Data mining techniques

Martin Brown

December 11, 2012

Many different data mining, query model, processing model, and data collection techniques are available. Which one do you use to mine your data, and which one can you use in combination with your existing software and infrastructure? Examine different data mining and analytics techniques and solutions, and learn how to build them using existing software and installations. Explore the different data mining tools that are available, and learn how to determine whether the size and complexity of your information might result in processing and storage complexities, and what to do.

Data mining as a process

Fundamentally, data mining is about processing data and identifying patterns and trends in that information so that you can decide or judge. Data mining principles have been around for many years, but, with the advent of big data, it is even more prevalent.

Big data caused an explosion in the use of more extensive data mining techniques, partially because the size of the information is much larger and because the information tends to be more varied and extensive in its very nature and content. With large data sets, it is no longer enough to get relatively simple and straightforward statistics out of the system. With 30 or 40 million records of detailed customer information, knowing that two million of them live in one location is not enough. You want to know whether those two million are a particular age group and their average earnings so that you can target your customer needs better.

These business-driven needs changed simple data retrieval and statistics into more complex data mining. The business problem drives an examination of the data that helps to build a model to describe the information that ultimately leads to the creation of the resulting report. Figure 1 outlines the process.
Outline of the process

The process of data analysis, discovery, and model-building is often iterative as you target and identify the different information that you can extract. You must also understand how to relate, map, associate, and cluster it with other data to produce the result. Identifying the source data and formats, and then mapping that information to our given result can change after you discover different elements and aspects of the data.

Data mining tools

Data mining is not all about the tools or database software that you are using. You can perform data mining with comparatively modest database systems and simple tools, including creating and writing your own, or using off the shelf software packages. Complex data mining benefits from the past experience and algorithms defined with existing software and packages, with certain tools gaining a greater affinity or reputation with different techniques.

For example, IBM SPSS®, which has its roots in statistical and survey analysis, can build effective predictive models by looking at past trends and building accurate forecasts. IBM InfoSphere® Warehouse provides data sourcing, preprocessing, mining, and analysis information in a single
package, which allows you to take information from the source database straight to the final report output.

It is recent that the very large data sets and the cluster and large-scale data processing are able to allow data mining to collate and report on groups and correlations of data that are more complicated. Now an entirely new range of tools and systems available, including combined data storage and processing systems.

You can mine data with a various different data sets, including, traditional SQL databases, raw text data, key/value stores, and document databases. Clustered databases, such as Hadoop, Cassandra, CouchDB, and Couchbase Server, store and provide access to data in such a way that it does not match the traditional table structure.

In particular, the more flexible storage format of the document database causes a different focus and complexity in terms of processing the information. SQL databases impose strict structures and rigidity into the schema, which makes querying them and analyzing the data straightforward from the perspective that the format and structure of the information is known.

Document databases that have a standard such as JSON enforcing structure, or files that have some machine-readable structure, are also easier to process, although they might add complexities because of the differing and variable structure. For example, with Hadoop's entirely raw data processing it can be complex to identify and extract the content before you start to process and correlate the it.

Key techniques

Several core techniques that are used in data mining describe the type of mining and data recovery operation. Unfortunately, the different companies and solutions do not always share terms, which can add to the confusion and apparent complexity.

Let's look at some key techniques and examples of how to use different tools to build the data mining.

Association

Association (or relation) is probably the better known and most familiar and straightforward data mining technique. Here, you make a simple correlation between two or more items, often of the same type to identify patterns. For example, when tracking people's buying habits, you might identify that a customer always buys cream when they buy strawberries, and therefore suggest that the next time that they buy strawberries they might also want to buy cream.

Building association or relation-based data mining tools can be achieved simply with different tools. For example, within InfoSphere Warehouse a wizard provides configurations of an information flow that is used in association by examining your database input source, decision basis, and output information. Figure 2 shows an example from the sample database.
Information flow that is used in association

![Diagram of data flow](image)

**Classification**

You can use classification to build up an idea of the type of customer, item, or object by describing multiple attributes to identify a particular class. For example, you can easily classify cars into different types (sedan, 4x4, convertible) by identifying different attributes (number of seats, car shape, driven wheels). Given a new car, you might apply it into a particular class by comparing the attributes with our known definition. You can apply the same principles to customers, for example by classifying them by age and social group.

Additionally, you can use classification as a feeder to, or the result of, other techniques. For example, you can use decision trees to determine a classification. Clustering allows you to use common attributes in different classifications to identify clusters.

**Clustering**

By examining one or more attributes or classes, you can group individual pieces of data together to form a structure opinion. At a simple level, clustering is using one or more attributes as your basis for identifying a cluster of correlating results. Clustering is useful to identify different information because it correlates with other examples so you can see where the similarities and ranges agree.

Clustering can work both ways. You can assume that there is a cluster at a certain point and then use our identification criteria to see if you are correct. The graph in Figure 3 shows a good example. In this example, a sample of sales data compares the age of the customer to the size of the sale. It is not unreasonable to expect that people in their twenties (before marriage and kids), fifties, and sixties (when the children have left home), have more disposable income.
Clustering

In the example, we can identify two clusters, one around the US$2,000/20-30 age group, and another at the US$7,000-8,000/50-65 age group. In this case, we've both hypothesized and proved our hypothesis with a simple graph that we can create using any suitable graphing software for a quick manual view. More complex determinations require a full analytical package, especially if you want to automatically base decisions on nearest neighbor information.

Plotting clustering in this way is a simplified example of so called nearest neighbor identity. You can identify individual customers by their literal proximity to each other on the graph. It’s highly likely that customers in the same cluster also share other attributes and you can use that expectation to help drive, classify, and otherwise analyze other people from your data set.

You can also apply clustering from the opposite perspective; given certain input attributes, you can identify different artifacts. For example, a recent study of 4-digit PIN numbers found clusters between the digits in ranges 1-12 and 1-31 for the first and second pairs. By plotting these pairs, you can identify and determine clusters to relate to dates (birthdays, anniversaries).

Prediction

Prediction is a wide topic and runs from predicting the failure of components or machinery, to identifying fraud and even the prediction of company profits. Used in combination with the other data mining techniques, prediction involves analyzing trends, classification, pattern matching, and relation. By analyzing past events or instances, you can make a prediction about an event.

Using the credit card authorization, for example, you might combine decision tree analysis of individual past transactions with classification and historical pattern matches to identify whether a transaction is fraudulent. Making a match between the purchase of flights to the US and transactions in the US, it is likely that the transaction is valid.
Sequential patterns

Often used over longer-term data, sequential patterns are a useful method for identifying trends, or regular occurrences of similar events. For example, with customer data you can identify that customers buy a particular collection of products together at different times of the year. In a shopping basket application, you can use this information to automatically suggest that certain items be added to a basket based on their frequency and past purchasing history.

Decision trees

Related to most of the other techniques (primarily classification and prediction), the decision tree can be used either as a part of the selection criteria, or to support the use and selection of specific data within the overall structure. Within the decision tree, you start with a simple question that has two (or sometimes more) answers. Each answer leads to a further question to help classify or identify the data so that it can be categorized, or so that a prediction can be made based on each answer.

Figure 4 shows an example where you can classify an incoming error condition.

Decision tree

![Decision Tree Example]

Decision trees are often used with classification systems to attribute type information, and with predictive systems, where different predictions might be based on past historical experience that helps drive the structure of the decision tree and the output.

Combinations

In practice, it’s very rare that you would use one of these exclusively. Classification and clustering are similar techniques. By using clustering to identify nearest neighbors, you can further refine your classifications. Often, we use decision trees to help build and identify classifications that we can track for a longer period to identify sequences and patterns.
Long-term (memory) processing

Within all of the core methods, there is often reason to record and learn from the information. In some techniques, it is entirely obvious. For example, with sequential patterns and predictive learning you look back at data from multiple sources and instances of information to build a pattern.

In others, the process might be more explicit. Decision trees are rarely built one time and are never forgotten. As new information, events, and data points are identified, it might be necessary to build more branches, or even entirely new trees, to cope with the additional information.

You can automate some of this process. For example, building a predictive model for identifying credit card fraud is about building probabilities that you can use for the current transaction, and then updating that model with the new (approved) transaction. This information is then recorded so that the decision can be made quickly the next time.

Data implementations and preparation

Data mining itself relies upon building a suitable data model and structure that can be used to process, identify, and build the information that you need. Regardless of the source data form and structure, structure and organize the information in a format that allows the data mining to take place in as efficient a model as possible.

Consider the combination of the business requirements for the data mining, the identification of the existing variables (customer, values, country) and the requirement to create new variables that you might use to analyze the data in the preparation step.

You might compose the analytical variables of data from many different sources to a single identifiable structure (for example, you might create a class of a particular grade and age of customer, or a particular error type).

Depending on your data source, how you build and translate this information is an important step, regardless of the technique you use to finally analyze the data. This step also leads to a more complex process of identifying, aggregating, simplifying, or expanding the information to suit your input data (see Figure 5).
Data preparation

Your source data, location, and database affects how you process and aggregate that information.

Building on SQL

Building on an SQL database is often the easiest of all the approaches. SQL (and the underlying table structure they imply) is well understood, but you cannot completely ignore the structure and format of the information. For example, when you examine user behavior in sales data, there are two primary formats within the SQL data model (and data-mining in general) that you can use: transactional and the behavioral-demographic.

When you use InfoSphere Warehouse, creating a behavioral-demographic model for the purposes of mining customer data to understand buying and purchasing patterns involves taking your source SQL data based upon the transaction information and known parameters of your customers, and rebuilding that information into a predefined table structure. InfoSphere Warehouse can then use this information for the clustering and classification data mining to get the information you need. Customer demographic data, and sales transaction data can be combined and then reconstituted into a format that allows for specific data analysis, as shown in Figure 6.
For example, with sales data you might want to identify the sales trends of particular items. You can convert the raw sales data of the individual items into transactional information that maps the customer ID, transaction data, and product ID. By using this information, it is easy to identify sequences and relationships for individual products by individual customers over time. That enables InfoSphere Warehouse to calculate sequential information, such as when a customer is likely to buy the same product again.

You can build new data analysis points from the source data. For example, you might want to expand (or refine) your product information by collating or classifying individual products into wider groups, and then analyzing the data based on these groups in place of an individual.

For example, Table 1 shows how to expand the information in new ways.

### A table of products expanded

<table>
<thead>
<tr>
<th>product_id</th>
<th>product_name</th>
<th>product_group</th>
<th>product_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>strawberries, loose</td>
<td>strawberries</td>
<td>fruit</td>
</tr>
<tr>
<td>102</td>
<td>strawberries, box</td>
<td>strawberries</td>
<td>fruit</td>
</tr>
</tbody>
</table>
Document databases and MapReduce

The MapReduce processing of many modern document and NoSQL databases, such as Hadoop, are designed to cope with the very large data sets and information that does not always follow a tabular format. When you work with data mining software, this notion can be both a benefit and a problem.

The main issue with document-based data is that the unstructured format might require more processing than you expect to get the information you need. Many different records can hold similar data. Collecting and harmonizing this information to process it more easily relies upon the preparation and MapReduce stages.

Within a MapReduce-based system, it is the role of the map step to take the source data and normalize that information into a standard form of output. This step can be a relatively simple process (identify key fields or data points), or more complex (parse and processing the information to produce the sample data). The mapping process produces the standardized format that you can use as your base.

Reduction is about summarizing or quantifying the information and then outputting that information in a standardized structure that is based upon the totals, sums, statistics, or other analysis that you selected for output.

Querying this data is often complex, even when you use tools designed to do so. Within a data mining exercise, the ideal approach is to use the MapReduce phase of the data mining as part of your data preparation exercise.

For example, if you are building a data mining exercise for association or clustering, the best first stage is to build a suitable statistic model that you can use to identify and extract the necessary information. Use the MapReduce phase to extract and calculate that statistical information then input it to the rest of the data mining process, leading to a structure such as the one shown in Figure 7.
MapReduce structure

In the previous example, we’ve taken the processing (in this case MapReduce) of the source data in a document database and translated it into a tabular format in an SQL database for the purposes of data mining.

Working with this complex, even unformatted, information can require preparation and processing that is more complex. There are certain complex data types and structures that cannot be processed and prepared in one step into the output that you need. Here you can chain the output of your MapReduce either to map and produce the data structure that you need sequentially, as in Figure 8, or individually to produce multiple output tables of data.
Chaining output of your MapReduce sequentially

For example, taking raw logging information from a document database and running MapReduce to produce a summarized view of the information by date can be done in a single pass. Regenerating the information and combining that output with a decision matrix (encoded in the second MapReduce phase), and then further simplified to a sequential structure, is a good example of the chaining process. We require whole set data in the MapReduce phase to support the individual step data.

Regardless of your source data, many tools can use flat file, CSV, or other data sources. InfoSphere Warehouse, for example, can parse flat files in addition to a direct link to a DB2 data warehouse.
Conclusion

Learn more. Develop more. Connect more.

The new developerWorks Premium membership program provides an all-access pass to powerful development tools and resources, including 500 top technical titles for application developers through Safari Books Online, deep discounts on premier developer events, video replays of recent O'Reilly conferences, and more. Sign up today.

Data mining is more than running some complex queries on the data you stored in your database. You must work with your data, reformat it, or restructure it, regardless of whether you are using SQL, document-based databases such as Hadoop, or simple flat files. Identifying the format of the information that you need is based upon the technique and the analysis that you want to do. After you have the information in the format you need, you can apply the different techniques (individually or together) regardless of the required underlying data structure or data set.

© Copyright IBM Corporation 2012
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)