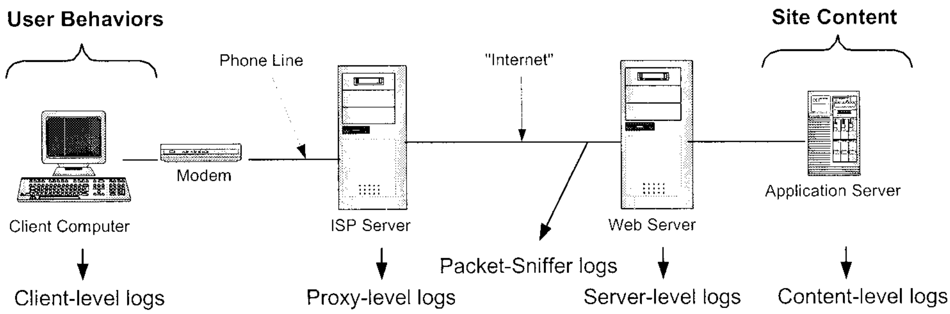

FIG. 25.3. Simplified World Wide Web.

server logs, which are collected at the server level. However, Web server logs were originally created to assist Web site administrators in debugging a site, and typically contain a line for every HyperText Transfer Protocol (HTTP) request that comes to a site. This means there is a line for every graphic, HyperText Markup Language (HTML) page, Common Gateway Interface (CGI) request, and so forth. When the Web started to gain popularity in the mid 1990s Web sites were relatively simple, and a server log was a reasonable picture of user behavior. Tools were created in both the research and commercial environments to analyze or mine these logs for usage patterns (Buchner & Mulvenna, 1998; Cooley, Tan, & Srivastava, 1999; Spiliopoulou & Faulstich, 1998; Wu, Yu, & Ballman, 1998). However, as the sites grew in complexity, and *application servers* were created to create *dynamic content* on the fly, the mapping between Web server logs and true customer behaviors became more and more tenuous. User behaviors occur at the browser or *client level,* and the site content is created by the application server. A Web server is no longer the end of the pipeline and is often unaware of important information maintained by the application server. An application server can maintain information about users, sessions, and the content being requested that is invisible to the Web server. For example, if an application server chooses to use the same file name for every dynamic page, from the point of view of a Web server the same content is being served over and over. Although information contained in CGI variables and HTTP header information can be used to disambiguate some of these problems, methods such as those described in (Cooley, 2000) are not always reliable.

In essence, most E-commerce log analysis systems are currently solving a much harder problem than necessary. An application server is able to accurately describe all events for which it is responsible. A clean and consistent event log can be recorded by an application server that is far superior to the information from a Web server log. More important, a true E-commerce event log can be created only by an application server. The information available to other logging methods is not guaranteed to be sufficient for E-commerce events. However, one drawback of collecting data from an application server is a potentially incomplete record of events. Browsers and proxy servers tend to keep a cache of files that have been requested recently. This means that pages can be viewed without recording an event in the application or Web server log. The caching of pages has become less of a problem with dynamic content, but an E-commerce ETL process must still take the possibility of an incomplete log into account. The major advantages and disadvantages of the various E-commerce data collection options are listed in Table 25.2.

Click-Stream Preprocessing. The practical difficulties in performing preprocessing are a moving target. As the technology used to deliver content over the Web changes, so do the preprocessing challenges. Although each of the basic preprocessing steps remains

TABLE 25.2
Web Data Collection Options

Collection Point	*Advantages*	*Disadvantages*
Client	Complete record of events Potentially multisite	Privacy concerns Can be disabled Not automatically created Can't track E-commerce events
Proxy	Multiuser & multisite	Large data volume Potentially incomplete Can't track E-commerce events
Packet-Sniffer	Can process HTTP headers	Large data volume Potentially incomplete Can't track E-commerce events
Web server	Widely available	Potentially incomplete Can't track E-commerce events
Application server	Can track E-commerce events Access to content information	Not automatically created Potentially incomplete

constant, the difficulty in completing certain steps has changed dramatically as Web sites have moved from static HTML served directly by a Web server to dynamic scripts created from sophisticated application servers and personalization tools. Both client-side tools (e.g., browsers) and server-side tools (e.g., application servers) have undergone several generations of improvements since the inception of the Web. *Data cleaning* is a site-specific step that involves mundane tasks such as merging logs from multiple servers and parsing the log into data fields. Typically, graphics file requests are stripped out at this stage. Next, user and *session identification* is performed through one of several methods, the most common being the use of cookies, user registration, or embedded session identifications in the uniform resource locators (URLs). *Page view identification* determines which page file requests are part of the same page view and what content was served. When frames are used, a single page view can consist of several HTML files. This step is highly dependent on knowledge of the Web site structure and content. Finally, *path completion* fills in page references that are missing due to local browser caching. This step differs from the others in that information is being added to the log. Each of these tasks is performed to create a server session file that is stored in the CRM data warehouse. In the case of MyWidgets, meta data for the Web sessions is stored in the "Session" table, as well as the clickstream in the "Click" table. Note that the server session file is usually only an estimate of what actually occurred due to techniques that obscure the data collection such as proxy servers and caching that are common in today's browsing environment. An overview of heuristics and algorithms that can be used to handle these problems are contained in (Cooley, Mobasher, & Srivastava, 1999). A detailed discussion on setting up and populating a clickstream data warehouse can be found in (Kimball & Merz, 2000).

DATA PREPARATION

By this point, all of the appropriate static and event data has been collected, extracted, translated, and loaded into a clean and consistent RDBMS schema. However, pattern discovery cannot be performed directly on the data in the warehouse. Different pattern discovery methods and algorithms require input data to be in different formats. Although there is no universally accepted term for getting data ready for input into a specific data mining algorithm, this chapter refers to this process as *data preparation,* to distinquish it from the preprocessing tasks associated with ETL. Many algorithms need all of the information associated with a customer to be in a single "line" of data. Other algorithms need all of the strings converted to numbers or all of the numbers within a certain range.

Data Aggregation

Data aggregation is normally something that is associated with ETL, reporting, or online analytical processing (OLAP). In fact, data mining is often discussed in terms of working with *unaggregated* data. Although it is true that the static information is rarely aggregated, it is not uncommon to aggregrate events. Methods such as classification, regression, and clustering typically require a single line or a fixed number of lines of data for each training example. There is usually more than one event associated with each customer for CRM mining. In the example schema, a customer can have several purchase events, E-commerce events, click events, session events, store events, or phone events. There are two major dimensions for aggregating CRM events—*time* and *value*.

An example of time-based event aggregation is shown in Fig. 25.4. In this case both the count and amount of purchases are summed up for three quarters, starting in Q3 of 1999. This type of aggregation can easily be performed and stored in the data warehouse, similar to what is done for some OLAP data cubes. However, there is a different form of time-based aggregation that is potentially more useful but difficult to precompute. This is what is referred to as *relative aggregation.* Aggregation based on a fixed set of dates, such as quarters in a calendar year, is not necessarily meaningful with respect to customers. New customers are added at different times and have different event histories. This is seen in Fig. 25.4, where none of the purchases for the two customers overlap. A relative aggregation works off of a variable date, such as the date a customer first purchased a product, or the date an offer went out by e-mail.

Customer ID	Product	Date	Total
1001	Widget A	1/1/00	$95.79
1001	Widget B	1/5/00	$48.87
1001	Widget C	1/5/00	$503.49
1002	Widget A	7/1/99	$95.79
1002	Widget D	10/5/99	$219.98
1002	Widget A	12/3/99	$95.79
----------	-------------	---------	----------
----------	-------------	---------	----------

Customer ID	Zip Code	Gender	Start Date
1001	94116	Male	1/1/00
1002	94103	Female	7/1/99
-----------	--------	--------	----

Cust. ID	Zip Code	Gender	# Events Q3 1999	$ Total Q3 1999	# Events Q4 1999	$ Total Q4 1999	# Events Q1 2000	$ Total Q1 2000
1001	94116	Male	0	$0.00	0	$0.00	3	$648.15
1002	94103	Female	1	$95.79	1	$315.77	0	$0.00
---	---	----	---	---	---	---	---	----

FIG. 25.4. Fixed time-based event aggregation.

Customer ID	Zip Code	Gender	Start Date
1001	94116	Male	1/1/00
1002	94103	Female	7/1/99
-----------	--------	--------	----

Customer ID	Product	Date	Total
1001	Widget A	1/1/00	$95.79
1001	Widget B	1/5/00	$48.87
1001	Widget C	1/5/00	$503.49
1002	Widget A	7/1/99	$95.79
1002	Widget D	10/5/99	$219.98
1002	Widget A	12/3/99	$95.79
----------	-------------	---------	----------
----------	-------------	---------	----------

Cust. ID	Zip Code	Gender	# Events Q1	$ Total Q1	# Events Q2	$ Total Q2
1001	94116	Male	3	$648.15	0	$0.00
1002	94103	Female	1	$95.79	2	$315.77
----	----	----	---	---	---	---

FIG. 25.5. Relative time-based event aggregation.

An example of relative aggregation is shown in Fig. 25.5. In this case the aggregation is based on the "start date" column in the customer table. The first two quarters of customer purchase history are shown. This type of aggregation is much harder to precompute and, more important, is very dependant on the business question being asked. The appropriate relative date and level of aggregation is likely to be different for various situations. As with data cubes, it is impossible to precompute every possible combination of aggregation levels and dates. As shown in both of the examples, in addition to a count of rows that fall into a given time period, a function can be applied to any continuous column. The "Total" column contains the sum of the purchase amount. Any number of functions could be applied to any of the continuous columns in the event table. Typically, functions that compute the sum, average, minimum, or maximum are applied.

An example of aggregation based on the second major dimension, *value,* is shown in Fig. 25.6. Here the values of a particular column or columns in an event log are used for the aggregation. In this case the count of each distinct value in the "Product" column becomes a separate column in the aggregated table. Any column that contains nominal or ordinal values is a candidate for value-based aggregation. Like time-based aggregation, the function is not limited to a simple count or sum.

In all of the examples discussed, one thing that becomes immediately clear is how quickly the number of dimensions for CRM mining can grow. Using value-based aggregation with a product catalog of 10,000 products would add 10,000 columns to the data mining process, assuming all 10,000 products actually appeared in the data being analyzed. Therefore, either a significant amount of feature space reduction must occur prior to the pattern discovery phase, or algorithms that can handle high-dimensional problems must be used. Feature space reduction techniques, such as information gain (Cover & Thomas, 1991) or principal component analysis (PCA) (Jolliffe, 1986), are an entire subject unto themselves and are discussed in chapter 15 and in Mitchell (1996). As mentioned earlier, this potential need for feature space reduction explains why event data often is overlooked when CRM mining is performed. Why spend time adding more dimensions with aggregation techniques if there is already a need to reduce the existing feature space?

Customer ID	Zip Code	Gender	Start Date
1001	94116	Male	1/1/00
1002	94103	Female	7/1/99
-----------	--------	--------	----

Customer ID	Product	Date	Total
1001	Widget A	1/1/00	$95.79
1001	Widget B	1/5/00	$48.87
1001	Widget C	1/5/00	$503.49
1002	Widget A	7/1/99	$95.79
1002	Widget D	10/5/99	$219.98
1002	Widget A	12/3/99	$95.79
----------	-------------	---------	----------
----------	-------------	---------	----------

Cust. ID	Zip Code	Gender	Widget A	Widget B	Widget C	Widget D
1001	94116	Male	1	1	1	0
1002	94103	Female	2	0	0	1
---	---	----	----	---	---	---

FIG. 25.6. Value-based event aggregation.

Feature Preparation

The final step before actually performing pattern discovery is *feature preparation*. Most data mining algorithms are very particular about the format of the input data. This means that missing, out of range, and outlying values must be taken care of, as well as the proper encoding of strings, nominal features, and ordinal features. Missing values usually need to be filled in with some sort of default value. Out-of-range values, such as -10 for an age, should be removed or grouped into a special category. Outliers, values that are not necessarily wrong but so far astray that they are suspicious, typically need to be removed or grouped together as well. When an algorithm requires the input variables to be numeric, an encoding strategy must be used. When a numeric target variable is available, this is commonly used to guide the encoding. Nominal variables (strings or numeric) can be replaced by the mean or median value of the target for each distinct value. Another common strategy is to use the rank order of the target mean for each value. Examples of these encoding strategies are shown in Table 25.3. The third encoding strategy shown in the table is referred to as *disjunctive Boolean*. This is a common

TABLE 25.3
Example Encoding Schemes

Original Column	*Target Column*	*Target Mean*	*Target Mean Rank*	*Disjunctive Boolean*		
				Red	*Green*	*Blue*
Red	5	6	1	1	0	0
Blue	10	11	2	0	0	1
Green	20	19	3	0	1	0
Blue	12	11	2	0	0	1
Blue	11	11	2	0	0	1
Green	18	19	3	0	1	0
Red	7	6	1	1	0	0
Red	6	6	1	1	0	0
Blue	11	11	2	0	0	1

scheme when a numeric target does not exist. Each distinct value in the column is broken out into a separate column with a zero or one indicating which value was assigned to a specific row. Like the aggregation schemes previously discussed, disjunctive boolean encoding can greatly increase the dimensionality of a data set.

Also commonly included in feature preparation is the notion of variable compression or *binning*. Instead of passing every distinct value of a continuous variable to a data mining algorithm, labels can be assigned to different ranges of values, converting the column into a nominal feature with far fewer distinct values than the original data. The same process can be applied to nominal or ordinal features. Because many algorithms scale with the number of distinct values, or *cardinality* of a feature, fewer distinct values can greatly reduce the processing time. In addition, by removing some of the noise from a data set, variable binning can increase the generalization performance of an algorithm. As with feature space reduction, each of the potential feature preparation steps is a topic unto itself. Chapters 14, 15, and Pyle (1999) provide in-depth discussions of many of the topics discussed in this section.

PATTERN DISCOVERY

Once the appropriate data has been pulled out of the data warehouse and properly prepared, any number of data mining methods can be applied to answer just about any business question that comes to mind. This section briefly discusses seven of the more popular methods—classification, clustering, regression, sequential patterns, association rules, time series, and collaborative filtering. Detailed discussions of the methods and algorithms can be found in Part I of this book.

When discussing pattern discovery, a lot of confusion exists about terminology. As shown in Fig. 25.7, there are four major layers involved in pattern discovery, the top layer being the actual *business question*. The base layer contains *algorithms*. There are thousands of data mining, machine learning, artificial intelligence, and numeric algorithms that can be applied to pattern discovery. Some classes of algorithms, such as neural networks or Markov models, are broad enough to merit separate discussions (such as Chapters 3 and 11). However, most algorithms are discussed in terms of general *methods,* which make up the second layer of

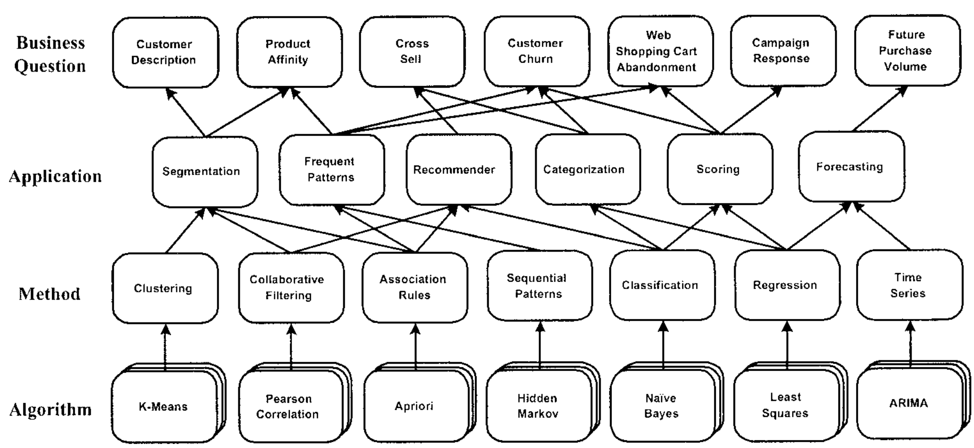

FIG. 25.7. Pattern discovery layers.

Fig. 25.7. A method is defined by the type of output obtained as a result of processing input data. The details of how the data is processed are not relevant when discussing methods. As shown, any number of algorithms can be used for a given method. For example, clustering could be performed using the *k*-means (Forgy, 1965), CLARANS (Ng & Han, 1994), or BIRCH (Zhang, Ramakrishnan, & Livny, 1996) algorithms, just to name a few. The third layer is the *application* layer. Applications wrap a method (or several methods) in business logic to solve a particular type of business question. For example, a categorization application answers questions about to what category a particular example belongs. The obvious choice of method for this application would be a classification method, but there is no reason a regression method could not be used instead. The methods simply output a score, probability, or yes/no for each possible category. A categorization application must handle a threshold for continuous outputs, the possibility of multiple categories, hierarchical categories, and so forth. As shown, a given business question can be answered by a number of different applications, which in turn can be solved by a range of methods. The connections and values in each layer of Fig. 25.7 are not meant to be an exhaustive list, but simply provide examples.

Classification methods (Friedman, 1994; Mitchell, 1996; Quinlan, 1993) determine how to assign examples to a set of classes. A training data set with the class labels already present is required for a classification method to learn. Clustering methods (Kaufman & Rousseeuw, 1990; Ng & Han, 1994) are used when a description of the various groups in a data set is required, and there is not necessarily a predefined set of class labels. Once a clustering is performed, the clusters often become the labels for a subsequent classification. Regression methods (Breiman, Friedman, Olshen, & Stone, 1984; Friedman, 1991) try to predict a continuous output by building a function that is an estimate of the “true” underlying pattern in the data. As previously mentioned, regression methods can be used for a binary classification application and are also commonly used for scoring and forecasting. Like regression, time-series methods predict continuous values, but do so over time, typically modeling both trends and cyclic behaviors in addition to the actual values. Both association rules (Agrawal & Srikant, 1994) and sequential patterns (Mannila, Toivonen, & Verkamo, 1995; Srikant & Agrawal, 1996) describe events that occur frequently in a data set. Association rules find correlations between items in a data set without taking order or time into consideration. For sequential patterns the order of the occurrence of the events is taken into account, as well as the actual values of the events.

A variant of clustering that is very useful for CRM mining is *targeted clustering,* sometimes referred to as supervised clustering. Because clustering is considered to be an *unsupervised* method, this chapter will use the term targeted clustering to avoid confusion. Targeted clustering groups the data with respect to a target variable. With an untargeted clustering method the best groups are found according to some generic distance metric that considers each dimension to be equally important. Although these groups can be differentiated from each other based on at least one of the features, such as age, gender, or behaviors, the groups may not be different in a meaningful business context. If all of the groups have the same average customer lifetime value or all tend to buy the same set of products, this is not necessarily useful. Targeted clustering attempts to find groups that are different with respect to the value of a target variable. This can be done by increasing the weight of the target variable in the distance metric or encoding the variables with respect to the target. In the extreme case, when the target is the only variable included in the distance metric, targeted clustering closely resembles classification. The important distinction between targeted clustering and classification is that there are no predefined categories for targeted clustering.

Collaborative filtering methods (Resnik, Iacovou, Suchak, Bergstrom, & Riedl, 1994; Shardanand & Maes, 1995) analyze events to find sets that are similar. In a CRM context, events are usually grouped by customer. For each customer in the data set collaborative filtering methods find a group of other customers that are similar with respect to the occurrences in the event logs. Once the group is formed, events can be ranked by how popular they are across the entire group. The output of a collaborative filtering method can be both the customers that are similar and the event rankings. For example, if the events are product purchases, the event rankings can be used for product recommendations. Any product that is highly ranked and has not already been purchased by a customer can be recommended. The idea is that if other customers with similar purchase histories have bought a particular product, it is likely the first customer will buy that product as well. Collaborative filtering is similar to clustering in that it discovers natural groups in a data set. The important difference in collaborative filtering is that there is a distinct group formed around each item (customer) in the data set. In effect each customer is the center of a unique group, and group memberships are not necessarily transitive. Customer A may appear in Customer B's group, but not vice versa. This makes collaborative filtering methods ideal for personalized recommender system applications.

PATTERN ANALYSIS AND DEPLOYMENT

As shown in Fig. 25.1, the discovery of patterns is not the final step for CRM mining. Although the patterns themselves may be the desired outcome for some questions, in most cases the patterns must be applied to new data. Pattern analysis includes simple reporting, graphical displays, and even further computations on the patterns. For patterns that will potentially be deployed, one of the most important analysis techniques is evaluating the *robustness*. A measure of robustness indicates how well a pattern or model will perform on new data. For patterns that are merely descriptive and will not be used on new data, a measure of interestingness is often useful as well as robustness.

Robustness

Whether the discovered pattern is a classification, clustering, or some other application, one of the most important metrics for CRM mining is how well the pattern will perform on new data. A classification model that is 90% accurate on existing data but only 50% accurate on new incoming data is not necessarily useful, especially if 70% accuracy is needed to realize a profit for a particular scenario. A model that is 80% accurate on existing data and 80% accurate on new data is probably preferable. The measure of the relationship between training error and deployment or test error is referred to as robustness. The more consistent the performance, the more robust the model. Even for patterns that are mainly used to describe a data set, such as clustering or association rules, it is important to determine whether the discovered patterns will hold in general or are specific to a particular data set. In statistics this often is referred to as finding the balance between *underfitting* and *overfitting*. As with many of the steps discussed in this chapter, measuring or guaranteeing robustness is a topic unto itself. One of the most common techniques is *cross-validation*. Here several models are built using different subsets of the existing data. The models are either compared or combined to try to verify that the model is a true pattern inherent in the data, instead of random noise. When looking for patterns that occur in only a small portion of the data, like association rules, it is easy to mistake random correlations for true patterns. Another way of ensuring robustness is to use a framework such as Vapnik's statistical learning theory (Vapnik, 1995) when discovering patterns.

Interestingness

When the discovered patterns are being used to get a description of the customers, a common problem is a lack of "interesting" patterns. The most frequent or highly correlated patterns tend to highlight relationships that are obvious or already known. Although confirmation of expected patterns is not completely useless, the real goal of CRM mining is to discover patterns that were previously unknown or unexpected that can then be leveraged to increase the ratio of profitable events. To ensure that all patterns of potential interest are discovered, the thresholds of many pattern discovery algorithms must be set low enough such that hundreds or thousands of patterns are output. Although the number of items to analyze has been reduced by several orders of magnitude, a list of several thousand patterns is only marginally more useful than the millions of lines of raw data. To make matters worse, the notion of what is interesting is very dependent on the goals of the analysis. This means that interestingness for CRM mining is not necessarily an objective measure, but is more subjective. The concept of *subjective interestingness* for data mining has been studied extensively in Cooley (2000), Liu, Hsu, and Chen (1997), and Padmanabhan and Tuzhilin (1998). Usually, the idea is to define what is known or expected. The interesting patterns are then the ones that do not meet the definition of an expected pattern.

Deployment

Finally, a predictive pattern or model that meets the acceptable robustness criteria must be deployed. Figure 25.8 depicts a typical deployment scheme for a new data set. The first two steps, ETL and data preparation, are identical to the standard CRM mining process. The only difference is that the target values are unknown. Once the new data is prepared, an existing model is used to generate predictions. Depending on the application, the predictions may be categories, clusters, scores, probabilities, or other continuous values.

In some situations, such as a call center, a single data point or datum must be scored, instead of entire set of data. This is shown in Fig. 25.9. A single datum does not need to go through a full ETL process. The need to calculate a prediction for a single data point is usually associated with a real-time situation. For example, a customer calls to purchase some items. Any cross-sell offers or a discount to prevent churn must be made while the customer is still on the phone. This requires a streamlined data collection and preparation process followed immediately by a prediction.

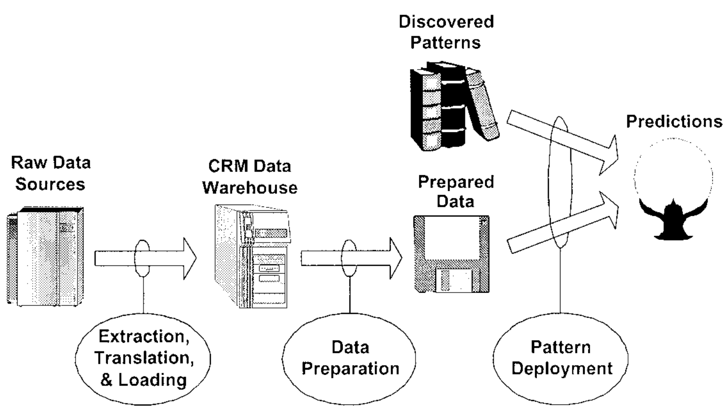

FIG. 25.8. Pattern deployment.

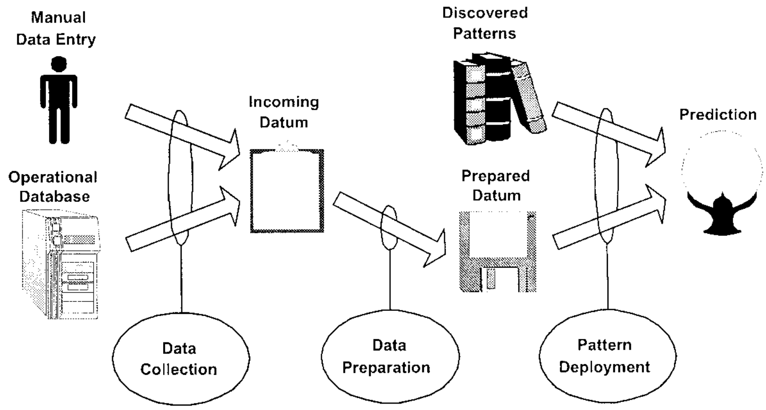

FIG. 25.9. Real-time deployment.

SAMPLE BUSINESS PROBLEMS

CRM business questions can be broken down into two major types—*strategic* and *operational*. Strategic questions are focused more on the descriptive power of CRM mining, whereas operational questions are aligned more closely with the predictive capabilities. This section uses the MyWidgets example for a number of common strategic and operational CRM mining questions.

Strategic Questions

A strategic question is one for which the answer is used to guide or influence a decision. When using CRM mining to answer a strategic question, it is the descriptive qualities of the discovered patterns that are the most important. An example would be an analysis to determine the key drivers of customer satisfaction. Actions can be taken based on a report summarizing the results of the analysis, but there is no model to be directly "deployed." The descriptive properties of the analysis itself is the end goal of CRM mining for strategic questions. MyWidgets is planning its next advertising and promotional campaign. The goals for the campaign are to bring in new "high-value" customers and increase the effectiveness of the Web channel. The three top strategic questions have been identified as "What is the description of the high value customers?", "What products are frequently purchased together?", and "Why are people abandoning Web shopping carts?" The answers will help determine where to place advertisements and what special offers should be made.

MyWidgets does not have a standing definition of "high value" customers. It is generally difficult to define a hard cutoff for concepts such as high- and low-value customers. This makes the question more suited to a targeted segmentation application. By clustering with respect to the total number of purchases each customer has made, the definition of "high value" will be made by the algorithm. In addition, the analysis will describe medium- and low-value customers as well. Figure 25.10 shows sample data for the analysis after data aggregation has been performed on the MyWidgets data warehouse from Fig. 25.2. The data includes information from the Web session, phone call, store visit, and purchase event tables. A relative time-based aggregation has been used, with the "start date" column from the "customer" table. The results of the analysis could be used to determine whether one channel provides higher

Cust. ID	Age	Zip Code	Gender	Web Sessions Year 1	Web Sessions Year 2	Purchase Calls Year 1	Purchase Calls Year 2	Support Calls Year 1	Support Calls Year 2	Store Visits Year 1	Store Visits Year 2	Total Purchase
1001	35	94116	Male	3	6	1	1	3	2	0	2	$648.15
1002	26	94103	Female	0	0	0	2	0	1	1	2	$95.79
------	---	----- --	----	---	---	---	---	---	---	---	---	--------

FIG. 25.10. Sample customer value clustering data.

value customers, or whether high-value customers have different demographic attributes. This will assist MyWidgets in choosing where to place the ads.

A description of what products are purchased together will assist MyWidgets in putting together promotional offers, such as "Get 15% off Widget A when Widget C is purchased." If products are not normally purchased together, a campaign trying to increase their cross-selling may be ineffective. The only data necessary for the analysis is the purchase data, as shown in Fig. 25.11. Association rules are the natural choice for answering this business question. However, as discussed previously, a subjective interestingness filter will probably need to be applied to reduce the number of discovered rules to a manageable number.

Shopping cart abandonment (SCA) occurs on a Web site when a potential buyer performs one or more *product add* E-commerce events with no subsequent *product purchase* event. A description of things that tend to cause SCA is an important first step in trying to reduce abandonment. For the MyWidgets campaign, the results of the analysis may suggest what offers to put on certain Web pages. A well-placed offer can entice a customer to proceed with a purchase instead of abandoning a cart. Alternatively, it may turn out that the checkout process itself is confusing. A significant number of abandonments could be customers who intended to purchase but were unable to navigate the checkout process. A sequential pattern analysis of the E-commerce events should be used for this question. If possible, the click events could be included as well. Figure 25.12 shows a sample pulled from the E-commerce table.

Operational Questions

An operational question is one for which the results can be directly applied to increasing the ratio of profitable events. For example, in direct mail campaign response modeling the model scores themselves are the most important output. Although it is desirable to have an explainable pattern or model, the model itself is the goal of the analysis. The "answers" to operational questions must be deployed to bring a return on investment. In conjunction with the new promotional campaign, MyWidgets wants to recommend products for cross-selling

Transaction ID	Product ID
541	A
541	B
541	C
542	A
542	D
543	B
---	---

FIG. 25.11. Sample purchase affinity data.

Session ID	DateTime	Event Type	Event Value
10321	1/5/2001 10 :43:32	View	A
10321	1/5/2001 10 :43:54	View	B
10321	1/5/2001 10 :44:05	Click-through	B
10321	1/5/2001 10 :44:46	Add	B
10321	1/5/2001 10 :45:29	Checkout 1	
10321	1/5/2001 10 :47:01	Checkout 2	
10321	1/5/2001 10 :47:56	Purchase	B
10566	1/7/2001 8:07:01	View	C
10566	1/7/2001 8:07:34	Click-through	C
10566	1/7/2001 8:08:18	Add	C
10566	1/7/2001 8:08:39	View	A
10566	1/7/2001 8:09:14	Click-through	A
---	---	---	---

FIG. 25.12. Sample shopping cart abandonment data.

and up-selling for customers who visit the Web site or order products on the phone. In addition, MyWidgets wants to make a special promotional offer to any customer who is in danger of churning.

Because MyWidgets tracks E-commerce events from its Web site, product recommendations can be made based on what customers looked at as well as what was purchased in the past. MyWidgets has decided that a product view is an indication of first level interest, a product click-through is second level, product adds are third level, and product purchases are fourth level. Product purchases can come from any of the MyWidgets channels, but the first three levels of interest are available only from the Web site. Sample data for this question is shown in Fig. 25.13. The data has been pulled from the E-commerce, phone, and purchase tables. Only the highest level of interest shown for each product was recorded when preparing the data. A collaborative filtering method is ideally suited for this question. For a given customer each product gets a score on how likely the customer is to be interested in that product. In the sample data, customers 1002 and 1004 have very similar histories. They both viewed Widget D, clicked-through Widget A for more information, and purchased Widget E. Because customer 1004 also purchased Widget C, collaborative filtering would rank this highly for recommendation to customer 1002.

For the churn question, MyWidgets has decided that any customer who has not had any events in the past 3 months should be considered to have left, or "churned." The data sources

Cust. ID	Widget A	Widget B	Widget C	Widget D	Widget E
1001	4	0	1	3	0
1002	2	0	0	1	4
1003	1	0	0	1	4
1004	2	0	4	1	4
---	---	---	---	---	---

FIG. 25.13. Sample product recommendation data.

Cust. ID	Age	Zip Code	Gender	Web Sessions Month -2	Web Sessions Month -1	Purchase Calls Month -2	Purchase Calls Month -1	Support Calls Month -2	Support Calls Month -1	Store Visits Month -2	Store Visits Month -1	Churn
1001	35	94116	Male	3	0	1	1	5	10	0	2	True
1002	26	94103	Female	1	2	0	2	0	1	1	2	False
------	---	-------	----	---	---	---	---	---	---	---	---	--------

FIG. 25.14. Sample churn data.

that will be included in the analysis are the Web sessions, phone calls, and purchase events. For aggregation purposes the date of the last event will be used as a relative aggregation point. Hopefully, there will be a significant pattern in the behavior of the customers who churn, prior to the last event. Figure 25.14 shows a sample of the training data after data aggregation has been performed. The target is binary, "true" for customers who have churned and "false" for those who are still active. This makes the churn question a binary classification application. Like the high-value customer description question, an algorithm that handles high-dimensional data will need to be used for predicting customer churn.

SUMMARY

This chapter has given an overview of the steps required for CRM mining. The process starts with a data warehouse and continues through data preparation, pattern discovery, pattern analysis, and in some cases pattern deployment. The bulk of the work involved with CRM mining is in the data preparation and pattern analysis. This is because the situation for every enterprise is unique, with different data sources and business questions to be answered. Although pattern discovery makes use of some very complex data mining and machine learning algorithms, there is no additional work that has to be done to apply them to CRM data once the data has been properly prepared.

The ideal data sources for CRM mining are customer behavior data. Because the ultimate goal of CRM mining is to increase ratio of profitable to unprofitable customer behaviors, data that directly measure those behaviors are superior to data that are merely surrogates. E-commerce data is especially valuable because it is the only data source that logs every interaction a customer has with a touchpoint.

As the examples in the sixth section show, show, a properly designed CRM data warehouse can support data mining for a wide variety of business questions. However, with data from a real CRM warehouse the number of dimensions in the prepared data can quickly grow into the hundreds or thousands. This means that CRM mining needs to take advantage of advanced feature space reduction techniques or algorithms that scale and perform well in high-dimensional spaces.

To be cost effective, a CRM mining system must be used to answer a variety of business questions. If the fixed costs associated with data preparation, data analysis, and pattern deployment are not amortized over several projects, it is unlikely that the CRM mining system will be profitable. Not every analysis leads to a million-dollar saving that will cover the costs of the data mining project. It is more likely that each analysis will lead to a more modest return on investment, and some may not result in any savings at all. A successful CRM mining system must be set up to answer questions in a timely manner, for example, days not weeks. Only then will the increased profits from maximizing the customer event ratio outweigh the costs of CRM mining.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference* (pp. 487–499). San Francisco: Morgan Kaufmann.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.

Buchner, A., & Mulvenna, M. D. (1998). Discovering Internet marketing intelligence through online analytical Web usage mining. *SIGMOD Record, 27,* 54–61. New York: ACM Press.

Cooley, R. (2000). *Web usage mining: Discovery and application of interesting patterns from Web data*. Unpublished doctoral dissertation, University of Minnesota.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. *Knowledge and Information Systems, 1,* 5–32.

Cooley, R., Tan, P.-N., & Srivastava, J. (1999). Websift: The Web site information filter system. In *International WEBKDD Workshop*. New York: ACM Press.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory.* New York: John Wiley & Sons.

Ehrenberg, A. S. C. (1988). *Repeat-Buying: Facts, theory and applications*. London: Charles Griffin.

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics, 21,* 768–769.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics, 19,* 1–141.

Friedman, J. H. (1994). *Flexible nearest neighbor classification* (Tech. Rep.). Stanford, CA: Stanford University.

Jolliffe, I. T. (1986). Principal component analysis. In *Principal component analysis.* New York: Springer-Verlag.

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: John Wiley & Sons.

Kimball, R., & Merz, R. (2000). *The data webhouse toolkit*. New York: Wiley.

Liu, B., Hsu, W., & Chen, S. (1997). Using general impressions to analyze discovered classification rules. In *Third International Conference on Knowledge Discovery and Data Mining* (pp. 31–36). Menlo Park, CA: AAAI Press.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1995). Discovering frequent episodes in sequences. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 210–215). Menlo Park, CA: AAAI Press.

Mitchell, T. (1996). *Machine learning*. New York: McGraw-Hill.

Ng, R., & Han, J. (1994). Efficient and effective clustering method for spatial data mining. In *Proceedings of the 20th VLDB Conference* (pp. 144–155). Santiago, Chile.

Padmanabhan, B., & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In *Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 94–100). Menlo Park, CA: AAAI Press.

Pyle, D. (1999). *Data preparation for data mining*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Resnik, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Conference on Computer Supported Cooperative Work* (pp. 175–181). New York: ACM Press.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth." In *ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 210–217). New York: ACM Press.

Spiliopoulou, M., & Faulstich, L. C. (1998). WUM: A Web utilization miner. In *EDBT Workshop WebDB98* (pp. 109–115). Berlin: Springer-Verlag.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Fifth International Conference on Extending Database Technology* (pp. 3–17). Berlin: Springer-Verlag.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag,

Wu, K.-l., Yu, P. S., & Ballman, A. (1998). Speedtracer: A Web usage mining and analysis tool. *IBM Systems Journal, 37,* 89–105.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. In *ACM-SIGMOD International Conference on Management of Data* (pp. 103–114). New York: ACM Press.