Chapter 5

Cluster Analysis

Chapter Outline

Introduction, 218

Business Situation, 220

Model, 221

Distance or Dissimilarities, 222

Combinatorial Searches with K-Means, 224

Statistical Mixture Model with EM, 227

Hierarchical Clusters, 229

Other Statistical Methods, 233

Data, 236

Attributes and Observations, 236

Continuous and Discrete Data, 237

Missing Data, 238

Clustering on Products: Cars, 238

Goals, 238

Data, 239

Microsoft Clustering, 241

Results from Microsoft Clustering, 243

Prediction, 246

Larger Model and Parameter Changes, 248

Traditional EM Clustering, 251

Goals and Data, 252

Results, 254

K-Means Clusters, 255

Comparison, 256

Customer Clustering with Categorical Data, 258

Data, 258

Microsoft Clustering Results, 259

Weka Clustering Results, 260

Summary, 262

Key Words, 263

Review Questions, 263

Exercises, 264

Additional Reading, 267

What You Will Learn in This Chapter

- Are all customers the same?
- Why would a business use clustering analysis?
- How does clustering work and how are the results interpreted?
- What type of data is used in clustering?
- How are clusters identified with multiple dimensions?
- How are traditional EM methods different from Microsoft Clustering?
- Are the results from the multiple tools and methods really different?
- How does categorical data change the results and interpretation of clusters?

Alberta

Cluster analysis is a common tool in marketing. In particular, it is used to categorize customers into groups. By identifying common features, it becomes possible to find people with similar features and target advertising to their needs. For example, two professors categorized five types of domestic tourists in the province of Alberta, Canada. Domestic tourists are those from within the region (Alberta) who travel and spend money on vacations and visits within the same region. The researchers used focus groups to identify potential attributes—particularly reasons given for traveling. Telephone surveys were used to collect data, and the cluster analysis identified five major groups of travelers as outlined in the table. Clustering software identifies the five clusters based on the attributes. The researchers added the descriptive titles of the clusters. The Alberta marketing organization then used these clusters and resulting preferences to create TV, radio, and newspaper ads to target each of the groups. [Hudson and Ritchie 2002]

1. Young Urban Active Outdoor N=520,730	2. Indoor leisure traveler N=586,285	3. Children-first traveler N=446,516	4. Fair-weather friends N=445,280	5. Older cost-conscious traveler N=754,501
M=49% F=51% Avg. age: 37.5 Married: 66% >$100,000: 16%	M=31% F=69% Avg. age: 40.2 Married: 70% >$100,000: 9%	M=50% F=50% Avg. age: 42.5 Married: 75% >$100,000: 19%	M=59% F=41% Avg. age: 44.2 Married: 62% >$100,000: 13%	M=44% F=56% Avg. age: 44.8 Married: 62% >$100,000: 11%
School holidays Cost/value Safe/secure	Safe/secure Cost/value Weather	Children's sports Safe/secure Cost/value	Family/friends Weather	Safe/secure Cost/value Weather

Row 1: label assigned by researchers and the count.
Row 2: Cluster rules and percentages
Row 3: Top key words

Clusters are a key tool for unsupervised learning. With minimal configuration, the tools can find groups of items that are similar. These groups reduce complexity, allowing decision makers to focus on a few key attributes.

Simon Hudson and Brent Ritchie, 2002, "Understanding the Domestic Market Using Cluster Analysis: A Case Study of the Marketing Efforts of Travel Alberta," *Journal of Vacation Marketing*, 8(3), 263-276.

Introduction

Are all customers the same? It is unlikely that all customers are the same. If so, the organization needs to work on expanding its offerings. But, are there groups of customers that act similarly? Several marketing companies provide services to interview and identify types of customers. These groups are given names and sample pictures so managers and salespeople can visualize each customer group. In many cases, product lines are designed specifically for each target group. Customer grouping is a classic application of **clustering**. The goal is to find clusters or groups so that people who fall within a specific group have more in common with members of that group than with any other cluster.

The concept of "more in common" is defined by the attributes or dimensions available. The values of the attributes are evaluated in terms of a distance measure. For example, if age is an attribute and two customers have the same age, then the distance measure is zero and they will be placed in the same group based on that attribute. Customers who are farther apart in age are likely to be placed into different clusters. Of course, with multiple attributes distance has to be measured across all attributes. Typically, the distance measures are summed across attributes.

Clusters are useful to summarize or reduce the number of dimensions. A company could have millions of customers with dozens of attribute measures. But if clustering can reduce them into 5 – 10 categories, then marketing can focus on those groups. The same concepts apply to other topics, such as products, inputs, regions, or almost anything with multiple dimensions. Instead of trying to treat every dimension separately, clustering looks for internal correlations where collections of various attributes are shared by a large enough number of items.

Figure 5.1

Simple cluster example with two attributes. The two clusters were artificially created so the separation is clear.

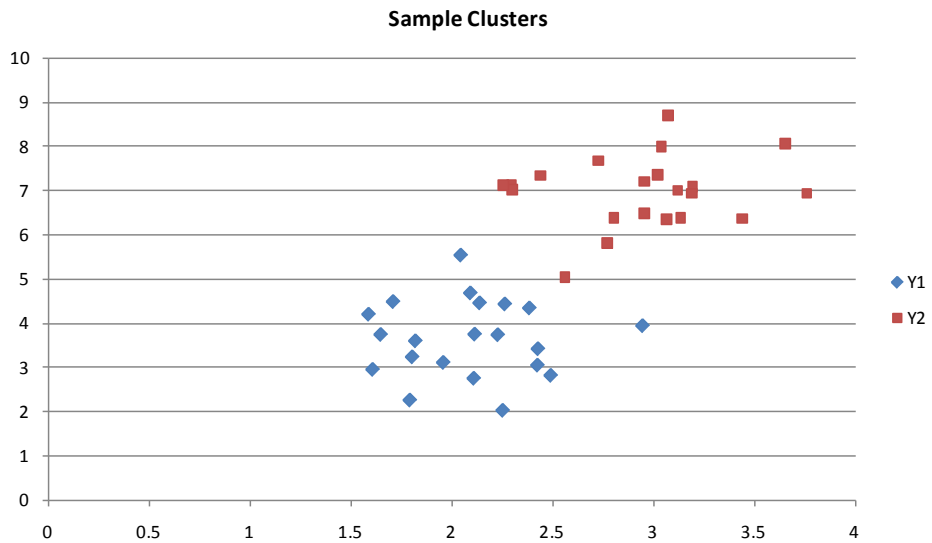


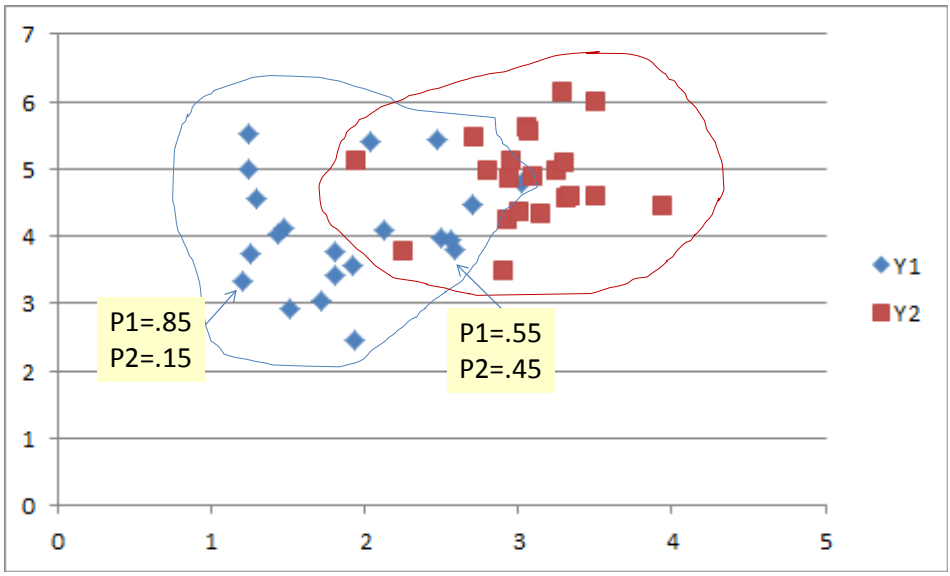
Figure 5.1 shows an example of a simple problem with two attributes and two clusters. The data values were deliberately created to ensure a clean separation of the clusters. Most real-world problems are less clear and the lines between clusters are not always clear. Consequently, different measurement methods can lead to different definitions of the clusters. The choice of method used to estimate the clusters also affects the results. With some methods, a point can be associated with more than one cluster. The association is defined in terms of a probability level. These statistics-based methods lead to grayer definitions, but the results are often more realistic. In many cases, people or items will fall into middle-of-the-road categories and be defined in terms of characteristics of multiple groups.

Figure 5.2 shows a more complex and more realistic version of two clusters. Clusters often contain significant overlap. Some clustering tools will force a point to belong to exactly one cluster. Other methods assign a probability to each point for each cluster. Points that are highly separated will have a higher probability of belonging to a single cluster (say 85 percent). Points that are within the overlap area will have almost equal probabilities (say 55 percent) of belonging to a cluster. In some ways, this approach is more realistic. Think about asking a person which group he or she prefers. Many people will classify themselves as being in the "middle," where they feel the ability to select either group depending on other circumstances.

Most of the clustering tools use automated **unsupervised learning** to obtain the results, so the process of performing clustering analysis is relatively straightforward. However, interpreting the results can be challenging. Also, different

Figure 5.2

Overlapping clusters. Each point has a probability of belonging to each cluster. The sample numbers are for demonstration, they were not computed. But points that are more highly separated will have a higher probability of belonging to one cluster. Points with more overlap will have more-equal probabilities.



tools, diverse measures, and the choice of attributes can all lead to different cluster definitions. Another potential problem is that some of the tools require long processing times as the number of dimensions (attributes) increases. Analysts and business managers need to focus on the ultimate question of whether the resulting clusters are useful and provide meaningful information that can be used to increase profits. Often, the analysis requires exploration and experimentation with various attribute combinations and methods until the results make sense and provide valuable information.

Business Situation

Why would a business use clustering analysis? Clustering is useful for any situation where objects or events are defined in terms of attributes and there are useful reasons to identify groups of objects by those attributes. Certainly, customer groups are the most common example in business. But customers are not the only useful application. Any collection of people or companies would be a candidate for clustering analysis. Many attributes exist on employees that could provide useful classifications. For example, employee evaluations, job training, team memberships, specific skills, years of experience, and so on might be used to cluster or categorize employees. How those clusters are used depends on the situation—from promotions to salaries to teamwork assignments. Similarly, suppliers can be evaluated on attributes of timeliness, quality, pricing, and associated measures. Of course, clustering is valuable only if there are many members in the original group. Also, as you will see, sometimes clustering produces only a small number of obvious groups.

Beyond people, clustering can also be useful in various aspects of production. For example, products might be grouped already into categories, but are those the correct categories? Perhaps products have changed over time and some items were simply thrown into categories and production runs based on a hasty decision that was convenient or made sense at the time. Cluster analysis based on item attributes can more precisely define which products are similar and dissimilar. Likewise, entire production lines and factories could be evaluated and grouped according to various measures, such as speed, quality, and cost. Remember that the key to obtaining useful results is to use attributes that match the goals of the problem.

Every area in business can define objects in terms of attributes and benefit from identifying clusters. Clusters are used to evaluate financial investments using common attributes such as term, risk, and return (or price). Accountants might group fixed assets or other cost structures. The legal department might look for clusters of cases. Beyond customers, marketing might examine customer complaints or problem reports on products. MIS could find groups of similar users to identify software and support needs. The technique can also be used to evaluate projects or even security threats.

Keep in mind that clustering has also been used successfully in several science disciplines. If the organization conducts research or production that requires scientific observations, clustering can be a powerful tool for solving problems.

Clustering has other important uses in data mining. Many tools encounter difficulties in estimating models when the number of attributes (dimensions) is too large. Clustering is a useful method for identifying which collections of attributes are the most important. The problem is simplified by creating a small number of clusters that identify different groups. These clusters can be used as dimensions—

instead of the dozens or hundreds of raw attribute values. As a side note, clustering is also used in some data-compression systems. By finding similar clusters of pixels, images can be reduced in size by storing the cluster data instead of the raw data. These methods are lossy and the quality degrades somewhat but clustering results in substantial space savings.

Model

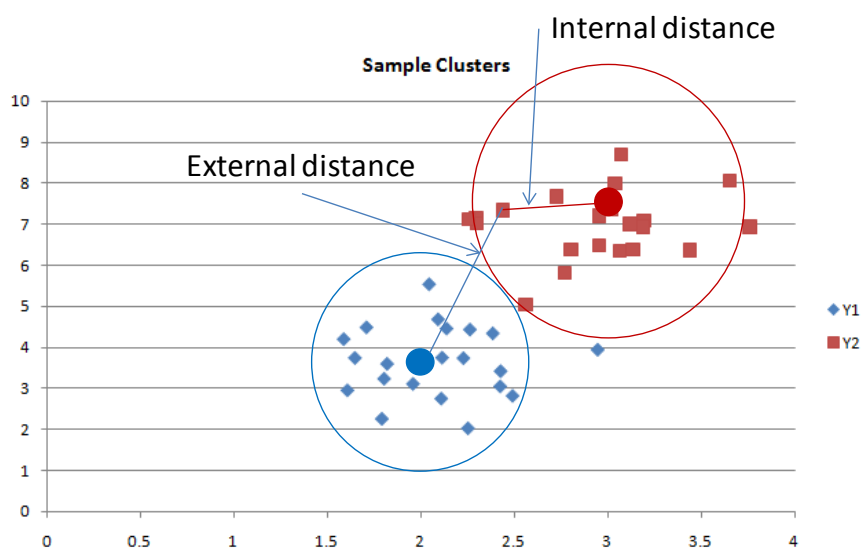
How does clustering work and how are the results interpreted?

The question of how clustering works is somewhat complicated because several versions of clustering tools exist. The differences are significant not just variations in search algorithms. The overall goal is the same: Find items that fall into groups with similar attributes. But the fundamental differences influence the interpretation of the results. Additionally, the definition used to measure distance can vary even within the same tool, changing the results.

Three types of clustering tools are examined in this chapter: (1) Combinatorial, exemplified by the K-means algorithm, (2) Statistical exemplified by the EM mixture algorithm, and (3) Hierarchical. Hierarchical clustering can be applied to most methods but it deserves special consideration. Other specialized clustering methods exist, including the patient rule induction method (PRIM) which seeks groups with the highest frequencies, and statistical models such as factor analysis and principal components.

Figure 5.3

The importance of distance in clustering. A point is placed into a cluster when the attribute distance with the center of the cluster is small relative to the distance to other clusters.



Distance or Dissimilarities

Distance is a key element in clustering. The definition alone indicates the importance—an item is placed into a cluster where it is similar to other items in that cluster and dissimilar to items in other clusters. Figure 5.3 shows the basic concept. Assume two potential clusters exist and the system needs to determine where to place a new point. Compute the distance from that point to the center of each cluster. The point should fall into the closest cluster. The processes for identifying the number of potential clusters and where they should be centered are different questions that need to be answered. But, the concept of distance measures on attributes is critical to all clustering methods.

Measuring distance is a classic issue in mathematics and statistics. Clustering complicates the problem in a couple of areas. One important area is that distance needs to be measured across multiple attributes. The standard approach is to treat the attributes independently and simply sum the difference measures. With a total of p attributes, the distance from one point (x_i) to another (x_k) is the sum of the distances of each attribute:

$$D(x_i, x_k) = \sum_{j=1}^{p} dj(x_{i,j}, x_{k,j})$$

Some systems weight the attributes (multiply by w_j), and the weights often add to one. In most cases, it is better to avoid using weights. The goal is to find the natural clusters and applying weights distorts the value of the individual attributes, hiding natural differences. Of course, it is still necessary to define the distance measure for each attribute value (d_j). The most common approach is to use a **Euclidean** measure—square the difference of the values:

$$d_j(x_{i,j}, x_{k,j}) = (x_{i,j} - x_{k,j})^2$$

The squared-difference is used because (1) it ensures values are always positive, and (2) it is mathematically convenient because it is differentiable and the derivative is continuous. On the other hand, large differences are treated more importantly than small differences. Because of the exponential (squared) term, large differences quickly dominate the results. This approach might cause conceptual problems in some applications. It is possible for one or two observations to dominate and define entire clusters. Consequently, more recent tools offer the use of absolute value to define the difference:

$$d_j(x_{i,j}, x_{k,j}) = abs(x_{i,j} - x_{k,j})$$

With absolute value, the measure is always positive, and larger differences are still important. However, larger differences are not exponentially more important. An outlier can still influence the location of a cluster, but the effect is not as dramatic. Should all systems switch to using absolute value? The major drawback to absolute value is that it is more difficult to handle mathematically, so the computations and cluster search are slower than with the Euclidean definition. Yes, computers are fast today, but clustering problems can require huge amounts of processing time as the number of clusters and number of observations increase. And, for many problems, the effect of outliers is relatively weak. Still, if a tool provides the option to measure in linear instead of squared differences, it is worth compar-

ing the results. If the clusters change substantially, the next step is to investigate the outliers to see which measure handles them the best for the specific problem.

Several other measures have been proposed, and some systems provide choices of five or more distance measures. It can be difficult to choose among them. Unless there is a specific objective to the search that closely matches a different measure, it is best to stick with the traditional squared differences, linear/absolute differences, or possibly the **correlation coefficient**:

$$\rho(x_i, x_k) = \frac{\sum_j \left(x_{ij} - \bar{x}_i\right)(x_{kj} - \bar{x}_k)}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{kj} - \bar{x}_k)^2}}$$

The means in the formula are computed across the attributes for a single observation. The correlation coefficient incorporates interaction effects among the attributes through the multiplication in the numerator. When the attributes are not independent but combine to produce unique differences, the correlation coefficient is a more accurate measure of the distance between observations.

Ordinal Attributes

The basic distance definitions assume that the attribute data is continuous. The standard distance measures only work with real-valued attributes. Yet, business objects often include discrete attributes. It is possible to define distance measures in these cases, but there are some drawbacks. One type of attribute that is straightforward is an **ordinal measure**: 1, 2, 3, and so on. Perhaps items are ranked or a survey response consists of 5-levels, such as the Likert scale: 1=strongly agree, 2=agree, 3=neither agree nor disagree, 4=disagree, and 5=strongly disagree. These attributes arise relatively often and can be converted to continuous measures by dividing by the highest value:

$$\frac{i - 1/2}{M}, i = 1, 2, \ldots, M$$

The one-half term is a standard correction to center the interval. However, this particular conversion does not seem to be popular in terms of availability. Some systems might simply interpret the raw data as continuous or treat it as categorical data. Yet, if the data is known to be an ordinal measure, this transformation is the best approach. To ensure it is applied, you might have to build a query and compute the new column using SQL.

Categorical Attributes

How is it possible to measure distances between **categorical attribute** values such as "Male" and "Female?" Perhaps for gender, the system could use the physical distance between Venus and Mars, but that number would be large compared to other attributes, so gender would dominate any combined measure. The example seems facetious, but it highlights the problem with categorical data—any measurement system is arbitrary. The key is to create one that adds the least amount of distortion. Categorical attributes are often called **nominal** dimensions because the distance values are assigned arbitrarily.

The simplest approach is to categorical data is to define a square matrix that lists all possible values of the categorical attribute. Values on the diagonal rep-

Distance	Female	Male
Female	0	1
Male	1	0

Distance	Dept A	Dept B	Dept C	Dept D
Dept A	0	1	1	1
Dept B	1	0	1	1
Dept C	1	1	0	1
Dept D	1	1	1	0

Figure 5.4

Distance tables for categorical data. When attribute values are equal the distance is zero. Off the diagonal, the distance is one. This choice works well for two attributes. For larger value sets it might be possible to assign different weights but they would be based on subjective values.

resent matching attribute values, so the distance is zero. Values off the diagonal could have different weights, but the most neutral is to assign a value of one to each difference. Figure 5.4 shows two examples. The gender matrix with only two values is the simplest. Choosing 1 for the off-diagonal distance does not affect the results because it could be scaled without affecting any interrelationships. The larger problem with four departments could be more complicated. Choosing 1 for every off-diagonal distance is the most neutral, but it might not be accurate. It is conceivable that the various departments are highly dissimilar. Perhaps C and D are more similar than C and A. But, the computer has no way of knowing these relationships. If the differences are important to the problem, the appropriate paired-distances could be altered to reflect the dissimilarities; but these changes are subjective and need to be made manually. If a particular tool does not support entering the matrix values by hand, the same effect can be created by assigning numbers to the categories and basing the cluster analysis on the numbers instead of the categories. This process effectively converts the categories directly into continuous data that reflect the subjective distances.

The point is that analysts and managers need to think about the data attributes-particularly for categorical data. If each category is roughly equal to the others, the default distance measures used by the clustering tools will suffice. If managers are aware of additional information that differentiates the values, the category should be converted into numerical data that reflect these valuations.

Combinatorial Searches with K-Means

Combinatorial search methods begin with a target of finding K clusters, hence the reason for calling it **K-means**. The mean is the center of each cluster. The goal is to find the best way to split the data to assign each point to exactly one cluster. In one sense, the search method attempts to compare all possible combinations of points into each cluster to determine the best groupings. Of course, with any reasonable-sized problem, it is impossible to test every possible combination. Hence, the algorithms find shortcuts that reduce the number of comparisons. Figure 5.5 illustrates the basic result. The objective is to minimize the total within-cluster

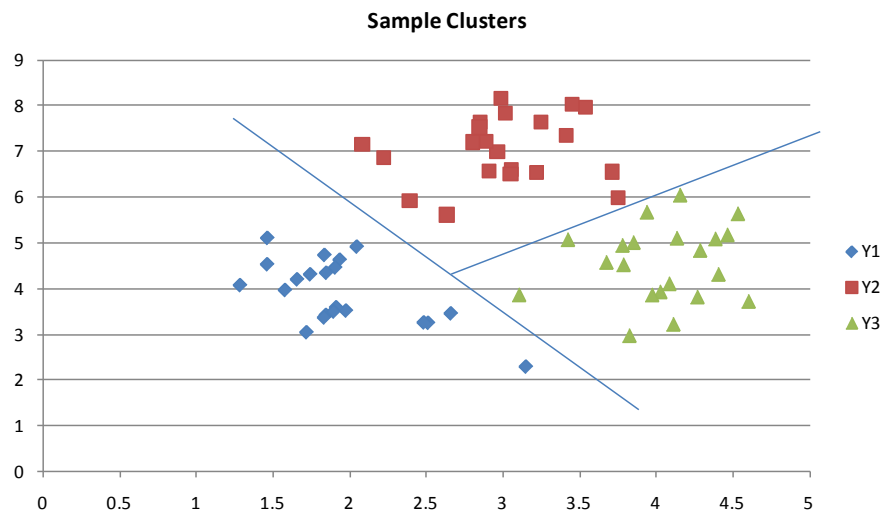


Figure 5.5

K-means goal. Begin with number of clusters (K) and assign each point to one cluster based on distance measures. Minimize the within-cluster distances.

distances. The Euclidean squared distance measure is most commonly used. The algorithm is iterative and begins with the number of desired clusters. The system is initialized with K prospective clusters—typically defined in terms of the means or central points. From the starting point the routine follows two basic steps:

1. For given cluster assignments, minimize the total cluster variance to find the central point or mean with respect to all *p* attributes.
2. Given the current set of means, assign each of N points to a cluster so that the within-cluster Euclidean distance is minimized.

The process repeats the steps until the points no longer shift to new clusters. The purpose of listing the steps in this chapter is to highlight a few important elements of the process. The most obvious question is how to set the number of clusters (K). A related question is how to set the starting point for each of the clusters. A much subtler issue is that the search method cannot guarantee that the best set of clusters is found. The process finds a local optimum, but, particularly with many attributes, the local optimum can be considerably different from the global optimum.

The question of setting the number of clusters (K) has a couple of answers. Sometimes there is a business reason for choosing a specific value. Perhaps managers already believe a certain number of clusters exist. Or, in the classic case, there are K salespeople and clustering is used to identify K groups of similar customers so that each salesperson deals with a similar group of clients. Alternatively, a heuristic method is often used to automatically select the number of means. The system starts with K=1 and computes a measure of the within-cluster distances. It then tests for K=2, up to Kmax. The within-cluster distances will decrease as K increases. In the extreme situation, if K=N (the number of observations) then

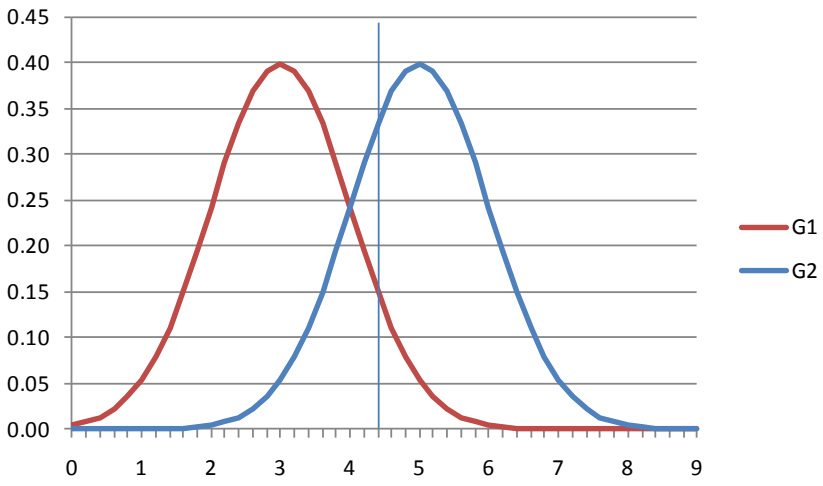
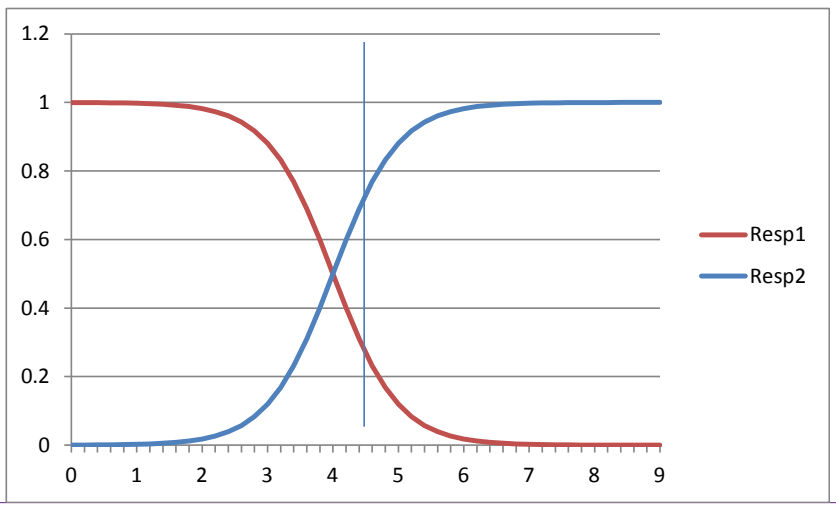


Figure 5.6

Statistical model of clusters. A simple example on one dimension with two potential clusters. The two clusters have different means but the same standard deviation. Consider a point to be classified sitting at 4.2 as shown by the vertical line.

Figure 5.7

Responsibility functions. They are the relative density functions: g0/(g0+g1). It is now possible to read the responsibilities of each cluster for the specified point (0.31 and 0.69).



every point belongs to a separate cluster and the distance is zero. However, somewhere along the way from 1 to Kmax, the system should hit the "natural" number of clusters. Moving beyond this point simply subdivides the natural clusters and there is little gain. Plotting the distance total on a chart against K should reveal a kink at the point where the natural number of clusters is exceeded. Tools can automate this process by computing a **gap statistic** that numerically identifies the break point and selects K. It is not a perfect measure, but it does provide an automated solution.

The question of starting points also lacks a definitive answer. Some systems randomly choose starting points. Others begin with one cluster point and add a new one by minimizing the distance measure assuming the other points are fixed. This process repeats until all K initial points are set. In either case, the choice can be automated, and is rarely altered by the analyst. Additionally, some tools experiment with different starting points. The goal is to find solutions from a variety of starting points. By running the algorithm from multiple starting points, it has a better chance of finding a global optimum.

Overall, K-means is a popular method for identifying clusters. It can be automated and run with minimal help from the analyst. The analyst chooses attributes, possibly redefining categorical variables to incorporate subjective knowledge. A key point is that the K-means algorithm always assigns each point to a single cluster. These assignments are based purely on minimizing the variation within each cluster, and the algorithm assigns observations by comparing various combinations.

Statistical Mixture Model with EM

The K-means algorithm assigns points to a cluster by attempting to find the cluster that fits best with each point. A different way to approach the problem is to assume that the underlying population consists of unseen clusters that follow some probability distribution. Each cluster has its own distribution, and any observed item must have come from some combination of these distributions. Essentially, the method assigns a probability to each point for belonging to each cluster. The distributions still evaluate distance measures on the attributes. A **mixture model** defines a linear combination of the probability functions, where the density functions are combined with weighted averages and the weights sum to one:

$$f(x) = \sum_{i=1}^{K} \alpha_i G(x, \mu_i, \sigma_i)$$

Typically, the distributions are assumed to be Gaussian (normal), and each cluster can have a different mean and standard deviation. To understand how this process differs from K-means and how it affects the results, it is easiest to diagram a simple problem with a single dimension (attribute) and two possible clusters.

Figure 5.6 shows the distributions for two clusters. The distributions are both Gaussian and have the same standard deviation (1.0) but different means or centers. The point to be classified is at 4.4 as shown by the vertical line. This point appears to be most strongly within the second cluster, but there is still a fairly high probability that it belongs to the first group. Instead of assuming that the point must belong to only one group, the mixture model defines a mixture parameter that indicates the association with both clusters. The main purpose of the method is to identify the value of this mixture parameter for each point and use these values to determine the means and standard deviations of each cluster distribution.

Start with initial mixture value ($\pi = 0.5$), and initial distribution parameters for each cluster.

$\hat{\gamma}_i = \frac{\hat{\pi} G(\hat{\mu}_2, \hat{\sigma}_2, y_i)}{(1-\hat{\pi}) G(\hat{\mu}_1, \hat{\sigma}_1, y_i) + \hat{\pi} G(\hat{\mu}_2, \hat{\sigma}_2, y_i)}$	Compute responsibilities for each observation and compute parameter.
$\hat{\mu}_1 = \frac{\sum_{i=1}^{N} (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^{N} (1-\hat{\gamma}_i)}$	Use parameters to estimate mean and variance for cluster distributions, and new estimate of mixture value.
$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^{N} (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^{N} (1-\hat{\gamma}_i)}$	Repeat until parameters are stable.
$\hat{\mu}_2 = \frac{\sum_{i=1}^{N} (\hat{\gamma}_i) y_i}{\sum_{i=1}^{N} (\hat{\gamma}_i)}$	
$\hat{\sigma}_2^2 = \frac{\sum_{i=1}^{N} (\hat{\gamma}_i)(y_i - \hat{\mu}_2)^2}{\sum_{i=1}^{N} (\hat{\gamma}_i)}$	$\hat{\pi} = \sum_{i=1}^{N} \hat{\gamma}_i / N$

Figure 5.8

EM Algorithm for two clusters. Step one uses expectations to compute responsibilities. Step two, maximization, computes the mean and standard deviation of the cluster distributions, along with the mixing parameter.

Instead of working with the probability density functions, the methods define responsibility functions. In the simplest form, **responsibilities** are the relative density functions: $g_0/(g_0+g_1)$ and $g_1/(g_0+g_1)$. Figure 5.7 shows the responsibility functions for the small example. When x=4.4, the responsibilities are 0.31 and 0.69. If the observed point falls further to the right, the responsibility for cluster 2 increases to 1, and the point would be classified as completely within the second cluster.

The **expectation maximization (EM)** algorithm uses this statistical foundation to determine the means and standard deviations of each cluster and to assign points to the clusters. Figure 5.8 outlines the basic steps for a two-cluster model. Similar to the K-means algorithm, EM begins with starting values of the means and standard deviations of the clusters. It also needs a starting value for the mixture parameter for each point, but this value is commonly set to 0.5 as an unbiased starting point. Once initialized, the responsibilities are computed for every observation at the expectation stage. These values then update the means and standard deviations for the clusters, and define the overall mixture parameter. The process repeats until the estimates stop changing. The mathematics and optimization are more complex with more than two clusters, but the concepts are the same.

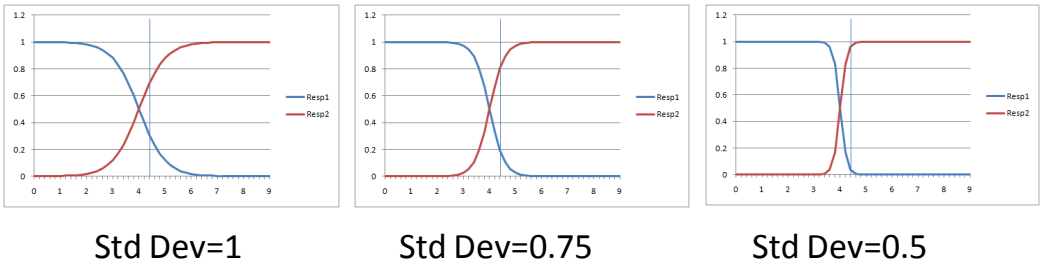


Figure 5.9

The effect of standard deviation on responsibilities. As standard deviation approaches zero, the separation of the responsibilities increases and each point is associated more closely with a single cluster.

There is no reason to memorize the equations, but they highlight the role of statistics in the process. In particular, note that the clusters are defined by means and standard deviations. The standard deviations play an important role in understanding the results. Figure 5.9 shows the one-dimensional example with three levels of standard deviation, which is set to the same level in both distributions. As the standard deviation approaches zero, the responsibilities become more rectangular and each observation is associated more closely with a single cluster. Conversely, when the standard deviation is high, the responsibility values approach 0.5. At zero, the EM method is essentially equivalent to the K-means approach and each observation is assigned to a single cluster.

When interpreting the results, the primary difference with the EM approach is that observations can be associated with more than one cluster. It is a softer association than with K-means. But, unless you look at individual observations, this effect is somewhat hidden. A more obvious difference is that clusters are defined by means and standard deviations, which are reported for attributes with continuous measures. Figure 5.10 shows a portion of the results from applying Microsoft's Clustering EM tool on the Corner Med patient attributes. One cluster definition is highlighted in terms of the Age attribute. Notice the use of the mean and standard deviation. Also, notice that Gender seems balanced across the clusters, so Gender is probably a weak classifier. This issue is covered in more detail in the Microsoft results section.

The EM algorithm faces the same basic issues as the K-means algorithm. The number of clusters must be determined in advance. The starting means and standard deviations of the clusters must be specified. Because of this second issue, EM converges to a local optimum, which might not be the best global solution. Most algorithms use processes similar to those for K-means for choosing starting points. Testing various starting points can also be used to help find better solutions.

Hierarchical Clusters

The question of number of clusters or means can be difficult to solve. Each tool uses different methods, so the final results are different depending on the tool. If necessary, the results can be made similar by forcing tools to use a specific number of clusters, but it does not solve the problem of determining the best number

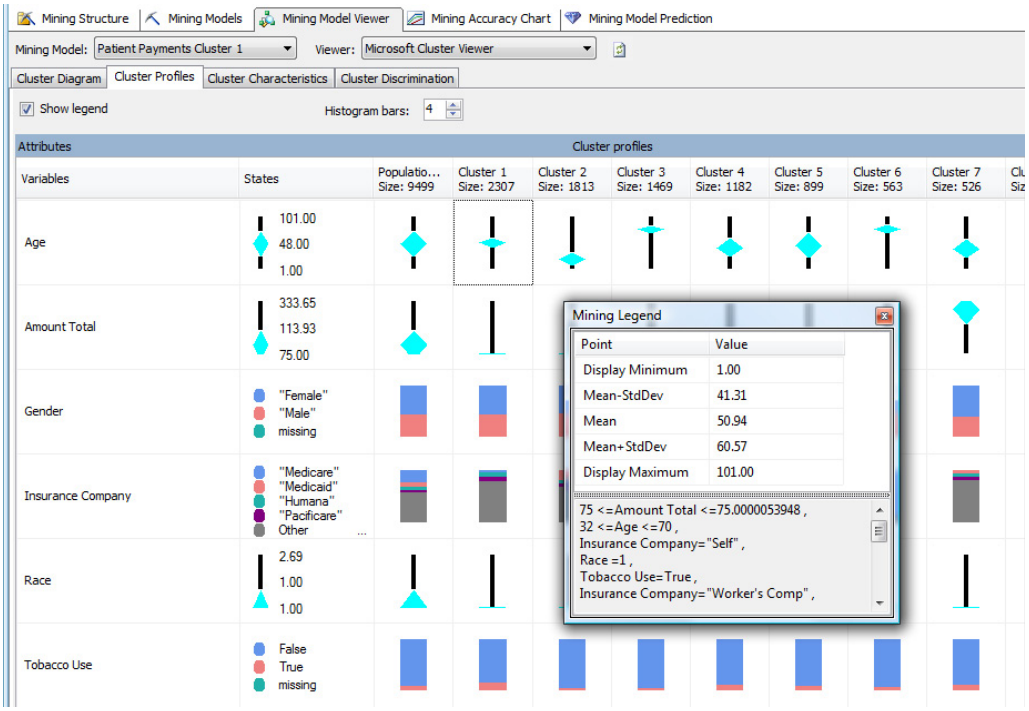


Figure 5.10

Sample EM results from Microsoft Clustering on Corner Med patient attributes. Note the mean and standard deviation report for the continuous Age variable.

of clusters. One potential answer is to use hierarchical clustering. With **hierarchical clustering**, the tools start at the top (all data in one cluster) and work down; or they start at the bottom (each observation is a separate cluster) and work up. Either way, the tools create a collection of clusters for all possible values of K.

Several variations exist of both hierarchical clustering methods. Consider the top-down approach first. Top-down hierarchical clustering is usually called **divisive** because the top cluster contains all of the data observations. At each level, the algorithm examines each existing cluster and divides it into two new clusters. The most interesting approach looks at the cluster to be split and finds the one observation that is the most dissimilar or whose average distance is farthest from the center. This observation becomes the center of the new cluster. The process then looks among the remaining elements to compute the average distance from the original cluster to the new cluster. The element with the greatest net average distance is moved to the new cluster. The process repeats until net gains no longer exist. Divisive hierarchical clustering can continue to the end where each observation is in a separate cluster, or it can be cut off early once a specified number of clusters has been reached. For this reason, divisive clustering is useful when it is important to hold the total number of clusters to a relatively small number. Analysts could choose the maximum number of clusters, or they could examine the re-

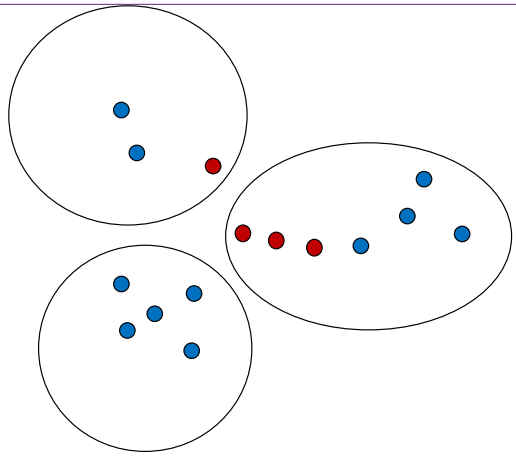


Figure 5.11

Single linkage problem is large-diameter clusters. Some items within a cluster are not very close to the others. Problems arise because only the smallest distance is used and items can chain their way into an inappropriate cluster.

sults at each step and decide if the gains from one more split are important enough to justify a more complex result. The technique is often used in data compression systems, where the goal is to obtain a close approximation to the data with as little storage (fewer clusters) as possible.

Hierarchical clusters can also be built from the bottom—starting with each observation in its own cluster. At each step, the **agglomerative** algorithm finds the two most similar clusters and combines them to form a new cluster at a higher level. The trick is to find the "most similar" clusters. This step requires a distance measure of dissimilarity between clusters. Earlier measures were defined between a single point and a cluster. Agglomeration can use the fundamental distance measures but it needs to apply to each point in the two clusters. Several measures have been defined, including single linkage, complete linkage, and group average. All methods compare each point in the first group (G) to each point in the second group (H), so the distances are pair-wise measures. The average simply computes each pair-wise distance, adds them up, and divides by the total number of pairs ($N_g N_h$).

Single linkage selects the smallest distance between any pairs (nearest neighbor): min d(c1, c2). The single linkage, smallest distance, measure tends to create clusters with large diameters. A point needs to be close only to one other point in the cluster. Figure 5.11 shows that points can enter a cluster by chaining onto a near neighbor. Once one point is in the cluster, then another close one enters, and the rest fall as part of the chain. But, on average, the points at the end of the chain can be considerably different from the other points at the far end of the cluster.

Complete linkage selects the largest distance between any pairs (furthest neighbor). If the data clusters are well separated, all three measures should produce about the same results.

Complete linkage tends to produce clusters with small diameters, and can exclude observations that are "close" in average distance. The points within a cluster are all closely related, but the measure tends to exclude points that are also close

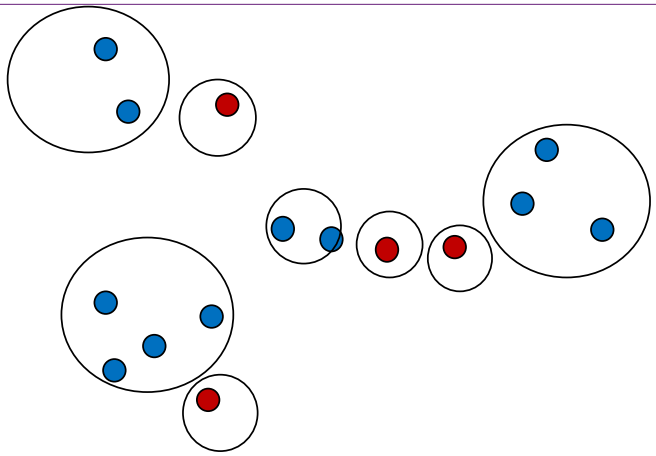


Figure 5.12

Complete linkage builds clusters based on the largest difference between two points. It tends to produce many small-diameter clusters and excludes points that are close on average.

on average and should probably be included. Figure 5.12 shows the problem. The points within the clusters are all close, but some nearby items are being excluded because the distance to one point within the cluster is too large. Ultimately, as the agglomeration continues, the points will be incorporated into the clusters, but the final cluster definitions can be different from those generated by other methods.

On balance, the group average measure presents a balance between the two extremes. It tends to produce relatively compact clusters with relatively good inclusion of close items. It also has nice mathematical properties that relate well to the statistical definition of clusters. But it is not a perfect measure. Because all attributes are numerically averaged, the distance measure is subject to the scale of the numbers. For example, one attribute might be measured in relatively small values from 0 through 100. A second attribute (such as income) could be measured in tens of thousands. This higher scale is going to dominate the effect of the other attributes. And, if the scale is changed (convert the income to thousands), the distance measure changes and the resulting clusters will be different.

Why should you care about the different methods? The answer is that the tools that support these methods have options. As an analyst, you have to select the options that most closely match the problem being investigated.

As shown in Figure 5.13, hierarchical clusters are sometimes displayed on dendrograms. A **dendrogram** is a compressed display of the clusters created at each level of the hierarchical clustering process. The height of each node (cluster) is proportional to the dissimilarity of its children. In the example, the dissimilarity between C and D is larger than it is between clusters A and B. Large dissimilarities are generally good because they indicate the need for different clusters. If dissimilarities are low, there is little need to split into new clusters. The dendrogram presents an interesting picture of how the clustering behaves at each level. However, keep in mind that the actual clusters, and the dissimilarities, are highly dependent on the distance measure chosen. Hence, the dendrogram is not a picture of the underlying data—merely a picture of the clustering choices. Some systems

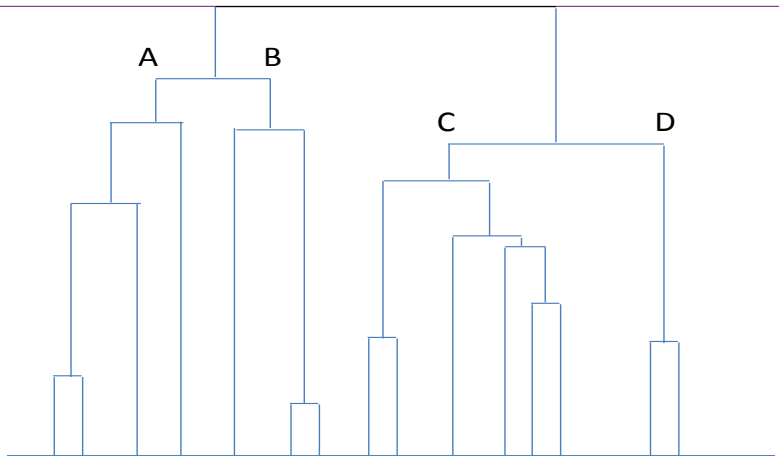


Figure 5.13

Dendrogram example. The diagram shows all of the clusters created at each level. The node's height is proportional to the dissimilarity between the node's children. So C and D are more dissimilar than A and B.

always draw the ending lines to the bottom of the chart. Some provide color capabilities to highlight higher-level clusters. In the example, everything below cluster C could be one color and below D would be a second color—indicating two of the final clusters chosen.

Other Statistical Methods

One of the main purposes of clustering is to reduce the number of dimensions or even the number of data points in a problem. Several other tools perform related tasks that can also be used to reduce the dimensions of a large problem. Many of them were designed for specific tasks, such as latent variables, exploratory projection pursuit, self-organizing maps, and multidimensional scaling. These techniques are useful for some data reduction problems but are not covered in this book. However, one older statistical technique is commonly used in data mining tools. **Principal components analysis (PCA)** is a technique used to reduce the number of dimensions or attributes in a problem. The method searches for a smaller number of components that are linear combinations of the existing data that can approximately match the original data. It is particularly useful when the attributes have a high degree of **multicollinearity**. When attributes are closely related, most statistical methods have difficulties separating the effects of the variables and often cannot find a good solution. PCA defines the internal correlations and uses them to create a smaller number of variables that define the same output data.

Figure 5.14 shows a small example with attributes X1 and X2. The first principal component is a vector that identifies the direction that contains the highest amount of variation. It is shown as the longer line in the diagram. The second component is **orthogonal** (perpendicular in two dimensions) to the first vector. It is substantially shorter than the first component because the data points have less variation in that direction. This set of data could possibly be reduced to a single dimension by projecting all of the points onto the first principal component. It

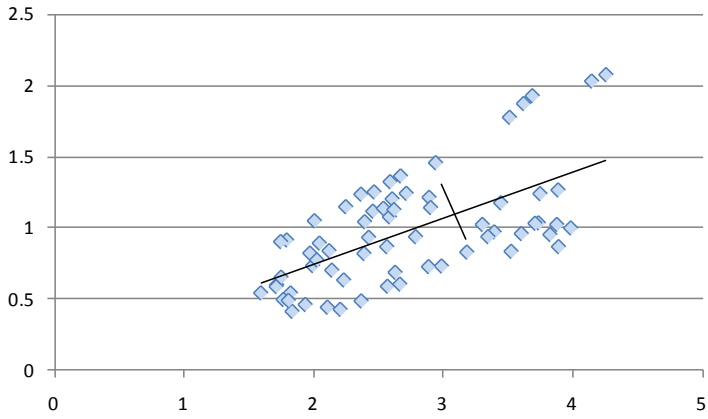


Figure 5.14

Sample principal components in two dimensions. The first component identifies the direction of most variance (X1). The second is smaller and orthogonal.

would incorporate variations from both the X1 and X2 dimensions (axes). The results would be an imperfect representation of the data. Some of the variation would be lost. The amount of information lost is visually seen by the length of the shorter line. If it is relatively short, less information is lost by projecting the points onto the longer line. With each problem, the analyst needs to choose how much variation should be lost. No fixed answer exists, but the tools do provide information to help make the decision.

The mathematics behind PCA are straightforward but require knowledge of linear algebra (matrices). It is probably unnecessary for business analysts to understand all of the mathematics, but some of the commonly-used terms come from linear algebra. One way to approach the problem is to define the matrix **X** which contains the N rows of data consisting of p columns of attributes. These values are centered for each attribute by subtracting the mean from each observation. An efficient method for finding the principal component vectors is to decompose the X matrix into products of three new, specially-constructed matrices:

$$\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V'}$$

Matrix **U** is an N x p matrix such that $\mathbf{U'}\,\mathbf{U} = \mathbf{I}_p$, the identity matrix. Similarly, V is a p x p matrix where $\mathbf{V'V} = \mathbf{I}_p$. **D** is a diagonal matrix where all elements are zero, except those on the main diagonal, which are sorted in descending order. This factorization of **X** is known as the singular value decomposition (SVD), and it is a commonly-used tool in statistics and numerical analysis. Numerical analysis is the branch of mathematics that deals with using the computer to solve mathematical problems. The point of the equation is that the columns of **UD** are the principal components of **X**. SVD is particularly useful for finding the coefficients in a linear regression model. Consequently, efficient computer algorithms exist for finding the principal components.

Many data mining and most statistical packages have the ability to perform principal components analysis. It is not necessary to know the underlying mathematics. Figure 5.15 shows the results of running PCA on a small sample dataset

eigenvalue	proportion	cumulative	
2.87027	0.57405	0.57405	0.536X3+0.497X5+0.479X2+0.475X1+0.101X4
1.35739	0.27148	0.84553	0.826X4+0.414X5-0.337X1-0.137X3-0.116X2
0.44212	0.08842	0.93396	-0.824X2+0.459X3+0.285X5-0.163X4+0.05 X1
0.29819	0.05964	0.99359	0.811X1-0.391X3+0.301X4-0.278X2-0.147X5

Figure 5.15

Sample results for PCA with five variables (X1-X5). The eigenvalues are used to choose the number of vectors to keep. Every component with an eigenvalue above one needs to be kept. Other components might be useful. Check the cumulative proportion of variation explained. Keeping all four would account for almost 99.4 percent of the variation, so the reduction of one dimension would yield very little loss of information.

with five attributes (X1-X5). The system has identified four principal components that are evaluated in terms of the **eigenvalues**. Technically, a principal component is an eigenvector, and each eigenvector has an association eigenvalue. The terms come from linear algebra and arise from a way of rewriting the problem. The eigenvalues are the key because they measure the variation explained by the matching eigenvector. In any problem, the eigenvalues sum to the number of dimensions (5 in the example). Dividing each eigenvalue by the total gives the percentage of variation explained. Eigenvalues are listed in descending order and most tools report the cumulative proportion. Kaiser's criterion (Kaiser 1960) states that components with an eigenvalue greater than one should always be included. This rule would include the first two vectors. From the cumulative column, those two dimensions together explain 84.6 percent of the variation. Some people might consider a 15 percent loss of information too great of a price to pay for reducing a problem from five dimensions down to two. Ultimately, the decision depends on the problem, the analyst, and how critical it is to reduce the number of dimensions. Continuing down the table, including all four components saves one dimension and covers almost 99.4 percent of the variation—a reasonable tradeoff. As a side note, the results fairly accurately depict the data. The columns were generated using two independent basis columns, with some additional randomness added to each of the columns.

Once the number of components has been selected, the vectors shown in Figure 5.16 are used to compute the new data columns. For example, the new variable V1 is computed as: 0.475 X1 + 0.479 X2 + 0.536 X3 + 0.101 X4 + 0.497 X5. The computation is performed for each data row in the original **X** matrix. Data mining tools that implement PCA can usually generate the new data columns automatically. The selected analysis is then performed on the modified (V1…V4) data columns. One difficulty with this approach is that it can be harder to evaluate and understand the results of the final tool. The end results will be expressed in terms of these principal components. To understand the role of the original variables, the analyst has to trace the impacts through these multipliers. For instance, X1 has a strong influence (0.81) on V4, a relatively strong effect on V1 (0.48), a tiny effect on V3 (0.05), and a negative effect through V2 (-0.34). Results expressed in terms of V1, V2, V3, and V4 have to be interpreted carefully.

V1	V2	V3	V4	
0.4752	-0.3374	0.0499	0.8110	X1
0.4788	-0.1161	-0.8245	-0.2783	X2
0.5363	-0.1366	0.4586	-0.3907	X3
0.1007	0.8260	-0.1625	0.3012	X4
0.4971	0.4145	0.2846	-0.1468	X5

Figure 5.16

Principal component vectors. New data columns are created by multiplying each coefficient by the original matching attribute (X1…X4) and adding the terms.

PCA is typically used only if it is required because of huge datasets and problems with too many dimensions. On the other hand, Ding and He (2004) showed that PCA is equivalent to a continuous solution for K-means. The principal component vectors and K-means centers are essentially defining the same concepts. Although, principal components categorizes points in terms of a mixture or percentage instead of forcing each point to a single cluster. The other interesting result of their work is that PCA provides a method to improve K-means searches to help ensure global optima instead of stopping at local points.

Data

What type of data is used in clustering? Clustering relies on a distance measure to determine whether an observation is closer to one cluster instead of others. Distance is more precise when it evaluates continuous variables. Discrete attributes can be used, but the results can be quite different than those from continuous variables. Huge databases can be a problem for some clustering algorithms. The number of observations (N) is an issue, but the number of attributes and clusters is also a problem.

Attributes and Observations

Clustering algorithms expect observations of data to be stored in a table or matrix where the columns represent different attributes or dimensions and each row contains one observation. This structure matches database tables and queries, so data mining tools based on a DBMS are easy to configure for clustering. Other tools usually read data from text files where each new line contains one row of observations. The data within a row are separated by a unique character—usually a comma or a tab character. Figure 5.17 shows a few rows of data from the Corner Med database. It contains some basic attributes about patients that might be used to identify possible groups of customers. Tobacco use is a binary (true/false) variable. Race is a categorical attribute but it is coded numerically. The Age was computed from the date of birth to a specific point in time at the end of the year. It could have been defined as the age at the time service was provided, but that approach runs into problems for patients who visit more than once a year. The insurance company is the name of the organization that is paying for visits. Technically, it could change during a year as well, but for most patients, it will be constant. The amount is the total amount of service billed for that patient over the year. Notice that each row represents one patient. The database contains thousands of patients but only a few are shown here.

PatientID	Age	Race	Gender	Tobacco	Insurer	Amount
1	58	1	Female	False	Cigna	75
2	2	1	Male	False	United Health	75
3	1	3	Male	False	Cigna	75
4	45	1	Female	False	UniversalCare	75
5	35	1	Female	False	Assurant	150

Figure 5.17

Standard data layout. A column represents a single dimension attribute. Each row is one observation, such as a purchase or a customer. Typically, a key column attribute identifies each row.

Data for tables is often stored in simple text files to make it easier to transfer the data to other systems. The **comma-separated values (CSV)** file is one of the most common formats. Most programs and DBMSs can read this format. Files in this format are simple text files, where data for one observation is stored in a single row. The data for each column are usually separated by commas or tab characters.

Continuous and Discrete Data

Whenever possible, it is best to use continuous data for attributes used in clustering. Most tools can handle categorical data, but the distance measures are limited. Typically, distance is measured as either zero or one. Zero distance is defined if two categories match exactly; otherwise the value is set to one. A few variations exist, but there is no automated method to define a more precise measure.

If the managers and analysts know that subjective differences exist among the categories, then the variables should be recoded into numerical data that reflect these differences. In the healthcare example, managers might want to weight the insurance companies differently. In particular, the Medicare and Medicaid governmental programs are often perceived quite differently from private insurers. Specifically, the government programs place tighter caps on fees and pay lower rates to physicians than most private firms pay. Perhaps the reason for including the insurance companies in the analysis is to consider these payment issues. If so, SQL can be used to recode the data. For example:

```
SELECT Visit.InsuranceCompany,
      InsV =
      CASE Visit.InsuranceCompany
            WHEN 'Humana' THEN 10
            WHEN 'Self' THEN 15
            WHEN 'Medicaid' THEN 1
            WHEN 'Medicare' THEN 1
            WHEN 'Blue Cross/Blue Shield' THEN 11
            WHEN 'Charity' THEN 0
            ELSE 9
      END
FROM Visit
```

Each insurance company is assigned a different number that somehow represents the payment history and value from the company. Only a few companies

are listed in the example. The default condition (ELSE) covers the others, but it would be better to include each company separately for most cases. This approach can be used when these subjective valuations exist. If no one knows appropriate values, then the neutral distance of one assigned to categorical data will have to suffice. It would also be possible to create a new table holding the names of the insurance companies and the new values. SQL would be used to join the tables together on the insurance name.

Missing Data

Clustering routines do not work with missing observations. Most tools delete observations (rows) that contain missing data. If the dataset contains only a few isolated missing values, this approach is reasonable. However, if one attribute happens to have many missing values, it would probably be better to drop that attribute from the analysis. Otherwise it will cause many rows of potentially useful data to be discarded.

Some tools provide options to replace missing data with new values. These new values can be a constant, an average, or perhaps an interpolated value computed from the two nearest points. For some problems, replacing missing data with the mean is relatively neutral. But, if a large percentage of the observations are replaced, the results may be altered. Many of the clustering tools automatically fill missing data points with the overall mean.

An interesting situation exists with missing data for categorical variables. In many cases, the missing value can be treated as just another attribute value. For example, Gender might contain: Female, Male, Missing. In some case, this interpretation might be difficult to understand. In those cases, filters or SQL queries can be used to drop the rows with missing data.

Clustering on Products: Cars

How are clusters identified with multiple dimensions? The simple example of clusters in two dimensions is relatively easy to see—as long as the clusters are relatively distinct. If clusters are weak, with considerable overlap, they can be difficult to see even in two dimensions. Now, imagine what happens in three, four, or more dimensions. The problem gets worse with hundreds, thousands, or millions of observations. Even when the mathematical algorithms identify clusters, they can be difficult for analysts and managers to understand. Software vendors try multiple methods to display clustering results, and these tools represent the major differences between software products.

Goals

To illustrate the challenges, this section uses real-world data on automobiles for sale in the U.S. in the 2012 model year. The purpose of this data is to illustrate the basic challenge of interpreting clustering results using a product that has multiple attributes that are likely to be recognizable to most students. Two tools are used to illustrate the process and the results: Microsoft Clustering and the free **Weka** data mining software (http://www.cs.waikato.ac.nz/ml/weka).

The goal is to determine which vehicles are similar and which are different. For instance, a marketing manager of an automobile company might want to know which cars are the closest competitors. A potential customer might ask the same question. More complex questions could also be addressed, but they generally require more data. For example, policymakers and long-term planners might want

ID	Year	Make	Model	Sec	Category	MPG	Price	Wt.	Cyl.	HP	Seat
769	2012	Mitsubishi	i-MiEV	11.9	Hatch	126	29125	2579	0	66	4
781	2012	Nissan	Leaf	7.9	Hatch	106	35200	3385	0	107	5
595	2012	Chevrolet	Volt Hatch	8.53	Hatch	95	39145	3781	0	149	4
839	2012	Toyota	Prius Plug	10.9	Hatch	95	32000	3165	4	98	5
615	2012	Fisker	Karma	5.9	Sedan	52	95900	5300	0	403	4
838	2012	Toyota	Prius	9.7	Hatch	51	23015	3042	4	98	5
648	2012	Honda	Civic Hybd	5.7	Sedan	44	24050	2853	4	93	5
705	2012	Lexus	CT 200h	10.4	Hatch	43	29120	3206	4	98	5
832	2012	Toyota	Camry Hybd	7.2	Sedan	43	25900	3435	4	156	5
630	2012	Ford	Fusion Hybd	8.7	Sedan	41	28775	3720	4	156	5

Figure 5.18

Sample car data. The CarID is random and is used as the key column. Year, Make, Model identify specific vehicles. Category is a human-assigned definition that should not be used in the analysis. The other columns are standard attribute measures. Sec is the number of seconds to reach 60 mph, MPG is miles per gallon in city driving, Price is the list price of the base model, weight is in pounds, Cyl. is the number of engine cylinders, HP is horsepower (bhp), Seats is the number of passenger seats to indicate size. Data was derived from various car Web sites and magazines.

to examine trends in clusters over time—which requires detailed data for multiple years or decades. Similarly, manufacturers would want consumer opinions on similarities and differences, but that information changes the way the question is analyzed and is more suitable for other chapters. From a personal standpoint, you could simply think about the data as an exploration of searching for a new car. The clusters will provide groups of cars that have specific similarities.

Data

The data consists of a simple CSV text file with 311 data rows. Each car is given a simple ID number to serve as the key. Cars are identified by the Year (2009 for all), Make (Toyota, Chevrolet, Ford, and so on), and Model (Prius, Cobalt, Fusion, etc.). Cars are often assigned to predetermined categories (SUV, Compact, Truck, and so on). This Category variable is included, but it should not be used or it might interfere with other clusters. Measures on the cars include: Seconds to 60 mph—a measure of performance, miles per gallon in city driving—reported by the EPA, price—typically the price of a base vehicle, weight—a measure of size in pounds, number of cylinders—which is likely correlated with performance and mileage, horsepower—a measure of performance, and number of seats—a measure of size.

Figure 5.18 shows a small portion of the data file. The data consists of standard measures on cars and was collected from various car Web sites and car magazines. A few vehicles have missing data. For instance, heavy trucks are not required to report gas mileage. Prices are the least useful because the listed price applies only to the base trim-level with few options. Some vendors use the trim levels to nudge cars into different categories—such as putting a more powerful engine in

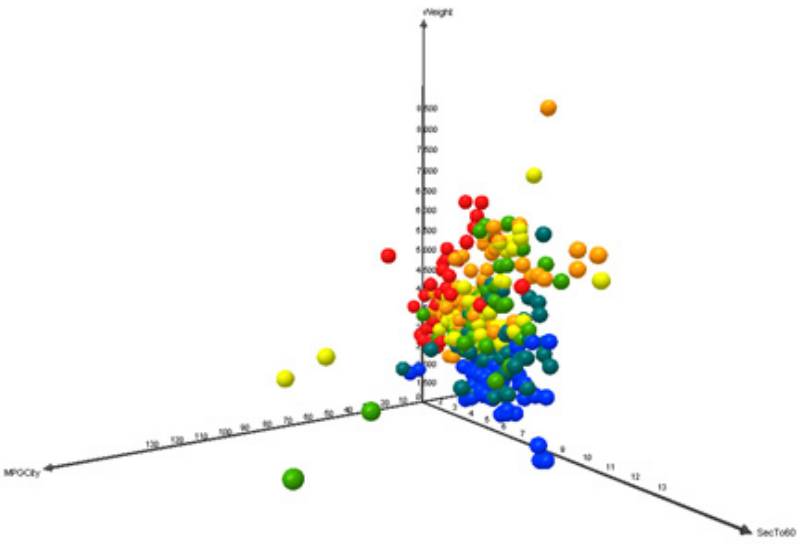


Figure 5.19

Sample car data in 3D plot. The axes are: MPGCity, SecTo60 (acceleration), and Weight. The color coding is assigned based on list price of the vehicle. Miner3D was used to plot the data, which has nice plots but is a relatively expensive tool. With the tool, the chart can be rotated and zoomed interactively to explore the data relationships.

the higher-trim level to improve performance. But, these changes blur the cluster definitions making the results even harder to interpret so they are ignored here.

Microsoft Clustering requires that the data be imported into a database table. A SQL script is available that creates the table and loads the data from the comma-separated values file. Once the data is in the database, a new Analysis Services project can be created using the Visual Studio Business Intelligence template. Add a **Data Source** by right-clicking the Data Sources entry in the Solution Explorer and choose the New Data Sources option. Follow the prompts to select the correct server and database.

Figure 5.19 shows the car data using a 3D plot which shows relationships across three dimensions. Actually, a fourth dimension (price) is displayed by color-coding the items. The tool (Miner3D) is relatively expensive, but it also supports clustering—in which case the colors can be assigned based on cluster values. In the sample data, the price coding is actually a decent clustering mechanism across the other three attributes (weight, acceleration, and MPG).

A tricky element of the data source is specifying the impersonation option. Obtaining data from the DBMS requires permissions on SQL Server to retrieve the data. Microsoft provides several options to set the appropriate level of security. One option is to set up a specific account on the SQL Server computer, or through Kerberos on a company network. Each person who uses the data source will use that account. A weaker but easier option is to use the built-in service account. This option is the simplest for testing, but weakest security for a production database. The third option is to use the Windows account of the user running the applica-

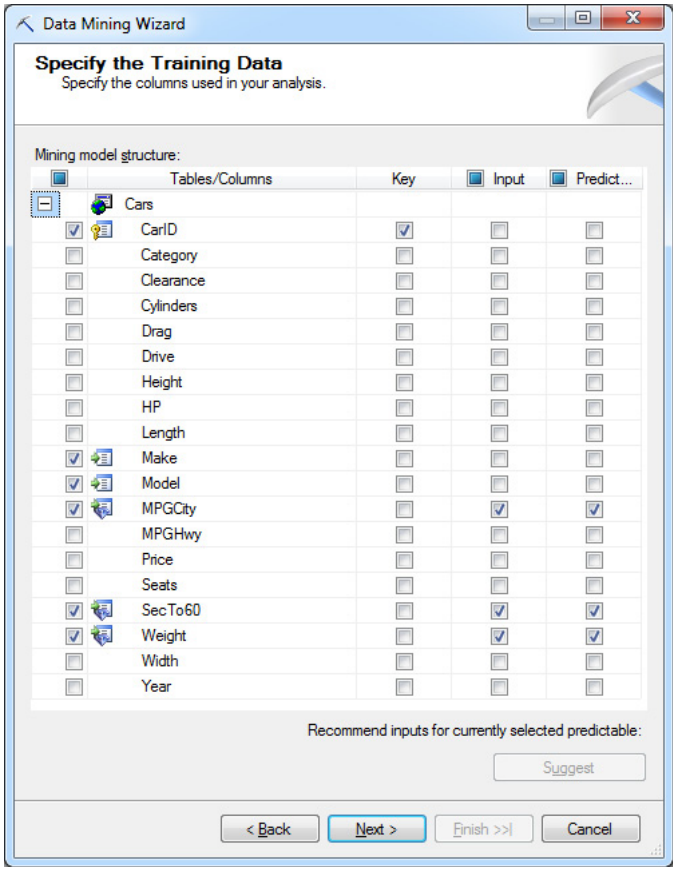


Figure 5.20

Selecting columns for Microsoft Clustering. Include Make and Model but be sure they are not used as Key, Input, or Predictable columns. For now, pick only the three measures: Weight, MPGCity, and SecTo60. Set them as Input and Predictable.

tion. This option enables detailed access control over the data, but requires that the server and client computers be connected through a security system (Windows Active Directory), and that permissions be assigned to each user on the database server. For now, use the service account option but remember to review security if the application is moved into a production environment and shared with multiple users.

The second step is to create a Data Source View with a right-click on the Data Source View entry. A **Data Source View** is typically a collection of tables and named queries. It defines the tables and relationships for data needed for processing. In this case, the data simply consists of the entire Cars table. Note that a Data Source View provides the ability to create named queries, which function similarly to SQL views.

Microsoft Clustering

Microsoft Clustering defaults to the EM algorithm, but it has an option to run K-means clustering. For the first pass, stick with the default EM algorithm—and

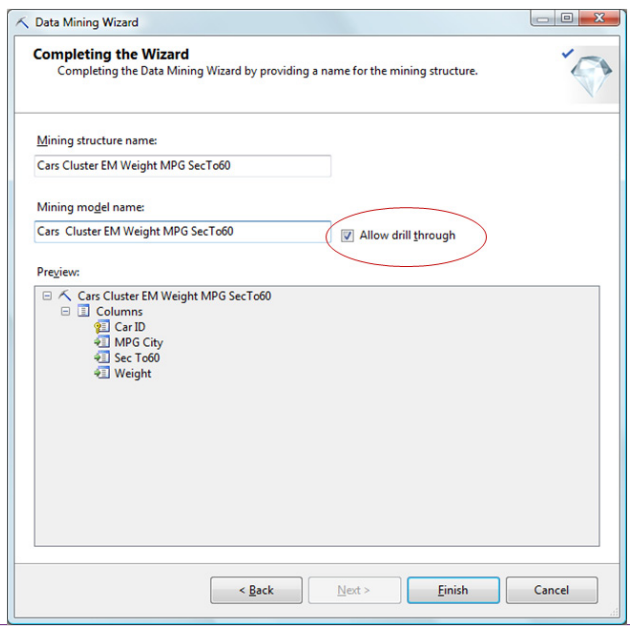
remember that it assigns a mean and standard deviation to each cluster attribute. Points can belong to multiple clusters.

It is straightforward to create the default data mining model. To simplify the initial results, begin with three attributes: MPG, Weight, and SecTo60 time. Right-click the Mining Structures entry in the Solution Explorer and choose the option to add a new mining model. Follow the basic prompts and select Microsoft Clustering as the tool. Use the default data source and Cars table as the Case table. Figure 5.20 shows one of the trickier steps—choosing the columns for the analysis. It should be set by default, but be sure CarID is the only key column. Select the three attribute measures to be used in the clustering as Input and Predictable types: MPGCity, SecTo60, and Weight. Finally, select Make and Model by setting the checkmarks in front of the names. Be careful not to assign them any other roles (Key, Input, or Predictable). They are included here only so that they will be available later for drill-down analysis providing human-recognizable names instead of the meaningless CarID value. For clustering there is often no need to reserve data for testing. With a limited number of observations in this dataset it is best to use all of them. Set the percentage of data for testing to zero.

Figure 5.21 shows the other tricky part of the configuration. Enter a name and description for the analysis that will be unique and recognizable later. Also, be sure to check the box "Allow drill through." Combined with the Make and Model columns, this option makes it possible to see exactly which vehicles fall into each cluster. Without this check box, the results return only the statistical data. It is much easier to understand results—particularly for products—when the actual

Figure 5.21

Setting the drill through option. This option is important for understanding clustering results. It makes it easy to get a list of exactly which products fall into each cluster.



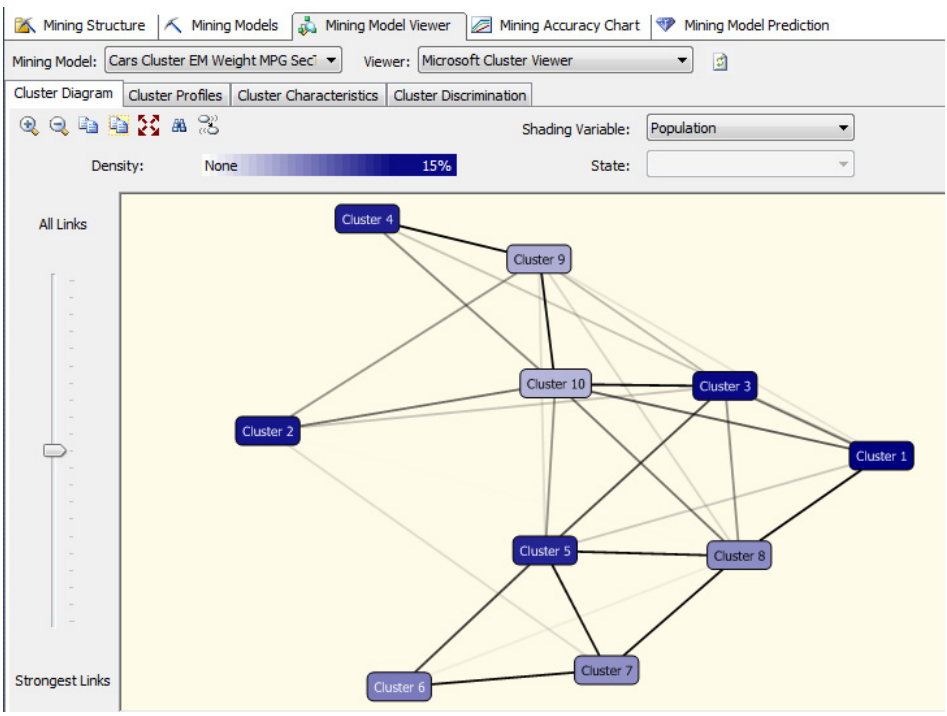


Figure 5.22

Cluster Diagram. Initially, the clusters are color-coded based on the number of observations—darker implies more data. The links indicate clusters that are close to each other.

values can be displayed for each cluster. Finish the wizard and accept the default values.

Mining models in Microsoft BI need three additional steps: (1) Transfer the model to the Analysis Server, (2) Process the model on the server—run the analysis, and (3) Browse the results. Steps 1 and 2 are often combined. However, for large, time-consuming models, it is possible to transfer everything to the server and then schedule the analysis job to run at a time when the server might be less busy. The models in this chapter are small enough to run in a few seconds. Right-click the new mining model entry in the Solution Explorer list and choose the Process option. Accept the default options on the pop-up screens to upload the data and Run the analysis. When it is finished close the setup forms. Right-click the data mining model entry and choose the Browse option to begin examining the results.

Results from Microsoft Clustering

Because of the challenges of understanding results in higher dimensions, Microsoft Clustering provides several tools to examine estimated clusters. Figure 5.22 shows the top-level Cluster Diagram. This diagram color codes the clusters based on the number of cases in each group. Darker clusters are larger. The lines attempt to show which clusters are close to the others. Selecting a single node will high-

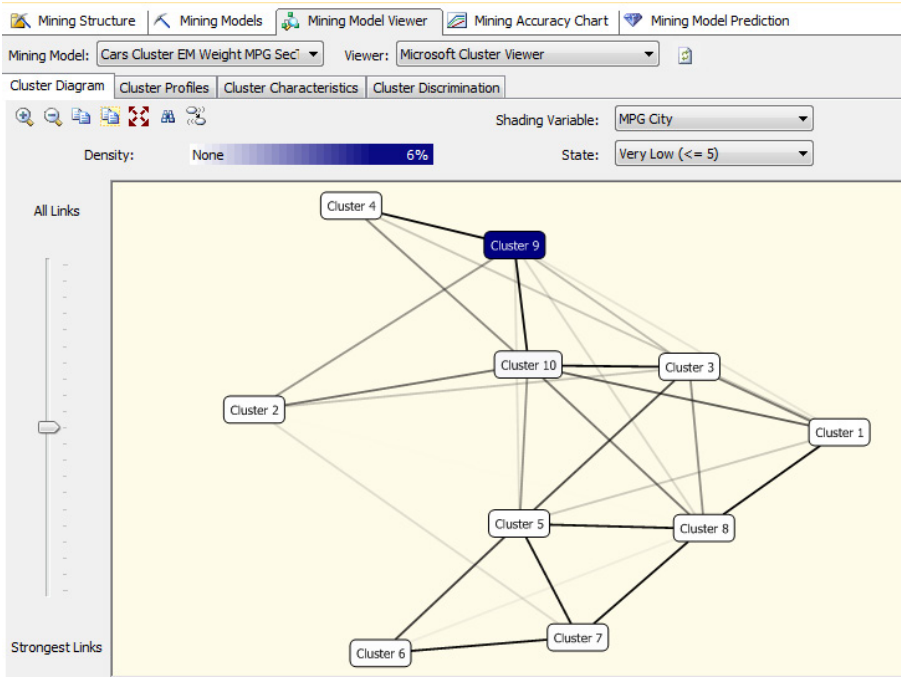


Figure 5.23

Cluster Diagram with MPG set to low values. Choose an attribute and a range to see which clusters consist of items with that attribute level.

light the nearest clusters to the one selected. The diagram provides minimal information for analysis, but it makes it easy to examine all of them at the same time.

The shading variable can be changed to one of the attributes using the drop-down list. The option also provides a selection box for various levels of the attribute. Figure 5.23 shows the clusters that contain cars with the lowest MPG. Cluster 9 is the strongest. Conversely, Clusters 4 and 10 represent higher MPG cars. This exploratory tool is useful for gaining an initial understanding of the clusters. The goal of the analyst is to understand what each cluster represents. Along those lines, the clusters can be given an explanatory name once the primary characteristics are identified.

One drawback to the Cluster Diagram is that it focuses on a single attribute. The Cluster Profile shown in Figure 5.24 is designed to provide information on all of the attributes across all of the clusters. It is a daunting task for a problem with hundreds of attributes and dozens of clusters. In some ways this tool provides too much detail, but the graphics provide some visual clues. The height of the blue diamond indicator represents the variation within a cluster for the specified attribute. Check the MPG attribute horizontally to see that most of the bars are small—indicating that each cluster represents cars within a narrow range of MPG. The glaring exception is Cluster 9, which appears to be a catch-all category, because the system defaults to 10 clusters.

The Mining Legend shows the details for any specific cluster. In the case of Cluster 7, MPG is between 10 and 22; 0-to-60 times are between 5.5 and 11.9 seconds, and weight ranges between 3880 and 7002 pounds. The mean and standard

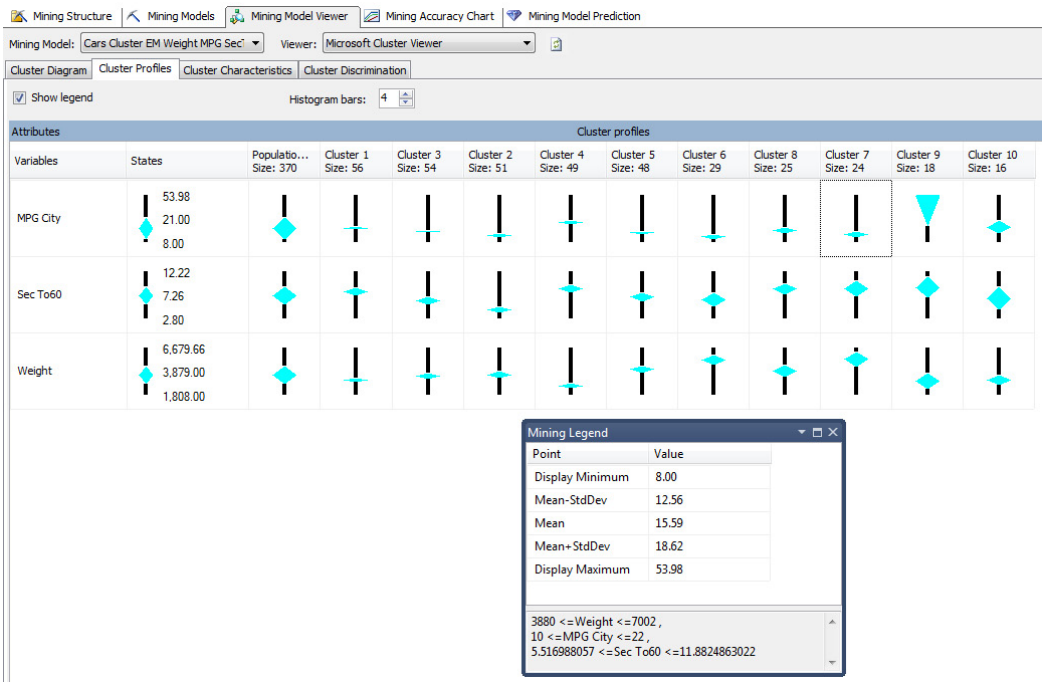


Figure 5.24

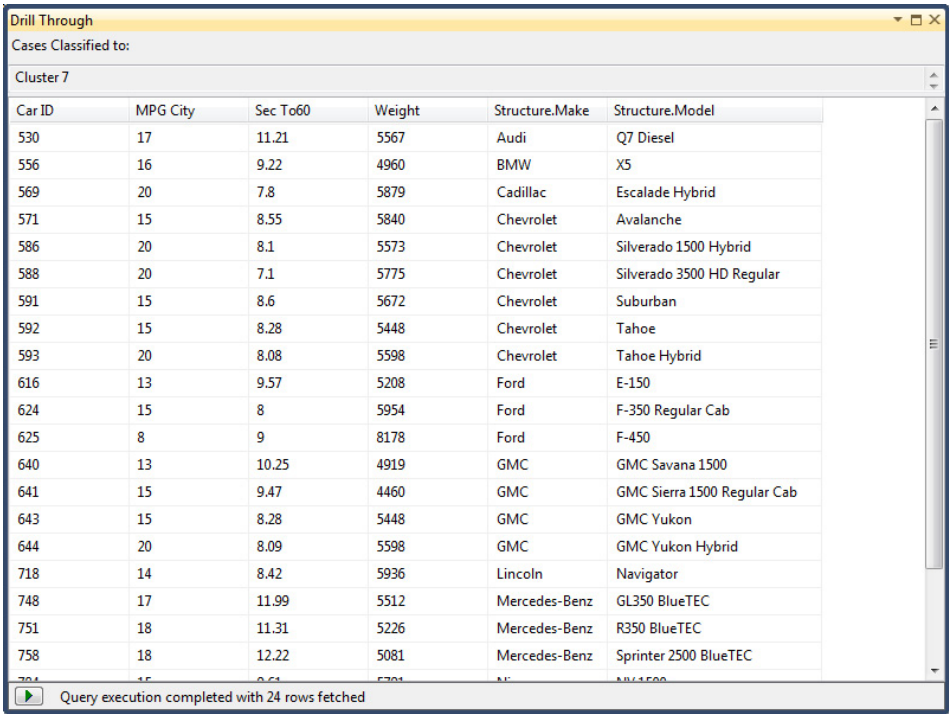
Cluster Profile. The objective is to show all of the attributes across all of the clusters. A difficult task if the problem contains hundreds of attributes and dozens of clusters. The Mining Legend shows details for one cell at a time.

deviation of a specific attribute are also shown in the top of the legend. Compare Cluster 7 to the others and it is clear that it represents heavy cars, with low gas mileage, and medium-low acceleration.

To better understand the cluster, drill through the cluster to see the observations within it. This option was checked during set up. Right-click the cluster and choose the option to drill through including the structure. The structure includes the Make and Model attributes that were specifically included but not set as Input data. Figure 5.25 shows the list of items in Cluster 7. Most of the vehicles are trucks or large SUV models. Think about the results for a minute. The system was able to group these items together based on only three factors: acceleration, weight, and MPG. In some clusters and in other problems, it will be necessary to add more dimensions to separate out some of the vehicles. For example, check out Cluster 6 which has some of the lowest MPG numbers and heaviest vehicles. That list includes several Bentleys and Maybachs, along with a few trucks.

Before moving on, look for clusters similar to Cluster 7. Notice that Cluster 2 vehicles have low MPG, fast acceleration, and at least medium weight. Drill through to see the list of vehicles and note that it consists of performance cars, so yes; the two groups are similar; yet different.

Microsoft provides additional tools to compare one cluster to another. The Cluster Characteristics provide a graphical display of the attribute values assigned to each group. Figure 5.26 shows the rules for Cluster 7. Most of the attributes



Car ID	MPG City	Sec To60	Weight	Structure.Make	Structure.Model
530	17	11.21	5567	Audi	Q7 Diesel
556	16	9.22	4960	BMW	X5
569	20	7.8	5879	Cadillac	Escalade Hybrid
571	15	8.55	5840	Chevrolet	Avalanche
586	20	8.1	5573	Chevrolet	Silverado 1500 Hybrid
588	20	7.1	5775	Chevrolet	Silverado 3500 HD Regular
591	15	8.6	5672	Chevrolet	Suburban
592	15	8.28	5448	Chevrolet	Tahoe
593	20	8.08	5598	Chevrolet	Tahoe Hybrid
616	13	9.57	5208	Ford	E-150
624	15	8	5954	Ford	F-350 Regular Cab
625	8	9	8178	Ford	F-450
640	13	10.25	4919	GMC	GMC Savana 1500
641	15	9.47	4460	GMC	GMC Sierra 1500 Regular Cab
643	15	8.28	5448	GMC	GMC Yukon
644	20	8.09	5598	GMC	GMC Yukon Hybrid
718	14	8.42	5936	Lincoln	Navigator
748	17	11.99	5512	Mercedes-Benz	GL350 BlueTEC
751	18	11.31	5226	Mercedes-Benz	R350 BlueTEC
758	18	12.22	5081	Mercedes-Benz	Sprinter 2500 BlueTEC

Figure 5.25

Cluster 8 Drill Through with Structure. Examine the items that appear in the cluster. Most of the vehicles are expensive, and large, with huge engines.

are evenly balanced, which makes it harder to identify the important elements for this cluster. If some bars are much longer than others, those become the primary dimensions. The lowest element in this chart is the MPG City 8-13 range, so this cluster excludes vehicles with very low gas mileage. But, acceleration and weight cover most categories equally.

Particularly in the case of this cluster, it is easier to evaluate the cluster in comparison to other clusters. The Cluster Discrimination tool is an interactive display that compares one cluster to any other. It can also compare a cluster to all others, by choosing the complement as cluster 2. Figure 5.27 shows the comparison between Cluster 7 and Cluster 2. Cluster 7 represents heavier vehicles with medium-to-slow acceleration. Cluster 2 contains slightly lighter-weight cars and really fast acceleration. The differences in MPG are negligible.

Prediction

Sometimes it is helpful to examine individual cases to see which cluster they would fall into. With EM clustering, it is also useful to look at the probability a specific point falls into a given cluster. Prediction is handled through a set of data mining functions. The Analysis Services provides an easy-to-use link to these functions that can be applied to a table of possibilities or to just one item. Setting up a table of inputs requires several steps, so the tool is easiest to use with just one item at a time.

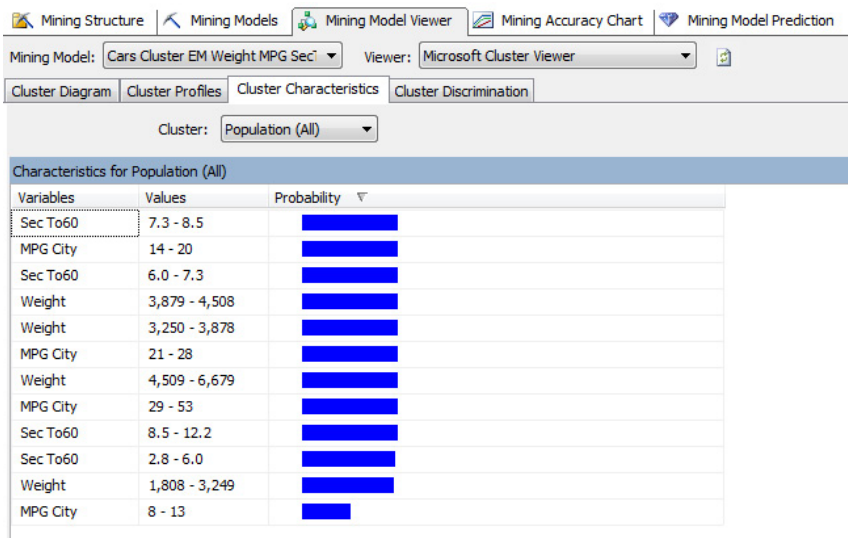


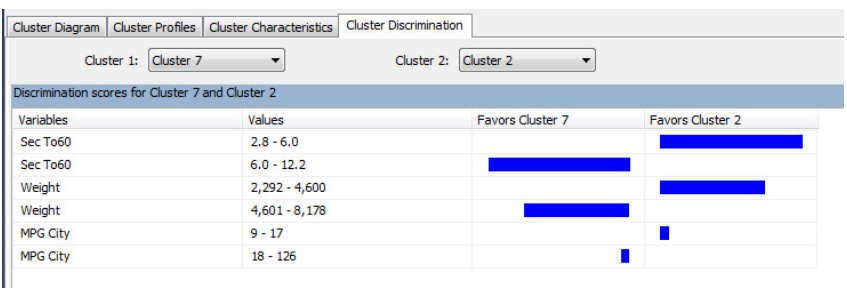
Figure 5.26

Cluster Characteristics. A graphical perspective of the cluster rules.

Figure 5.28 shows the basic process for predicting the cluster for a specific car. Begin by selecting the Mining Model Prediction tab. By default, the tool is configured to select input data from a table. Click the (third) icon (or right-click) to switch to a singleton query. Enter the input attributes for a specific car. The example here is: MPG=18, SecTo60=6.1, and Weight=3174. In the grid at the bottom of the form, select Prediction Function as the source and set Cluster as the field. Run the query by clicking the first icon (display results). The answer should be Cluster 3. Return to the design screen and change the Field to ClusterProbability. Run the new query and the result is 0.7946; so this particular car is associated with Cluster 3 with about 80 percent probability. Remember that with EM clustering, an individual point could be associated with other clusters. It is straightforward to obtain these probabilities. On the query design grid, under the column for Criteria/

Figure 5.27

Cluster Discrimination. Compare one cluster to (a) any other, or (b) all others. Useful for comparing clusters that have similar but slightly different attributes.



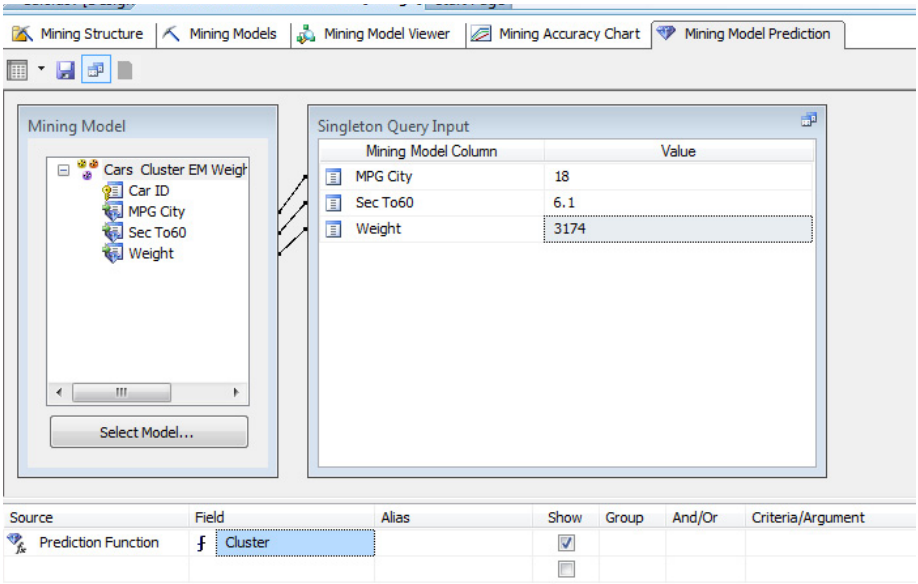


Figure 5.28

Prediction for one item. Select The Mining Model Prediction tab. Choose the Singleton query icon in the toolbar. Enter values for the input attributes. At the bottom of the form, choose the Prediction Function as the Source and Cluster for the Field. Click the first icon to switch to Query Result view. The result for this example is Cluster 3.

Argument, enter 'Cluster 8' and run the query again. You must include the single quote characters around the cluster name. The probability for Cluster 4 should be 0.0125. Other cluster names can be tested as well, and each cluster can be tested separately. It is probably useful to return to the Cluster Diagram and select Cluster 3 to highlight the clusters most closely associated with Cluster 3 (1, 2, 8, 9, and 10) and test those values first.

Larger Model and Parameter Changes

It is time to make the problem a little more realistic and more complex. The initial model used only three measurable attributes on the cars, because the tradeoffs among those three are relatively easy to see. The full model contains a few more attributes that might be important differentiators for clusters. Close the existing model.

Because the data source view contains all of the attributes for the cars, a new data mining model can be created using the same data source and view as the earlier model. Create a new model by right-clicking the Mining Structures entry. Follow the wizard and choose the Microsoft Clustering technique. Stick with the Cars table and verify that CarID is set as the Key attribute. As before, select Make and Model by marking the checkboxes in front of the columns; but do not set them as Key, Input, or Predictable. They are only going to be used as lookup values for the drill through option. Select most of the other attributes as Input and Predictable: Cylinders, HP, MPGCity, Price, Seats, SecTo60, and Weight. Year is not necessary

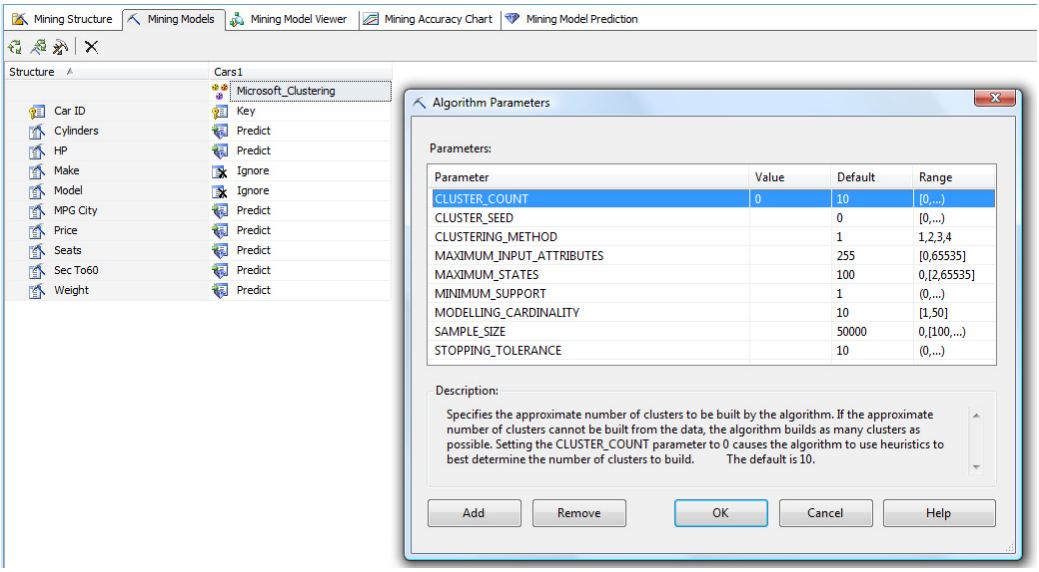


Figure 5.29

Set CLUSTER_COUNT to heuristic. By default, the clustering tool always creates 10 clusters. Right-click the Microsoft_Clustering cell and pick the option to Set Parameters. Change the CLUSTER_COUNT value to zero, which tells the system to use heuristics to pick the best number of clusters.

because all of the vehicles are from the same year. In a more extensive database across multiple years, the Year attribute would be useful. Work through the rest of the Wizard screens, enter a unique and memorable name and be sure to check the Drill Through option. Finish the wizard steps but do not process the model yet.

Remember that the initial results contained 10 clusters. This value is the default, and the Clustering technique will always try to fit 10 clusters regardless of the problem. Sometimes the managers and analysts will have a better idea of the number of desired clusters. Other times, the heuristics described in the Model section are useful to help determine the appropriate number of clusters. To set these options, click the Mining Models tab. Right-click the Microsoft_Clustering cell and choose the option to Set Algorithm Parameters. As shown in Figure 5.29, change the **CLUSTER_COUNT** entry to zero (0) to ask the system to heuristically find the appropriate number of clusters.

Take a closer look at some of the parameters that can be set. The **CLUSTERING_METHOD** is one that might be useful. It accepts four values: 1=Scalable EM, the default; 2=Non-scalable EM, 3=Scalable K-means, and 4=Non-scalable K-means. For most cases, the non-scalable choices (2 and 4) have limited value. Although they are a little more complete at testing various combinations, they can be used only with problems with a relatively small number of observations. The primary choices are the EM method (1) and K-means (3). As explained in the Model section, analysts might choose K-means if it is important to assign observations to exactly one cluster. For now, leave the CLUSTERING_METHOD at the default EM value.

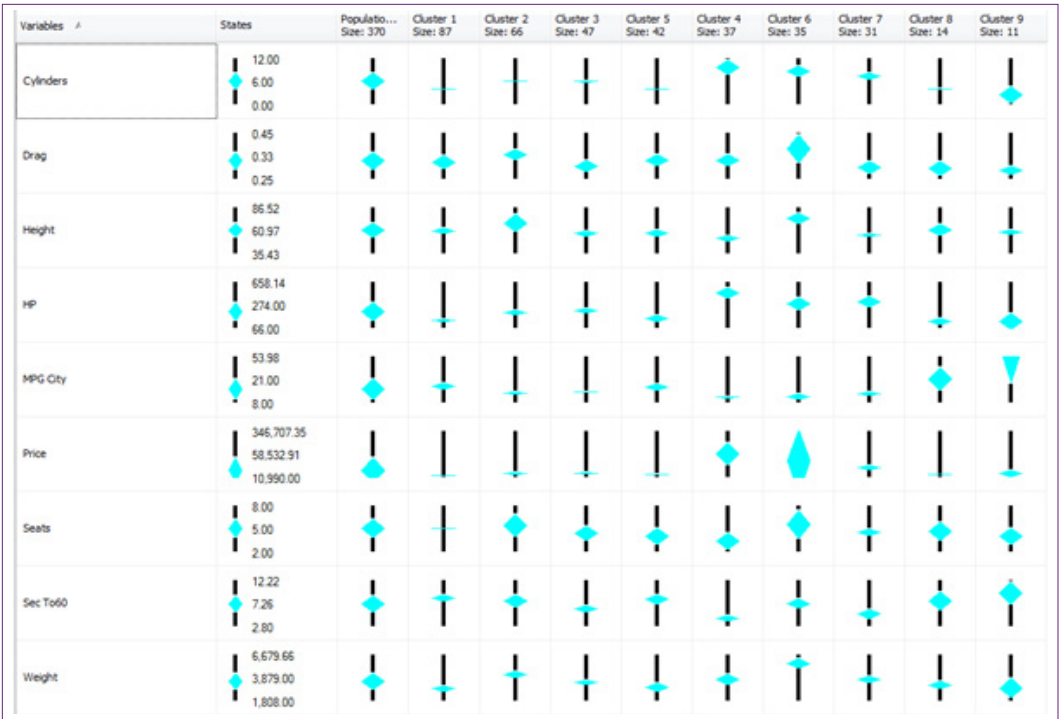


Figure 5.30

Complete results. Note the number of clusters (9), and notice that the internal variance is low, indicating tight clusters in almost all cases.

The other parameters can be used to fine-tune results or performance. Maximum values for inputs and states place limits on the size of the problem and are rarely changed. Cluster seed, modelling cardinality, and sample size are used to control the estimation process and should not be changed unless the problem is too large to be solved with the defaults. The only other value analysts might change is minimum support which specifies the smallest size cluster to be allowed. It defaults to one observation, which is not much of a group.

When the cluster count has been set to zero, process the new model and browse it. The first thing to notice is that it contains 9 clusters instead of 10. And the relationships among the clusters are thinner than before. Adding more attributes made it easier to classify the vehicles. Figure 5.30 shows the profile results. The increased number of attributes makes the chart more complex, but it appears to have improved the overall clustering process. The variance within each cluster for most attributes is low—indicating tight clusters. Examine a few of the clusters to see which vehicles fall into each category. The vehicles appear consistent and it should be possible to assign meaningful names to each cluster.

For comparison, the same data was evaluated using K-means clustering (option 3 in the parameters). Figure 5.31 shows the cluster profiles. Note first that the heuristic chose only three clusters to summarize over 300 vehicles. Also, notice the first cluster contains 3443 vehicles, or almost all of them. Drill through to check out the vehicles in each cluster—the second cluster is generally large trucks and SUVs and the third cluster contains one car, which is probably there because of

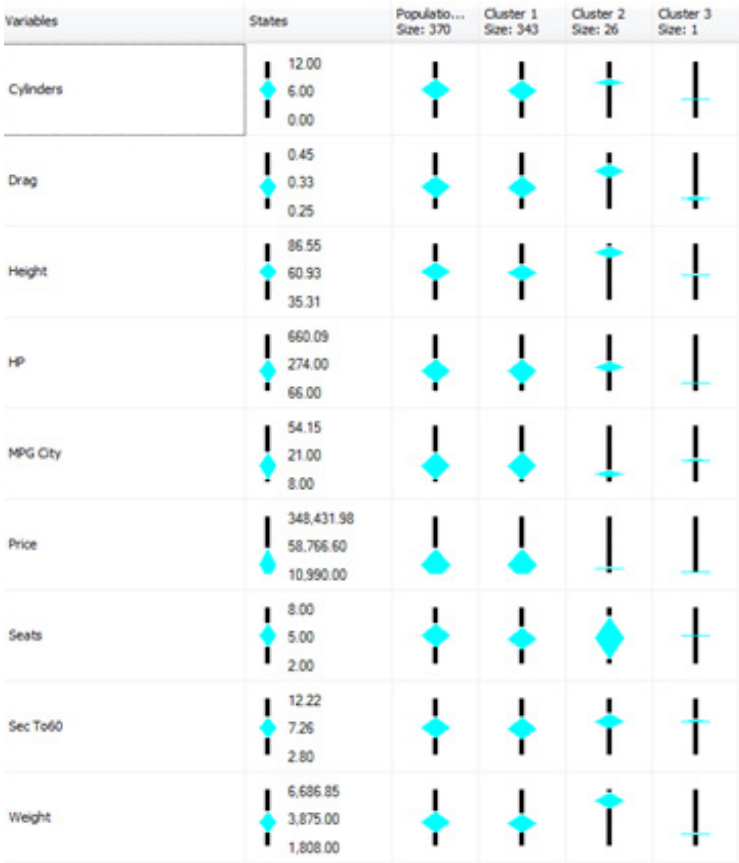


Figure 5.31

K-means clustering on all attributes. The heuristic resulted in choosing only three clusters. But almost all vehicles are in Cluster 1.

the lower price. If K-means is selected as the clustering method, it will be best to specify the desired number of clusters to override the poor default choices. But even when 9 clusters are manually specified to match the number from the EM results, almost half of the observations (173/370) are lumped into one cluster. Overall, the EM clustering technique seems better suited for this data.

Note that it is possible to assign names to each of the clusters. Whenever possible, clusters should be given meaningful names that describe the attributes represented by points in that cluster. With the car data, names might include "large SUVs," "Sports cars," and "Hybrids."

Traditional EM Clustering

How are traditional EM methods different from Microsoft Clustering? Many of Microsoft's data mining tools have additions or tweaks that make them different from the pure model. Throw in the fact that clustering is heavily dependent on (1) the distance measure, (2) the selection of the number of clusters, and (3) the way the algorithms seek a global instead of a local optimum.

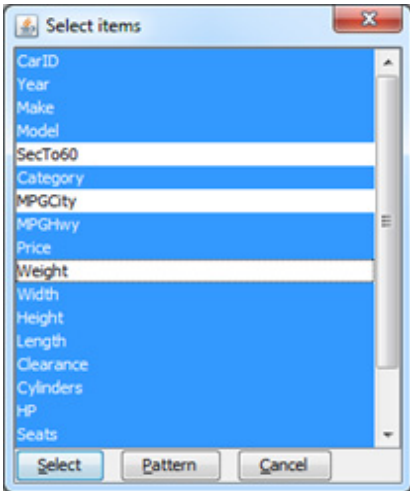


Figure 5.32

Weka Import. Attributes to ignore. Use Ctrl+Click to remove columns from the analysis.

It quickly becomes possible to obtain different results from every tool tested. For comparison, the Weka tool from the University of Waikato in New Zealand (http://www.cs.waikato.ac.nz/ml/weka) was used to examine the Cars dataset.

Goals and Data

The Weka tool is written in Java and runs on most computers. It imports data from CSV files and can save projects in a proprietary format. It has the ability to

Figure 5.33

Traditional cluster results from Weka. The content of each cluster is defined by the mean and standard deviation.

	C0	C1	C2	C3	C4	C5	C6
N/Pct	0.15	0.26	0.07	0.06	0.14	0.19	0.13
SecTo60							
mean	8.293	6.629	8.424	6.204	4.458	8.231	8.693
std. dev.	1.446	1.032	2.181	1.342	0.603	0.896	0.899
MPGCity							
Mean	16.490	17.917	46.397	11.601	13.905	21.722	27.284
std. dev.	2.353	1.208	27.227	1.308	1.834	1.350	1.983
Weight							
mean	5048	3929	3317	5722	3849	3362	2717
std. dev.	582.5	421.8	792.6	751.9	406.0	295.3	234.8

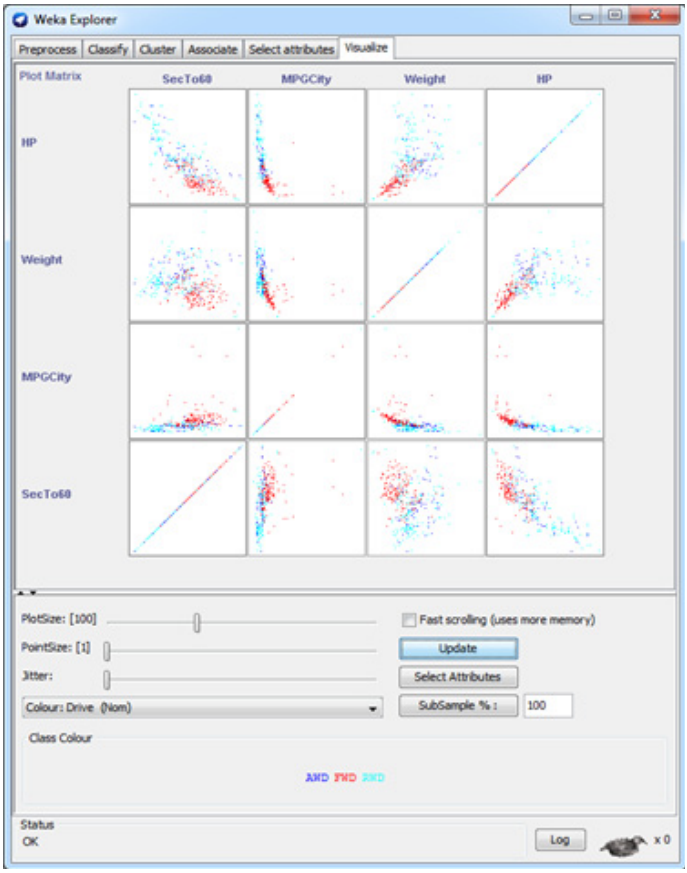


Figure 5.34

Weka 2D visualization. The data are plotted in two-dimensional charts for each combination of attributes.

directly connect to a DBMS, but the connection requires the use of Java ODBC. For large problems, it is probably worth the time to configure this connection. For small problems similar to the simple Cars table, it is easier just to import the CSV file. Weka is a tool that contains basic algorithms for several data mining tasks. It is free and relatively easy to use.

The basic Cars.csv file can be imported directly into the Weka Explorer. Start Weka, open the Explorer and click the Open file button. Navigate to the appropriate folder, change the file type list to CSV and input the file that includes the titles. The attributes should be displayed in the main list. Switch to the Cluster tab. By default, the Explorer uses all of the attribute columns in the file. Some of these are unnecessary, and it will be best to start with the smaller model that uses only the SecTo60, CityMPG, and Weight attributes. Click the Ignore attributes button to open the selection window. Figure 5.32 shows the selected items to be ignored. Hold the Ctrl key down and click on all but the three desired entries to remove them from the analysis.

Figure 5.35

Weka cluster results shown with color coding. Any pair of attributes can be selected. The clusters are highlighted. Selecting any point with the mouse displays the details for that specific object.

Results

The basic Cars.csv file can be imported directly into the Weka Explorer. Start Weka, open the Explorer and click the Open file button. Navigate to the appropriate folder, change the file type list to CSV and input the file. Figure 5.33 shows the results of the analysis of the three primary attributes. As a first point of differentiation, note the automatic selection of seven clusters. Based on counts of observations, the clusters seem reasonably balanced. Examine the clusters by looking at the clusters with the fastest acceleration (lowest 0-60 times): Cluster 3 and Cluster 4. At first glance, the clusters appear similar, but compare the MPG and Weight attributes. Vehicles in Cluster 3 are much heavier (larger) with lower MPG—probably SUVs and trucks. The vehicles in Cluster 6 are likely to be sports cars. Although the means and standard deviations provide useful data, it can be difficult to interpret.

Weka provides an integrated visualization tool to make it easier for humans to evaluate the attributes. Figure 5.34 shows the two-dimensional plots of four attributes. The tool shows combinations of all four attributes. Including more attributes in the file leads to a large set of comparison charts. The goal of the visualization is to spot pair-wise correlations and clusters. The attributes can be selected via the button on the form. The purpose of the large number of small charts is to provide a quick overview. Pair-wise correlations are relative easy to spot—as lines in the charts. Clusters appear as concentrations of dots, but the overlap of points makes them harder to spot. Selecting one of the charts provides a larger

Attrib.	All	0	1	2	3	4	5	6	7	8
N	370	21	85	33	44	34	86	32	15	20
Sec To 60	7.26	4.48	6.54	4.67	7.72	9.42	8.36	8.67	8.50	4.92
MPG City	20.7	14.5	18.7	15.0	16.1	20.2	30.3	25.2	15.0	11.7
Price	58,533	127 K	40,632	89,370	42,203	31,426	21,042	25,440	27,856	330 K
Weight	3,879	3,421	3,977	4,020	5,124	3,943	3,030	2,948	5,278	4,947
Height	61.0	49.7	60.0	55.5	72.9	65.3	58.8	57.3	77.9	5.1
Cyls	5.9	7.9	5.7	7.7	7.0	4.5	3.9	3.9	6.8	12.0
HP	274	434	281	456	303	191	153	158	290	539
Seats	4.7	2.0	4.7	4.4	6.9	5.1	5.0	3.3	2.7	4.1
Drag	0.326	0.325	0.323	0.313	0.358	0.341	0.312	0.328	0.334	0.325

Figure 5.36

Weka K-means results on all attributes. Values are center points of the clusters.

view of the relationships between the pairs of attributes. A third dimension can be chosen to display as a new color.

Once the clustering analysis has been run, it is possible to display a cluster chart by the paired dimensions. On the main Explorer page, right-click the results line and choose the Visualize option. Figure 5.35 shows the clusters comparing MPG with acceleration. Select a few of the top-right marks and the cars will be hybrids and low-price sedans with small engines and good gas mileage. Select one of the items near the bottom-left of the chart and they will consist of high-performance ultra-luxury cars with huge engines, no gas mileage, and absurd prices. The drop-down lists at the top of the form make it easy to switch the chart to compare different attribute pairs.

Humans perform reasonably well when estimating correlations and clusters in two dimensions. Three dimensions work for some people, but it is difficult to display three dimensional charts without hiding most of the data. Most people cannot visualize relationships beyond three dimensions. Hence, the reason for creating an entire panel of pair-wise two-dimensional charts.

K-Means Clusters

Weka appears to do a better job with the K-means technique than the Microsoft Clustering tool. The tool is selected near the top of the Explorer window under the Cluster tab—click the Choose button and select the SimpleKMeans technique. Right-click the resulting selection and choose Properties to change the underlying parameters. Specify the numClusters to 9 to see how well the results match the Microsoft EM outcome. In the end, the observations are distributed better across the clusters than they were with the Microsoft K-means technique. The largest cluster holds 24 percent of the observations

Figure 5.36 shows the resulting centers of the K-means. Notice the relatively even distribution of the number of observations. Glancing at the results, Price seems to be a good starting point for evaluating the clusters because most of the

Attrib.	C1	C2	C3	C4	C5	C6	C7	C8	C9
Cyls.	4.0	6.0	5.9	9.6	4.0	8.6	7.2	4.0	2.4
Drag	0.32	0.35	0.30	0.33	0.33	0.38	0.30	0.30	0.29
Height	60.1	69.0	57.3	51.7	57.6	73.7	55.6	61.4	59.0
HP	159	263	282	516	184	375	397	150	150
MPG City	24.4	17.2	18.3	13.1	23.3	13.8	16.8	31.2	64.3
Price	19660	36225	38434	178041	28656	126770	77806	25946	35557
Seats	5.0	5.4	4.4	3.3	4.0	5.5	4.5	4.6	4.0
Sec To 60	8.51	7.93	6.30	4.40	8.23	7.39	5.21	7.97	9.47
Weight	3080	4562	3729	4012	3227	5743	4023	3424	3135
N	87	66	47	37	42	35	31	14	11
Sample	Toyota Camry	Jeep Liberty	Chevy Camaro	Merc. CL63 AMG	Mini Cooper	Cadillac Escal.	Ford Shelby GT500	Toyota Prius	Nissan Leaf
Descrip.	Basic Sedans	Basic SUVs	Mid-perf.	High-price perf.	Mid-price perf.	Big SUVs	High-Perf.	Hybrid	Elec.

Figure 5.37

Summary of Microsoft EM clusters. Means are listed. Sample vehicles are pulled from cluster list.

clusters have fairly strong separation based on Price. Weight might be a good secondary differentiator. Compare Clusters 1 and 3 that have similar price points—the weights are quite different. Based on the number of seats and price, Cluster 3 probably represents large SUVs.

Comparison

Are the results from the multiple tools and methods really different? Or, is one tool and method better than the others? This second question is easier to answer: No. Most of the differences across tools arise because of choices in the number of clusters. For the most part, it should be possible to specify the number of clusters to be the same in any tool, and using the same technique, obtain close to the same results. But there are no guarantees. The clustering algorithms are always slightly different. On the other hand, the tools are considerably different in how the data and results are presented and visualized. It is quite possible that analysts will develop a preference for a tool based on the way the results are presented and searched. Most tools automatically include all data by default. For a side-by-side comparison, the holdout size in Microsoft Clustering should be set to zero.

As a test, the Weka EM tool was used to generate the same number of clusters as found by the Microsoft tool (9). Figure 5.37 summarizes the Microsoft clusters. To save space and keep the table legible, only the means of the clusters are listed.

Figure 5.38 lists the clusters found using the Weka EM clustering tool. Using its own heuristics, the Weka EM tool found eight clusters on the full set of attributes.

Attrib.	C0	C1	C2	C3	C4	C5	C6	C7	C8
Cyl.	5.7	5.8	3.8	3.9	8.0	12.0	7.2	6.4	7.9
Drag	0.32	0.34	0.32	0.32	0.38	0.33	0.32	0.35	0.31
Height	59.6	69.3	58.5	59.6	76.3	54.4	50.1	74.2	55.8
HP	278	263	150	156	340	539	425	276	458
MPG City	18.9	17.8	25.4	27.9	14.7	11.7	15.3	15.9	14.8
Price	41,475	32,623	25,304	21,676	48,610	318,756	119,627	36,841	82,745
Seats	4.7	7.1	2.2	4.8	6.4	3.8	2.4	3.8	4.5
Sec To 60	6.69	7.70	8.57	8.54	7.75	4.83	4.46	8.88	4.73
Weight	3959	4489	2812	3114	5689	4791	3373	5152	4122
N	88	22	12	126	26	22	25	23	26
Sample	Volvo XC70	Ford Explor.	Mazda Miata	Toyota Prius	Lincoln Navig.	Bentley	Porsche 911	M-B. R350	Jaguar XK
Desc.	Mid-Level	Cheap. SUVs	Lighter Cars	Basic sedans	Big SUVs	Luxury	High Perf.	Luxury SUVs	Luxury Perf.

Figure 5.38

Summary of Weka EM forced to seven clusters. Means are listed. Sample vehicles are pulled from cluster list.

But it is easier to compare clusters if the same number is used for both cases. Try comparing the clusters. Start by looking at prices. Cluster 4 in the Microsoft table probably corresponds to Cluster 5 in the Weka table. But there are differences in the items—look at the counts. The largest clusters in both cases are labeled as "basic sedans," but the clusters are different. Notably, the Weka cluster includes the hybrids which are separated by the Microsoft algorithm. The Weka algorithm also separated SUVs (and large sedans) differently. Price seems to have played a greater role in the Weka clustering versus weight and MPG in the Microsoft clustering approaches. So, yes, different algorithms give different results. It might be useful to test multiple approaches on data to see if additional insights can be gained. The goal of clustering is not to find a single "perfect" answer—such a thing does not exist. Instead, the objective is to gain insight into the items involved.

These results are not meant to imply that one tool is better than the other. Both tools have configuration options to support multiple distance measures. Although, Weka probably has more options, and Weka does allow people to write custom code to create new models. The main purpose of the exercise is to show that the choice of a model and distance measure can make a big difference in the results. Regardless of the tool used, it is useful to try some of the other options to see if one measure might be more useful for a particular problem.

The choice between K-Means and EM techniques is much clearer. EM uses a soft cluster definition where an observation falls largely into one cluster, but can also be associated with other clusters. When this concept matches reality, probably most of the time, EM does a better job of defining clusters because it can handle these in-between cases better. The key feature of K-means is that it forces each observation into a single cluster. There might be times when this clarity is needed, but forcing data to behave to a strict model might not give the best results.

Customer Clustering with Categorical Data

How does categorical data change the results and interpretation of clusters?

The automobile dataset was created specifically because most people are familiar with the attributes and all of the attributes are measured with continuous data, which works best for clustering. But, business problems often involve categorical data so it is important to know how to interpret results involving categorical data.

Data

The Corner Med Patient database contains some relatively common personal attributes of customers, and a few that are unique to the medical world. Begin by examining the Patient table, which is equivalent to a Customer table for typical businesses. It contains attributes on DateOfBirth, Gender, Race, TobaccoUse, and location information. The bulk of the data for Corner Med is generated from government reports on anonymous patient visits that include most of those attributes. However, the location and phone number data (and names) are randomly generated. In a real case, it would be useful to include the location data. In this example, it is unlikely to be important. The Visit table also contains some useful information about patients, such as the InsuranceCompany, and blood pressure data. The blood pressure data is interesting, but it contains a high percentage of missing values so it should not be used. The insurance company data is important because it defines how the business gets paid with various limits on procedures and charges. In particular, the government Medicare and Medicaid programs have strict limits. One additional attribute might be useful to classifying patients—a measure of the patient's involvement with Corner Med. This dimension could be measured either by a count of the number of visits per year or the total amount billed per year by patient. The AmountCharged column in the VisitProcedure table provides that value when it is summed by patient.

To analyze the data with Microsoft Clustering, create a new Analysis Services project in Visual Studio. Add a new data source that connects to the CornerMed database. Create a new Data Source View that contains most of the tables: Patient, Visit, VisitProcedures, ICD9ProcedureCodes, VisitDiagnoses, ICD9DiagnosisCodes, and VisitMedications. Only three of these tables are needed for the clustering problem, but the other tables can be used later for different problems.

Because the attributes needed appear in three tables, it is necessary to build a named query to combine the tables into a single source. Right-click the main data source view screen and choose option for New Named Query. Provide a name for the query that is unique and describes the data, such as PatientCharges. The current data in CornerMed consists of a single year (2010), so no constraints are necessary for the date. Race, Gender, and TobaccoUse are straightforward because they are in the Patient table. The InsuranceCompany attribute could cause problems if people have multiple visits and change insurance companies at each visit—but that rarely happens within a single year, so it can be ignored for now. The total of the AmountCharged can be computed as the Sum of a GROUP BY query. The Age presents a challenge. The database holds date of birth, which is the best way to handle it. An Age column can be computed by subtracting the date of birth from a specific date. But which date should be used? That is, the age of the patient on which day is needed? It would be easy to pick the date on the day of the visit, but the subtotal for Amount runs across multiple visits and dates, so that Age

of Visit value will change. It is better to choose a fixed date so the patient's age is constant within that year. The easy solution is to use the end of the year or start of the next year (01-Jan-2011). The query is:

```
SELECT      dbo.Patient.PatientID, DATEDIFF(yyyy, dbo.
Patient.DateOfBirth,
      CONVERT(DATETIME, '2011-01-01 00:00:00', 102)) AS Age,
dbo.Patient.Race,
      dbo.Patient.Gender, dbo.Patient.TobaccoUse, dbo.Visit.
InsuranceCompany,
      SUM(dbo.VisitProcedures.AmountCharged) AS AmountTotal
FROM  dbo.Patient
INNER JOIN dbo.Visit ON dbo.Patient.PatientID = dbo.Visit.
PatientID
INNER JOIN dbo.VisitProcedures ON dbo.Visit.VisitID = dbo.
VisitProcedures.VisitID
GROUP BY dbo.Patient.PatientID, DATEDIFF(yyyy, dbo.Patient.
DateOfBirth,
      CONVERT(DATETIME, '2011-01-01 00:00:00', 102)), dbo.
Patient.Race,
      dbo.Patient.Gender, dbo.Visit.InsuranceCompany, dbo.
Patient.TobaccoUse
```

To use the same data in an external program, such as Weka, copy the SQL and paste it into a new query window in SQL Server. Run the query to obtain the results. Right-click the Results window and choose the option to Save Results As. Select a location and save the values in a CSV file. Before closing the name query editor, test the query to ensure it returns correct values. Once the query is created, right-click the PatientID and select the option to set it as the **Logical Primary Key**. The analytical tools require that one column uniquely identify each row. By setting it now, the tools automatically pick it up and save steps and reduce errors later.

Microsoft Clustering Results

Once the named query is created to define the data, the process of building the model and analyzing it is straightforward. Create a new mining model and choose the Clustering method. Set the holdout percentage to zero so that all cases are used in the clustering process. Also, before processing the model, switch to the Mining Models tab and set the algorithm parameters. Set the cluster count to 0 to have the system choose the number of clusters instead of forcing it to fit 10 clusters.

Figure 5.39 shows the results from Microsoft Clustering on the Corner Med patient attributes. Notice the use of the stacked bar charts for the categorical attributes. To evaluate the role of the attribute, look across the clusters for changes in the charts. For instance, tobacco use is slightly higher in some clusters and almost non-existent in others. Notice that Cluster 4 is basically non-smoking, but check the age to see why. So Cluster 4 could be called *Babies*, and it is the only cluster that covers that specific age group. Cluster 7 has a higher percentage of tobacco users than the other clusters. Race is 1 (white), Gender is biased towards female, age is slightly less than average, insurance does not look good (other), and this cluster has one of the highest charges per patient. As another example of categorical data, check out Clusters 3 and 10 which have the highest percentage of Medicare coverage. From a medical and business perspective, those two clus-

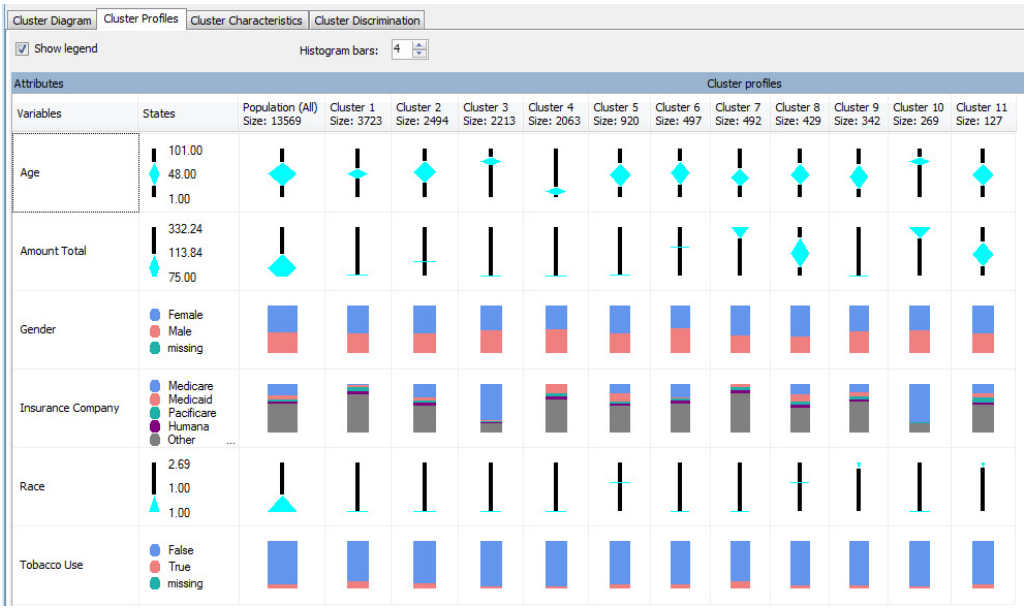


Figure 5.39

Corner Med patient clusters. Compare the categorical attributes in each cluster and to the population. Tobacco use is slightly higher in some clusters, Gender varies in some, and insurance is a major classifier.

ters deserve more investigation. Most of the attributes are the same, including the highest ages (which fits with the Medicare attribute), yet Cluster 3 has the lowest total spending and Cluster 10 the highest. Some missing factor must differentiate the clusters and it would be nice to track down that factor.

Weka Clustering Results

Figure 5.40 shows the results from the Weka EM clustering on the same Corner Med patient data. The CSV file was exported directly from the query run inside SQL Server. The results for the categorical attributes have been converted to percentages and highlighted using Excel. The percentages are easier to compare across clusters. Notice that the results are considerably different from the results generated by Microsoft Clustering. However, five of the Weka clusters focus on Medicare and Medicaid, which parallels the Microsoft results. The biggest difference is that five of the clusters here are driven by gender differences. Plus Clusters 2 and 4 represent tobacco users. In fact, the first five clusters are relatively easy to describe because of the dominant features in each cluster. The last two clusters are more difficult to understand, but because the last cluster holds only five observations, it is somewhat irrelevant.

As a side note, Weka can perform hierarchical clustering when the Cobweb clustering method is chosen. However, even with the smaller automobile dataset, the results are difficult to read. It could be useful for identifying the appropriate number of clusters for a problem. The tree view shows the effects of different numbers of clusters.

	0	1	2	3	4	5	6
Count	4309	7723	526	10	820	285	5
Age	80.4	42.5	41.3	74.6	51.5	31.5	51.4
TotalAmount	122.23	111.65	120.57	125.72	115.94	92.14	123.02
Race	1.04	1.20	1.30	1.04	1.08	3.43	1.02
TobaccoUse	0.00	0.00	0.99	0.00	1.00	0.23	0.27
Gender							
Female	0.996	0.626	0.989	0.001	0.416	0.229	0.592
Male	0.004	0.374	0.011	0.999	0.584	0.771	0.408
Insurance							
Cigna	0.002	0.062	0.021	0.050	0.062	0.201	0.455
United Healthcare	0.002	0.065	0.038	0.003	0.055	0.022	0.023
UniversalCare	0.002	0.068	0.062	0.010	0.063	0.028	0.047
Assurant	0.002	0.074	0.068	0.005	0.050	0.049	0.046
Humana	0.002	0.077	0.074	0.007	0.054	0.064	0.050
Blue Cross/Blue Shield	0.002	0.072	0.067	0.005	0.055	0.053	0.043
Pacificare	0.002	0.077	0.082	0.007	0.055	0.029	0.052
Aetna	0.002	0.061	0.115	0.048	0.040	0.162	0.028
Kaiser Permanente	0.002	0.069	0.049	0.008	0.049	0.040	0.045
Medicare	0.974	0.123	0.033	0.838	0.243	0.088	0.025
Medicaid	0.002	0.099	0.255	0.001	0.093	0.161	0.039
Self	0.002	0.060	0.059	0.003	0.079	0.016	0.043
Nationwide	0.002	0.072	0.043	0.008	0.057	0.055	0.045
Worker	0.002	0.015	0.021	0.005	0.036	0.022	0.034
Charity	0.002	0.004	0.013	0.001	0.010	0.008	0.025

Figure 5.40

Corner Med patient clusters via Weka. The categorical results were converted to percentages and highlighted to make them easier to read.

In the end, it makes sense to test multiple clustering methods. The results might vary depending on the tools measurement, search, and underlying assumptions. But, data exploration requires thinking about the data in different ways. The goal is rarely to arrive at a precise, repeatable definition. Instead, data mining seeks to shed light on relationships and provide insights. If multiple tools produce different perspectives, it should be considered a useful encouragement of further exploration.

Summary

Clustering is an unsupervised learning method that tries to find groups of items that are close to each other. Close is defined by a distance measure, and different clusters arise depending on the details of the distance measure. Choosing the number of clusters is a second challenge that has several different solutions. Sometimes the number of clusters can be specified within the business problem, other times a heuristic approach is used to estimate the appropriate number.

Three primary clustering methods are in common use: K-means, expectation maximization (EM), and hierarchical. The primary difference between K-means and EM is that K-means creates hard clusters where each observation can fall into only one cluster. EM uses probability to define softer groupings, so that an observation has a probability of belonging to a cluster and could be associated with multiple clusters. Results from EM clustering can be harder to interpret if the associated cluster probabilities are weak. Yet, it is often a more realistic approach because items being clustered often fall into gray definitions and can fit into multiple clusters. Hierarchical clustering attempts to solve the problem of choosing the number of clusters by finding all possible sets of clusters, from one cluster down to N clusters where each observation falls into its own group. The technique only works with a small number of observations.

Principal component analysis (PCA) is a different statistical tool that is sometimes used to simplify attributes. It is slightly different from clustering. Clustering attempts to collect observations into smaller groups. PCA attempts to reduce the number of dimensions or attributes by finding combinations that explain the data almost as well as the original set of attributes. Although the approach and the tools are different, in the end both tools attempt to reduce the complexity of a problem by finding smaller combinations of attributes that can represent the data.

Clustering works best with continuous data because of the reliance on distance measures. Categorical data can be analyzed as nominal measures that assign a distance measure to different attribute values. If the analysts and managers are aware of better measures, better clusters can be created by defining a new variable that contains an expert's assigned measure.

When examining results, analysts and managers should strive to assign names and descriptions to each cluster that accurately reflect the primary focus of the cluster. These groupings are used to examine the impact of decisions on the various groups and explain how each cluster might react to changes. For example, participants in one cluster might be more sensitive to price changes than another cluster. One challenge with analyzing cluster results is that they can vary greatly depending on the technique chosen as well as the specific tool and algorithm used to search the data. It can be easy to get into arguments over which tool or method generates the most accurate clusters. Clearly delineated data can lead to strong clusters—and these are generally consistent across tools and methods. It is the gray areas that lead to diverse results. Still, anything that helps managers and analysts see the data in new ways has the ability to lead to better understanding of the data. Instead of assuming that one cluster definition is better or worse than another, simply embrace the judgments involved and evaluate all potential clusters as knowledge.

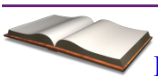
Key Words

agglomerative
categorical attribute
CLUSTER_COUNT
clustering
CLUSTERING_METHOD
combinatorial search
comma-separated values (CSV)
correlation coefficient
Data Source
Data Source View
dendrogram
distance
divisive
eigenvalues
Euclidean
expectation maximization (EM)
gap statistic
hierarchical clustering
K-means
Logical Primary Key
mixture model
multicollinearity
nominal
ordinal measure
orthogonal
principal components analysis (PCA)
responsibilities
unsupervised learning
Weka

Review Questions

1. What are the common business uses for clustering analysis?
2. What is the best distance measure to use for most clustering problems with continuous data?
3. How are categorical attributes handled in clustering algorithms? Is there a better approach?
4. How is the EM method different from the K-means approach to identifying clusters?
5. How is missing data handled by clustering?
6. How is principal components analysis different from clustering?
7. What features are provided by Microsoft Clustering to analyze results?
8. What features are provided by Weka clustering to analyze results?
9. What is prediction and how is it handled by Microsoft Clustering?

Exercises



Book

1. Pick a subset of interesting attributes for the car database and set up and run the cluster analysis for them. First use the EM method then run the K-means algorithm. Assign names to the resulting clusters and comment on the differences in the results between the two algorithms.

2. Run the EM clustering analysis for the cars database. Choose five vehicles and run predictions for them. List the clusters predicted for each vehicle, then find the probabilities for each of the other clusters. Comment on the results.

3. Run EM and K-means clustering for the Corner Med patient data. Assign descriptive names to each of the clusters. Comment on differences between the results from the two methods. Hint: Compute Age=DateDiff(yyyy, DateOfBirth,'12/31/2010').

4. Run the Weka clustering tool on the Corner Med patient data using the EM method. Use the cluster visualization tool and comment on the resulting clusters.

Rolling Thunder Database

5. Run clustering analysis on the Customer data, provide descriptive names to any clusters and comment on the results.

6. Run clustering analysis on the Bicycle data, including at least time to build (ShipDate-OrderDate), frame size, model type, year, and construction which represents frame material. Describe the clusters and comment on any results.

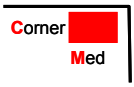
7. Run clustering analysis on the components, without the Category attribute. Identify the clusters and comment on how accurately they match the actual categories.

8. Run clustering analysis on the purchase orders. Describe the clusters created and comment on the results and how they might be used to alter purchase decisions.

Diner

9. Run EM clustering analysis on the diners and describe the resulting clusters. Use prediction tools to determine the most likely cluster and the probabilities of association with other clusters for five different observations.

10. Run K-means clustering analysis on the diners and describe the resulting clusters. Comment on how the clusters might be used to increase sales.



Corner Med

11. Run EM clustering analysis on the procedures performed for each patient. Describe the resulting clusters and comment on them.

12. Run EM clustering and K-means clustering analysis on the patient diagnoses. Describe the resulting clusters and comment on the differences between the two sets of results.

13. Run EM clustering analysis on the Visit that includes at least the diagnoses, procedure, and drug codes. Describe the resulting clusters and comment on the results.

Basketball

14. Pick a team and a season and run EM clustering on the players and player statistic averages and describe any resulting clusters.

15. Run clustering analysis against all of the player statistics for one season without including the player's position. Describe the clusters and compare them to the player positions. Comment on any differences.

16. Are some divisions better or worse than others in terms of winning? What about in terms of total points scored? Run clustering by division with won/loss and points scored.

17. Run clustering analysis on the teams and games for one season, without the conference or division, describe the clusters.

Bakery

18. Run clustering analysis on the products, without using the category. Describe the clusters and compare them to the actual categories.

19. Run clustering analysis on the Sale and SaleItem tables. Convert SaleDate to day of week, month, and split it into time of day: morning, noon, and evening. Describe the clusters.

Cars

20. Find data on more vehicles and add it to the database. Specifically, find data on at least 20 different trim levels for the existing vehicles. Run EM clustering and compare the results to the clusters found with the original data.

21. Run the Weka hierarchical clustering (Cobweb) on the car data and use the results to comment on the number of clusters that should be chosen.

Teamwork

22. In the dining database, assign a different month to each team member. Run clustering analysis for each month and compare the results. Comment on any differences.

23. In the basketball database, split the database into teams that made the playoffs and those that did not. Split your team into two groups and assign one set to each group. Run clustering analysis on the player statistics, describe the clusters, and compare the results across the teams.

24. In the bakery database, split the data into years. Assign one year to each team member. Convert SaleDate into month, day of week, and time of day (morning, noon, evening). Run cluster analysis on the sales data, describe the clusters, and compare the results from each team member.

25. In the cars database, add at least one more attribute to the analysis—such as ground clearance or height. Run the clustering analysis and compare the results to the original clusters.

Additional Reading

Ding, Chris and Xiaofeng He, 2004, K-means Clustering via Principal Components Analysis, *Proceedings of the 21st International Conference on Machine Learning, Banff*, Canada. [Highly mathematical but a proof that principal components are related to K-means.]

Kaiser, H. F. 1960, The Application of Electronic Computers to Factor Analysis, *Educational and Psychological Measurement*, 20, 141-151. [The original source of the rule to choose principal components when the eigenvalue is greater than one.]

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman, 2009, *The Elements of Statistical Learning/2e*, Springer: New York. [An outstanding book on data mining, with an emphasis on theory. A graduated-level book that requires a strong mathematics background. Excellent mathematical analysis of the clustering algorithms.]

http://www.cs.waikato.ac.nz/ml/weka [Free data mining software from The University of Waikato in New Zealand. The software is written in Java and runs on most computers.]