Data Mining Techniques: Theory and Practice

Course Notes
Table of Contents

Course Description ....................................................................................................................... v

Prerequisites ................................................................................................................................ vi

Chapter 1  Introduction .......................................................................................... 1-1
  1.1 Introduction .............................................................................................................. ........ 1-3

Chapter 2  Methodology ......................................................................................... 2-1
  2.1 Introduction .............................................................................................................. ........ 2-3

Chapter 3  Data Exploration ................................................................................... 3-1
  3.1 The Environment and Data Exploration .......................................................................... 3-3

Chapter 4  Regression Models............................................................................... 4-1
  4.1 Regression ................................................................................................................ ........ 4-3

Chapter 5  Decision Trees ...................................................................................... 5-1
  5.1 Decision Trees ............................................................................................................ ...... 5-3

Chapter 6  Neural Networks ................................................................................... 6-1
  6.1 Neural Network Models ................................................................................................... 6 -3

Chapter 7  Memory-Based Reasoning .................................................................. 7-1
  7.1 Memory-Based Reasoning ............................................................................................... 7-3

Chapter 8  Clustering .............................................................................................. 8-1
  8.1 Clustering ................................................................................................................ ......... 8-3
Chapter 9  Survival Analysis ................................................................. 9-1
  9.1  Survival Analysis ................................................................. 9-3

Chapter 10  Association Rules ................................................................. 10-1
  10.1  Market Basket Analysis and Association Rules ......................... 10-3

Chapter 11  Link Analysis ................................................................. 11-1
  11.1  Link Analysis ................................................................. 11-3

Chapter 12  Genetic Algorithms ................................................................. 12-1
  12.1  Optimization and Genomes ................................................................. 12-3
Course Description

Explore the inner workings of data mining techniques and how to make them work for you. Students are taken through all the steps of a data mining project, beginning with problem definition and data selection, and continuing through data exploration, data transformation, sampling, portioning, modeling, and assessment.

To learn more…

For information on other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the Web at support.sas.com/training/ as well as in the Training Course Catalog.

For a list of other SAS books that relate to the topics covered in this Course Notes, USA customers can contact our SAS Publishing Department at 1-800-727-3228 or send e-mail to sasbook@sas.com. Customers outside the USA, please contact your local SAS office.

Also, see the Publications Catalog on the Web at support.sas.com/pubs for a complete list of books and a convenient order form.
Prerequisites

No prior knowledge of statistical or data mining tools is required.
Chapter 1 Introduction

1.1 Introduction

.................................................................1-3
1.1 Introduction

This Chapter...
- introduces the course and the instructors
- reviews the course agenda
- defines the term data mining
- puts data mining in its business context.

Course Objectives
- Describe the major data mining techniques, especially differences among them and how they are applied to business problems.
- Describe how to use SAS Enterprise Miner for setting up data mining problems.
- Interpret the results of various data mining techniques.
- Spread enthusiasm for data mining techniques and their applications.
The First Day

1. Introduction
2. Methodology and Applications
3. Data Mining Tools and Data Exploration (with lab)
4. Regression (with lab)
5. Decision Trees (with lab)

Each day there is a break mid-day for lunch. There are snack breaks mid-morning and mid-afternoon. Labs for exercises are usually aligned with breaks so that students can take extra time if they want.

The Second Day

5. Decision Trees (with lab) (continued)
6. Neural Networks (with lab)
7. Memory-Based Reasoning (with demonstration)
8. Clustering (with lab)
The Third Day

8. Clustering (continued)
9. Survival Analysis (with lab)
10. Market-Basket Analysis and Association Rules
11. Link Analysis
12. Genetic Algorithms

This day is shorter than the other days because the instructors do not spend as much time on the last three topics.

How Will the Instructors Accomplish This?

- Lectures describing techniques
- Stories illustrating how data mining is used in practice from Data Miners' consulting experience
- Demonstrations using SAS Enterprise Miner
  - If you have not used SAS Enterprise Miner, then today you might feel as if you are drowning.
  - By the end of the course, you will have basic mastery over the tool.
- Hands-on exercises
  - Some students will finish all exercises, particularly those familiar with SAS Enterprise Miner.
  - Some students might not finish all of them.
This course is based on the book *Data Mining Techniques* by Michael Berry and Gordon Linoff.

Gordon and Michael are the founders of a data mining consulting company named (with great imagination) Data Miners, Inc. This course reflects years of practical data-mining experience. In this class, data-mining techniques are introduced in a business context. Part of what you learn is how these techniques are used to connect data to important business problems.

The focus of this class is the data-mining process and the tools, techniques, and algorithms that are used in that process. During the three days of class, you will have an opportunity to interact with data sets taken from different industries and to use SAS Enterprise Miner, the "flagship" solution for data mining.

Software is not the focus of the class, and it is not meant to replace a SAS Enterprise Miner class.
Instructors’ Background

- Studied mathematics (not statistics)
- Background in parallel processing and large databases since the 1980s
- Founded Data Miners, Inc. in 1997
- First used SAS Enterprise Miner (Version 2) in 1998

The instructors do not work for SAS and do not have expert knowledge of SAS products. They do, however, often use SAS products in their work.

The most important aspect of the instructor’s background is that he continues to be an active practitioner in the field of data mining and so brings a practitioner’s viewpoint to every topic.

Michael and Gordon Met at a Supercomputer Company in 1990

You probably never heard of Thinking Machines but possibly you saw the movie *Jurassic Park*.

In Michael Crichton’s book, Cray computers run the park.

The movie used massively parallel supercomputers instead.

Scene from *Jurassic Park*, courtesy of Dave Fisher (www.cca.org/dave/tech/images/jurassic-park-cm5.jpg)
Data Mining Is a Process for ...

“...the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules...” *


Data mining is a business process for maximizing the value of data collected by the business.

It is easy to fall into the trap of thinking of data mining as a collection of techniques and algorithms. In reality, the data-mining techniques discussed in this class are the tools used in the process of mining data.

All parts of this definition are important. Data mining is exploration and analysis. It is not guessing or trying to prove what you already know. It involves large quantities of data. In many cases, “large” simply means that the data is too big to fit into Excel. Most importantly, the meaning needs to be significant for the business. Data is filled with patterns; you want the ones that are relevant and actionable.

The Data Mining Challenge

Getting anything useful out of tons and tons of data

There is undoubtedly something useful in there, but it is not easy to reach.
How Does Data Mining Do This?  
By Creating Models

A model takes one or more inputs and produces one or more outputs:
- scores, rankings, quantiles
- visualizations, graphics, charts
- insight

*Training* is the process of creating a model from an initial set of data.
*Scoring* is the process of applying model scores to new data.

Data mining is largely about creating models. As used in this class, a *model* is anything that takes data in and produces a useful result, such as a score, an estimate, a classification, or a prediction. Models are meant to reflect something about how the world works. Changes in the inputs should produce changes in the output that are similar to what actually happens when those things change in the real world.

Sometimes the output of the model is the most important thing, and the process by which the outcome is produced is of little interest. Other times, the output of the model is never used, but the insights gained by building the model are of vital importance. Either way, it is all considered data mining.
Sources of Data for Data Mining

Data mining uses data from the past to affect future action.

Data from the past  Future Action

All data comes from the past. The job of data mining is to use past data in order to make better decisions about future actions. This process assumes several things. The most important assumption is that the future will be similar enough to the past that lessons learned from past data will remain applicable in the future. Another important assumption is that there are actually rules and patterns to be discovered and that the available data contains the required information.
When data mining becomes well integrated into a company’s way of doing business, it is a continuous process. You can describe this process as the "virtuous cycle of data mining" because data mining creates new information that leads to new action. Each new action is designed as an experiment, so that the results can be properly assessed. These experiments create more data to be mined and suggest new business opportunities.

Although most circles have no clear beginning, this one starts with the business problem or question.

1. Correctly identifying the business problem is a crucial first step. (In the next chapter, you see several examples of how this can go wrong.)

2. The next step is to identify data that can help answer the question posed by the business problem. (Extracting actionable information from the data is the part of the cycle that is the course focus.)

3. The final step is to act on the information. Actions must be taken in such a way that the results can be measured. This means giving careful thought to the design of experiments.
Business Cases and Data Sets

Product Penetration
- Census data for 1,006 towns of New York State

Response Modeling and Customer Segmentation
- House file with past ordering behavior for a catalog retailer
- Census data about the counties where the customers reside

Product Affinities
- Purchase transactions from the catalog data to be mined for association rules

Survival Analysis
- Start and stop data with contract information for wireless phone subscribers

Data is available for download at www.data-miners.com/materials.htm.

During this course, you follow four business cases, each of which has one or more data sets associated with it. The data is real. However, in some cases, it is modified to protect clients' proprietary information.
2.1 Introduction

This Chapter...

- motivates the need for a data mining methodology by explaining some of the ways that data mining can go wrong without a plan
- introduces the 12 steps of the methodology
- illustrates each step with a story.
Patterns might not represent any underlying rule.

Some patterns reflect some underlying reality.
- The party that holds the White House tends to lose seats in Congress during off-year elections.

Others do not.
- When the American League wins the World Series, Republicans take the White House.
- Stars cluster in constellations.

Sometimes, it is difficult to tell without analysis.
- In U.S. presidential contests, the taller man usually wins.

Correlation does not imply causation.

Humans are good at seeing patterns. Presumably, the reason that humans developed an affinity for patterns is that patterns often reflect some underlying truth about the way the world works. The phases of the moon, the progression of the seasons, the constant alternation of night and day, and even the regular appearance of a favorite TV show at the same time on the same day of the week are useful because they are stable and therefore predictive. You can use these patterns to decide when it is safe to plant tomatoes and how to program the VCR. Other patterns clearly do not have any predictive power. If a fair coin lands on heads five times in a row, there is still a 50-50 chance that it will be tails on the sixth toss. The challenge for data miners is to determine which patterns are predictive and which are not.
According to the article “Presidential Candidates Who Measure Up,” by Paul Sommers in *Chance* (www.stat.duke.edu/chance/), Vol. 9, No. 3, Summer 1996, pp. 29-31, the following is a list of presidential heights for the last century:

<table>
<thead>
<tr>
<th>Year</th>
<th>Winner</th>
<th>Height (inches)</th>
<th>Runner-Up</th>
<th>Height (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>McKinley</td>
<td>67</td>
<td>Bryan</td>
<td>72</td>
</tr>
<tr>
<td>1904</td>
<td>T. Roosevelt</td>
<td>70</td>
<td>Parker</td>
<td>72</td>
</tr>
<tr>
<td>1908</td>
<td>Taft</td>
<td>72</td>
<td>Bryan</td>
<td>72</td>
</tr>
<tr>
<td>1912</td>
<td>Wilson</td>
<td>71</td>
<td>T. Roosevelt</td>
<td>70</td>
</tr>
<tr>
<td>1916</td>
<td>Wilson</td>
<td>71</td>
<td>Hughes</td>
<td>71</td>
</tr>
<tr>
<td>1920</td>
<td>Harding</td>
<td>72</td>
<td>Cox</td>
<td>Not available</td>
</tr>
<tr>
<td>1924</td>
<td>Coolidge</td>
<td>70</td>
<td>Davis</td>
<td>72</td>
</tr>
<tr>
<td>1928</td>
<td>Hoover</td>
<td>71</td>
<td>Smith</td>
<td>Not available</td>
</tr>
<tr>
<td>1932</td>
<td>F. Roosevelt</td>
<td>74</td>
<td>Hoover</td>
<td>71</td>
</tr>
<tr>
<td>1936</td>
<td>F. Roosevelt</td>
<td>74</td>
<td>Landon</td>
<td>68</td>
</tr>
<tr>
<td>1940</td>
<td>F. Roosevelt</td>
<td>74</td>
<td>Wilkie</td>
<td>73</td>
</tr>
<tr>
<td>1944</td>
<td>F. Roosevelt</td>
<td>74</td>
<td>Dewey</td>
<td>68</td>
</tr>
<tr>
<td>1948</td>
<td>Truman</td>
<td>69</td>
<td>Dewey</td>
<td>68</td>
</tr>
<tr>
<td>1952</td>
<td>Eisenhower</td>
<td>70.5</td>
<td>Stevenson</td>
<td>70</td>
</tr>
<tr>
<td>1956</td>
<td>Eisenhower</td>
<td>70.5</td>
<td>Stevenson</td>
<td>70</td>
</tr>
<tr>
<td>1960</td>
<td>Kennedy</td>
<td>72</td>
<td>Nixon</td>
<td>71.5</td>
</tr>
<tr>
<td>1964</td>
<td>Johnson</td>
<td>75</td>
<td>Goldwater</td>
<td>72</td>
</tr>
<tr>
<td>1968</td>
<td>Nixon</td>
<td>71.5</td>
<td>Humphrey</td>
<td>71</td>
</tr>
<tr>
<td>1972</td>
<td>Nixon</td>
<td>71.5</td>
<td>McGovern</td>
<td>73</td>
</tr>
<tr>
<td>1976</td>
<td>Carter</td>
<td>69.5</td>
<td>Ford</td>
<td>72</td>
</tr>
<tr>
<td>1980</td>
<td>Reagan</td>
<td>73</td>
<td>Carter</td>
<td>69.5</td>
</tr>
<tr>
<td>1984</td>
<td>Reagan</td>
<td>73</td>
<td>Mondale</td>
<td>70</td>
</tr>
<tr>
<td>1988</td>
<td>Bush</td>
<td>74</td>
<td>Dukakis</td>
<td>68</td>
</tr>
<tr>
<td>1992</td>
<td>Clinton</td>
<td>74</td>
<td>Bush</td>
<td>74</td>
</tr>
<tr>
<td>1996</td>
<td>Clinton</td>
<td>74</td>
<td>Dole</td>
<td>74</td>
</tr>
<tr>
<td>2000</td>
<td>Bush</td>
<td>71</td>
<td>Gore</td>
<td>73</td>
</tr>
</tbody>
</table>

For the last century, the ratio is 16 taller versus 10 shorter (61% to the taller). If you include the cases where the candidates have the same height and consider the popular vote as opposed to the electoral outcome, then the tally is 21/26 (81%). (Gore had a higher popular vote in 2000.)

Copyright © 2009 by Data Miners, Inc.
The model set is the collection of historical data that is used to develop data mining models. For inferences drawn from the model set to be valid, the model set must reflect the population that the model is meant to describe, classify, or score. A sample that does not properly reflect its parent population is biased. Using a biased sample as a model set is a recipe for learning things that are not true. It is also difficult to avoid. Getting a truly representative sample requires effort. One explanation for the “Dewey Defeats Truman” headline is that telephone poles were biased, because in 1948 more Republicans than Democrats had telephones. Although the polling industry learned how to better estimate results for the population at large based on a sample, biases regularly occur when taking a poll over the Internet or by radio, and today there is again a sizeable number of households without landline telephones.
More than one novice data miner discovered that usage declines in the month before a customer stops using a service. This chart shows monthly usage for a subscriber who appears to fit the pattern. Appearances can be deceiving. Looking at minutes of use by day instead of by month shows that the customer continued to use the service at a constant rate until the middle of the month and then stopped completely. Presumably on that day, he or she began to use a competing service. The putative period of declining usage does not actually exist and so certainly does not provide a window of opportunity for retaining the customer. What appears to be a leading indicator is actually a trailing one.
Why Have a Methodology?

- To avoid learning things that are not true
- To avoid learning things that are not useful
  - Results that arise from past marketing decisions
  - Results that you already know
  - Results that you already should know
  - Results that you are not allowed to use
- To create stable models
- To avoid making the mistakes that you made in the past
- To develop useful tips from what you learned

The methodology presented here evolved from our experience. Early in our careers, we learned that a focus on techniques and algorithms does not lead to success. The methodology is an attempt by Data Miners Inc. to consolidate the “tricks of the trade” that were formulated over the years.

Data mining should provide new information. Many of the strongest patterns in data represent things that are already known. People over retirement age tend not to respond to offers for retirement savings plans. People who live where there is no home delivery do not become newspaper subscribers. Even though they might respond to subscription offers, service never starts. For the same reason, people who live where there are no cell towers tend not to purchase cell phones.

Often, the strongest patterns reflect business rules. Not only are these patterns uninteresting, their strength might obscure less obvious patterns.

Learning things that are already known does serve one useful purpose. It demonstrates that, on a technical level, the data mining effort is working and the data is reasonably accurate. This can be reassuring.
This 12-step methodology is intended to cover the basic steps needed for a successful data mining project. The steps occur at very different levels, with some concerned with the mechanics of modeling while others are concerned with business needs.

As shown in the slide, the data mining process is best thought of as a set of nested loops rather than a straight line. The steps do have a natural order, but it is not necessary or even desirable to completely finish with one before moving to the next. Things learned in later steps can cause earlier ones to be revisited. The remainder of this chapter looks at these steps one by one.
1) Define the Business Goal

- Improve the response rate for a direct marketing campaign.
- Increase the average order size.
- Determine what drives customer acquisition.
- Forecast the size of the customer base in the future.
- Choose the right message for the right groups of customers.
- Target a marketing campaign to maximize incremental value.
- Recommend the next, best product for existing customers.
- Segment customers by behavior.

A lot of good statistical analysis is directed toward solving the wrong business problems.

A data mining project should seek to solve a well-defined business problem. Wherever possible, broad, general goals such as “understand customers better” should be divided into more specific ones such as the ones on the slide.
Define the Business Goal

Example: Who is the yogurt lover?
What is a yogurt lover?
  - One answer is coupons printed at the cash register.
  - Another answer is coupons mailed to people’s homes.
  - Another is advertising.

Through a misunderstanding of the business problem, we created a model that gave supermarket loyalty card holders a yogurt-lover score based on the likelihood that they would be in the top tercile for both absolute dollars spent on yogurt and yogurt as a proportion of total shopping.

The model got good lift, and we were pleased. The client, however, was disappointed. “Who is the yogurt lover?” the client wanted to know. “Someone who gets a high score from the yogurt lover model” was not considered a good answer. The client wanted something similar to the following: “The yogurt lover is a woman between the ages of $X$ and $Y$ living in a ZIP code where the median home price is between $M$ and $N$.” A description such as that could be used for deciding where to buy advertising and how to shape the creative content of ads. Data Miners’ description, based as it was on shopping behavior rather than demographics, could not.
Big Challenge: Defining a Yogurt Lover

“Yogurt Lover” is not in the data. You can impute it, using business rules:
- Yogurt lovers spend a lot of money on yogurt.
- Yogurt lovers spend a relatively large amount of their shopping dollars on yogurt.

Define the Business Goal

What is the business problem?

Initial problem: Assign a churn score to all customers.
- Recent customers with little call history
- Telephones? Individuals? Families?
- Voluntary churn versus involuntary churn

How will the results be used?
Better problem: “By 24 September, provide a list of the 10,000 Elite customers who are most likely to churn in October.”

The new problem is much more actionable.

An instruction to calculate churn scores for all customers led to many questions. Only after understanding that the business goal was actually to choose candidates for a retention program aimed at members of the Elite Customer Club was it possible to refine the goal into something more actionable.
Define the Business Goal

When did a customer leave?

- When she has not made a new charge in six months?
- When she had a zero balance for three months?
- When the balance does not support the cost of carrying the customer?
- When she cancels her card?
- When the contract ends?

Before building a credit card attrition model, you must agree on the definition of attrition. There is no correct answer. Your choice depends on the nature of the actions that the model is designed to support. Typically, it is the data miner's responsibility to define attrition. That is a business decision, not an analytic one.

In the wireless industry, many customers begin with one-year contracts. When the year is finished, many of them leave, which is a phenomenon known as anniversary churn. For one client, anniversary churn seemed very high, with many people leaving on the day that they became eligible.

Is the cancellation date really the correct variable to examine? By looking at call center data, it became clear that many subscribers first placed a call to cancel months earlier but they were told that they were required to keep the service until the year ended.
2) Translate Business Problems into Data Mining Tasks

Do you already know the answer?

In directed data mining, the data has examples of what you are looking for, such as the following:
- customers who responded in the past
- customers who stopped
- transactions identified as fraud

In undirected data mining, you are looking for new patterns, associations, and ideas.

Data mining often comes in two flavors, directed and undirected. Directed data mining is about learning from known past examples and repeating this learning in the future. This is predictive modeling, and most data mining falls into this category. Undirected data mining is about spotting unexpected patterns in the data. To be useful, these unexpected patterns must be assessed for relevance.

Data Mining Goals Lead to Tasks and Techniques

<table>
<thead>
<tr>
<th>Goals</th>
<th>Tasks</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Acquisition</td>
<td>Exploratory Data Analysis</td>
<td>Decision Trees</td>
</tr>
<tr>
<td>Credit Risk</td>
<td>Binary Response Modeling</td>
<td>Regression</td>
</tr>
<tr>
<td>Pricing</td>
<td>Multiple Response Modeling</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>Customer Churn</td>
<td>Estimation</td>
<td>Survival Analysis</td>
</tr>
<tr>
<td>Fraud Detection</td>
<td>Forecasting</td>
<td>Clustering</td>
</tr>
<tr>
<td>Discovery</td>
<td>Detecting Outliers</td>
<td>Association Rules</td>
</tr>
<tr>
<td>Customer Value</td>
<td>Pattern Detection</td>
<td>Link Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hypothesis Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Visualization</td>
</tr>
</tbody>
</table>

To translate a business problem into a data mining problem, it should be formulated as one of the seven tasks mentioned on the slide.

The described data mining techniques provide the tools for addressing these tasks. No single data mining tool or technique is equally applicable to all tasks.
Example: Maximizing Profits

Example from the KDD Cup, a data mining competition associated with the KDD conference (www.sigkdd.org):

- Purpose: Maximizing profit for a charity fundraising campaign
- Tested on actual results from mailing (using data withheld from competitors)

Multiple approaches to the modeling:

- Modeling who will respond
- Modeling how much people will give
- Perhaps more esoteric approaches

However, the top three winners all took the same approach (although they used different techniques, methods, and software).
This example is taken from a data mining contest that was held in conjunction with the 1998 KDD Conference (knowledge discovery and data mining). What separated the winners from the losers was not the algorithms that they used or the software that they employed, but how they translated the business problem into a data mining problem.

The business problem was to maximize donations to a charity. The data was an historical database of contributions.

Exploring the data revealed the first insight. When someone contributes more often, the individual contributes less money each time. Usually, you think that the best customers are those who are the most frequent. In this case, though, it seems that people plan their charitable giving on a yearly basis. They might give it all at one time, or space it over time. More checks do not mean more money.

This insight led to the winning approach. People seem to decide how much money to give separate from whether they will respond to the campaign. This suggests two models:

- a model built on a training set that contains both contributors and non-contributors to determine who will respond to the campaign
- a model built only on contributors to estimate how much they will give

The three winning entries took this approach of combining models.
An Unexpected Pattern

An unexpected pattern suggests an approach.

*When people give money frequently, they tend to donate less money each time.*

- In most business applications, as people take an action more often, they spend more money.
- Donors to a charity are different.

This suggests that potential donors go through a two-step process:

- Shall I respond to this mailing?
- How much money should I give this time?

Modeling can follow the same logic.

3) Select Appropriate Data

What is available?
What is the right level of granularity?
How much data is needed?
How much history is required?
How many variables should be used?
What must the data contain?

Assemble results into customer signatures.

Of course, data mining depends on the data that is available. It is worth asking questions about the data to determine what types of questions it can answer.
Is the Right Data Available?

Elevated printing defect rates might be due to humidity, but that information is not in press run records.

Poor coverage might be the number-one reason for wireless subscribers to cancel their subscriptions, but data about dropped calls is not in billing data.

Customers might already have potential cross-sell products from other companies, but that information is not available internally.

In some cases, the right data might not be available. There is sometimes the option of purchasing external data; however, this rarely fills all the gaps. A company’s data about its own customers is a key competitive advantage that other companies do not have.
Types of Attributes in Data

Readily Supported
- Binary
- Categorical (nominal)
- Numeric (interval)
- Date and time

These are the standard attributes for data stored in data sets or databases.

Require More Work
- Text
- Image
- Video
- Links

These are case studies for text, image, and links in the chapters on memory-based reasoning (MBR), link analysis, and genetic algorithms.

Categorical columns contain discrete information, such as product type, county, credit class, and so on. SAS calls these nominal, after the Latin word for name. Notice that class columns might be represented as numbers (ZIP codes, product IDs, and so on). However, computation on these numbers makes no sense.

Binary columns, a special type of categorical columns, take on exactly two values. These are particularly important as targets. Notice that sometimes the two values are “something” and missing (particularly when data is appended from outside vendors).

Numeric columns contain numbers. SAS calls these interval because subtraction (the interval between two values) is defined.

Date/Time columns represent dates, times, and spans of time between them. SAS calls these intervals as well.
The customer signature is the basic unit of modeling. Each row describes a single customer (or whatever we are interested in). The columns contain features about the customer.
Chapter 17 of *Data Mining Techniques* is devoted to constructing customer signatures and preparing data for mining.

Some of the operations typically used to assemble a customer signature are listed on the slide. These operations are needed to take data that is stored at many levels of granularity and bring it all to the customer level.

Some data (credit score and acquisition source, for example) is already stored at the customer level and can be copied.

Some data (billing data, for example) is stored by month. In order to create a time series in the customer signature, such data must be pivoted.

Some data is stored as individual transactions. Such data must be aggregated to the customer level. Often, the aggregation is performed to the monthly level and the summarized data is then pivoted.

Some data finds its way into the customer signature through lookup tables. For example, the customer’s ZIP code might be used to look up such things as the median home price, median rent, and population density.

Still other values do not exist anywhere in the source data and are summarized from the available data, such as the historical sales penetration within a ZIP code or the churn rate by handset type.

Finally, many elements of the customer signature are derived from other elements. Examples include trends, ratios, and durations.
4) Explore the Data

Examine distributions.
- Study histograms.
- Think about extreme values.
- Notice the prevalence of missing values.

Compare values with descriptions.
Validate assumptions.
Ask many questions.

A data exploration demonstration occurs soon.

Good data miners seem to rely heavily on intuition, for example, somehow being able to guess what a good derived variable to try might be. The only way to develop intuition for what is happening in an unfamiliar data set is to immerse yourself in it. Along the way, you are likely to discover many data quality problems and be inspired to ask many questions that would not otherwise arise.

A good first step is to examine a histogram of each variable in the data set and think about what it tells you. Make a note of anything that seems surprising.

If there is a state code variable, is California the tallest bar? If not, why not? Are some states missing? If so, does it seem plausible that this company does not do business in those states? If there is a gender variable, are there similar numbers of men and women? If not, is that unexpected?

Pay attention to the range of each variable. Do variables that should be counts take on negative values? Do the highest and lowest values seem to be reasonable values for that variable to have? Is the mean much different from the median? How many missing values are there? Are the variable counts consistent over time?
2.1 Introduction

Ask Many Questions

Why were some customers active for 31 days in February, but none were active for more than 28 days in January?

How do some retail card holders spend more than $100,000/week in a grocery store?

Why were so many customers born in 1911? Are they really that old?

Why do Safari users never make second purchases?

What does it mean when the contract begin date is after the contract end date?

Why are there negative numbers in the sale price field?

How can active customers have a non-null value in the cancellation reason code field?

These are real questions we had occasion to ask about real data. Sometimes the answers revealed facts that we did not know about the client’s industry. At the time, New Jersey and Massachusetts did not allow automobile insurers much flexibility in setting rates, so a company that sees its main competitive advantage as smarter pricing does not want to operate in those markets. Other times we learned about idiosyncrasies of the operational systems. For example, a data entry screen insisted on a birthdate even when none was known, which led to many people being assigned the birthday of November 11, 1911 because 11/11/11 is the date you get by holding down the 1 key and auto-repeating until the field is full. Sometimes we discovered serious problems with the data such as the data for February being misidentified as January. In the last example, we learned that the process extracting the data had bugs.
The fact that a variable keeps the same name over time does not mean that its meaning stays the same. Credit classes might always be A, B, C, and D, but the threshold credit scores might change over time. Similarly, the definition of sales regions might change over time. Policy changes might cause changes in customer behavior and so on. Looking at how data values change over time can reveal some of these things. This graph suggests that there might be a price increase near the beginning of February.
5) Create a Model Set
- Set up a proper temporal relationship between the target variable and inputs.
- Create a balanced sample.
- Include multiple timeframes.

The model set contains all the data that is used in the modeling process. Some of the data in the model set is used to find patterns. Some of the data in the model set is used to verify that the model is stable. Some is used to assess the model’s performance. Creating a model set requires assembling data from multiple sources to form customer signatures and then preparing the data for analysis.

Modeling is done on flat data sets that have one row per item to be modeled. Generally, these rows are called customer signatures because in analytic customer relationship management (CRM), most analysis is at the customer level. Sometimes, however, another level is appropriate. If customers are anonymous, analysis might have to be done at the order level. If the relationship is with entire families, the household level might be most appropriate. This decision must be made before signatures can be constructed.
Prediction or Profiling?

The same techniques work for both.

Earlier

In a predictive model, values of explanatory variables are from an earlier timeframe than the target variable.

Later

Same Timeframe

In a profiling model, the explanatory variables and the target variable might all be from the same timeframe.

When you build a predictive model, data from the distant past is used to explain a target variable in the recent past. This separation of timeframes is very important; it mimics the gap that exists when the model is deployed and used. When a predictive model is used, data from the recent past is used to predict future values of the target variable.
Data mining algorithms find patterns in preclassified data. If you look at the same training data, the model becomes more complex (more leaves in a decision tree; more iterations of training for a neural network; higher degree polynomials to fit a curve), and it appears to fit the training data better. Unfortunately, it will be fitting noise as well as a signal. This can be seen by applying models to a second preclassified data set. In this graph, the error rate is increasing on the validation data as the model becomes overly complex, although it is still decreasing on the training data.

The tendency of models to memorize or *overfit* the training data at the expense of stability and generality is a topic that is discussed in the chapters on decision trees and neural networks.
When you build models, the model set is often partitioned into several sets of data.

- The first part, the *training set*, is used to build the initial model.
- The second part, the *validation set*, is used by the modeling algorithm to adjust the initial model to make it more general and less tied to the idiosyncrasies of the training set; some data mining algorithms use a test set, some do not.
- The third part, the *test set*, is used to gauge the likely effectiveness of the model when applied to unseen data.

Three sets are necessary because after data is used for one step in the process, it can no longer be used for the next step because the information it contains already became part of the model. Therefore, it cannot be used to correct or judge.

For predictive models that will be scored in the future, the test set should also come from a different time period than the training and validation sets. The proof of a model’s stability is in its ability to perform well month after month. A test set from a different time period, often called an *out of time* test set, is a good way to verify model stability, although such a test set is not always available.
Balancing the Model Set

A very accurate model simply predicts that no one wants a brokerage account:

- 98.8% accurate
- 1.2% error rate

This is useless for differentiating among customers.

Stratified sampling and weights (called frequencies in SAS Enterprise Miner) are two ways of creating a balanced model set. With stratified sampling, the model set has fewer records, which is often desirable. With weights or frequencies, every record counts, but the common ones count less than the rare ones.

Using stratified sampling or weights to balance the outcomes in the model forces the model to distinguish characteristics of the rare class rather than simply ignoring it. It is easy to return to the actual distribution of classes and probabilities after the fact.
### Explanatory Variables

Explanatory variables might form a time series within a customer signature.

The numbers represent the number of time periods before the excluded time period.

The target variable is recorded after the excluded period.

Excluding the time period before the target variable is recorded allows time for model deployment.

Excluding the time period before the target variable is recorded avoids “leakage” of future information.

---

### Dealing with Seasonality

Different rows of the model set can be taken from different time periods to avoid learning seasonal effects.

Seasonal information is added back using indicator variables.

- Back-to-school flag
- On promotion flag
A stable model works on both the model set and the score set.

6) Fix Problems with the Data
Data imperfectly describes the features of the real world.
- Data might be missing or empty.
- Samples might not be representative.
- Categorical variables might have too many values.
- Numeric variables might have skewed distributions and outliers.
- Meanings can change over time.
- Data might be encoded inconsistently.

All data is dirty. All data has problems. What is or is not a problem varies with the data mining technique. For some, such as decision trees, missing values and outliers do not cause too much trouble. For others, such as neural networks, they cause trouble. (For that reason, some of what we say about fixing problems with data is in the chapters on the techniques where the problems cause the most difficulty.)
No Easy Fix for Missing Values

Throw out the records with missing values?
- No. This creates a bias for the sample.

Replace missing values with a “special” value (-99)?
- No. This resembles any other value to a data mining algorithm.

Replace with some “typical” value?
- No. Replacement with the mean, median, or mode changes the distribution.

Impute a value? (Imputed values should be flagged)
- Maybe. Use distribution of values to randomly choose a value.
- Maybe. Model the imputed value using some technique.

Use data mining techniques that can handle missing values?
- Yes. One of these, decision trees, is discussed.

Partition records and build multiple models?
- Yes. This action is possible when data is missing for a canonical reason, such as insufficient history.

Some data mining algorithms are capable of treating **missing** as a value and incorporating it into rules. Others cannot handle missing values, unfortunately. None of the obvious solutions preserves the true distribution of the variable. Throwing out all records with missing values introduces bias because it is unlikely that such records are distributed randomly.

The IMPUTE node in SAS Enterprise Miner simplifies the task of replacing missing values, but with what? Replacing the missing value with some likely value such as the mean or the most common value adds spurious information. Replacing the missing value with an unlikely value is even worse because the data mining algorithms will not recognize that. For example, 999 is an unlikely value for age. The algorithms will use it. Another possibility is to replace the value with a common value, such as median or mean, or to pull a value from the distribution of known values. It is even possible to predict the missing value using a decision tree or neural network. Under some circumstances, this can be the only thing to do.

It is sometimes useful to distinguish between two different situations that can both lead to a null value in a database or in a SAS data set. A value is said to be **missing** if there is a value but it is unknown. An empty value occurs when there simply is no value for that attribute.

In the decision trees chapter, two approaches to building models in the presence of missing values are discussed:
- using surrogate splits
- treating missing as a legitimate value
After the model set is assembled and major data problems are fixed, the data must still be prepared for analysis. This involves adding derived fields to bring information to the surface. It might also involve removing outliers, binning numeric variables, grouping classes for categorical variables, applying transformations such as logarithms, and turning counts into proportions. Data preparation is such an important topic that a colleague, Dorian Pyle, wrote *Data Preparation for Data Mining* about it.

### Default Transformations

#### Standardize numeric values.
- All numeric values are replaced by the notion of “how far is this value from the average.”
- Conceptually, all numeric values are in the same range. (The actual range differs, but the meaning is the same.)
- Standardization basically works for all techniques.
- Although it often has no effect on the results (such as for decision trees and regression), it never produces worse results.
- Standardization is so useful that it is often built into SAS Enterprise Miner modeling nodes.

#### Replace categorical values with appropriate numeric values.
- Many techniques work better with numeric values than with categorical values.
- Historical projections (such as handset churn rate, penetration by ZIP code, and so on) are particularly useful.
8) Build (Train) Models

- Regression
- Decision trees
- Neural networks
- Memory-based reasoning
- Cluster detection
- Survival analysis
- Link analysis
- Association rules
- Genetic algorithms

Build models by applying a data mining technique to the model set.

The remainder of the class is primarily devoted to understanding these techniques and how they are used to build models.

9) Assess Models

Score the model and then measure the results.

Measures
- Misclassification rate
- Variance (mean squared error)

Pictures
- Classification table (confusion matrices)
- Lift charts
- Cumulative gains charts
- ROC curves
- Score ranking charts

There are a number of different ways to assess models. All of these work by comparing predicted results with actual results in preclassified data that was not used as part of the model building process.
Assessing Starts With Scoring

There are various ways to score models:
- Score in SAS Enterprise Miner by connecting a Score node to a model and data to score.
- Produce code in SAS, C, or Java to score remotely.

SAS Enterprise Miner can produce code for all transformations in a diagram.

For some types of models, such as decision trees and regression, the result can be a set of rules or equations that can be expressed in English or SQL.

You can then assess the model on the training, validation, and test sets.

Binary/Categorical Target Misclassification Rate

<table>
<thead>
<tr>
<th>TARGET</th>
<th>Fit statistics</th>
<th>Statistics Label</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESPOND</td>
<td>NOBS</td>
<td>Sum of Frequencies</td>
<td>12520</td>
<td>6389</td>
<td>6394</td>
</tr>
<tr>
<td>RESPOND</td>
<td>SUMW</td>
<td>Sum of Case Weights TimesFreq</td>
<td>25056</td>
<td>17776</td>
<td>16706</td>
</tr>
<tr>
<td>RESPOND</td>
<td>MISC</td>
<td>Misclassification Rate</td>
<td>0.204004</td>
<td>0.310046</td>
<td>0.320056</td>
</tr>
<tr>
<td>RESPOND</td>
<td>MAX_</td>
<td>Maximum Absolute Error</td>
<td>0.860529</td>
<td>0.860529</td>
<td>0.860529</td>
</tr>
<tr>
<td>RESPOND</td>
<td>SSE_</td>
<td>Sum of Squared Errors</td>
<td>4725.037</td>
<td>3577.705</td>
<td>3413.863</td>
</tr>
<tr>
<td>RESPOND</td>
<td>ASE_</td>
<td>Average Squared Error</td>
<td>0.050539</td>
<td>0.201243</td>
<td>0.204523</td>
</tr>
<tr>
<td>RESPOND</td>
<td>RASE_</td>
<td>Root Average Squared Error</td>
<td>0.454237</td>
<td>0.446092</td>
<td>0.452023</td>
</tr>
<tr>
<td>RESPOND</td>
<td>DYN_</td>
<td>Deviation for ASE</td>
<td>25056</td>
<td>17776</td>
<td>16706</td>
</tr>
<tr>
<td>RESPOND</td>
<td>DFF_</td>
<td>Total Degrees of Freedom</td>
<td>12520</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the object of a model is to classify things, the misclassification rate is a simple measure of how well the classifier performs. It measures the number of times that the classifier is correct on each of the partitions of the model set. The misclassification rate is one of several statistics provided in the fit statistics data set that is created as one of the results of building a model.
A classification table or confusion matrix provides more information than the simple misclassification rate. In many problems, the cost of a classification error is different for each value of the target variable. For example, in medical diagnosis, a false negative is more dangerous than a false positive because it might lead to not treating a serious illness.

SAS Enterprise Miner displays a classification table as a stacked bar chart. In this example, the two classes are balanced perfectly, so the two bars are the same height. The red area of each bar shows where the model made a correct classification. (This is the bottom part of the stacked bar on the left where the actual value is 0 and the top half of the stacked bar on the right where the actual value is 1.) The model shown here does a better job of predicting target=0 than it does predicting target=1.
2.1 Introduction

In this case, a table is usually more informative than the chart.

In SAS Enterprise Miner, there is an icon to change charts into tables.

---

**Binary/Categorical Target Classification Table**

In this case, a table is usually more informative than the chart.

In SAS Enterprise Miner, there is an icon to change charts into tables.

---

**Binary Target Cumulative Gains Chart**

---
Lift is another popular way of assessing models. *Lift* compares the probability of finding a member of the desired class in a group selected by the model to the prevalence of that class in the population. To create a lift chart such as the one shown, a list is sorted according to the model score with the records that score highest at the top of the list. Lift is therefore highest in the first deciles and declines to 1.0 when the entire list is taken.

The chart shown is for a model built on a balanced model set where 50% of the records are in the target class. The chart shows that at the first decile, the model has a lift of 1.5, which means that there are one and a half times as many records belonging to the target class than would be found by chance. The two lines show what happens when the model is applied to the training set and to the validation set. The fact that the lines are close together means that the model is stable. The behavior on the validation set is similar to that on the training set that was used to create the model.

A concentration chart is another way of looking at the same information. You can see the cumulative percent of the target class found as you go deeper and deeper into the list. The 45-degree line shows what happens if the model has no power. The gap between that baseline and the concentration curve shows that the model provides some benefit. The point where the vertical distance between the two lines is maximized is called the *point of maximum benefit*. This is also the point that maximizes the difference between the distribution of outcome classes between the two lists created by partitioning the sorted list at that point.
2.1 Introduction

Binary Target
Score Rankings

Binary/Categorical Target
Misclassification costs might be unequal.

<table>
<thead>
<tr>
<th>Outcome To</th>
<th>Data File</th>
<th>Target Var</th>
<th>Target</th>
<th>Outcome</th>
<th>Correct Test</th>
<th>Target Pred</th>
<th>Outcome Pred</th>
<th>Count</th>
<th>Total Pct</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTION</td>
<td>VALIDATE</td>
<td>RESPONSE</td>
<td>0</td>
<td>0</td>
<td>Correct</td>
<td>86,898.06</td>
<td>27,843.18</td>
<td>398</td>
<td>39.8038</td>
<td>5</td>
</tr>
<tr>
<td>PREDICTION</td>
<td>VALIDATE</td>
<td>RESPONSE</td>
<td>1</td>
<td>0</td>
<td>Incorrect</td>
<td>36,116.72</td>
<td>37,372.57</td>
<td>373</td>
<td>18.9567</td>
<td>0</td>
</tr>
<tr>
<td>PREDICTION</td>
<td>VALIDATE</td>
<td>RESPONSE</td>
<td>1</td>
<td>1</td>
<td>Correct</td>
<td>40,784.19</td>
<td>62,154.94</td>
<td>626</td>
<td>31.3943</td>
<td>1</td>
</tr>
</tbody>
</table>

For a medical diagnosis, a false negative is worse than a false positive.

SAS Enterprise Miner enables you to specify a cost matrix and to optimize models on misclassification cost rather than on a misclassification rate.
The average squared error (ASE) is another of the statistics provided in the **fit statistics** result data set when a model is built in SAS Enterprise Miner. This measure is most useful for evaluating an **estimator**, which is a model that estimates the value of a numeric target variable. The mean squared error is often called the **variance**. Its square root, reported by SAS Enterprise Miner as root average squared error (RASE), is also called the **standard deviation**. The RASE has the advantage of being measured in the same units as the target variable itself.
Numeric Target Score Rankings Chart

Rankings are shown for deciles (or 5%-tiles).
Both the actual and predicted target mean are shown.
For relative scoring, the lines should monotonically decrease (showing differentiation across different score groups).
For financial scoring, the lines should decrease sharply and be close together.
Models are meant to be used. Depending on the environment, that might mean scoring customers every month, every day, or every time that they initiate a transaction. In this picture, it might mean creating rules that are triggered when a customer looks at a particular item on a retail Web page.

Deploying a model means moving it from the data mining environment to the scoring environment. This process might be easy or difficult. In the worst case, the model is developed in a special modeling environment using software that runs nowhere else. To deploy the model, a programmer takes a printed description of the model and recodes it in another programming language so that it can be run on the scoring platform.

A more common problem is that the model uses input variables that are not in the original data. This should not be a problem because the model inputs are at least derived from the fields that were originally extracted from the model set. Unfortunately, data miners are not always good about keeping a clean, reusable record of the transformations that they applied to the data.
11) Assess Results

Design experiments.

- Compare actual results against expectations.
- Compare the challenger's results against the champion's.
- Did the model find the right people?
- Did the action affect their behavior?
- What are the characteristics of the customers most affected by the intervention?

The proof of the pudding is in the eating!

The real test of data mining comes when you can measure the value of the actions you took as a result of the mining. Measuring lift on a test set helps you to choose the right model. Profitability models based on lift can help you to decide how to apply the results of the model. However, it is very important to measure these things in the field as well. In a database marketing application, this requires always setting aside control groups and carefully tracking customer response according to various model scores.

When you think about designing an experiment to assess results, the following are some things to remember:

- the right size for the test
- being certain that any test groups are chosen randomly and receive the same treatment (same message, same offer, same timing, similar customers)
- being sure that operational systems can handle the process
Good Test Design Measures the Impact of Both the Message and the Model

<table>
<thead>
<tr>
<th>Message</th>
<th>“Control” Group</th>
<th>“Target” Group</th>
<th>Modeled Holdout</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Chosen at random; receives message.</td>
<td>Chosen by model; receives message.</td>
<td>Chosen by model; receives no message.</td>
</tr>
<tr>
<td>YES</td>
<td>Response measures message without model.</td>
<td>Response measures message with model.</td>
<td>Response measures model without message.</td>
</tr>
</tbody>
</table>

Impact of model on group getting message
Impact of message on group with good model scores

“Holdout” Group
Chosen at random; receives no message.
Response measures background response.

Picked by Model
NO
YES
This graph is taken from the assessment of an e-mail campaign by a bank. The e-mail was sent to customers who gave permission to be contacted in this manner. The e-mail suggested that they sign up for a particular product. Ten thousand people were selected at random to receive the e-mail. This is the control group, which had a 0.2% response rate. Another 10,000 people were sent the e-mail because they were scored by the model as likely to be interested in the product. This group responded at the rate of 0.7%, or three and a half times the rate of the control.

Clearly, the model did something. It found a group of people more likely to sign up for the product than the average customer. Was it the e-mail that caused them to sign up, or did the model simply identify people who were more likely to sign up with or without the e-mail? To test that, the bank also tracked the take-up rate of people who did not get the e-mail. It was virtually nil, so both the model and the message had an effect.
12) Begin Again

New hypotheses to test
New data
- Model scores
- Cluster assignments
- Treatments
- Responses

Example:
A model discovers that geography is a good predictor of churn.
- What do the high churn geographies have in common?

Every data mining project raises more questions than it answers. This is a good thing. It means that new relationships are now visible that were not visible before. The newly discovered relationships suggest new hypotheses to test, and the data mining process begins again.
Lessons Learned

Data miners must be careful to avoid pitfalls.

- Learning things that are not true or not useful
- Confusing signal and noise
- Creating unstable models

A methodology is a way of being careful.

Data mining comes in two forms.

- **Directed data mining** searches through historical records to find patterns that explain a particular outcome. Directed data mining includes the tasks of classification, estimation, prediction, and profiling.

- **Undirected data mining** searches through the same records for interesting patterns. It includes the tasks of cluster, finding association rules, and description.

The primary lesson of this chapter is that data mining is full of traps for the unwary, and following a data mining methodology based on experience can help you avoid them.

The first hurdle is translating the business problem into one of the six tasks that can be solved by data mining: classification, estimation, prediction, affinity grouping, clustering, and profiling.

The next challenge is to locate appropriate data that can be transformed into actionable information. After the data is located, it should be explored thoroughly. The exploration process is likely to reveal problems with the data. It will also help to build the data miner’s intuitive understanding of the data.

The next step is to create a model set and partition it into training, validation, and test sets.

Data transformations are necessary for two purposes:

- to fix problems with the data such as missing values and categorical variables that take on too many values
- to bring information to the surface by creating new variables to represent trends and other ratios and combinations

After the data is prepared, building models is a relatively easy process.

Each type of model has its own metrics by which it can be assessed, but there are also assessment tools that are independent of the type of model. Some of the most important of these are the lift chart, which shows how the model has increased the concentration of the desired value of the target variable, and the confusion matrix, which shows the misclassification error rate for each of the target classes.
Chapter 3  Data Exploration

3.1  The Environment and Data Exploration .................................................................3-3
3.1 The Environment and Data Exploration

This Chapter...
- introduces some background for SAS Enterprise Miner
- introduces some of the exploratory tools available in SAS Enterprise Miner
- demonstrates the use of SAS Enterprise Miner as a data exploration tool by exploring the NYtowns data set.

Data Mining Tool: Characteristics
You might want the following in a data mining tool:
- Wide variety of techniques, including tools for data exploration and standard statistical techniques
- Flexible data management to get data in and out of the tool
- Visual display of results
- Ability to score models with the tool and without the tool
- Ability to handle large sets of data, including those too large to fit in memory
- Ability to export results and data in other formats as necessary (Excel, PowerPoint, HTML, databases)
- Speed and accuracy
- Good documentation and training
SAS Enterprise Miner is a powerful client/server application. You use it in single-user mode on a single machine.

SAS Enterprise Miner consists of several processing components, the important ones of which are shown here. The client software is what you use in the course. It communicates with two other servers. One is responsible for doing the work (the server). The other is responsible for maintaining the data and information about the data (the Metadata server).

The reason for having this architecture is flexibility. The architecture facilitates the development of diagrams on workstations and then enables you to run them on large servers. It also makes it possible to work on the same diagram from two different computers. (This functionality is not important during this course.)

SAS Enterprise Miner uses data that is ultimately stored in SAS data sets. These data sets contain the data and describe it at the lowest level, that is, the fields and their types. For data mining purposes, you typically want more information, such as the role that variables play in modeling (for example, target, input, frequency, and rejected) and the level of the variables (nominal, interval, ordinal). This additional information is stored in the Repository. In addition, the Repository stores diagrams and other information needed by the client.

More complete descriptions of these components are available in the online Help.
SAS Enterprise Miner’s user interface consists of seven areas that are of interest for data mining.

**SEMMA Methodology**

Nodes are arranged in an order.

- Sample
- Explore
- Model
- Modify
- Assess
Metadata in SAS Enterprise Miner

Data sets have roles, which describe how the SAS Enterprise Miner diagram processes them.
- Raw, Training, Validation, Test, Score, and so on

Data columns have data types, which describe how values are stored and how they are shown.
- FORMAT, INFORMAT, and so on

Data columns have levels, which describe what they measure.
- Nominal, Ordinal, Interval

Data columns have modeling roles, which describe how the columns are used by the nodes.
- Target, Input, Rejected, and so on

Getting Started with Data Exploration

- Data set size
- Exploring column definitions
- Exploring values in individual columns
- Exploring columns with relation to the target
- Exploring with respect to geography and time
- Exploring important subgroups of products and customers
- Ask questions
- Think about mean, median, mode, missing values, extreme values, and other values.

Use tools available in SAS Enterprise Miner.
- Stat Explore node
- Segment Profiler node
- Scatter plots
- Decision Tree node
3.1 The Environment and Data Exploration

Scatter Plots: Useful for Seeing Patterns

Tip: Create scatter plots using symbols with hollow interiors to see density. In SAS Enterprise Miner, any graph style other than the default (such as Analysis or Statistical) does this.

Segment Profiles

*Segment profiles* compare sizes and distributions among different groups.

Towns with large land areas are concentrated in the extreme north and south of New York.
Correlation (with Target) Chart

Correlation is the Pearson correlation, which measures the extent to which a change in the value of an input leads to a change in the target.

*Positive correlation* means that as the input increases, so does the target.

*Negative correlation* means that as the input increases, the target diminishes.

Variable Worth Chart

*Variable worth* captures some relationships missed by correlation. It measures the ability of the input to discriminate between classes or values of the target, and you see the same measure used for decision trees.

Variable worth is positive and unbounded.
Of course, geographic patterns such as the one found by the segment analyzer above are more easily seen on a map. A simple scatter plot by latitude and longitude makes a crude but recognizable map. The larger towns are in the Adirondacks, in the Southern Tier, and on Long Island.

Demo Available: Exploring the NYTOWNS Data Source

Refer to Exercises for Chapter 3 in the handout.
Chapter 4  Regression Models

4.1  Regression
4.1 Regression

This Chapter...
- introduces models using the most commonly used technique for building models, that is, regression
- describes models, using an example familiar to many students
- discusses linear and polynomial regression
- lists the dangers to consider when you use regression
- discusses logistic regression.
History

In the late 1700s, two astronomers (Johann Titius and Johann Bode) noticed a particular pattern among the positions of the planets:

<table>
<thead>
<tr>
<th></th>
<th>Prediction</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>0.4</td>
<td>0.39</td>
</tr>
<tr>
<td>Venus</td>
<td>0.7</td>
<td>0.72</td>
</tr>
<tr>
<td>Earth</td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>Mars</td>
<td>1.6</td>
<td>1.52</td>
</tr>
<tr>
<td>??</td>
<td>2.8</td>
<td>2.77</td>
</tr>
<tr>
<td>Jupiter</td>
<td>5.2</td>
<td>5.2</td>
</tr>
<tr>
<td>Saturn</td>
<td>10.0</td>
<td>9.54</td>
</tr>
</tbody>
</table>

One planet was missing between Mars and Jupiter. This object is called Ceres, the largest asteroid in the asteroid belt.

The race was on to find this missing planet.

In the 1770s, astronomers observed a pattern among the distances between the known planets and the sun. The chart on the slide shows the distances measured in astronomical units (AU). One AU is the distance between the earth and the sun. The pattern is simply that the $n^{th}$ planet is $0.1 \text{ AU} + 0.3 \times n$. This is an example of a linear relationship, at least to the precision available to astronomers in the 18th century.

However, this clean model of the solar system had a problem. The formula failed for Jupiter, because it was at the distance of the sixth planet rather than the fifth. The obvious conclusion was that another planet was waiting to be discovered. For 20 years, astronomers looked for such a planet.
In the auspicious year of 1800, an Italian astronomer, Joseph Piazzi, found the planet. Unfortunately, he found the planet in the winter months in northern Italy, and was only able to get three relatively accurate measurements of its position. In late February, the planet disappeared behind the sun, so it was no longer visible from the earth. The race was on among European astronomers to determine where it would reappear.

Carl Friedrich Gauss, whom some call the greatest mathematician who ever lived, had recently graduated from university and set about solving this problem. A big part of the challenge was the inaccuracy of the observed measurements. However, the three measurements actually "overfit" the data. The orbit of planets is an ellipse with the sun at one focal point; only two positions are needed to determine the ellipse. Gauss had three positions rather than two. He invented the idea of finding the ellipse that minimizes the sum of the squares of the distances between the observed points and the points on the ellipse.

This is the key idea behind least squares regression. You might assume that he chose least squares to simplify the computations. Why not simply choose the distance? In fact, this is possible, and it is called robust regression. However, finding the coefficients is much more computationally intensive than for ordinary least squares regression.
This example starts with a simple scatter plot of a sample of customers, comparing tenure versus the total amount paid.

One feature of this scatter plot is worth mentioning. The symbols used are hollow disks. This makes it possible to see areas where the points cluster, such as in the low tenure and low dollar amount areas. This is especially true when different colors are used to compare different groups of customers.

This plot shows an expected relationship between tenure and the amount paid, that is, when a customer has a long-term association with a business, the customer pays more. This is a simple relationship that suggests a linear relationship.
In fact, there is a line that fits the data closely. The formula for this line says that you multiply the tenure by $0.56 and subtract $10.14. For a tenure of 100, you expect a customer to pay $56 – $10.14 = $45.86.

This line has an important property. Of all possible lines, it is the one where the vertical distances from the observed data points to the line are as small as possible. The black lines show the distance from a few data points to this line. The distance used for this calculation is the geometric distances that you learned in high-school geometry. Fortunately, there are simple equations that determine the values of the coefficients. This method of finding the best line is known as the least squares method, because it depends on the formula for distances as the square root of the sum of the squares.

An alternative method is to ask the question, “Which line makes the observed data points the most likely?” This formulation uses a technique known as maximum likelihood estimation to calculate the probability of getting the observed values, assuming that the model is correct. In the case of a line, these two methods produce the same results. However, in other situations, it is not possible to use least squares, so another technique, such as maximum likelihood, is necessary.

The R-square value varies between 0 and 1 and it tells you how strong the relationship is. When the value is near 1, then the relationship is strong. When the value is near 0, then the relationship is weak. One way of thinking about R square is that it tells you the proportion of the pattern that is in the data. That is, the relationship between tenure and amount paid explains 89% of the relationship seen in the data.

R square is calculated by taking the ratio of the sum of the squared errors (distance of the $y$ values from the line) to something known as the total squared error (the sum of the squares of the difference between each $y$ value to the average $y$ value) and subtracting it from 1. In other words, R square tells you how much better the model’s estimate is than simply guessing the mean value.
With the formula, it is easy to calculate the residuals. These are the differences between the observed values and the expected values; that is, the differences between the data points and the line.

In general, the residuals should be symmetric around the $x$-axis. There should be as many high values as low values. They should also cluster near the $x$-axis, and hopefully not show any obvious pattern. For instance, as the $x$-values increase, you do not want to see the residuals becoming systematically larger. The biggest and smallest residuals occur about halfway through the data.
Statisticians have studied residuals since the 19th century. One result is that a good model should result in residuals that follow a normal distribution. If not, then there are likely to be other effects that are not accounted for.

This chart shows a histogram of the residuals and of the corresponding normal distribution with the same mean and standard deviation (mean = $0 and stddev = $24.16). The residuals are not following a normal distribution. This is especially notable because the peak is skewed a bit on the high side. This suggests, not surprisingly, that tenure alone does not fully explain spending.
A Model

\[
\text{amount} = 0.56 \times \text{tenure} - 10.34
\]

A line represents a simple model. It takes an input (in this case, tenure) and produces an output (in this case, amount paid). The following are some properties:

- The slope of the line represents change. Statisticians call this \textit{beta} and mathematicians call it \textit{m}.
- Applying the model produces an \textit{expected value}.
- How well the model fits is measured by R square. When this is close to 0, the fit is poor, and when it is close to 1, it is very good.
- The averages of the expected values are the same as the averages of the original data points.
- The inverse relationship is easy to calculate.

The best fit provides a good example of a simple model. The model takes an input (tenure) and produces an output (amount). Such a model has several interesting properties.

First, the slope of the line gives the amount of change in the \textit{y}-variable for each unit of the \textit{x}-variable. This is particularly useful when the \textit{x}-variable represents something over time.

Statisticians refer to this value as the \textit{beta} value, after the coefficient in the formula for lines \((y = \beta x + a)\), and mathematicians refer to it as \textit{m}, after the coefficient \((y = mx + b)\). With this formula, you can calculate the expected value of the amount paid for any tenure.

How good is the model? One measure is the R-square value, which is a measure of how much better the model is than simply using the mean value as the estimate. When the actual points are close to the line, the numerator is small and R square is near 1. When the actual points are far from the best fit line, the numerator is large and R square is close to 0. Be aware that because the model is a best fit line, the numerator can never be larger than the denominator because in the worst case, the best fit line would simply cut through the points at the average value.

For this particular model, the value of R square is 0.87, which is good.

One way of thinking about R square is that it tells you how much of the linear pattern is in the data, for example, how much of the variance of amount paid is explained by the variable tenure. That is, the relationship between tenure and amount paid explains 89% of the relationship seen in the data.

One nice feature of the linear model is that it preserves some characteristics of the data. In particular, the expected values of all the input data have the same average as the original data. Another feature is that the inverse relationship is easy to calculate.
The four charts above show four different sets of data with different characteristics. The two on the left have low values of R square, meaning that the data is dispersed and not well described by the best fit line. The two on the right have high values of R square, meaning that the data is close to the best fit line.

Another way of thinking about R square is how stable the data is. If you take another subset of data, will you get a similar model? This is important because there are often subtle choices in the data that you use and how you modify fields. When the R square value is low, then different subsets can behave very differently. Put another way, a few more data observations might dramatically change the coefficients in the model. When the R square value is high, this is not likely and the model is more stable. Good patterns in data should result in stable models.

Patterns Exist Even When R Square = 0

In such cases, the best fit line is always horizontal.
The original data omitted all the customers who never paid. Here, these additional points are shown as new points along the horizontal axis and the best fit model looks different. Previously, the estimated value for a customer at approximately 18 months was about $275. Now, the estimate increased to nearly $350 and an increase of almost 30%. This could have dramatic implications for how much money you are willing to spend to retain customers.

What happened is that the initial points “pushed” the line down, resulting in the long-term trend going up at a higher rate. This resulted in something that might initially seem counterintuitive. By adding more points whose values are $0, you actually increased the estimates for many customers. Ironically, the average expected value for the line with the new $0 points is less than the average expected value for the rest of the points (because the additional zero-valued data pushes down the average).
Best Fit Lines

POSITIVES
- Easy to apply to continuous data
- Easy to explain
- Easy to make pretty pictures
- Explains importance of each variable and overall estimate
- Fast

NEGATIVES
- Can produce unstable models
- Might not be expressive enough as a model
- Difficulty using categorical data types
- Difficult to add variables
- Possibility of nonsensical estimates
- Provides too much confidence in what might be misinformation

The model is only as good as the data going into it.

The Regression Equation

Formally, what you did is fit the observed data points to the following equation to minimize the sum of the square of the differences between the observed data points and the expected values:

\[ Y = \beta_0 + \beta_1 X_1 \]

Fitting means that you found values for the parameters \( \beta_0 \) and \( \beta_1 \) that minimize the sum.

For this type of equation, fitting requires some relatively simple matrix calculations.

Statisticians call the Xs and Ys \textit{constants}, and the betas \textit{variables}.

Lines find global patterns. The effects of the input variables are the same regardless of their values.

This slide introduces the mathematics behind regression. There is an algebraic corollary to the geometric (visual) interpretation of the data. Although the geometric visuals are useful in the simplest case, they stop being useful when you add more variables and calculate some of the more refined statistics.

The purpose is not to go into the mathematics and subtle statistics of regression analysis. However, we do want to emphasize that the line is an important \textit{interpretation} of how the data will work. In scientific work, the interpretation is often based on scientific theory, that is, there are underlying scientific laws related to the variables in an equation. You do not generally have this luxury in data mining.
More Variables: Multiple Regression

There can be any number of predictor variables:

\[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \]

This is known as multiple regression.

Two warnings:
- New variables only work when they are linearly independent of the other variables.
- Adding a variable almost always changes the coefficients of the other variables in the equation.

Multiple regression extends the notion of linear regression to more than one predictor variable. This is significant. After all, most data describing customers has dozens, if not hundreds, of variables.

When you use multiple variables, there are additional statistics for each variable that determine how important the variable is in the calculation. Understanding these statistics is important. It is easy to add variables. However, when any group of variables already has a linear relationship with a new one, then the coefficients are meaningless. Any change in the coefficient of the new variable can be offset by modifying the coefficients of the other variables. Which particular values these coefficients have is not relevant.
This data is similar to the data shown earlier, except that this is for two different products with slightly different price points. The next two slides show two different ways of handling this situation, which is common in data mining problems.

The two different groups shown above are shaded differently. The top group is for the more expensive product, and the bottom group is for the less expensive one.

Notice that the R-square value increased for each of these groups. Using the information about the product to estimate total amount paid is better than using information only about tenure. However, the improvement is not great, because the price differential is not large.
Alternatively Use an Indicator Variable

Introduce an indicator variable that is 1 for the more expensive product and 0 for the less expensive one:

\[ amount = -13.41 \times indicator + 0.55 \times tenure - 0.33 \]

This compares to the two models without the indicator:

\[ amount = 0.61 \times tenure - 10.62 \]
\[ amount = 0.49 \times tenure - 6.50 \]

To express these models, a single equation requires adding yet another term, \( tenure \times indicator \):

\[ amount = 4.12 \times indicator + 0.61 \times tenure - 0.12 \times indicator \times tenure - 10.62 \]

An alternative approach is to add an indicator variable for the product. The coefficient for the indicator is a negative value, which confirms that customers with the less expensive product indeed pay less.

If you look at the individual variables, you see some interesting relationships. For instance, the coefficient of tenure in the top equation is nearly equal to the average of the coefficients of the other two. Another way of looking at this is that the overall trend is the average of the trends for the other two.

The top model produces results different from the bottom model. In order to get exactly the same results, you need to add an additional variable, which is an indicator for the tenure, that is, \( indicator \times tenure \), and an additional coefficient, which is shown in the bottom equation. Notice what happens to this equation when the indicator takes on the value of 0. The terms with the indicator disappear, and you have \( 0.61 \times tenure - 10.62 \) or the first of the stratified equations. When the indicator is 1, then the equation becomes \( (0.61 - 0.12) \times tenure - (10.62 - 4.12) \), which reduces to the second of the equations. Stratification is equivalent to adding the indicator term and \( indicator \times tenure \) term.
Variable Selection

Regression models do not work well with large numbers of input variables. Here are some common ways to limit the number of inputs:

- Human intelligence.
- Forward selection – Do the regression on all remaining variables and add the one that performs the best.
- Step-wise selection – Do the regression on all remaining variables, add the one that performs the best, and perhaps remove previous variables.
- Step-wise with validation set – Use step-wise selection to create a family of models and then choose the model that performs best on a validation set.
- Select using a Decision Tree or Variable Selection node.

Regression wants as few variables as possible to work effectively. When human intelligence fails, the standard technique is based on the idea of adding one variable at a time, as in the previous example where you added an indicator variable after adding the tenure variable. There are several variants on this idea.

The first is simply to keep adding variables and to stop when some statistical measure is no longer valid. The second is to add new variables and to consider removing old ones, to keep the models as parsimonious as possible. Finally, you can use regression to create a family of models with different variables. Of these candidate models, the best is the one that works best on the validation data set.

There are other ways, such as using decision trees or the Variable Selection node to select variables.

Measuring Relative Importance of Variables

Easy Method:

- Standardize all input variables.
- Coefficients show relative strength.
- This method does not change the results of the models.

Complicated Method:

- Use fancy statistics to understand the relative importance of each coefficient.
  - Actually, “under the hood,” this method actually uses the previous technique.
General Linear Models

The *linear* in *linear regression* does not refer to the fact that the regression curve is a line. It refers to the fact that the coefficients in the regression equation are linear.

The following are examples of functional forms for general linear models:

\[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \ldots + \beta_n X_1^n \]
\[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1 X_2 + \beta_3 X_2 \]
\[ \ln(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \]

The *linear* part of linear regression involves the coefficients, rather than the functional form itself. That is, a model is a linear model when its coefficients (the betas) are linear.

Here are three examples of linear models:

- The first is the general form of a polynomial model in one variable. Although the variables are raised to powers, the coefficients are not. If you add new variables that are transformations of the existing variables (namely powers), then you can change this into a simple linear model. When you look at a scatter plot, it is fun to fit higher-order polynomials to the data. Although fun, higher-order models are not necessarily more useful.

- The second equation introduces an *interaction* term; this is the term where \( X_1 \) and \( X_2 \) are multiplied by each other. Such interactions are important, because often variables have more than additive effects on the output.

- The third form shows that the prediction variable can be transformed, as long as the rest of the variables retain their linear form. Linear models are very general, so as long as the variables can be transformed so that the resulting form looks linear, then the coefficients are easy to calculate.
Sometimes, you want to know more than the best fit equation. What if the question is, "Where is the minimum of some function, given some observed data points?" The best fit line does not provide guidance, because a line has no minimum values. By using a more complicated equation, you can fit the points better; the model is more flexible.

This example was chosen on purpose. When you train neural networks, one of the challenges is finding the set of parameters that minimizes the error. The neural network function is so complicated, that it is not possible to find the minimum value directly, so you have to search for it. The search proceeds by evaluating the error rate for various parameters (think of this as the vertical axis) and then fitting a multidimensional parabola to these points. The overall minimum is assumed to be in the region of the minimum of the multidimensional parabola. This technique of training is called conjugate gradient, which is discussed in the section on neural networks.
This example has only a small number of data points. However, these points do not seem to have a linear trend, as seen by the fact that the R-square value is so low for the line shown in green. By using a more complicated equation, you can fit the points better; the model is more flexible.

The purpose of this example is to show a somewhat different use of the results. Consider that the goal is no longer to measure expected values. Instead, you want to find the point that minimizes the function contained in these points. Obviously a line does not have a minimum value, because you can continue to extend it on either side. However, a parabola does have a single minimum (or a maximum).

In addition, there is a simple function that determines where the minimum occurs. This is at the point $X = -\beta_1 / (2\beta_2)$, which in this case is $-(-2.20 / 2 \times 1.14)$ or 0.97. This agrees with the value that you see.

You see this chart again in the chapter on training neural networks. Polynomial regression is one technique used for optimizing neural network weights.
Flexibility has its limits. A parabola has a better fit than the line. In fact, this is always true for any set of data, even for data that closely fits a line. The form of the parabola contains the line plus it has an additional parameter. The additional parameter is not used unless it improves the model.

Actually, when the degree of the polynomial is higher, the fit improves, because there are more parameters in the equation. In fact, for any set of $n$ points, there is a polynomial of degree $n-1$ that fits the points exactly. In practice, the goal is not to only minimize the error. You want to learn from the data points, and merely memorizing the data points does not provide any advantage when you look at new data points.
Binary Targets

What happens when the target is binary?

Linear regression does not work, because whatever the form of the equation, the results are generally unbounded.

Instead, you work with the probability \( p \) that the event will occur rather than a direct classification.

This is an interesting situation:

- The observation is discrete… “yes” or “no”.
- The estimation is numeric… a probability.

Consider the event of whether a customer ever paid by the tenure of the customer.

When you have a binary classification problem, then linear regression is not appropriate, because the results are, in general, unbounded as \( X \) approaches infinity, and with a binary classification there are simply two values.

If you use the values 0 and 1 for the target variable, it is possible to fit a linear regression model. However, this model does not work well because the error terms are large.
Has a Customer Paid?

The following chart is a scatter plot of tenure by whether a customer ever paid:

Tenure versus Ever Paid

There is a regression line, but it does not make sense. After all, how can the probability exceed 100%?

Here is an example of a binary classification problem. Has a customer ever paid? The scatter plot shows tenure versus the payment flag. Non-paying customers are terminated after a certain period of time. Initially, there are payers and non-payers for the small tenure groups. However, as tenure increases, soon everyone pays.

This type of situation shows how linear regression is not appropriate. First, notice that the model has a very poor fit. Second, even within this range of data, the model values exceed 100%, which is rather excessive for a probability.
Odds Instead of Probability

Consider the probability $p$ of an event (such as a horse winning a race) occurring.

The probability of the event not occurring is $1-p$.

The odds of the event happening is $p:(1-p)$, although you more commonly express this as integers, such as a 19-to-1 long shot at the race track.

$$\text{odds} = \frac{p_{\text{win}}}{p_{\text{loss}}} = \frac{p}{1-p}$$

The ratio 19:1 means that the horse has one chance of winning for 19 chances of losing, or the probability is $1/(19+1) = 5\%$.

As you saw, the probability poses some problems for regression. So, statisticians devised a way to use odds instead of the probability.

If you say that the odds of a horse winning are 3:1, you are saying that for every three times that the horse loses, the horse will win one time. There is a slight confusion, because the horse race odds are presented in the opposite order. More formally, you should say 1 to 3 in this case. Actually, you should present the odds as 1:3, because it is one win for three losses.

To calculate the probability, consider four races. In four races, the horse should win once. The odds of 1:3 correspond to the probability of 25% because $25\%/(1-25\%) = 1/3$.

Properties of Odds and Log Odds

Odds is not symmetric, varying from 0 to infinity.

Odds is 1 when the probability is 50%.

Log odds is symmetric, going from minus infinity to positive infinity, like a line.

Log odds is 0 when the probability is 50%.

It is highly negative for low probabilities and highly positive for high probabilities.
Logistic Regression

The log odds is used as the target for logistic regression:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

Notice that you are modeling $p$, the probability, even though the data contains discrete outcomes.

Through some simple mathematical formulations, this is equivalent to the following:

$$P = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

The resulting formula is not linear in $\beta_0$ and $\beta_1$.

Fitting the coefficients requires calculation-intensive methods. (The most common is called maximum likelihood estimation.)

The log-odds ratio provides a motivation for logistic regression. However, the formula can be expressed in terms of the original probability, as shown here. This expression is inherently non-linear, because the coefficients are in the exponents.

This has great theoretical significance. For instance, the parameters for a logistic regression are calculated using maximum likelihood, rather than least squares. In practice, though, these details are relatively hidden, and the difference is really between choosing PROC LOGISTIC rather than PROC GLM. The SAS Enterprise Miner Regression node decides whether to use logistic regression or linear regression based on whether the target variable is binary or interval.

With two possible outcomes, changing the target from one to the other merely flips the signs on the coefficients.
The logistic function has a characteristic S shape. The parameters on the model shift the curve left or right and stretch or compress the curve.

This function has some nice properties. Around 0, the slope is about 45% and the curve approximates a line from near the region of -1 to 1. Beyond this range, it gradually flattens out, and saturates at 100% or 0%. These properties make the logistic a natural curve for expressing probabilities.
To finish this section, look at the example mentioned earlier. This chart shows the logistic fit for the data. After approximately 120 days, the logistic regression determined that people paid their bills.

This example shows logistic regression being used for binary, discrete outcomes. It can also be used for continuous outcomes. This is especially useful when the data contains aggregated probabilities for different groups of customers.

**Lessons Learned**

- Regression is a general purpose modeling technique that has been available for centuries.
- It is available in many tools, and creates easy-to-use score code.
- Regression finds global patterns in data.
- Regression works best on numeric inputs.
- Regression works best when there are only a handful of inputs.
- Adding a new variable to a regression model can change the interpretations of other variables in the model.

**Demo Available: Regression**

**Refer to Exercises for Chapter 4 in the handout.**
5.1 Decision Trees

This Chapter...

…introduces decision trees, a tool that is useful for both data exploration and modeling.
Topics include the following:
- how trees are used for data exploration
- how trees are used for classification
- how trees are used for scoring
- splitting criteria for categorical targets
- splitting criteria for numeric targets
- different approaches to missing values
- recognizing instability and overfitting

Directed Data Mining with Decision Trees

A decision tree is a set of rules represented in a tree structure.
- Easy to understand how predictions are being made (The tree says how.)
- Easy to build and visualize
- Powerful data mining technique

Decision trees are one of our favorite data mining techniques because they combine data exploration with building models. Decision trees are powerful because they work easily on large amounts of data and on diverse data types.

A decision tree is a set of rules that is optimized for directed data mining. A tree is built on data with known examples; it can then be applied to new data to attach scores.
Decision trees are especially useful when there is more than one way to become a member of the target class. For example, a model to find profitable credit-card holders might identify three types:

- High-balance revolvers
- High-volume transactors
- Convenience users who make occasional large purchases and pay off the balance over time

Each of these can be represented by different paths through the tree.
The path through the decision tree starts at the root node, which is at the top, and moves to the leaf nodes, which are at the bottom. (One of the jokes that we make is that statisticians do not get outside much, so they do not realize the way that real trees grow.) Each node has two or more children. In this example, the root node has three children, two of which have two more.

The path specifies a set of rules. Each edge of the path adds another clause, such as “lives in a wealthy county” or “sends kids to public school” or “has spent less than $100 in the past year.” The combination of all these rules leads down to a specific leaf node. The leaf nodes have no children.

These paths through the trees explain how a tree is used for scoring. However, along the way, the rules also provide insight into which variables are important and help you understand what is happening.
Each leaf of a decision tree contains information that can be used for scoring. This leaf, for instance, says that 11,112 training instances landed here. Of those, 3.5% were YES. In this case, that means that 11,112 churners landed here. If the tree is being used for classification, then new records that land here are classified as NO. If the tree were also used to estimate a score, then the score would be 0.965, indicating a very high likelihood of NO.

The score can then be used for marketing campaigns. For instance, knowing that some customers are not likely to churn means that these customers can be removed from retention efforts. This saves money that can perhaps be used to retain other customers.
One of the downsides to scoring with a decision tree is that there are usually a small number of leaves. All records that land in each leaf receive the same score, which means that there are only a small number of different values that the scores take on. Often, decision trees have some large leaves; so many records have the same score. This lack of differentiation in the score values can be a problem in some applications.
Decision trees are a great way to explore a data set to begin to understand how the various input variables relate to a target variable. In this scenario, the business question is to determine the characteristics of towns where the product of the imaginary company does well and what distinguishes those towns from towns where the product does poorly. By understanding where you do well in the current market, that is, the state of New York, you hope to do make a success of the planned expansion across Lake Champlain into Vermont and New Hampshire.

The data source NYTowns contains demographic and geographic variables describing all the towns (somewhat arbitrary subdivisions of a county) in New York along with penetration, a variable reporting what percentage of households in a town currently use a product in the category. Using the quantile transformation in the Transform Variables node, you split the product penetration variable into terciles so you can label towns as high, medium, or low penetration. Throw away the mediums, and use a decision tree to see what separates the highs from the lows. In the tree, the lows have a value of 0 and the highs a value of 1.

The first split in the tree is on the number of households in urban areas. The product seems to have greater appeal in towns with very few such households. The appeal is further enhanced when mortgage payments and other costs of ownership are low.
The resulting tree has the following rules:

IF mortgages with payment less than $700 < 7.1% AND PopUrban < 161
• 0 : 77.8%
• 1 : 22.2%

IF mortgages with payment less than $700 >= 7.1% AND PopUrban < 161
• 0 : 4.5%
• 1 : 95.5%

IF PopUrban >= 1,457
• 0 : 99.0%
• 1 : 1.0%

IF Median Monthly ownership cost with mortgage < $876.50 AND 161 <= PopUrban < 1,457
• 0 : 10.0%
• 1 : 90.0%

IF Median Monthly ownership cost with mortgage >= $876.50 AND 161 <= PopUrban < 1,457
• 0 : 91.3%
• 1 : 8.7%

Decision trees are more than an exploration tool, of course. Here you see a decision tree used as a classification tool. The data for this example comes from another one of the course data sets, Catalog. This data comes from a catalog retailer. The target variable, Respond, is 1 for people who placed an order from the previous year’s Christmas catalog. Using that historical data, you build a tree to classify customers as likely (1) or unlikely (0) to place an order if mailed this year’s Christmas catalog. Each leaf of this tree is labeled with either a 1 or a 0. Records reaching a leaf labeled 1 are classed as likely responders. Those reaching a leaf labeled 0 are classed as likely non-responders.
In this tree, the target variable is an interval variable, the `ordernum`. The tree finds rules that govern order size. In this case, the findings are not particularly interesting. People who spent a large amount in the past continue to spend a similar amount in this order.

Trees can also be used to identify the variables that are important. This is simply a matter of selecting the variables out of the tree, starting at the root node. SAS Enterprise Miner makes it easy to pass the chosen variables to other modeling techniques, such as regression.

This example also shows that not all trees are binary. The maximum number of branches to be created at each node is one of the parameters that you can set in SAS Enterprise Miner.
Comments on Multiway Splits

- Very few tools enable splits on continuous variables to be other than binary.
- Multiway splits are more common on categorical variables.
- There is no inherent advantage to the number of splits (that is, more is neither better nor worse).
- However, when you compare the purity of two candidate splits that produce different numbers of children, there is inherently a bias in the comparison. (For this reason, binary splits are preferred.)
- Also when more splits are allowed at a node, each child node is smaller. Bigger children enable further splits down the tree.

Each node in a decision tree can have two or more splits. Some decision tree algorithms, such as CHAID, can create separate children for each value of a categorical input. The result is a proliferation of children. Such trees are not very deep. However, most algorithms (and even CHAID with the right parameters) can limit the number of children at each node.

In general, there is no inherent advantage in allowing more splits rather than fewer. However, when you consider two different splits such as a binary split compared to a three-way split, the comparison is necessarily biased either in favor of more splits or in favor of fewer splits. (Although this bias is not actually consequential, we tend to prefer binary splits.) In addition, binary splits ensure that the children are as large as possible, which helps to improve the stability of the split.
There are many variations on the decision tree algorithms. However, all have the same structure. A training set consisting of preclassified data is used to build the tree. The entire training set is at the root node. The decision tree algorithm then looks at each variable and decides the best way to split that variable to model the target. After doing this for all variables, the algorithm chooses the best split among them. In this example, the root node has three children. The number of children is arbitrary, although binary splits are the most common.

Now, the data is split among the children. Any record lands at exactly one child. The process is repeated for each child until no more useful splits are found. Some algorithms stop here. However, more commonly, the decision tree algorithms remove extra leaves that do not work on the validation set. This process is called pruning.

The decision tree algorithm can be computationally intensive, because the algorithm looks at every field in every record to determine the best split. SAS Enterprise Miner enables users to specify that the splits are determined using a sample of the data instead of all the data. With this option, the tree algorithms run quickly, even as the size of the training set increases.
At the start of the process, there is a training set consisting of preclassified records, that is, the value of the target variable is known for all cases. The goal is to build a tree that assigns a class (or a likelihood of membership in each class) to the target field of a new record based on the values of the input variables.

The tree is built by splitting the records at each node according to a function of a single input field. The first task, therefore, is to decide which of the input fields makes the best split. The best split is defined as one that does the best job of separating the records into groups where a single class predominates in each group.

The measure used to evaluate a potential split is purity. The next slides show specific methods for calculating purity in more detail. However, they are all trying to achieve the same effect. With all of them, low purity means that the set contains a representative distribution of classes (relative to the parent node), while high purity means that members of a single class predominate. The best split is the one that increases the purity of the record sets by the greatest amount. A good split also creates nodes of similar size, or at least does not create nodes containing very few records.

These ideas are easy to depict visually.
This is a good split. The parent node is half dark and half white. One child is almost all dark and the other is almost all white. The challenge is to find purity functions that recognize this as a good split as easily as you can.

**Purity Measures**

Purity measures for categorical targets
- Gini (also called population diversity)
- Entropy (also called information gain)
- Information gain ratio
- Chi-square test

Purity measures for numeric targets
- Reduction in variance
- F test

The choice of purity measure depends solely on the target, not on the input variables.

Examine how some of these tests evaluate the split in question.
Why All the Gory Details?

The split criteria are not actually very difficult to understand and understanding them makes the technique seem less magical.
Tree models are easier to understand and interpret when you understand how they are constructed.
You can understand the difference between various named algorithms such as CART and CHAID.

Which Is the Better Split?

Two children: Both are 64% pure.
The sizes are exactly the same.

Two children: One is 80% pure and the other 57%
However, the sizes are quite different.

This example shows two candidate splits for a tree. Which is the better one? The data begins perfectly balanced between light and dark. One candidate split produces the children on the left, where one child is quite pure but is rather small. The other produces the children on the left. In this case, each child is exactly the same size but is less pure.

Although decision tree algorithms choose the splits automatically, it is worthwhile to look inside and see what is happening. The next slides carry out the calculations for the Gini, entropy, and CHAID purity measures and show how they compare on these two candidate splits.
Gini: Easy Measure to Explain

Gini, used in the social sciences and economics, is the probability that two things chosen at random from a population will be the same (a measure of purity).

A pure population has a Gini index of 1.

If there are two groups equally represented, then the Gini index is 0.5.

The Gini index is the sum of the square of the proportions:

\[ p_1^2 + p_2^2 \]

The goal is to maximize Gini.

The Gini score is named after an Italian economist and statistician, Corrado Gini, who worked in the first half of the 20th century and studied populations and the distribution of income. (In 1932, Gini resigned from his position as the president of the Central Institute of Statistics in Rome as a protest against fascism.)

The Gini score is perhaps best understood using an analogy from ecology. Consider a wildlife refuge that has two types of animals, roadrunners and coyotes. To measure how pure the population is, you might enter the refuge, hunt for an animal, and photograph it. You might then go again, and take another picture. The Gini score is the probability that these two pictures are of the same type of animal.

If there are only roadrunners, then both pictures will be of roadrunners, so the chance is 100%. The same is true if there are only coyotes in the refuge. If the population is split between the two, then there is a 50%-50% chance that both pictures will be of the same animal, so the Gini score in this case is 0.5. In fact, for two outcomes, the Gini score always varies between 0.5 and 1.0. A higher Gini score indicates a purer population.

Expressed as an equation, the Gini index of a node is the square of the proportions of each outcome in the node.
The Gini score of a particular Decision Tree node is the sum over all the classes represented in the node of the proportion of records belonging to a particular class squared. The Gini value of a split is the sum of the Gini scores of all the nodes resulting from the split, weighted by each node's proportion of the records. When Gini is chosen as the splitting criterion, the algorithm searches for the split that increases the Gini score by the greatest amount. When there are multiple splits, then there is a correction factor for Gini, because Gini scores tend to increase as the number of children increases.

For a binary target, such as the one shown here, the formula for Gini is the square of the proportions of each value of the target. For the parent node, the Gini score is 0.5, because \(0.52 + 0.52 = 0.25 + 0.25 = 0.5\).

This calculation is repeated for each of the children. Because they have the same proportions of the target values (although the proportions are reversed), each child has the same Gini score, which in this case is 0.541. The overall Gini score of the split is the weighted sum, which is easy in this case, because the two have the same size and the same score. The score of the split is 0.541.
Comparing the Splits with Gini

Candidate split B:
- Gini for the left child is
  \((\frac{1}{5})^2 + (\frac{4}{5})^2 = 0.04 + 0.64 = 0.68\).
- Gini for the right child is
  \((\frac{10}{23})^2 + (\frac{13}{23})^2 = 0.189 + 0.319 = 0.508\).
- Gini for the split is the weighted average:
  \((\frac{5}{28}) \times \text{Gini}_\text{left} + (\frac{23}{28}) \times \text{Gini}_\text{right} = 0.539\).

As you saw, the Gini score for each of the two children in the first proposed split is 0.541.

What about the second proposed split? In this case, the Gini score is more complicated because the two children have different sizes and different proportions. Consequently, the calculation must be done for each child. The total score is 0.539.

Which Is the Better Split?

When you look at the two candidate splits, it is difficult to determine which is the best. However, the Gini score provides a single value for comparison. In this case, the Gini score for Candidate A is better than the split on Candidate B. However, the Gini scores are close, so the preference is not that strong.
Entropy is more difficult to explain because it comes from information theory. The idea is that entropy encodes the amount of information that can be stored in a given set of bits. For this reason, the entropy calculation uses logarithms.

Entropy runs in the opposite direction from the Gini score. A pure population has entropy of 0, whereas one where two groups are equally represented has entropy of 1. The formula is the sum of the proportion of each class times the log, in base two, of the class. Because the proportions are less than 1, the logs are negative. To make this a positive value, entropy is often multiplied by -1.

Entropy and Gini are actually similar as measures. Each is a weighted sum of proportions. In the case of Gini, the weighting is the proportion itself. In the case of entropy, the weighting is the log of the proportion.
Entropy for Candidate Split A

The entropy is the weighted sum of the entropy of each child (weighted by the size of the split).

Entropy for the root node is 1 \( -(0.5\log(0.5) + 0.5\log(0.5)) \).

Entropy value for either child:
\[-((5/14)\log(5/14) + (9/14)\log(9/14)) = -(-0.5305 + -0.4098) = 0.9403 \]

Evaluate the split with the weighted average of entropy values for all children:
\[0.5*0.9403 + 0.5*0.9403 = 0.9403 \]

Information gain is \[1 – 0.9403 = 0.0597\].

The entropy of a particular Decision Tree node is the sum over all the classes represented in the node of the proportion of records belonging to a particular class times the log, in base two, of the proportion. The entropy of a split is the sum of the entropy values of all the nodes resulting from the split, weighted by each node's proportion of the records. When entropy is chosen as the splitting criterion, the algorithm searches for the split that reduces the entropy by the greatest amount. When there are multiple splits, then there is a correction factor, because entropy tends to go to 0 as the number of children increases.

The entropy of the parent is 1, because \[\log(0.5) = -1\] and \[-1\*(0.5\log(0.5) + 0.5\log(0.5)) = -1\*(0.5*-1 + 0.5*-1) = 1\]. For binary targets, this is the maximum value of the score.

This calculation is repeated for each of the children. Because they have the same proportions of the target values (although the proportions are reversed), each child has the same entropy score, which is 0.9403. The overall entropy of the split is the weighted sum, which is easy, because the two have the same size and the same score. The score of the split is 0.9403.
Comparing the Splits with Entropy

Candidate split B:
- Entropy for the left child is
  \[-1^*\left((1/5)^*\log(1/5) + (4/5)^*\log(4/5)\right) = -1^*(-0.4644 + -0.2575) = 0.7219.\]
- Entropy for the right child is
  \[-1^*\left((10/23)^*\log(10/23) + (13/23)^*\log(13/23)\right) = -1^*(-0.5225 + -0.4652) = 0.9877.\]
- Entropy for the split is
  \[\frac{5}{28}*\text{Entropy}_{\text{left}} + \frac{23}{28}*\text{Entropy}_{\text{right}} = 0.9402.\]

You can carry out the same calculation for the second proposed split. In this case, the calculation is more complicated because the children have different sizes and different proportions. However, the calculation remains the same.
The entropy scores are close, but the minimum value is for Candidate B rather than for Candidate A.

You choose this example because it is the smallest example on a balanced set of data where entropy and Gini differ in their choices of what is best. From this example, you can see that entropy has a slight preference for smaller and purer nodes, whereas Gini has a slight preference for nodes that are similar in size.

For any balanced data set with fewer than 28 records, the two purity measures always choose the same splits. However, as the number of records increases, the two start to disagree more and more often.
Chi-Square Is from Statistics

The chi-square test is an important test in statistics to measure the probability that observed frequencies in a sample are due only to sampling variation.

Chi-square is always relative to the proportion in the original population (the parent).

If the proportions in both children are the same as the parent, then the chi-square value is 0.

If both children are pure, then the chi-square value is high. (For a 50%-50% population, the value is the population size.)

The calculation for chi-square is done for each child:

\[ \frac{(c_1-\text{expected}(c_1))^2}{\text{expected}(c_1)} + \frac{(c_2-\text{expected}(c_2))^2}{\text{expected}(c_2)} \]

In this calculation, \(c_1\) is the number of instances of class 1 in one child and \(\text{expected}(c_1)\) is the expected number given the proportion in the parent.

The goal is to maximize chi-square.

Chi-square is a standard statistical measure devised by the British statistician Karl Pearson to test the significance of observed values. A significance test produces a probability that observations are due solely to sampling variation and not to some underlying cause. The chi-square test is one such test that is applicable to categorical variables. As such, it is a natural choice for decision trees. The original work on such trees was performed the 1970s.

For a given target class in a given child, the chi-square is calculated as the square of the difference between the observed number and the expected number divided by the expected number. The expected number is calculated by multiplying the size of the child by the proportion of the class in the parent. The overall chi-square value for the child is the sum of the chi-square values for each class.

If the class proportions in the child are the same as the class proportions in the parent, then the chi-square value is 0, because the observed values are the same as the expected ones. When the children are pure, then the chi-square value is maximized as the size of the parent population.

Assume that there are \(X\) light and \(X\) dark in the original. Then the expected value of each is \(X/2\). The chi-square of each class is \((X-\text{X}/2)^2/(\text{X}/2)\) and \((0-\text{X}/2)^2/(\text{X}/2)\). Each of these reduces to \(\text{X}/2\); so the chi-square of each child is \(\text{X}/2 + \text{X}/2 = \text{X}\). For both children, it is \(2\text{X}\), which is the size of the original population.

When you use chi-square as a splitting criterion, the goal is to maximize the value.
Chi-Square for Candidate Splits

The expected value of dark or light is 7 in each child. So, the chi-square value for each child is as shown below:

\[(9-7)^2/7 + (5-7)^2/7 = 4/7 + 4/7 = 1.1429\]

The overall chi-square value is the sum for each child:

\[1.1429 + 1.1429 = 2.2857\]

The expected value of dark or light is 2.5 for the left child and 11.5 for the right child. The chi-square values are as follows:

Left: \[(4-2.5)^2/2.5 + (1-2.5)^2/2.5 = 2.25/2.5 + 2.25/2.5 = 1.800\]

Right: \[(10-11.5)^2/11.5 + (13-11.5)^2/11.5 = 2.25/11.5 + 2.25/11.5 = 0.3913\]

Overall: \[1.8000 + 0.3913 = 2.1913\]

Here you carry out the chi-square calculation for each of the candidate trees.

Which Is the Better Split?

Chi-Square = 2.2857
(Entropy = 0.9403)
(Gini = 0.541)

Chi-Square = 2.1913
(Entropy = 0.9402)
(Gini = 0.539)

The maximum chi-square agrees with the maximum Gini score in this case.
Synonymous Splits

There are many ways to describe the same thing, such as decreasing usage.

Different variables might be essentially synonymous.

At each node in a decision tree, the algorithm finds the best split.

When the children are split, the synonymous variables are also split, so they have little to contribute. The algorithm finds better splitters.

Decision trees do an effective job of finding the variables that are most important for the outcome.

Building a decision tree model is one way of identifying a set of input variables that, taken together, convey most of the available information. The variables chosen this way tend to be fairly independent. However, because no branch pays any attention to what is happening in other branches, it might happen that a rule on one side of the tree chooses one variable while a rule on the other side of the tree chooses another, essentially synonymous variable. In these cases, the human modeler can arbitrarily select one of the two for inclusion in other models.

For example, the customer ID might be found as a split inside a decision tree. This is because these IDs are often created sequentially, so the ID is synonymous with the start date. However, the ID provides a finer level of granularity, because the split can occur anywhere (as opposed to only on date boundaries). For this reason, the ID might be chosen rather than the start date or tenure, even though the two are essentially synonymous.

The advantage of the decision tree approach is that after one variable is chosen, the data is actually split into two different sets.
Handling Synonymous Splits

Synonymous splits are often caused by different variables capturing the same phenomenon, such as declining usage.

- Low recent usage
- Ratio of recent usage to historical < 1
- Low recent bills
- Beta value of usage < 0
- and so on

A better approach is to add to the customer signature variables that describe interesting phenomena.

For instance, there might be several variables that represent declining telephone usage. The recent call volumes are low. The recent billing amounts are low. The ratio of recent amounts to historical amounts is less than 1. The beta value is negative. All of these variables might be tested for a particular split in a decision tree and all might be close. However, it is likely to be chosen somewhat arbitrarily.

A better approach is to explicitly put variables for known phenomena into the customer signature.

Handling Missing Values

Decision trees can handle missing values in several different ways:

- The algorithm can ignore records that have missing values. (not recommended because it introduces bias)
- The algorithm can treat missing values as legitimate values and assign them to one of the children.
- The algorithm can save alternative splits on other variables that approximate the actual split. In SAS Enterprise Miner, these are called surrogate splits.

Decision trees do a good job of handling missing values by sending them to one of the children. Using surrogate splits is one of several approaches to missing values allowed by SAS Enterprise Miner.
Watching a Decision Tree Grow

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50.0%</td>
<td>50.0%</td>
</tr>
<tr>
<td>0</td>
<td>50.0%</td>
<td>50.0%</td>
</tr>
<tr>
<td>N in Node</td>
<td>80000</td>
<td>59998</td>
</tr>
</tbody>
</table>

The model set is balanced so that 50% quit and 50% remained active. (1=active, 0=stopped)
The training set is larger than the validation set.
All decisions on making splits use the training set.
You can see the effects of splits on the validation set.

First Split on Handset Churn Rate

Handset churn rate can be measured each month before scoring.
The handset models are associated with high churn change over time, but the fact that some handsets are more desirable than others remains constant.
People with high-churn handsets and no monthly fee are very likely to churn. The lack of a monthly fee might indicate that they are not on a contract.

On the lower churn side of the tree, the next split after the handset churn rate is Dealer Group. Some dealers are much more likely to churn subscribers.
And the Process Keeps Going...

Falling Record Count
As record count falls, splits become unstable.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>64.6%</td>
<td>69.0%</td>
</tr>
<tr>
<td>0:</td>
<td>15.4%</td>
<td>31.0%</td>
</tr>
<tr>
<td>N in Node</td>
<td>39</td>
<td>42</td>
</tr>
</tbody>
</table>

SAS Enterprise Miner default leaf size is not large enough.

Often, the simplest way to improve a decision tree model in SAS Enterprise Miner is to set the number of records required for a split search and the smallest number of records allowed in a node to much higher than the defaults provided by the software. For many applications, 100 is a reasonable minimum leaf size. For other applications, where the final number of segments should be small, a higher minimum is appropriate.
The Worst Case

In the worst case, the leaves are totally wrong.

This leaf in the decision tree has exactly one row from the training set, and this row is N. The leaf has a score of 100% for N. However, three rows of the validation set happen to land at the leaf, and they are all Y. The leaf is wrong 100% of the time.

The Chaotic Error Rate
Pruning is a way of increasing model stability by decreasing model complexity.

In an earlier version of this chart, the X axis was labeled “complexity.” For trees, one way of measuring complexity is the number of leaves. On the training data, every split decreases the error rate. As the leaves become more numerous, they contain fewer and fewer records, and it becomes less and less likely that the distribution of outcomes within a leaf will be similar from one data set to another. The validation data is used to find the point where added complexity begins to hurt rather than help.
One Traditional Pruning Algorithm

Create a new measure to punish trees for being complex.

- The adjusted error rate takes into account the size of the subtree: \( AE(T) = E(T) + \alpha \cdot \text{leaf\_count}(T) \)

Create a set of candidate subtrees of different sizes using the adjusted error rate.

- Start with the largest tree. (This is the first subtree.)
- Increase the value of \( \alpha \) until some subtree has an adjusted error rate less than the tree as a whole.
- Keep increasing the value and get more (and smaller) subtrees, until only the root node remains.

Measure the performance of each subtree on the validation set.

- Select a subtree based on the validation set error rate.

The pruning algorithm used by CART has two parts:

- First the algorithm finds a set of subtrees.
- Then it evaluates the subtrees to find the one that works best on the validation data set.

Finding the subtrees uses the *adjusted error rate*. This error rate treats the tree as a classification tree and adjusts the error rate of the whole tree by punishing the tree for complexity. The measure of complexity is the number of leaves in the tree. At first, the value of \( \alpha \) is 0, so there is no punishment for complexity. Gradually increasing the value produces subtrees that work better than the overall tree. Each of these subtrees is remembered for testing against the validation set. The tree with the best performance on the validation set is the one returned by the algorithm.
Sometimes there is a clear inflection point where the validation set error begins to increase. Other times the validation set error levels off while the error on the training set decreases.

One of the problems with the CART pruning algorithm is that entire subtrees disappear because of the dependence on the classification error rate. If all the leaves under a given node have the same classification as the node, then they are never considered in the pruning algorithm. The node itself might have a confidence of 60%, and one leaf is 98% and another is 51%. However, because the classification error rate is used, all are classified the same way, and the punishment for complexity prevents any of the leaves from being considered.
Three-Way Target

The credit class determines the type of churn.

However, building binary models (one for churn, the other for the type of churn) also works.

In this tree, people with high-churn handsets all tend to leave. Those in credit classes A and B leave voluntarily. Those in credit classes C and D leave involuntarily. Might not paying be the way that some people choose to say “I quit!”? 
The Icicle Diagram

The icicle diagram shows the whole tree.

The top level in an icicle diagram corresponds to the root node of the tree. Each level of the icicle diagram then corresponds to the leaves at a particular depth in the tree. Each box in the icicle diagram corresponds to a leaf node in the tree. The size of the box corresponds to the size of the node, and the color to the node’s purity.

This diagram is called an icicle diagram because as the tree grows deeper, the icicle diagram starts to resemble icicles hanging down.

Sometimes Rules Are Best

IF 17.5 <= tot units demand AND $29 <= $ last 24 months THEN
NODE : 7
N : 1018
1 : 75.0%
0 : 25.0%

IF tot $ 98Q3 < $10 AND lifetime orders < 3.5 AND $ last 24 months < $29 THEN
NODE : 8
N : 2042
1 : 34.7%
0 : 65.3%

Rules are often the most useful results from a decision tree. These can be expressed as English rules, or even as SQL WHERE clauses.
The cumulative gains consists of a series of line segments. Each segment corresponds to one of the leaves of the tree. The slope of the line corresponds to the lift at that leaf. The length corresponds to the number of records that land there. The steepest segments correspond to the leaves with the biggest lift (highest density of the desired outcome).

There is a relationship between the decision tree and the concentration curve. In fact, you can look at the concentration curve and learn about the model that produced it.

**Lessons Learned**

Decision trees are a powerful tool for both data exploration and model building.

- Decision trees can use both categorical and numeric inputs.
- Decision trees can work on any number of inputs. In fact, they are a good tool for selecting variables for other purposes.
- Decision trees work best on classification tasks. They can be used for estimation, but produce a very limited number of scores.

**Demo Available: Building a Decision Tree Model Using the VCS Data**

**Refer to Exercises for Chapter 5 in the handout.**
Chapter 6  Neural Networks

6.1  Neural Network Models .................................................................................................. 6-3
6.1 Neural Network Models

This Chapter...

…introduces neural networks, a powerful technique that was originally developed to simulate biological networks but is now used for modeling. The chapter discusses the following topics:

- the biological metaphor
- the relationship of neural networks and ordinary regression models
- what goes on inside a neural network
- how the network is trained
- data preparation issues for neural networks
- principal components
The basic building blocks of a biological brain are the neurons, which are connected together in brains and other nerve tissue. The neuron is a device for combining input signals and producing output signals.

In the 1930s, Dr. Warren McCulloch, a neurophysiologist at Yale University, proposed a simple model to explain the behavior of neurons. When the combined level of the inputs rises above some threshold, the neuron fires and other neurons downstream receive an input. Each neuron has a slightly different chemistry (which can change over time), so the responses of different neurons are different. Of course, this is a simplification because neurons actually respond to both electrical and chemical responses. It is a useful simplification.

Within a decade, the first digital computers became available for military research and in academic institutions. The simplified neurons were among the first things that scientists wanted to model on computers.

First consider that a neural network is similar to a black box for producing estimates. Then peek inside the box to see what is going on.
In the brain of a frog, there are millions of connected neurons. Frogs are particularly adept at catching flies. There might be, for example, a neuron within the visual system of a frog that fires in response to fly-like movements, and another that fires in response to things similar to the size of a fly. These two units could be connected to a neuron that fires when the combined value of these two inputs is high. This neuron is an input into yet another that triggers tongue-flicking behavior.

This basic example of a biological neuron proved far beyond the state-of-the-art in the 1950s and 1960s. Although you can explain the frog’s behavior in terms of neurons, and you might even be able to “see” the neurons in operation, such observations do not explain how the neurons detect fly-like movements or how they detect something the size of a fly. That is, you can observe the result of training, but you do not know how the biological neuron is trained.
Inside the Frog Brain

Before

- Fly-like motion
- Fly-like size
- Hunger

Tongue control

After

- Fly-like motion
- Fly-like size
- Hunger

Tongue control

academic.emporia.edu/sievertl/verstruc/brain.htm
The origins of artificial neural networks go back to work in the 1940s when researchers tried to simulate the workings of actual biological nervous systems. This work later caught the attention of machine-learning and artificial-intelligence researchers who were interested in making computers “think” the way that people do. What better way to make computers think than to have them emulate biological thought processes? This has not happened yet, but neural networks have nonetheless proven themselves useful.

In the 1980s, there was a breakthrough in neural network research, that is, the development of the backpropagation algorithm. This provides a way to train a neural network so that it can be used for practical applications such as detecting credit card fraud and recognizing hand-written numbers on checks. At that point, neural networks began to overlap the territory of traditional statistical models. At first, the statisticians were suspicious of neural networks because of the odd (from a mathematical point of view) language used to describe them.

After neural networks were recognized as close cousins of traditional modeling techniques such as logistic regression, the statisticians and mathematicians started to pay more attention and were able to greatly improve the efficiency of their training, or as statisticians would say, model fitting.
The appraiser or real estate agent is a good example of a human expert in a well-defined domain. Houses are described by a fixed set of standard features that can be taken into account by the expert and turned into an appraised value.

The Federal Home Loan Mortgage Corporation (Freddie Mac) uses a neural network to do the same thing for homes throughout the United States. A neural network takes specific inputs (in this case, the information describing the mortgaged property) and uses them to estimate the value of the house.

The list of inputs is well-defined because of the extensive use of the multiple listing service (MLS) to share information about the housing market among real estate agents and due to the standardization of housing descriptions for mortgages sold on secondary markets. The desired output is also well-defined, that is, a specific dollar amount. In addition, there is a wealth of experience, in the form of previous sales, for teaching the network how to value a house.
The artificial neuron is similar to the biological one in that it is a device for combining inputs to create an output. This picture shows the important features of the artificial neuron. The unit combines its inputs into a single value, which it then transforms to produce the output. These together are called the activation function.

The activation function has two parts:

- The first part is the combination function that merges all the inputs into a single value. As shown in this slide, each input has a weight. The most common combination function is the weighted sum, where each input is multiplied by its weight and these products are added together.

- The second part of the activation function is the transfer function, which gets its name from the fact that it transfers the value of the combination function to the output of the unit. The transfer function drawn here is S-shaped, which is typical. The shape leads to the useful behavior that small changes in a very small or very large value have little effect, while changes in value in the middle of the range have near linear effect.

The process of training the network is the process of adjusting the weights.
Combination and Transfer Functions

Neural networks can have arbitrary combination and transfer functions. The combination function can be MAX or MIN or something else. In practice, the weighted SUM works well. The transfer function can also have many shapes, such as a line, sigmoid, tanh, step function, or something else.

This picture shows four possible transfer functions.

- The **linear transfer function** is the least interesting. If a neural network is designed so that all the transfer functions are linear (and all the combination functions are weighted sums), then the network represents a linear regression.

- The **logistic function** and the hyperbolic tangent are essentially the same shape. The major difference is that the hyperbolic tangent is centered on 0 instead of on 0.5, which sometimes makes it a bit more convenient. Both of these functions are saturated for large or small values. For instance, for large values, both produce a number very close to 1, when the inputs are bigger than 4 or 5. This is an example of nonlinear behavior, because the output does not depend on the input.

  However, when the input is near 0, then small changes in the inputs have corresponding changes in the output. This is an example of linear behavior.

- The **activation functions** combine both linear and nonlinear behavior, which is part of the reason for neural networks’ incredible flexibility.

- The fourth possible activation function is the **step function**. In some ways, this comes closest to the biological model. However, researchers found that neural networks that use step functions as activation functions are very difficult to train, because step functions are not continuous.
This is the general form of the most popular type of neural network for directed modeling, the *multilayer perceptron*. Data coming from the database is first passed to an input layer where it is transformed into the proper range. The inputs are then fed to the nodes of the hidden layer. Although it might at first appear as if each node in the hidden layer is doing the same thing because they all receive the same inputs and contain the same combination and transfer functions, this is not the case because each node receives a different set of weights for the inputs. In other words, each hidden node represents an equation of the same form, but with different parameters.

The results of the calculations performed in the hidden nodes are then combined in the output node. The link function is applied to the combined output to produce an estimate for the value of the target variable.
Neural Network Used for Classification

A neural network whose output layer has a logistic transfer function can be used for binary classification.

When a neural network is used to model a binary outcome, the result can be interpreted as the probability that the record in question belongs to the class labeled 1 (when the network uses logistic activation functions).

Describing Models as Neural Networks: Linear Regression

\[ E(y) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \]

The combination function is the weighted sum.
The transfer function is logistic.

This picture shows the way that a familiar regression equation can be rendered as a neural network diagram. The independent variables become the input nodes. The parameters are drawn as weights on the synapses connecting the inputs to the output node, which here consists of a simple combination. Notice that \( w_0 \), the intercept, is called the bias in neural network terminology. The bias is usually not drawn on the network diagram.
Describing Models as Neural Networks: Logistic Regression

\[ E(y) = \text{logit}(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3) \]

The combination function is the weighted sum.
The transfer function is logistic.

There is no difference between the network diagram for linear regression and the network diagram for logistic regression. The only difference between the two is the transfer function.

Multilayer Perceptron (tanh Activation)

\[ E(y) = w_0 + w_1 H_1 + w_2 H_2 \]
\[ H_1 = \tanh(w_{01} + w_{11} x_1 + w_{21} x_2 + w_{31} x_3) \]
\[ H_2 = \tanh(w_{02} + w_{12} x_1 + w_{22} x_2 + w_{32} x_3) \]

Here is a somewhat more complicated network. This time, the inputs go to each of two nodes, called hidden nodes. Each hidden node has a combination function (weighted summation) and a transfer function (hyperbolic tangent). The outputs of the hidden nodes are combined again inside the output node and a link function is applied to map the output value back to the proper range. This diagram represents the most common type of neural network for directed data mining: the multilayer perceptron.
Neural Networks

Neural networks are universal approximators. A sufficiently complicated neural network can approximate any continuous multi-variable function.

- This is a theoretical property and not necessarily useful in practice.

Neural networks are not rigid models, so they can find local patterns.
- They can fit local variations without having an impact in other areas.
Even with two hidden nodes and one input, the response of the neural network can vary greatly in shape. The response is determined by adding up the sigmoidal output of the hidden nodes.

In the top example, the output starts and ends around 1, which indicates that extreme values in either direction are treated the same way. However, in the middle region, the output varies between its maximum and minimum.

The second example is more similar to a step function, where the value starts off very small, increases gently to a middle value, and then increases sharply to a high value.

Both of these are examples of the same neural network with different weights, showing how the network can learn to adjust to very different patterns in the data. Of course, this is the simplest example. With additional inputs and additional nodes, the network is more complicated.
Each hidden node is a nonlinear function of its inputs. In the case that you are following, the transfer function is sigmoidal and so forms a sigmoidal surface as shown in the picture. The final result is a linear combination of these sigmoidal surfaces. It was demonstrated that such a combination of sigmoidal surfaces can approximate any continuous function to an arbitrary degree of accuracy. For this reason, multilayer perceptrons are said to be universal approximators. (Of course, there is no guarantee that you can find the right set of curves to approximate your function, nor is there any guarantee that there even exists a function of your target variable in terms of the inputs you chose.)

The greatest strength of neural network models is their ability to approximate any continuous function without making any assumptions about the underlying form of the function to be approximated.
The goal of training a neural network is always the same, that is, find the best set of weights for the network. There are generally many weights in a network, at least a dozen. This makes finding the optimal solution a difficult problem, because there are a variety of algorithms for solving it. In fact, none of the training techniques is guaranteed to find the global best solution. They do all find solutions that are better than any small adjustment to the weights, and which might be globally optimal. One way to approach finding the globally best solution is to train the same network several times using different sets of weights at initialization.

The procedure for using neural networks is similar to the one that you already saw for decision trees in that you first fit the model to a training set resulting in an overfit model and then used a validation set to retreat to a more general model. Neural networks require more data preparation, however.

After all of the inputs are in the proper form and there are no missing values, the model is trained. Training continues until there is no further improvement as measured by the objective function or until a preset number of iterations is reached or a preset time limit is reached. At the end of the training, the model is likely to be overfit. In other words, it memorized the training set and so made very good estimates on the training data, but did not generalize well to other data sets.

To avoid this, the system not only remembers the final weights; it also keeps track of the weights from each iteration of training. Each set of weights is a candidate model. The candidate models are applied to the validation set and the one that minimizes the objective function on the validation data becomes the final model.

Because the validation data has a role in choosing the model, a third data set, the test set, is used to judge the model’s accuracy on unseen data. For further confidence in the model’s generality, it should also be tested on an out-of-time data set (that is, one taken from a different time period than the training data).
By now you understand that after the diagram for a neural network is drawn, nothing changes but the weights. The diagram is a representation of an equation and all you have to do is find the right parameters to the equation. How?

The original method, described here, is called back propagation. At the heart of back propagation are the following three steps:

1. The network receives a training example and, using the existing weights in the network, calculates the output or outputs.
2. Back propagation calculates the error by taking the difference between the calculated result and the expected (actual result).
3. The error is fed back through the network and the weights are adjusted to minimize the error, hence the name, back propagation, because the errors are sent back through the network.

The back propagation algorithm measures the overall error of the network by comparing the values produced on each training example to the actual value. It then adjusts the weights of the output layer to reduce, but not eliminate, the error. However, the algorithm is not finished. It assigns the blame to earlier nodes in the network and adjusts the weights connecting those nodes, further reducing overall error.

Given the error, how does a unit adjust its weights? It estimates whether changing the weight on each input would increase or decrease the error. The unit then adjusts each weight to reduce, but not eliminate, the error. The adjustments for each example in the training set slowly nudge the weights toward their optimal values. After the network sees enough training examples during enough generations, the weights on the network no longer change significantly and the error no longer decreases. This is the point where training stops; the network learned to recognize patterns in the input.

Back propagation was the first algorithm used for training neural networks. Besides being interesting for historical reasons, it is relatively easy to explain without difficult mathematics. However, it performs poorly. The conjugate gradient is relatively easy to explain, although the mathematics can become complicated. Other techniques (explained in more detail in SAS Help) are useful for the most advanced users and researchers.
Neural networks are trained in generations, starting from an initial set of weights (which can be randomly generated or might use some other technique).

In general, training a generation is the process of applying the network to the input data and adjusting the weights to optimize performance.

All the training algorithms process the data in basically the same way, which creates generations of weights. Although there are differences in the training techniques, you can think of them as reading through the input data and adjusting the weights.

Some training techniques are incremental, meaning that they adjust the weights for every record. Some training techniques are batch, meaning that many records (perhaps all the training set) are read, before the weights are adjusted. You generally expect to have dozens or hundreds of generations before the network converges on a solution.
Beyond Back Propagation

Training a network is really the process of assigning the optimal weights to the network. Some other optimization algorithms are listed below:

- Conjugate gradient
- Quickprop
- Quasi-Newton
- Levenberg-Marquardt

None is guaranteed to return the global optimum.

The backpropagation algorithm works by calculating the gradient, the vector of partial derivatives of the objective function with respect to each of the input parameters. This gradient vector points in the steepest direction. Backpropagation is a hill-climbing technique. One method of looking for the highest point in a landscape is to always follow the steepest path until you cannot go up any more. (If you try this in the Adirondacks, you are likely to find yourself on the top of a steep hill gazing up at the high peaks all around you.) In an attempt to overcome this problem of finding local optima, there are parameters that can be set for the learning rate and the momentum. The learning rate controls the size of the steps you take while climbing the hills. The Jolly Green Giant is more likely to find the top of a nearby mountain than an ant is. Momentum makes you tend to continue in the same direction that you were traveling instead of turning fully to the new steepest direction after every step. Higher momentum makes it less likely that you remain at a local optimum, but it also makes it more difficult to stop at the top after you reach it.

Despite much fiddling around with backpropagation, it simply is not a very good optimization technique. We started with it for the same reason that the machine learning people did, that is, it is relatively easy to understand and explain. After neural networks fell into the hands of mathematicians familiar with numerical analysis, backpropagation was replaced with other, more efficient optimization techniques that were already developed for other purposes. SAS Enterprise Miner provides several choices of optimization algorithms, but there is generally no reason not to use the default.
Quickprop is an alternative method for training neural networks. It is similar to using a parabola to find a minimum at a few data points. The idea is that Quickprop tests a few different sets of weights, fits a multi-dimensional parabola to them, and then goes directly to the minimum of the parabola.

Actually, this is an oversimplification of the algorithm. It actually works by taking various partial derivatives and fitting the multidimensional parabola using them. Quickprop often converges quickly.

**Note about Weights**

The number of weights in a neural network represents the degrees of freedom of the neural network.

When there are more weights, it is easier for the neural network to overfit the data.

In a neural network that has \( N \) inputs and \( H \) hidden nodes, there are \( (N+1)H+H+1 \) weights in an MLP neural network. This grows quickly as either \( N \) or \( H \) increases.
Can Neural Networks Be Explained?

The fact is that the coefficients explain exactly what is happening in the neural network.
However, people do not readily understand what the complicated equation means.

Attempts to Explain Neural Networks

Many things that people do to try to explain neural network models do not work very well.
- A decision tree with the neural network score as the target variable can characterize high scores and low scores, but does not really explain what the neural network does.
- Sensitivity analysis can also be misleading. Variable $X_i$ might have little effect when other variables are set at their means, but have a large effect as part of some other interaction.
- Visualize which hidden units dominate the output for different regions of the input space.

If explaining how the model produces results is important, then neural networks are not the right technique.
Neural Networks Are Sensitive to Data

Works best on fewer (dozens, rather than hundreds) inputs
- Each new input means several new weights.
- You should spend time choosing the best inputs.

Categorical Inputs
- Use caution when converting to numeric.
  - Use one of $n$-1 coding.
  - Find a natural ordering.

Continuous Inputs
- Pay attention to the distribution.

Null Values
- These are a significant problem.

In a multilayer perceptron, the number of connections grows faster than the number of nodes. The number of nodes grows with the number of inputs. One additional input variable can be responsible for many additional weights to be optimized. The number of training examples needed is largely a function of the number of weights to be optimized. When there are too many input variables, the training process becomes lengthy and the chances of getting a good model diminish. In SAS Enterprise Miner, the variable selection node can be used to pick a reasonable set of variables. Another approach, using the decision tree node for this purpose, has the advantage of producing a model that might have some explanatory value.

The process of transforming input values into the form required for a neural network has the potential to either destroy information or add in spurious information.

Because neural networks do actual math on the inputs, they are sensitive to outliers. A decision tree split, by contrast, only cares whether the value being tested is greater than the split value. It might be greater by one or by one million; the result is the same.

The next slides address these data preparation issues.
Do Not “Use” Any Variables... Almost

In SAS Enterprise Miner, set the Use column value for all variables to No. Then, add variables one at a time. Remember to set the target variable to Use!

The Problem with Spurious Ordering for Categorical Variables

An arbitrary assignment of states to numbers makes Washington closer to West Virginia than to Oregon.

Neural networks use the actual values to create the model. The order (especially) and magnitude (a bit less) can affect the resulting model.

When categories are mapped arbitrarily to numbers, unordered data becomes ordered. This adds spurious information. It is a common mistake to map the states of the United States to the numbers 1 through 50, for example. This makes it seem that Alabama is much closer to Alaska than to Mississippi or Tennessee. There is no way to negate this falsehood.

Copyright © 2009 by Data Miners, Inc.
Eliminating Spurious Ordering Using N-1 Coding

<table>
<thead>
<tr>
<th>Flavor</th>
<th>AK</th>
<th>AL</th>
<th>AR</th>
<th>AZ</th>
<th>unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

One solution is to introduce a separate variable for each possible value. This is called “1 of n-1” encoding.

Problems:
- This method creates a large number of input variables.
- The input variables are highly correlated.
- Most neural network tools do this automatically.

Generally, this is a bad idea.

One solution is to create a series of indicator variables, one for each flavor. No separate variable is needed to encode other because that is represented by all the flavor indicators having zero values. This technique solves the problem of spurious ordering, but introduces a new problem, a proliferation of variables.

SAS Enterprise Miner automatically uses 1 of n-1 coding for categorical inputs to a neural network. Notice that this can greatly multiply the number of inputs, so categoricals should be used sparingly, if at all, for neural network inputs.
Finding a Natural Ordering

Previous response by State

<table>
<thead>
<tr>
<th>State</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AK</td>
<td>1.2%</td>
</tr>
<tr>
<td>AL</td>
<td>1.4%</td>
</tr>
<tr>
<td>AR</td>
<td>2.0%</td>
</tr>
<tr>
<td>AZ</td>
<td>4.1%</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

In the Vermont Country Store data, for instance, you can take the proportion of responders to catalogs in 1997 and 1998 by state and use this as an input variable for modeling.

When possible, the best solution is to look for a natural ordering of the categories, one that has something to do with how the categories relate to the target variable. In this example, coupons were distributed for boxes of ice cream treats in various flavors. These coupons have different redemption rates. If redemption rate can be shown to be related to the target variable in some way, then it makes sense to replace the categorical flavor variable with the numerical redemption rate. Sometimes there can be several interesting numerical variables associated with a single categorical variable. You might discard the telephone handset model, but keep its weight, its relative popularity, and the churn rate associated with it. Similarly, you might discard ZIP codes, but keep the population density, median home price, and percent of owner-occupied homes.
A few very high net-worth individuals can cause a problem. The simplest transformation of these values into a small range would be to divide all of the values by the largest. That would be a mistake because most values are very close to zero in order to allow for Bill Gates to be at 1. Sometimes it makes sense to discard records that are extreme outliers. Other times it makes more sense to apply some sort of transformation.

Perhaps the best way to handle skew is to use standardized values rather than raw values for numeric inputs. Alternative transformations, such as scaling or taking the log, can also be useful.
The name thermometer code arises because the sequence of ones starts on one side and rises to some value, similar to the mercury in a thermometer. This sequence is then interpreted as a decimal written in binary notation. Thermometer codes are useful when the difference on one end of the scale is less significant than differences on the other end. For instance, for many marketing applications, having no children is very different than having one child. However, the difference between three children and four is somewhat negligible. Using a thermometer code, the number of children variable might be mapped as follows:
- 0 (for 0 children)
- 0.5 (for one child)
- 0.75 (for two children)
- 0.875 (for three children)
- and so on

For categorical variables, it is often easier to keep mapped values in the range from 0 to 1. This is reasonable. However, to extend the range from –1 to 1, double the value and subtract 1.

Thermometer codes are one way of including prior information into the coding scheme. They keep certain code values close together because you have a sense that these code values should be close. This type of knowledge can improve the results from a neural network, that is, the knowledge does not make the network discover what you already know. Map values onto the unit interval so that codes close to each other match your intuitive notions of how close they should be.
Variable Selection for Neural Networks

Because neural networks do not like a large number of variables, how can you choose the best one?

- Use good judgment to choose variables.
- Use decision trees to choose them.
- Avoid categorical variables!
- Standardize numeric variables.

Stepwise and forward selection are not yet available.

Use clever ways to reduce the number of variables, such as principal components.

Additional Neural Network Nodes

In addition to the basic Neural Network node, SAS Enterprise Miner contains two interesting extensions:

- Automatically finding the best parameters for a given problem (AutoNeural node)
- Combining neural networks with principal components (DMNeural node)
AutoNeural

The AutoNeural extension does what computers do best:
- tries out different network architectures and sizes to find the best network for a given problem
- does not try out different combinations of inputs, so inputs still must be chosen

Automatic Neural Networks

A diagram similar to the one above uses a decision tree to choose variables and then uses the AutoNeural node to choose the best parameters for a neural network (and to build the network too).

Using SAS Enterprise Miner, it is possible to automate much of the process of building neural networks. Decision trees, for instance, are a good way to select variables. When connected to another modeling node, the second node knows to choose the variables that the decision tree chose. The AutoNeural node then takes these variables and builds a neural network, after testing different network topologies.
Variable Reduction Using Principal Components

Instead of selecting only a few variables, an alternative approach is to combine the inputs using principal components.

You can do this explicitly with the Principal Components node, or by using the DMNeural node (which uses principal components but also tries different transfer functions).
Principal Components

The first principal component is similar to the best fit line, and it has a similar definition. It is the line that minimizes the square of the distance from each point to the line. This is also called *orthogonal regression*.

Remember that the best fit line minimizes the square of the vertical difference rather than the overall distance.

Statistically, the definition is phrased differently as the line that maximizes the variance of the expected values.

Principal component analysis is an area of statistics that reduces dimensionality of inputs by introducing linear combinations of the inputs. That is, given a set of variables, how can you capture as much information as possible in a single variable? This is an example of reducing the dimensionality of the data.

The first principal component can be thought of as the long diagonal of the data points in $n$-dimensional space. This is the line that minimizes the sum of the distances from all the points to the line. This is similar to linear regression, but different because you are looking at total distance rather than distance in only the Y-direction.

As with linear regression, you can calculate residuals (which are shown as the short black lines in the chart) using vector subtraction. In this way, you can continue to find the second principal component and the third up to the number of dimensions of the data.

These principal components have some useful properties. For instance, they are linearly independent of each other, which make them a good choice for reducing a large number of inputs to a small number of inputs. It is also a good property for simple regressions. As you recall, when talking about linear regression, you want the inputs to be independent of each other. This is exactly a property of principal components.
Properties of Principal Components

Although they look similar, principal components are very different from best-fit lines.

- Principal components treat all the dimensions in the same way.
  There is no special dimension that corresponds to the target.
- You usually apply principal components only to inputs.

The distance from each point to the first principal component is a vector. That is, it has a magnitude and direction. These can be plotted for a second principal component, and a third, and so on.

Properties of Principal Components

The number of principal components matches the dimensionality of the data.
Each principal component is perpendicular to all the others.
Each principal component is a linear combination of the input variables.
Principal components are a good way to reduce the number of input variables. In fact, they are used in SAS Text Miner for handling text.
Other Uses: Visualizing Clusters

This example from the clustering node shows the centers of clusters on a two-dimensional plot using the first two standardized principal components.

Other Uses: Variable Clustering

Variables are grouped by how much their first principal component dominates the principal components. Such groups are clusters. The procedure is repeated, and uses available variables and clusters at each stage. This is one method for reducing the number of variables.
Lessons Learned

Neural networks are a powerful tool for building models.

Neural networks do not provide an explanation of how they work, or more accurately, the explanation is a complicated equation that humans do not understand.

When using neural networks, you need to reduce the number of input variables.

One method of reducing the number of inputs is to use principal components.

Refer to Exercises for Chapter 6 in the handout.
Chapter 7  Memory-Based Reasoning

7.1  Memory-Based Reasoning ........................................................................................................7-3
7.1 Memory-Based Reasoning

This Chapter...

...introduces memory-based reasoning (MBR), one of several data mining techniques that depend on the concept of distance. The chapter discusses the following topics:

- distance and similarity
- choosing a suitable training set
- many examples of MBR, including for text classification, medical diagnosis, and identifying songs
- combining input from several neighbors
- collaborative filtering
Memory-Based Reasoning

Memory-based reasoning (also known as k-nearest neighbor or case-based reasoning) predicts unknown values for records in a data set based on a combination of values for the K records most similar to it in an historical data set.

Humans often reason by example, and find situations similar to new ones that they encounter. MBR operates on the same principle.

People use memory-based reasoning all the time in formal ways such as diagnosing a disease by comparing its symptoms to the symptoms that are usual for various diseases, and in informal ways such as guessing the prices charged in a restaurant by the presence or absence of candles and tablecloths. The basic idea is simple. To classify an unknown example, compare it to various known examples that are similar. If most of these known examples belong to a particular class, assign the new record to that class as well. If the patient has a circular rash and has recently been bitten by a tick, chances are good he or she has Lyme disease because that circular rash is the first symptom many patients notice. If the patient later develops fever and joint pain, the diagnosis becomes more certain because these are symptoms that often follow the initial rash in Lyme disease.
Local Probability Density Estimation

What is the probability that A is a circle?
What is the probability that B is a circle?

This is somewhat related to the idea of paired tests.

An unclassified record near A is likely to be a red circle. An unclassified record near B might also be a red circle, but has a good chance of being a green triangle instead.
This scatter plot shows that the familiar towns of New York fall plotted against two variables: the log of the town population and the median home value. Viewed through this particular prism, Manhattan has no near neighbors; it has a huge population and outlandish home values. Brooklyn and Queens are very close neighbors of each other with large populations and similar home values, which are significantly above the state median, but well below those in Manhattan and the richer suburbs.

This scatter plot also shows the median rent. The top (most expensive) quintile is represented by circles. The bottom quintile is shown as triangles. The middle three quintiles are squares.
You can use this scatter plot to illustrate the basic ideas of memory-based reasoning by trying to understand one town based on what happens in neighboring towns. In this case, neighboring means similar along these two dimensions of population and median home value, but not similar geographically. This example looks at median rent.

The first step is to find the K nearest neighbors by gradually expanding a circle around Tuxedo, the target town, until the desired number of neighbors is found. In this case, K=2 and the nearest neighbors are Shelter Island in Suffolk County and North Salem in Westchester County. These three towns are dissimilar in many ways, but along the two chosen dimensions, they are close. That similarity depends on the variables used to measure something is an obvious but important point that should be remembered when you use any nearest neighbor approach. All three towns have a population close to the state median and home values that rank near the top of a list of towns sorted by home value.
Depending on which variables are used to describe them, the towns of Chenango and Fenton are more or less close to one another.
General Procedure for MBR

General procedure for memory-based reasoning (MBR):

- Save all of your pre-classified data.
- For a new record, find the nearest records in the pre-classified data. These are the neighbors.
- Use the neighbors to decide on how to assign a value to the new record.

Two Fundamental Concepts

- Distance – How do you measure the distance between two things?
- Combination function – How do you combine the results from the neighbors?

You determined what the nearest towns are by drawing circles on a plot that has axes scaled in a seemingly arbitrary way. Clearly, a more precise definition of distance is required. In fact, a distance function is the fundamental requirement for using a nearest neighbor approach such as MBR.

However distance is defined, after you have a distance metric, you can say that all towns within a certain radius will be used as neighbors. Alternatively, you can say that the K nearest towns will be used as neighbors. This example takes the second approach with K being set to 2.

After the neighbors are found, you must somehow combine them to determine a classification or estimate. In this case, to estimate rents, you could average the median rents of the neighboring towns to calculate an estimated median rent for Tuxedo, but that is by no means the only possibility. One possible variation is to use a weighted average with the nearest neighbors having greater weight than those farther away.
To build a distance function for multi-field records, start with distance functions for each field. These field-level distances are the distance along a single dimension of a multi-dimensional space.

Not all fields are numeric, of course. Here are two examples of distance functions for categorical fields:

- The first is the count of matching fields. In each dimension, if the items match, they are distance 0. If they do not match, they are distance 1.
- The second metric is good for items that can be placed in a hierarchy such as the hierarchies used by grocery stores to categorize the things they sell.

The hierarchy is a tree with many branches. Things that are closely related should be near each other in the tree. Things that are not related at all will be connected only through the root.

After the field-to-field distances are computed for each field, all that remains is to combine them to form the record-to-record distances. This can be done in several ways.
In MBR, the training set is the model, so creating a good training set is very important. As with other data-mining techniques that you studied, balancing the training set so that there is a similar number of each class leads to a better model. If this is not done, a majority of a record’s neighbors are likely to be the majority class.

Similarly, the technique cannot work unless there are many examples of each class.

Because distance is involved, all of the numeric values should be comparable. Converting them all to Z scores is the approach taken with the SAS Enterprise Miner MBR node.

Although distance functions can be constructed for categorical variables, it is difficult to make them comparable with the numeric distances in other dimensions. A better approach is to substitute related numeric variables for categorical ones.
In SAS Enterprise Miner

Combination options:
- Sum or weighted sum
- Number of neighbors

Distance options:
- Euclidean distance

Outside the node
- Standardized inputs
- Principal components
- No validation set
MBR Case Study: 
Classifying News Stories

Text mining is important and different.
- Much of the world’s information is in the form of text.
- Text is free-form, which is different from structured data.
- Features can be extracted and represented in a structured way.
  - Term-document matrix
  - Parts of speech

The News Classification task is important and different from others.
- Assigning the right keywords helps users find the right stories.
- There are multiple correct labels for each story.
- Different stories have different numbers of labels.

Another text-mining example is in the section on genetic algorithms.

An example of MBR in action is the classification of news stories. This work was sponsored for Dow Jones, which publishes the *Wall Street Journal*.

Editors assign each story a set of codes that describe the contents of the article. The codes themselves are in several categories, such as I for industry and R for region. The purpose of the codes is twofold.

- First, the codes enable readers to search more specifically for the type of article they are looking for. Articles about IPOs in Japan can be found more easily using the codes rather than the search terms **IPO Japan**.
- Second, the codes also allow stories to be routed automatically, for instance, to users who are interested in receiving articles about IPOs in Japan via e-mail.

As a classification problem, this is complicated.

First, you can assign any number of codes to a story, so no single classification code is the right answer. You also have to determine how many codes.

Second, the data that you are using is free text data, which is typically handled by extracting specific features from the data. In this case, however, there is special software that can calculate the distance between two articles using a technique called *relevance feedback*. A consequence is that this complicated example provides a good example of MBR in action.
What the Codes Look Like

Use a major code, such as I for Industry, R for Region, or S for Subject, followed by a detail code.

Examples:
- I/INS for articles about the insurance industry
- R/JAP for articles relevant to Japan
- S/IPO for articles discussing initial public offerings (IPOs)

An article about all three would presumably be about insurance company IPOs in Japan.

An article can have one or more codes in multiple categories.

How Can MBR Help?

MBR starts by identifying a set of neighbors.
MBR helps by finding neighbors of a news story and looking at the codes for these neighbors.

The training set consists of about 50,000 stories from many different news sources (not only Dow Jones). These are news articles, containing examples of over 300 codes. Each story has several codes assigned to it, often with more than one code in a category.

Relevance Feedback

To assign keywords, we borrowed from a project whose goal was to eliminate the need for keywords.

In relevance feedback retrieval, the user initially provides a small number of query terms. These are used to produce an initial list of candidate articles that might or might not be relevant.

The user indicates which articles are relevant and the entire text of the relevant articles provides a new, much-enlarged set of query terms.

“Get me more like this one.”

Which Articles Are “Like This One”?

Count the number of words that the documents contain in common, and eliminate common words such as the and and that do not affect the meaning. Optionally, weight the words so that sharing a rare word counts more than sharing a common word. Use the count of shared words to establish a distance between 0 and 1.

To find the neighbors, you need to know the distance. The distance function uses a technique called relevance feedback, which measures the similarity in vocabulary between the two articles (when common words are eliminated).

How SAS Enterprise Miner’s Text Node Works

The SAS Text Miner node gives the retained terms in each document a weight to form a term-document matrix. There are several choices for calculating the weights. The dimensions of the term document matrix might be reduced by singular value decomposition. The distance between two documents is calculated as the cosine of their respective weighted term vectors, or as the Euclidian distance.

The Text Miner node supports a variety of sophisticated distance functions.
Back to the Story

The training set had three months of news stories (49,652 stories from over 100 news sources) already containing codes.

The stories contained an average of 2,700 words.

There were 361 different codes in six categories. Some codes were fairly common and others were rare.

An additional evaluation set consisted of 200 more stories.

For testing the results, a specially coded set from 200 additional stories was used. The codes on these stories were re-reviewed by the editors.

How Were the Results Combined?

Simple majority
- Look at all the codes at nearby neighbors and assign codes that are in the majority of neighbors.

Weighted majority
- Look at all the codes at nearby neighbors and give each one a weight that is higher for closer neighbors. Assign the codes with the highest weights.

Use the weighted majority and, by experimentation, use 11 neighbors.

Each of the nearest articles also had multiple codes. After different groupings of combination functions were investigated, the final one chosen was a weighted majority voting. Each neighbor voted for its own codes using the inverse of the distance as a weight. Codes whose total votes exceeded some threshold were assigned to the news story.
How Results Were Measured

Two hundred stories were sent back to a panel of senior editors for re-coding.  
- An expensive process

The original training set consisted of unverified codings. Only codes agreed upon by a majority of the panel were included.

The assigned codes were compared to the codes originally assigned by the editors. This comparison is difficult because of the following reasons:
- Some stories are assigned by both methods.
- Some stories are only assigned by editors.
- Some stored are only assigned by MBR.

The results were tested against human editors.

MBR Comparison with the Experts

The experts did not always agree on codes to be assigned to a story. MBR did better than novice editors and as well as intermediate editors. MBR did better on categories with many examples for each code, and got more than 90% of the codes in some cases.

The results for the 200 stories were compared with the results from the editors. The MBR system did as well as intermediate editors. MBR found 88% of the codes assigned by humans; in addition, it added about a fifth more codes not found by the humans. Overall, the results were impressive, given that no information about the coding system itself was used to assign codes.
Another Text Mining Application of MBR: Classifying Census Responses

Remember the industry and job fields that helped to predict product penetration in the New York towns?

Historically, the Census Bureau gets them from longhand text answers to questions on the census' long form.

MBR can be used to make the same assignments.

Assigning industry and job codes has been a challenge for the U.S. Census since 1850, when occupation information was first included. The definitions of the codes changed over time. Now they are more formalized and standardized than ever before. On the census forms, people are asked open questions such as “What type of work was this person doing?” and “What were this person's most important duties?” The text answers need to be coded to be more useful.

Starting with 2010, the U.S. Census will no longer use the long form of the census. Instead, the agency will do continuous sampling around the country, and ask questions similar to those on the long form.

Images and Video: More Difficult than Text

MBR is used for facial detection and recognition.
Applications range from identifying actors and actresses in film clips to finding bad guys in surveillance videos.
Detecting faces in images is the first step to recognizing them, and it is a more complicated problem than first appears. Many data-mining techniques were used to try to solve this problem. When you use MBR, one goal is to reduce the size of the training set, because the search for faces in a picture requires looking at many different parts of the picture.

When you reduce the size of the training set, the goal is to find outlier examples of faces and non-faces. Here the non-faces show traits similar to faces, but not quite enough detail. Similarly, the faces are recognizable, but small changes would make them disappear.

This example comes from the paper “Example-based Learning for View-based Human Face Detection” by Kah-Kay Sung and Tomaso Poggio and is available at ftp://publications.ai.mit.edu.ai-publications/pdf/AIM-1521.pdf.
The challenge of detecting faces in pictures is a major computational task, but one that is quite important as a first step toward recognizing the faces. Faces can appear anywhere in the picture, be of any size and orientation, and there can be any number of them. Visual detection is something that humans do very well; it is something that computers do not do as well.

Nearest neighbor techniques were applied to this area since at least the early 1990s. The idea is to compare every area of a picture to a database of faces and non-faces to determine where the faces are. Comparisons using MBR work quite well, but are computationally intensive. This is an example of a problem where reducing the size of the training set is important for creating a practical system.
Increasing the Efficiency of MBR

One problem with video data is that it is very large. Nearest neighbors need not be very near if they are still the right class.

One problem with MBR is the large number of records kept in the training set that must be consulted for each classification. This picture shows that many of those records are superfluous. By keeping only those near an anticipated boundary, the same classification can be made with fewer calculations.

MBR for Medical Diagnosis

Dr. Georgia Tourassi of Duke University Medical Center developed a diagnostic aid for breast cancer based on MBR.

Mammogram images are compared to a database of thousands of images with known diagnoses.

The system not only produces a diagnosis; it also shows the physician the neighboring cases on which the diagnosis was based.
The Basic Idea

Query Mammogram

Knowledge Database

- Normal Mammogram
- Abnormal Mammogram

K most similar
normal cases

K most similar
abnormal cases

DECISION ALGORITHM
The tool is identifying two factors. First, are the observed features in the mammogram “true masses”? Second, when there is a true mass, are the corresponding images benign or malignant?
Identifying Songs

Shazam is software that enables users to recognize songs, particularly from mobile devices. Let your phone "listen" to the music for a little while, and it recognizes the tune. The methodology is robust, with respect to many things that affect sound quality:

- different acoustic abilities of equipment
- compression
- noise
- starting at random points in the song

Shazam is a remarkable application for mobile devices that enables a user to record a part of a song, and then Shazam recognizes the song. Although a popular and impressive demonstration at a party, this technology definitely has more profitable uses. For instance, it is possible to automatically listen to many broadcasts and identify the rights holders to the works. This facilitates both supporting rights holders financially as well as detecting unauthorized use of such works.

Shazam is candid about how the technology works. All figures in this section come from “An Industrial-Strength Audio Search Algorithm,” by Avery Li-Chun Wang (www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf) or from his presentation at ISMIR 2003 (ismir2003.ismir.net/presentations/Wang.pdf).

Using MBR to Identify Songs

Define a “distance” metric.

- The metric is the number of consecutive seconds that an audio clip matches a known song.
- Take into account acoustic issues and background noise.

Calculating this “distance” requires delving a bit into acoustics.

- Use the “frequency” domain rather than the “time” domain.
- Only consider peaks in the frequency domain.
- Do some further tricks to make the algorithm run faster.
- The problem of “seconds of overlap” becomes “seconds of overlapping peaks in the frequency domain.”
Spectrogram of the Sound

It starts with a spectrogram of the sound (created from a .wav file).

The spectrogram captures the details of the audio file in the frequency domain for each one-second period of the sound file.

The algorithm starts by converting all songs and excerpts into a spectrogram. The spectrogram records the frequencies that are used, and they are sampled periodically throughout the song. These spectrograms are large, because they contain enough information to reconstruct the song.

Peaks in the Spectrogram

Peaks in the spectrogram become features. This is the audio signature.

The peaks in the spectrogram form a constellation. These uniquely identify the song. An excerpt should exactly match the constellation. However, searching for such a map exhaustively is computationally prohibitive.

Peaks are extracted from the spectrogram. These peaks form a unique pattern that describes the song. Presumably, any excerpt would have a subset of these peaks. However, searching for such a match is computationally prohibitive. This is an example of a nearest-neighbor algorithm, where the nearest neighbor is described by the number of matching peaks.
A clever method is used to reduce the search space. Each peak is considered an anchor point, and for each anchor, a target region is chosen. The target region is later in time and does not include all frequencies (to filter out the effects of noise). The data is then the pairs of points between the anchor and each peak in the target.

This information is calculated both for the songs (where it is kept in a data store) and for the excerpt. The problem is matching the excerpt to the song.

Now it is a question of matching anchor points to anchor points. When the excerpt matches enough consecutive anchor points, then a match is found.
In the end, the matches are essentially lined up in a scatter plot, where the points represent the offset by second in each of the sound files (the sample and the song).

A match is indicated by a diagonal line in the scatter plot.
Collaborative filtering is a variant of memory-based reasoning particularly well-suited to the application of providing personalized recommendations. A collaborative filtering system starts with a history of people’s preferences. The distance function determines similarity based on an overlap of preferences, that is, people who like the same thing are close. In addition, votes are weighted by distances, so the votes of closer neighbors count more for the recommendation. When someone is nearer, the vote counts more in making a recommendation. In other words, it is a technique for finding music, books, wine, or anything else that fits into the existing preferences of a particular person by using the judgments of a peer group selected for their similar tastes. This approach is also called social information filtering.

One of the hardest things about collaborative filtering is establishing the initial set of ratings used to establish the neighborhoods. Many Web sites that offer customers the ability to rate purchases find that very few do. Another difficulty is that in a large space such as books, any one person can rate only a small fraction of the titles. This results in sparse ratings vectors with little overlap between people. One solution is to get everyone to rate the same core list of popular titles. On the other hand, ratings of rare items might provide more information.
The Global Recommendation

The global recommendation might not work for the individual.

Sometimes There Is No Need to Be Clever

The data mining book you are looking at might be a better clue than the dozens of spy stories that you bought in the past.

amaznode.fladdict.net/
Lessons Learned

Memory-based reasoning combines information from neighbors to arrive at a prediction.

The “nearest” neighbors are defined by whatever input variables you want.

MBR can be used for non-traditional problems, such as classifying news stories and medical images.

MBR is powerful, but requires storing all the training data. In a sense, the data is the model.
Chapter 8  Clustering

8.1  Clustering
8.1 Clustering

This Chapter...

- revisits the idea of distance between records
- introduces some uses for clustering
- introduces the $k$-means clustering algorithm
- introduces divisive clustering algorithms
- introduces agglomerative clustering algorithms
- introduces self-organizing maps
- discusses data preparation for clustering
- demonstrates finding clusters with the SAS Enterprise Miner Clustering node.
**Clustering Does Not Use a Target Variable**

Clustering finds groups of records that are similar to one another.

Clustering can help you do numerous things:
- make sense of large amounts of data
- find customer segments
- reduce the records by finding prototypical examples
- divide large data sets into smaller more homogeneous pieces

Understanding clusters requires human intelligence.
However, it can be interesting to build clusters on input variables and see what happens to a target variable in the cluster.

Clustering is an example of undirected data mining. Clustering algorithms look for similarities within data records, not based on a target variable but simply based on self-similarity. Clustering might be used for several purposes. One purpose is simply to make sense of large amounts of data. What patterns are lurking inside the records? The clusters might then be modeled separately for some purpose.

Clustering and customer segmentation are related, because they are often doing similar things. However, clusters are not necessarily customer segments, because they do not have the human intelligence that goes into segmentation. In fact, clusters are only useful insofar as they provide insight.

Sometimes patterns that are not apparent in a large, noisy data set can be found within simpler clusters. Clusters can also represent naturally occurring customer segments. Sometimes, most records fall into one big cluster. Outliers might represent fraud, disease, or some other anomaly.
Clustering and Segmentation

Clustering produces segments. There are also many other ways to create segments:

- geography
- product
- revenue
- attitudes expressed to market researchers
- demographics

Segments created through clustering might not have any obvious or useful interpretation.
Astronomical Clustering

The Hertzsprung-Russell diagram clusters stars by temperature and luminosity.

Early in the 20th century, astronomers trying to understand the relationship between the luminosity (brightness) of stars and their temperatures, made scatter plots similar to this one. The vertical scale measures luminosity in multiples of the brightness of the sun. The horizontal scale measures surface temperature in degrees Kelvin.

Two astronomers, Ejnar Hertzsprung and Norris Russell, thought of doing this at nearly the same time. They observed that the stars fall into three clusters. This observation led to further work and the understanding that these three clusters represent stars in very different phases of the stellar life cycle. The relationship between luminosity and temperature is consistent within each cluster, but the relationship is different between the clusters because fundamentally different processes generate the heat and light. The 80 percent of stars that fall on the main sequence generate energy by converting hydrogen to helium through nuclear fusion. This is how all stars spend most of their active lives. After some number of billions of years, the hydrogen is consumed. Depending on the star's mass, it then begins fusing helium or the fusion stops. In the latter case, the core of the star collapses, generating a great deal of heat in the process. At the same time, the outer layer of gasses expands away from the core, and a red giant is formed. Eventually, the outer layer of gasses is stripped away and the remaining core begins to cool. The star is now a white dwarf.
Clusters of people with similar behavior are naturally occurring market segments.

In the business world, one common use for clustering is to find groups of customers with similar characteristics. The different clusters might have different product affinities, different channel affinities, different product usage patterns, and so on. These differences can suggest different marketing treatments.

The usual approach to customer segmentation is to use business rules based on common sense. Such rules can be considered hypotheses to be tested. Clustering offers a way, based directly on data, of segmenting customers.
Clusters of Products

Clusters of products that sell together can form the basis of recommendations.

Products often fall into clusters of things that are frequently purchased together. This information can be used to make recommendations. When a customer purchases a partial cluster, you can try to fill in the missing pieces.

Clusters Used to Guide Marketing Effort

In the 1990s, Bank of America’s National Consumer Assets Group embarked on a pilot data mining program that resulted in increased adoption rates for home equity loans.

Before the project, home equity loan marketing targeted two groups:
- parents of college-age children
- people with high but variable incomes

Marketing materials and outbound call lists reflected this focus.

Perhaps the consultants had variable incomes and children in college.

Some of this material comes from a Bank of America case study written by Larry Scroggins based on work by Lounette Dyer and Bob Flynn, then of Casablanca Information Services. Used by permission.
A Scoring Model Came First

Using a training set of current customers, Bank of America developed a model to score customers on their propensity to respond to a home equity loan offer.

- Eleven percent of customers fell into the “most likely to respond” category.

Then the bank used clustering to learn more about the customers who had a high response likelihood.
Here is an example of how clustering helped one large bank improve its marketing of home equity lines of credit. The picture is a parallel dimension plot that shows some of the customer clusters found by the bank. This type of graph is called parallel coordinates.

The highlighted cluster revealed a marketing opportunity for home equity credit lines. The cluster included many examples of people who had both personal and business-oriented banking products under the same tax ID. Because only small, sole proprietorships use the owner’s Social Security number as a tax ID, these are small businesses. The existence of this cluster led the bank to the realization that people use home equity credit lines to fund the startup of a small business. None of the bank’s marketing material was oriented that way. A new campaign based on this idea received much better response than the previous one that focused on using home equity credit lines to pay for college tuition.
Response Rate More than Doubled

In all, 14 different clusters were discovered. Many of them had no discernable use. One of them was notable for the high number (30%) of cluster members who had business as well as personal accounts. (Sole proprietorships often use the owner’s Social Security number as the business Tax ID number.) This cluster accounted for 27% of the 11% of overall customers who scored highest for propensity to respond to a home equity offer. An analyst noted the pairing of home equity lines and small businesses and hit upon the idea of marketing home equity lines as a way to get start-up capital for a small business.
Clustering for a Better Fit

The U.S. Army wanted to reduce the number of uniform sizes in inventory. Cornell researchers clustered a database containing more than 100 measurements for 3,000 women. A small number of variables describe a small number of distinct body types.


Ordinary ways of measuring sizes of women’s clothes consist of graduated sizes where all dimensions increase together. An 8 is bigger than a 6 and smaller than a 12 in all dimensions. Different body types are acknowledged through the use of odd and even sizes, petite sizes, junior sizes, and others.

Ashdown’s approach was more radical. She found clusters in a database of detailed body measurements of female soldiers. In the end, the clusters could be characterized by only a few variables. Each cluster represents a different basic body type with different relative proportions. The result is better fitting uniforms with a smaller number of sizes.
The \( k \)-Means Clustering Algorithm

The \( k \)-means clustering algorithm assumes a geometric interpretation of the data. That is, the records are points in an \( n \)-dimensional data space.

- Assume that there are \( k \) clusters.
- Start with \( k \) clusters with centers chosen arbitrarily.
- Iteratively improve them.

The \( k \)-means clustering algorithm depends very much on how distance is measured.

A popular clustering algorithm that is available in SAS Enterprise Miner is \( k \)-means. Although computationally intensive, this algorithm existed since at least the 1960s. In the previous slide, \( k \)-means is the clustering algorithm that was used to cluster female soldiers.

Data Preparation for Clustering

Same as for MBR:

- Standardize numeric values.
- Replace categorical variables with related numeric ones.
- Keep input fields independent of one another.

Consider using weights to provide domain knowledge.
Consider using principal components.

Because both rely on distance metrics, the data preparation issues for clustering are similar to those for MBR.
The $k$-Means Algorithm

The $k$-means algorithm starts by randomly choosing a certain number ($k$) of seeds to be the centroids of clusters. These are usually taken from the data.

How the initial seeds are chosen is not important. They are only there to provide a starting point for the algorithm.

Assigning Each Point to a Centroid

All the points are now assigned to the nearest centroid. This is the initial cluster assignment.

Each record is provisionally assigned to a cluster based on which seed is closest.
Moving the Centroids

The centroids are moved to the center of their respective clusters.

After all the points are assigned to a cluster, the centroid of each cluster is recalculated. All records are reassigned to the cluster whose centroid is the closest. This process continues iteratively until the centroids stop moving.
New Cluster Assignments

The process of moving the centroid and assigning clusters is repeated until the clusters are stable. This usually happens after a handful of iterations.

Notice that the cluster boundaries are straight lines.

You can think of this as a geometric process.

The three lines on the above diagram divide the plane. The leftmost line divides the plane into those points that are closest to the bottom left (purple) cluster and those that are closest to the upper middle (green) cluster. The middle line divides the plane between the bottom left and rightmost cluster (blue), and the central line between the upper middle and rightmost cluster.

Using these lines as a guide, it is easy to assign points to the cluster center closest to the point.

The algorithm usually converges after a handful of iterations to a stable set of clusters.
Decision trees are a good way to explain what clusters are doing. The splits for decision trees are straight lines (or hyperplanes) that are parallel to the axes. The boundaries between clusters are straight lines (or hyperplanes), but typically angled. Even so, decision trees do an impressive job of describing the clusters.

In this decision tree, which is predicting a cluster ID, all the leaves have a density of 100%.
Gaussian Mixture Models

Gaussian mixture models are similar to k-means clustering. The overall distribution of points is assumed to be described by a combination of \( k \) different probability density functions. The parameters (mean and variability) of the \( k \) Gaussians are iteratively adjusted.

Each Gaussian has some responsibility for each point.

Gaussian mixture models are a probabilistic variant of k-means. The name comes from the Gaussian distribution, a probability distribution often assumed for high-dimensional problems. (It only applies the normal distribution to more than one variable.) As before, the algorithm starts by choosing \( k \) seeds. This time, however, the seeds are considered to be the means of Gaussian distributions. What happens next is an expectation maximization procedure that tries to optimize the parameters of each Gaussian distribution and the weights used to combine them to maximize the likelihood of the observed points.

The algorithm iterates over two steps called the estimation step and the maximization step.

The estimation step calculates the responsibility that each Gaussian has for each data point. Each Gaussian has strong responsibility for points that are close to it and weak responsibility for points that are distant. In this picture, the line thickness represents responsibility. The responsibilities will be used as weights in the next step.
The algorithm alternates between two steps: the estimation step and the maximization step.

- In the estimation step, each Gaussian is assigned some responsibility for each point. Thicker lines indicate greater responsibility.
- In the maximization step, the mean of each Gaussian is moved toward the centroid of the entire data set, weighted by the responsibilities.

These steps are repeated until the Gaussians are no longer moving. The Gaussians can grow as well as move. However, each Gaussian is constrained so if it shows a very high responsibility for points close to its mean, then there is a sharp drop-off in responsibilities. If the Gaussian covers a larger range of values, then it has smaller responsibilities. Because the distribution must always integrate to one, a Gaussian weakens as it enlarges.

The reason this is called a mixture model is that the overall probability distribution is the sum of a mixture of several distributions. At the end of the process, each point can be labeled as belonging to the cluster that claims it most strongly. Alternatively, the competing assignments can be normalized to provide the probability that each point belongs to each of the clusters. This is sometimes called soft clustering.
The Result: Soft Clustering

Sometimes it is useful to enable more than one cluster to claim a record.

- If a cluster has high fraud propensity, you might want to know about anyone who belongs even a little.
- If a cluster has high response rates, you might want to vary the threshold for inclusion to adjust the size of a mailing.

How Many Clusters Are There?

- It is perfectly sensible to find two clusters in a deck of cards.
  - a red one and a black one
- It is equally sensible to find four clusters.
  - one for each suit
- You can also find 13 clusters.
  - one for each card value

The choice is left to the user.

If there is no a priori reason to assume that a particular number of clusters exists, you might want to try several values of $k$ and evaluate the resulting clusters by, for example, finding the value of $k$ that yields clusters with the lowest average variance.
Additional Clustering Algorithms

$k$-means and its soft cousin, Gaussian mixture models, are by no means the only available approaches to clustering.
A few of the others are discussed:
- agglomerative clustering
- self-organizing maps
- divisive clustering
- $k$-means combined with divisive clustering
  - used in the case study

Agglomerative Clustering

1. Create a similarity matrix.
2. Merge the two closest clusters.
3. Update the new (smaller) similarity matrix.
4. Continue until all records are in one cluster.
Each step produces a candidate clustering.

This is available with the SAS CLUSTER procedure.

Agglomerative clustering starts with each record forming its own cluster. Nearby clusters are combined to make fewer, larger clusters. This process continues until all records are in one large cluster.
Agglomerative Clustering Example
The numbered boxes represent the ages of family members at a picnic.

Here, a group of people are clustered by age. The distance function is simply the difference in ages.

Agglomerative Clustering Example
In the first step, any two or more people who are only one year apart are clustered (agglomerated).
Single linkage distance means two clusters are as close as their closest members.

Single-linkage distance is used. This means that the distance between two clusters is the distance between their closest members. In this example, the distance between the cluster with an eight-year-old and the cluster with a 13-year-old is two.
Agglomerative Clustering Example

The next step looks for groups that are two levels apart. After this step, there are six clusters.

Agglomerative Clustering Example

After six steps, three stable clusters form. It will be a long time before the next merge.

At the point shown here, three distinct clusters are found: children, parents, and grandparents.
Agglomerative clustering depends on the distance between clusters to decide which clusters to combine. There are three different ways this can be done and, as shown in the picture, these can lead to different results.

In the single linkage method, the distance between two clusters is given by the distance between the closest members. This method produces clusters with the property that every member of a cluster is more closely related to at least one member of its cluster than to any point outside it.

In the complete linkage method, the distance between two clusters is given by the distance between their most distant members. This method produces clusters with the property that all members lay within some known maximum distance of one another.

In the third method, the distance between two clusters is measured between the centroids of each. The centroid of a cluster is its average element.
Divisive Clustering

1. Start with all records in one cluster.
2. Split into two clusters, each with a lower variance.
3. Continue until every record is in its own zero-variance cluster.
4. Evaluate each level and pick the best one for your application.

Agglomerative clustering starts with all the points in their own clusters and builds the clusters step by step. Divisive clustering takes the opposite approach, with all the data in a single cluster that will then be split. The process is similar to building decision trees. In fact, you can think of decision trees as divisive clusters where the purity measure is the similarity metric.

Divisive Clusters and Decision Trees

Decision trees are a directed version of divisive clustering.

- Divisive clustering reduces the average distance within child nodes by considering all possible splits.
- Decision trees reduce the variance or diversity of target variables within child nodes.

In a sense, decision trees form clusters with respect to the target variable.

People who think that they want clustering would often be happier with the segments created by a decision tree.
Self-organizing maps (SOM) are a neural-network-based approach to clustering developed by Finnish researcher Dr. Teuvo Kohonen.

The basic SOM has an input layer and an output layer. Each unit in the input layer is connected to one source, as in the networks for predictive modeling. As in other neural networks, each unit in the SOM has an independent weight associated with each incoming connection.

In contrast to the multilayer perceptrons that you already studied, the output layer consists of many units instead of only a handful. Each of the units in the output layer is connected to all of the units in the input layer. The output layer is arranged in a grid, as if the units were in the squares on a checkerboard.
When a member of the training set is presented to the network, the values flow forward through the network to the units in the output layer. The units in the output layer compete with each other, and the one with the highest value “wins.” The reward is to adjust the weights leading up to the winning unit to strengthen its response to the input pattern. Not only is the path to the winning node strengthened, but the paths to its neighbors in the grid are strengthened as well. This way, the number of clusters found is not determined by the number of output units because several output units can together represent the cluster. Clusters similar to each other should be closer together than more dissimilar clusters.
Evaluating Clusters

What is the correct value of $k$ in $k$-means?

What is the right level of clustering when you use agglomerative or divisive clustering?

Measures of the value of a set of clusters:

- cluster sizes being somewhat equal (for some applications)
- average variance
- distance between cluster formation and cluster merging
- ratio of average distance of members from cluster centroid to average distance between centroids

Because the point of clustering is to find groups of records that are similar, one obvious measure is the variance (the sum of the squared differences of each element from the mean). To take cluster size into account, you can divide the total variance by the number of elements in the cluster. The best clustering is the one whose clusters have the lowest average variance.

A problem with using variance is that for agglomerative and divisive clustering techniques that start or end with single-record clusters, these clusters are always preferred because they have zero variance. For agglomerative techniques, a better measure of cluster worth is the time that elapses between when the cluster is formed and when it is merged into another, larger cluster.

One measure that works for all clustering techniques is to compare the average distance between cluster members and the cluster centroid with the average distance between cluster centroids (using whatever distance metric was used to create the clusters initially).
Interpreting Clusters

Find the mean of each variable.
- This average member typifies the cluster.
- The centroid is particularly valuable for $k$-means clustering.
- The means can be plotted on a parallel dimensions plot.

Sort variables by z-scores.
- The cluster is defined by the variables that most distinguish it from the population.
- Build a decision tree with the segments as the target variable.
  - The SAS Enterprise Miner Segment Analyzer does this, among many other things.

After clusters are found, human interpretation is required to decide whether the clusters are meaningful and how to make use of them.

Comparing Clusters Using a Segment Profiler

The Segment Profiler is a good way to compare different clusters. For each cluster, the Segment Profiler determines which variables are most important for defining the cluster. It then compares the overall distribution of the variable (the gold outline) to the distribution in the cluster (the green). The first and second clusters are very similar in the variables that they choose. However, notice that the first cluster is disproportionate because $\text{household}$ has a value of 1, and the second disproportionately has $\text{household}$ with a value of 0. Notice also that the third cluster is defined by very different variables.
Visuallizing Clusters Using Principal Components

This example from the clustering node shows the centers of clusters on a two-dimensional plot using the first two principal components (actually, standardized principal components). Notice that clusters are not actually circles and might not overlap when all dimensions are included.

Interpreting Clusters Using Centroids

The catalog data set used in this course contains a series of variables dept* that record the number of times a person made a purchase in each of 27 departments. Here, SAS Enterprise Miner was used to cluster customers into four groups based on these variables. This graph shows what proportion of each cluster's total purchases was found in each department. The variable that shows the greatest separation is dept25, which is food. Cluster 2 shoppers have 37.5% of their purchases in that category while cluster 1 customers have only 11% of their purchases in the food category.

Apart from food, looked at this way, these clusters do not show much differentiation. All the ribbons tend to rise and fall together although dept2, women's underwear, separates clusters 2 and 3 from clusters 1 and 4, and cluster 3 has a higher peak at dept 13, kitchen.
Interpreting Clusters Using Average Counts of Purchases in Departments

This graph shows the actual raw count of purchases in each category for the clusters seen on the previous slide. It goes a long way toward explaining why the clusters seemed to behave so similarly in the last graph. The main difference between clusters is not what people buy, but how much. Cluster 3 contains the big spenders. These shoppers bought more than the others in almost every category, but the difference is most pronounced in three departments: food, kitchen, and household. Cluster 2, with the second most purchases, is even more skewed toward food. This suggests trying to attract more foodies to the catalog because they seem to buy more. It also suggests that clustering should not be done on unadjusted counts!
Another Attempt at Clustering

These four clusters were made after passing all of the *dept* variables through the standardization transformation in the SAS Enterprise Miner Transform Variables node. Two of the new clusters look interesting. Cluster 2 has people who buy more bath supplies than anyone else. Cluster 3 has people who buy many items in a number of departments, including household, kitchen, beauty, health, food, and outdoor. Unfortunately, nearly all the customers are in cluster 1. That is the one that hugs the 0 line, meaning that these are people who are close to the mean in every department. The two interesting clusters have only a handful of people each, and that is not worth printing a special bath products catalog.

A Clustering Case Study

In an attempt to compete with local town papers, *The Boston Globe* publishes regional editions with slightly different editorial content.

- Town meeting reports, high school sports, …

There are constraints on the definitions of the regions.

- They must be contiguous so that trucks can drive a sensible route.
- They must align with advertising regions.

Within these constraints, make towns within a zone as similar as possible.
The Boston Globe prints a number of regionalized editions with a few pages of local editorial content. In this study first the towns were clustered based on demographic variables, the distance from Boston, and other factors. Although penetration (the percentage of households subscribing to the Globe) was not one of the clustering variables, penetration does vary widely between clusters.

**k-Means Combined with Divisive Clustering**

Use \( k \)-means to create two clusters. Divide the larger cluster into two clusters, also using \( k \)-means. Repeat this process until you are finished.
Characterizing the Clusters

Although penetration was not one of the clustering variables, the clusters have very different penetration rates.

Cluster 2 (A Good Cluster)

- **Home value 500k+**
  - most of any cluster

- **Home value 250-500k**
  - most of any cluster

- **College degree (35%)**
  - most of any cluster

- **Median years of school (15.66)**
  - most of any cluster

- **% Blue collar occupations**
  - lowest of any cluster
Most Zones Have a Dominant Cluster

<table>
<thead>
<tr>
<th>Town Name</th>
<th>Editorial Sub-Zone</th>
<th>2002 Home Delivery Subscribers</th>
<th>2002 Home Delivery Penetration</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chelsea</td>
<td>North 1</td>
<td>1,028</td>
<td>7%</td>
<td>1-B</td>
</tr>
<tr>
<td>Everett</td>
<td>North 1</td>
<td>2,407</td>
<td>14%</td>
<td>1-B</td>
</tr>
<tr>
<td>Lynn</td>
<td>North 1</td>
<td>3,553</td>
<td>9%</td>
<td>1-B</td>
</tr>
<tr>
<td>Malden</td>
<td>North 1</td>
<td>4,202</td>
<td>16%</td>
<td>1-B</td>
</tr>
<tr>
<td>Melrose</td>
<td>North 1</td>
<td>4,401</td>
<td>38%</td>
<td>1-B</td>
</tr>
<tr>
<td>Revere</td>
<td>North 1</td>
<td>3,683</td>
<td>18%</td>
<td>1-B</td>
</tr>
<tr>
<td>Saugus</td>
<td>North 1</td>
<td>2,913</td>
<td>29%</td>
<td>1-B</td>
</tr>
<tr>
<td>Wakefield</td>
<td>North 1</td>
<td>3,643</td>
<td>36%</td>
<td>1-B</td>
</tr>
<tr>
<td>Winthrop</td>
<td>North 1</td>
<td>2,430</td>
<td>30%</td>
<td>1-B</td>
</tr>
</tbody>
</table>

Clusters and Zones

<table>
<thead>
<tr>
<th>Town Name</th>
<th>Editorial Sub-Zone</th>
<th>2002 Home Delivery Subscribers</th>
<th>2002 Home Delivery Penetration</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brookline</td>
<td>City</td>
<td>7,805</td>
<td>28%</td>
<td>2</td>
</tr>
<tr>
<td>Boston</td>
<td>City</td>
<td>37,273</td>
<td>15%</td>
<td>1-B</td>
</tr>
<tr>
<td>Cambridge</td>
<td>City</td>
<td>9,172</td>
<td>19%</td>
<td>1-B</td>
</tr>
<tr>
<td>Somerville</td>
<td>City</td>
<td>5,395</td>
<td>16%</td>
<td>1-B</td>
</tr>
<tr>
<td>Needham</td>
<td>West 1</td>
<td>4,524</td>
<td>39%</td>
<td>2</td>
</tr>
<tr>
<td>Newton</td>
<td>West 1</td>
<td>18,345</td>
<td>56%</td>
<td>2</td>
</tr>
<tr>
<td>Wellesley</td>
<td>West 1</td>
<td>2,387</td>
<td>26%</td>
<td>2</td>
</tr>
<tr>
<td>Weston</td>
<td>West 1</td>
<td>2,389</td>
<td>56%</td>
<td>2</td>
</tr>
<tr>
<td>Waltham</td>
<td>West 1</td>
<td>5,894</td>
<td>25%</td>
<td>1-B</td>
</tr>
<tr>
<td>Watertown</td>
<td>West 1</td>
<td>4,541</td>
<td>29%</td>
<td>1-B</td>
</tr>
</tbody>
</table>

Data Miners Inc. next looked at the mapping of demographic clusters to the geographic zones. In the City zone, all but one town has similar demographics.

In the zone immediately to the west of the City zone, two towns, both of which border the City zone, fall into the same cluster as the majority of City towns. If there is reason to believe that readers in the same demographic cluster will be interested in similar editorial content, both City and West 1 could be made demographically purer by swapping Waltham and Watertown for Brookline.
Zone Modification
Based on these results, zones were modified.

A Successful Study
We validated the editorial zones designed by the editors, which already tended to be fairly homogeneous in terms of the demographic clusters that were found.
Clustering helped to show what the “good” towns have in common.
There were opportunities to tweak the boundaries of the editorial zones to increase their demographic purity.
Lessons Learned

Clustering is a powerful tool for finding patterns in data. Clustering does not, however, build predictive models to estimate a target.

One of the most common algorithms is \(k\)-means clustering, where the user specifies the number of clusters.

Human judgment is needed to determine whether the clusters are good.

Refer to Exercises for Chapter 8 in the handout.
Chapter 9  Survival Analysis

9.1  Survival Analysis
### 9.1 Survival Analysis

**This Chapter...**

...introduces survival analysis and using SAS statistical functionality in SAS Enterprise Miner using the SAS Code node.

Topics include the following:
- time-to-event problems
- calling SAS procedures from SAS Enterprise Miner
- the difference between survival and retention
- time-zero covariates
- advanced concepts, such as competing risks, proportional hazards, and left truncation

---

### Traditional Approach

A traditional approach to data mining uses predictive modeling.

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Set</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>✗+1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score Set</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>✗+1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data to build the model comes from the past.
- Predictions for some fixed period in the future
- Present when new data is scored
- Use of training, validation, and test sets
- Models built with decision trees, neural networks, logistic regression, and so on

The traditional approach to data mining builds predictive models for a period of time in the future, using data from a model set in the past. A data mining algorithm can determine patterns in the past and apply these to the future. The approach corresponds well to running marketing campaigns, because the campaigns happen during a specific period of time.
Survival Data Mining

Survival data mining adds the element of when things happen.

- Time-to-event analysis
- Terminology from the medical world
  - Estimating how long patients survive based on which patients recover, and which patients do not
- Can measure effects of variables (initial covariates or time-dependent covariates) on survival time
- Natural for understanding customers
- Can be used to quantify marketing efforts

The traditional approach, however, only looks at whether specific events occur rather than when they occur. Survival data mining adds the elements of when things happen. Survival is particularly valuable for gaining insight into customers and for quantifying this insight. An estimate of how long customers will remain is useful for customer value calculations, as well as for direct comparisons of various groups. This estimate can be refined based on initial characteristics of customers, as well as on events that happen during a customer’s lifecycle. This places survival data mining in the heart of customer-centric initiatives.

These two approaches are complementary. It is not that one approach is better than the other. For a specific campaign, the traditional approach usually works better than the survival approach, because the campaign is happening during a particular period of time. For customer insight and for quantifying results over time, survival is often preferable.
### Why Survival Is Useful

You can compare hazards for different groups of customers.

You can calculate the customer half-life.
- How long does it take for half of the customers to leave?

You can calculate truncated mean tenure.
- What is the average customer tenure for the first year after starting? For the first two years?
- This can be used to quantify effects.

Survival curves are useful because they show retention information at a glance. Stratifying groups based on initial start conditions facilitates comparisons. Business people understand the idea of survival, which makes survival curves a useful communication tool.

One way to use survival curves is simply to compare survival at a given point in time, such as at one year. Although this is useful, there are some other measures that you can take from the curves.
How Long Will a Customer Survive?

Survival always starts at 100% and declines over time.
If everyone in the model set stopped, then survival goes to 0; otherwise, it is always greater than 0.
Survival is useful for comparing different groups.

Breaking customers into different groups based on initial conditions is called stratification.

This chart shows a survival chart for two groups of customers, high-end customers and regular customers. It plots survival values on the vertical axis against customer tenure on the horizontal axis. Placing customers into different groups based on initial conditions (such as high-end products versus regular products) is called stratification. It is important to remember that stratification only works well when the grouping variable is defined at the beginning of the customer lifetime.

Survival curves always start at 100% because all customers are alive at the beginning of the time period. They are also monotonically decreasing; that is, survival curves never curve upward. This is because survival does not enable reincarnation. After someone stopped, he is not allowed to return. If everyone in the source data stops, then the survival drops to 0 at the end.

These survival curves are smooth because they are based on several million customers. As shown in an example, survival curves based on smaller samples are more ragged.
The first of these measures is the median tenure or median customer lifetime. This is the point where the curves cross the 50% line. At this point, half the customers survived and half stopped. Because survival curves are monotonic decreasing, there is only one median value, so the median is well-defined. If the curve happens to be flat around the 50% line, then you can choose the middle value as a reasonable median.

As this example shows, the median tenure can show quite dramatic differences for different groups of customers. Half the regular customers are gone in 22 months, whereas the high-end customers stay for more than 6 years. (The difference is about 58 months.) However, the median lifetime suffers the same problems as medians in general. They tend to tell you about the one customer in the middle and do not say anything about the others. For example, all the customers could leave on the day after the median tenure, or they could all stay for another 10 years. Either is possible.
Perhaps a more useful measure is the average tenure (which is also called truncated mean tenure). This is the average tenure of customers over a period of time. In this chart, you see that the average regular customer is expected to stay for approximately 44 months, whereas the average high-end customer is expected to stay for 73 months. The difference of 29 months, shown as the upper area, is less striking than for the median.

However, the average tenure is more useful, because this is an average for all customers. That is, it can be used for customer value calculations in a way that the median cannot be. If customers create $100 in net revenue per year, then the difference between the two groups is $290 in the 10 years after acquisition, or $29/year. When you determine how much money to spend on acquisition, such information is very informative.

How do you calculate the area under the curve? In this case, it is easy. The area under the curve is simply the sum of the value of the survival points. This is apparent if you draw rectangles around each survival point. The vertical length is the survival; the horizontal width is one unit (in this case, months). The area of the rectangle is the survival value at that point times 1. A good estimate of the area under the curve is the sum of all the rectangles, which is the sum of all the survival values.
Calculating the area under a survival curve does not require the use of complex mathematics, such as calculus. Instead, divide the area into rectangles. The sum of the areas of the rectangles is close to the area under the curve (and quite sufficient for your purposes).

What is the area of one rectangle? Each has a base of one unit and the height is the survival value at that point. So, the area is the survival value. The sum of the areas of the rectangles is simply the sum of the survival values.

This implies that the average truncated survival is simply the sum of the survival values up to that point.
Original Statistics

Life table methods were used by actuaries for a long time.

- Referred to in an article in 1693 in *Philosophical Transactions* by Edmund Halley, “An Estimate of the Degrees of the Mortality of Mankind, drawn from curious Tables of the Births and Funerals at the City of Breslaw; with an Attempt to ascertain the Price of Annuities upon Lives”
- Applied with vigor to medicine in the mid-20th century
- Applied with vigor to manufacturing during the 20th century as well
- Took off in 1972 with Sir David Cox's Proportion Hazards Models that provide effects of initial covariates

The methods presented here are life-table methods that were in use by actuaries for several centuries. Perhaps the earliest description of the basic algorithm for calculating life tables is in the January, 1693 edition of *Philosophical Transactions* (www.pierre-marteau.com/c/boehne/halley.html). (Pierre Marteau's Publishing House has a Web site devoted to virtually publishing works from the 17th, 18th, and 19th centuries.) At that time, the challenge was collecting the correct data; Halley’s purpose was to understand, among other things, how many male citizens of Breslaw (now Breslau) were between 18 and 56 (and hence potentially part of a fighting force) and why women of child-bearing age (16-45) only have one child every six years. Halley, incidentally, was also an astronomer who predicted the return of the comet that bears his name.

Times change. However, the methods themselves remain appealing because they enable the data to speak. There are few assumptions and the assumptions are transparent.

In the 20th century, survival analysis started being used extensively for medical research after World War II. The purpose was to learn what treatments worked best or the difference in life span for smokers versus non-smokers. When medical researchers say that one form of treatment works better than others on a deadly disease, survival analysis is probably being used in the background.

The manufacturing world is another area where survival analysis is used, although it is called *failure analysis* (or something similar). When you buy a light bulb with a rating that it will survive for 2000 hours, this is an example of failure analysis. When you buy a disk drive with a MTTF (mean time to failure) of 300,000 hours (over 30 years), the result is determined using survival analysis. In this case, you assume that a particular distribution of failures, the mean time to failure, and the median time to failure have the same value.

In 1972, survival analysis took a step forward. Sir David Cox developed a method for measuring the effect of initial covariates on survival, without having to calculate the survival values themselves. Cox Proportional Hazards, as his method is called, was quickly extended to time-dependent covariates (things that happen during the customer lifetime).

It is said that his original paper is one of the 10 most cited papers of all time. This is not surprising, because almost all medical research studies use proportional hazards or some variant.
Survival for Marketing

Survival for marketing has some differences.

- Discrete time (probability versus rate)
  - Traditional survival analysis uses continuous time.

- Hundreds of thousands or millions of examples
  - Traditional survival analysis might be done on dozens or hundreds of participants.

- A wealth of data
  - Traditional survival analysis looks at factors incorporated into the study.

- Window effects due to business practices and database reality
  - Traditional survival ignores left truncation.

Perhaps because the connotations of survival analysis are so dire (death or failure), these techniques were slow to be accepted in the marketing world. However, there are some other reasons.

Medical studies generally involve few participants (in the dozens or hundreds) rather than the millions that you see in customer records. For this reason, much of the analysis focused on getting the last iota of information from a limited set of examples. The problems that you face are how to get any information from millions of rows. Survival analysis provides a very useful set of measures for looking at a customer base; at one level, they summarize the information into a comprehensible form.

Another issue is time. Many of the original methods assumed that time was continuous and calculated hazard rates. Although the results are quite similar to using hazard probabilities, rates are a more difficult concept to apply to the business world. In addition, the methods sometimes assumed that there were no ties in the data; that is, that no two patients died at the same point. With the large volume of records common in the business world, this assumption is not realistic.

Because you work with large volumes of data, traditional statistical concerns, such as confidence bounds, do not apply. The bounds are very narrow because the sample is so large. Of course, there might be some cases where such concerns are important, but they are the exception rather than the rule.

Another way that survival in the marketing world differs from traditional survival analysis is left truncation, which occurs when stops are not available before a certain date. For instance, a data warehouse might be loaded with data about active customers. However, customers who stopped in the past might not be included. A similar situation occurs when two companies merge. Often, the resulting database systems include only active customers, including their start dates. Customers who were not active at the time of the merge might be discarded. This action results in a bias, called left truncation, because stops on a time line before the truncation date are discarded (truncated in statistical terminology).

Paul Allison’s classic book, Survival Analysis Using the SAS® System: A Practical Guide (SAS Publishing, 1995), is an excellent resource on survival analysis for the medical world. However, on page 5 he dismisses the importance of left truncation, which is one of the most important considerations in the business world.
Empirical Hazard Probability Definition

Hazard, $h(t)$, at tenure $t$ is the probability that a customer who survived to tenure $t$ will not survive to tenure $t+1$,

$$h(t) = \frac{\# \text{ customers who stop at exactly tenure } t}{\# \text{ customers at risk of stopping at tenure } t}$$

(or that a one-time event that did not yet occur at time $t$ will occur before time $t+1$).

The value of a hazard depends on units of time: days, weeks, months, or years.

Hazard in this context differs from the traditional definition because time is discrete (hazard probability, not hazard rate).

Hazard is a probability between 0% and 100%, but never negative!

The challenge is creating an unbiased estimate.

The survival calculation starts with hazards. This is the probability at any point in time that a customer will stop between time $t$ and before time $t+1$. The hazard is a probability, so it always has a value between 0 and 1. The specific value depends on the units of time being used.

When you use all the data, the hazard probability is the number of customers who stopped with a particular tenure divided by the number of customers with that tenure or larger. This makes the calculation particularly easy to understand and visualize.

One assumption about this calculation is that the hazard rate is stable over time. One of the big issues with customers is how the hazards change. As an example, though, consider a company that stops customers who never pay at 50 days. The hazard at 50 days will be high, assuming that this group is somewhat sizeable. If the company changes its policy to stop these customers at 40 days, the hazard at 50 days changed. The assumption that the hazards are stable is not true, because now the peak will be at 40 days.
This timeline shows customers who start and stop at different times. Three of the customers stopped in the past. Two stopped today, and three are still active. You know the tenure of the stopped customers, because you have both a beginning and an end date. For the active customers, you do not know if they will stop tomorrow or if they will continue for the next 10 years. That is, you only have the lower bound on their tenure. You can say that their tenure is *censored*.

When you look at the most recent data, the censor date is today (or the most recent update date). For some purposes, such as testing, it is sometimes desirable to set the censor date at a point in the past. When you do this, you must be sure to exclude all future information. For instance, if the censor date were set at the first bar, it changes the data. Ann’s tenure is shorter. Bob, Cora, Emma, Fred, and Hal are not customers because they did not start. Gus is still active with a shorter tenure. The only customer who remains the same is Diane.
The hazard calculation switches timelines from calendar time to tenure time. On this scale, all customers start at time 0 and continue until they are either stopped or censored. Notice that the active customers are censored at different points.

From this perspective, you can gather the two pieces of information needed for the hazard calculation: the customer tenure and whether the customer stopped.
In practice, it is cumbersome (or impossible) to expand each customer for each day.

The calculation itself is easiest to understand if you show what happens at each tenure time unit. For instance, the top line in this chart shows that Ann is active for all tenure times. The second line shows that Bob stopped at Tenure 6. The gray boxes indicate that Bob is not included in the calculation for those times.

For each time unit, you count how many customers are active, how many are stopped, and the total number of customers. The population at risk is the total. The population who succumbed to the risk is the stopped group. Their ratio is the hazard at that time.

In practice, the calculation of the hazard rates does not require creating a matrix of all time units by all customers. Even on today’s computers, this would not be feasible for large numbers of customers. However, this method of looking at the calculation does provide insight into how the calculation proceeds and helps in understanding what is happening.

To calculate the hazards, you cannot directly estimate the hazards for tenures greater than any tenure seen in the data. In order to extrapolate hazards to longer tenures, it is necessary to make assumptions about hazards in the future, perhaps that the hazards are constant or that they increase or decrease at some fixed rate.
Infant mortality rate, about 7 per 1000

Hazard is bathtub-shaped, because it starts high, goes down, and then rises again.
This example is from a less well-known domain, that is, customers with a subscription to a particular service. The hazards are shown on a daily basis.

The first feature that you notice is a high initial hazard at time zero. The reason for this is that some customers never start or stop immediately after they start. In some cases, you might not want to include these customers. However, at this company, the rate of never-starts varies by the acquisition channel, so it is an interesting measure.

Between 60 and 120 days, two peaks are close together. The first peak is for non-payment. Customers who never pay are sent escalating collection letters as their dunning levels increase. (The dunning level is a term used in billing systems to measure the lateness of non-paying customers.) This is an example of a hazard caused by a specific business policy.

The second of these peaks is due to the end of the initial promotion. Most customers are enticed with a promotion for a period of time. Customers who are not very committed take the promotion but never intend to pay full price.

As a note, these hazards are based on the customer’s commitment date, that is, the first day that he incurs charges. Because of this, the peaks for non-payment and the promotion ending are clear. If you used another day, such as the customer contact date, the peaks might not be so well-defined.

After these peaks, the hazards have a general trend with some bumpiness. Part of the bumpiness is due to within-week variation. Also, there are peaks that correspond to the typical monthly billing cycle. Customers are more likely to stop when they receive their bills.

The general trend in the hazards is a gradual decrease over time. What this means is that when customers stay with a company longer, they are less likely to stop. This is as good a measure of loyalty.
This is another example of hazards from another business. In this business, customers tend to sign up for year-long contracts, and they are encouraged to stay until their contract date by the threat of a fee.

These hazards show features for initial non-payment, for stopping after one year, after two years, and for non-payment after one and two years. It is interesting that the non-payment in the second year is the biggest spike. This is interesting because these customers presumably paid for their first year.

In this case, you also notice a decline in hazards in the several months before the end of each year. These hazards were calculated using the stop-commitment date (rather than the stop-request date). What might be happening, in practice, is that customers are calling to stop several months before the end of their service years. However, they are persuaded to stay until their service dates.

Understanding this scenario is important. It means that intervention campaigns that are targeted in the 10th and 11th months of service might be directed to customers who already made a commitment of stopping. These campaigns might be more effective in months 7 and 8.
9.1 Survival Analysis

Survival Calculation

What proportion of customers survive to time \( t \)?

- Survival at time \( t \), \( S(t) \), is the probability that a customer will survive at least to time \( t \).

Calculation:

- \( S(t) = S(t-1) \times (1 - h(t-1)) \)
- \( S(0) = 100\% \)

\[
S(T) = \prod_{t=1}^{T} (1 - h(t-1))
\]

\[
S(0) = 1
\]

The hazard calculation starts the story; the survival calculation completes it. Survival is a cumulative measure that a customer does not succumb to the risk of stopping. This is expressed as the result of one minus the hazard probabilities.

If you consider the way that you calculated hazard probabilities, all customers have a risk of stopping at \( t=0 \). All survival values start accumulating from \( h(0) \). For this reason, every survival value uses information from all the customers. In particular, survival uses the information about whether customers did or did not stop at a particular point in time.
You can see this calculation in the same table shown earlier.

Notice that at Tenure 3, the hazard is 12.5%, but the survival is still 100%. This is due to the definition of the hazard. Diane, the customer who stopped at Tenure 3, stopped between Tenure 3 and before Tenure 4. For this reason, she is considered active at Tenure 3 and the stop takes effect at Tenure 4.

Retention

Another way to answer the question “How long will a customer survive?” is to do a retention calculation.

- Of customers who started 1 time unit ago, how many are still active?
- Of customers who started 2 time units ago, how many are still active?
- ...

Retention is another measure related to survival. One common way of measuring retention is to look at the customers who started at a particular time in the past, for example, 20 weeks ago, and count the number that are currently active, for example, 65%. This is the retention value for 20 weeks.

Retention is easy to calculate and it is readily understandable.
This is a picture of a retention curve. Notice that the curve is jagged. In addition, the curve is not monotonic. Sometimes it goes up; sometimes it goes down. What does it mean if the retention at 21 weeks is higher than the retention at 20 weeks? This is counterintuitive. You expect that the number of 21-weekers who stopped would include all those who stopped at 20 weeks plus an additional number.

One reason could be due to sampling error. However, there are other reasons that shed more light on what might be happening. For instance, the mix of customers being acquired might change between the two weeks. Consider a telemarketing center that has a “good customer” score. Perhaps the best of the good customers were called 21 weeks ago. These are customers you expect to retain better. A week later (20 weeks ago), fewer good customers were called and these customers have a higher rate of stopping.

Another factor that might vary over time is the number of customers being acquired. Maybe new-customer starts decreased almost to zero in one of the weeks. In this case, sampling variation can play an important factor in the variability of retention.
Retention only uses customers from a narrow period of time to calculate any data point.
Survival uses as much information as possible from all customers to calculate each data point.

This chart shows two curves, one for retention and one for survival, based on the same data. The survival curve almost resembles a moving average of the retention, although it is not.

A good way to look at the comparison is to define what the two curves mean. The survival curve provides a benchmark of customer churn behavior in this data. As mentioned earlier, each point on the survival curve incorporates information from the entire data set.

The retention curve, on the other hand, provides the most recent information about each point. After all, what is a more recent measure of 50-week retention than customers who started 50 weeks ago?

With this perspective, you can now interpret the two curves. During the first few weeks, retention is higher than survival. This means that new customers in the past few weeks performed better than customers as a whole during the first few weeks of tenure. Customer acquisition brings in better quality. Much of the difference occurs during the first week. Perhaps the difference is a sharp reduction in initial “buyer's remorse.”

During other periods, retention is lower than survival. This implies that customers during those weeks were worse than customers overall. This is an example where the retention curve and the survival curve are similar. The various differences are small and could be due only to random variations.
Survival can be applied to many different types of events. This example shows survival and hazard values for reactivations. Here, the beginning of the time is when a customer stops, and survival means that the stop continues to be in force. The upper curve measures the proportion that reactivates against the scale on the left. The hazard for reactivation is shown on the lower curve against the scale on the right. The curve shows that most reactivations occur within the first ninety days. This is not surprising. This example comes from a wireless phone carrier. They reserve a customer’s telephone number for 90 days after the customer stops. After 90 days, the number can be assigned to another newer customer.

This curve is based on the telephone number, so you do not have to worry about matching newly started customers to previous customers. In businesses that collect a good identifier, such as the Social Security number, the matching does not pose a major problem. Other businesses, however, do need to be concerned with name and address matching in order to understand reactivation rates. Fortunately, householding algorithms are available to make the connection between stopped accounts and newly started ones.
Time to Next Order

In retail businesses, customers make multiple purchases. Survival analysis can be applied here, too. The question is how long until the next purchase.

- The initial state is the date of purchase.
- The final state is the date of next purchase.

It is better to look at 1 – survival rather than survival (the cumulative probability that a purchase does happen).

Another example for the application of survival is the time to next purchase for retail events. In this case, the start is when a customer makes a purchase. The stop is when (if ever) he makes another purchase.

Because survival is measuring the likelihood of a customer not making a purchase, it makes more sense to look at the inverse, that is, the chance that a customer will make a purchase.

This chart shows time to next purchase for all customers. In this case, the event is having a purchase, so survival is really the survival of the not-purchase state. Instead of survival, 1 – Survival is more interesting.
This chart shows the time to next purchase, stratified by the number of purchases in the past. The heavy gray line is the overall average for any purchase.

**Doing This in SAS**

For this, you need to add a SAS Code node and use the LIFETEST procedure.

SAS has good support for survival analysis. However, to use these functions in SAS Enterprise Miner, you need to use the SAS Code node.

**Demo Available: Creating Survival Curves**
Advanced Topics

- Proportional Hazards
- Competing Risks
- Left Truncation
- Forecasting
When looking at survival calculations, you are often interested in the effect of other variables besides tenure on the survival calculation. These variables fall into two categories: time-zero covariates and time-dependent covariates.

- Time-zero covariates are known at the beginning of the customer relationship. These are things such as the day of the week of the start, the acquisition channel, the geography, the initial product, and so on.
- Other covariates are those that occur during the customer lifetime, such as whether he signed up for a loyalty offer or whether he was late paying a bill. These are time-dependent covariates.

The most straightforward method is stratification. If there is a large amount of data and few covariate values, divide the data into discrete sets and directly measure the survival of each. This approach fails to work as the number of covariate values increases. Unfortunately, stratification does not work easily on time-dependent covariates.

The instructor described a non-parametric measure for calculating the hazard probabilities. There are other parametric methods where initial covariates can be included in the model. One of these, PROC LIFEREG, estimates the hazards based on their following particular functional forms. This procedure enables you to include initial covariates in the modeling process. However, the estimate of the hazards does not include much of the detail that is so useful for understanding customers.

Another non-parametric method uses splines to estimate the hazards. Anyone interested in splines should take the Business Knowledge Series course *Survival Data Mining: Predictive Hazard Modeling for Customer History Data*.

Finally, a technical method, proportional hazards, measures the effect of covariates. This is called *proportional hazards regression.*
Proportional Hazards Regression

- Developed by Professor David Cox in 1972 (now Sir David Cox)
- Reputedly, one of the 10 most cited scientific papers, because proportional hazards is very important for research purposes
- Ingenious and insightful
- Semi-parametric, that is, it determines the effects of covariates without first calculating the survival
- Provides size and level of significance for different factors

To understand how it works, return to a small set of customers.

What is the probability that Diane stops at Tenure 3?
**Personal Hazard Function**

Assume that Diane has her own personal hazard probability function, $h_{\text{diane}}(t)$.

What is her probability of stopping at exactly Tenure 3? She did not stop at Tenures 0, 1, or 2:

$$(1-h_{\text{diane}}(0))*(1-h_{\text{diane}}(1))*(1-h_{\text{diane}}(2))*h_{\text{diane}}(3)$$

Diane does, sort of, have her own personal hazard probability function. It is a function of her covariates.
**Proportional Hazards Regression**

Proportional hazards regression uses this information.

The likelihood of Diane exactly stopping at Tenure 3:

\[(1-h_{ann}(3)) \times (1-h_{bob}(3)) \times (1-h_{cora}(3)) \times h_{diane}(3) \ldots\]

What you want to know is which covariates influence Diane to stop and the others not to stop.

```plaintext
proc phreg data=&EM_IMPORT_DATA;
    model tenure*censored(1)=
        isms ismg ismm /selection=stepwise;
    isms=market='Smallville';
    ismg=market='Gotham';
    ismm=market='Metropolis';
run;
```
Stratification Is an Alternative Approach

Survival for the first year:
- Top: 93.4%
- Bottom: 85.6%
- Increase for Top: 7.8%

Average improvement in a lifetime for the first year is 3.9% (assuming that 7.8% would stay, on average, six months).

Assuming a revenue of $400/year, each customer in the top group contributes an additional revenue of $15.60 during the first year.

Cox proportional hazards has many advantages, because it works for both time-dependent and time-zero covariates and it measures the effects of the covariates on survival. However, it does not give a specific numeric value for quantifying results.

For this, stratification is often the best approach for time-zero covariates. This example shows two groups of customers and their survival for the first year. In the first group, 93.4% of the group survives; in the second, 85.6%.

Remember that the area under the curves gives the average tenure for the first year. You can extend this to say that the area between the curves measures the difference in average tenure during this period of time. In this case, approximate the area as a triangle. You can perform the calculation to say that an average customer in the top group contributes an additional 3.9% “customer-years” during the first year as compared to the second group. If these customers raise $400 per year, then this corresponds to additional revenue of $15.60 during the first year. Of course, these customers contribute even more after that, because more survive the first year.
Competing Risks

Customers can leave for many reasons.
- Customers might cancel.
  - Voluntarily
  - Involuntarily
  - Migration
- Survival analysis can show competing risks.
  - Overall \( S(t) \) is the product of \( S_r(t) \) for all risks.
- Consider an example.
  - Overall survival
  - Competing risks survival
  - Competing risks hazards
  - Stratified competing risks survival

So far, you considered that all stops are equivalent. In practice, customers stop for different reasons. Two distinct reasons are voluntary and involuntary churn, that is, the difference between a customer deciding to stop and a customer who is stopped for lack of payment. There are other situations as well. Customers might upgrade to a different product, or be sold to a competitor.

In the language of survival analysis, these are competing risks. A good way to think of them is the risks are competing for the attention of every active customer. Any active customer might succumb to any of these risks at any given time.

Conditional Hazards - Conditional Survival

Conditional hazards lead to conditional survival.

What is the probability of not succumbing to one particular risk at tenure \( t \)?

Notice that this is different from survival, because it is a conditional probability. Conditional survival is always larger (or as large as) overall survival.
This chart shows the customers that you saw several times before. Now, though, the stopped customers are either red for involuntary churn or green for voluntary churn.

Competing risks calculates hazard values and survival values for each risk. When you look at one risk, customers who succumb to other risks are censored at the point when they stop. This introduces the fact that censored customers are not necessarily active; maybe they stopped for other reasons.

The tenure timeline provides more information about what is happening. Now, in addition to recording whether customers stop, you also record the reason for the stop. This will impact later calculations.
Consider Diane who stopped at Tenure 3 for involuntary churn. When calculating the risk for voluntary churn, you use as much information as you can. She did not churn voluntarily at Tenure 1, at Tenure 2, nor at Tenure 3. However, after that, you do not know what happened. The information is censored, without her stop ever counting.

There are some things to notice about survival for a given competing risk. The value is always greater than the survival for all the risks, because stops caused by other risks are not counted in the calculation.

The meaning of the competing risk survival value might at first seem counter-intuitive. The value of 41.7% at Time 12 does not mean that 59.3% of the customers stopped at this point. Instead, it means that given that a customer does not stop for any other reason, the probability of stopping voluntarily is 59.3%. This is a conditional probability and you must remember the conditional part.

---

**Calculation for Voluntary Churn**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ann</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td><strong>Bob</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cora</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Diane</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Emma</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Fred</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gus</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hal</strong></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ACTIVE</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>STOPPED</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Hazard</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Survival</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compute the hazard by counting only the voluntary stops. Customers who stop by other means are at risk until they stop. Then they are censored.
This is an example of the overall survival for a customer population. In this survival curve, there is a sharp drop around 60 days. The next slides use competing risks and stratification to elucidate what is happening.
This slide shows the competing risks for voluntary and involuntary churn. The steep drop around 60 days is associated entirely with involuntary churn; the non-payment period has little effect on voluntary churn.

The shapes of the curves are interesting. The involuntary churn curve drops steeply during the first few months. However, after that period, the risk of involuntary churn levels off almost to nothing. On the other hand, voluntary churn is important throughout the customer lifecycle.

The overall churn follows the shape of the dominant factor. Initially, it resembles the voluntary churn curve. After a few weeks, it starts to match the involuntary curve. In the long term, it follows voluntary churn. The overall churn is the product of the two other curves.
What the Hazards Look Like

As expected, a steep drop has a very high hazard.

The hazards follow the same patterns as shown before. However, the hazards more clearly show the large number of customers who leave for involuntary churn reasons.

The hazards for voluntary churn show a declining pattern in the long term. This pattern has two cyclical subpatterns. The tall peaks that occur about once every month are due to customers stopping when their bills arrive. The other peaks are weekly cycles, because customers are more likely to stop on certain days of the week. As mentioned earlier, the longer term decline in hazards for voluntary churn is a good measure of the loyalty of the customers.
This chart shows four survival curves. Each of the competing risk curves is broken into two pieces, depending on the payment method. In this case, credit card payers are better than non-credit card payers, because the thin lines have the best survival. The top line shows that credit card payers have a very low rate of non-payment, which makes sense because bills are paid automatically. For payment to be denied, the credit card must be denied or a customer must dispute the charge.

Overall, this chart shows that the steep drop in involuntary churn is associated with non-credit card payers. This has consequences for customer acquisition and customer care. Converting customers to credit card payment would probably improve retention. It would have a dramatic effect on involuntary churn; it would also affect voluntary churn.
This chart shows what happens after a group of customers are expected to start. The green part represents customers who remain active, the red is for customers who stopped involuntarily, and the blue is for customers who stopped voluntarily. You can see that, at the end of the diagram, about 60% of customers are still active, about 15% stopped involuntarily, and about 25% stopped voluntarily.
Left truncation is a property of the data set. It occurs when stopped customers are not stored in the data before a particular point in time. Left truncation is a problem because

- companies load data warehouses but do not keep historical data
- different customer groups (regions, market segments, and so on) might have different truncation dates
- nobody tells you that this is happening
- it does not occur in well-designed studies.

Left truncation is a problem that is faced in corporate data that does not exist in well-defined studies. Companies lose historical data. Sometimes, a company never had the data. For instance, when merging new customers from another company, an organization might only keep active customers. Previous customers are not of interest. The same thing can happen during an initial load of a data warehouse.

The real problem with left truncation is that no one tells you that it is happening, and yet it can have a significant impact on the hazards.
Left Truncation on the Calendar Timeline

On the calendar timeline, you see the effect of left truncation. For customers who stopped during the truncation period, you have no information. For customers who started during the truncation period, you usually have their start dates, although this is not always true. You saw databases where the start information is lost even for active customers.
On the tenure timeline, left truncation is much less obvious. What happens is that customers are missing, but you do not know that they were supposed to be there in the first place, so there is no reason to suspect missing data.

This has a direct effect on the hazards. Now, at Tenure 3, you do not have any customers who stopped, so the hazard is 0% instead of 12.5%. The left truncation introduced a bias into the hazard calculation.
This histogram shows the number of starts and stops by month. You see that the starts go back at least to 1996 (actually to 1988). However, the stops abruptly start in January 2004. The likely explanation for this is that some database was moved and only active customers were kept in the database. That is, there is a systematic reason why the stops are not present, and this systematic reason likely biases the hazards.
Fixing Left Truncation

Filter the data so that only records that started after the truncation date are used for the survival calculation.

- This is easy. However, if the truncation date is recent, then survival will not extend far into the future.
- The WHERE statement in SAS and SQL is an easy way to do this.

More sophisticated approaches are available when you use the PHREG procedure or custom DATA step code.

```sas
proc lifetest data=&EM_IMPORT_DATA
   method=act intervals=(0 to 1000 by 1)
   plots=(h,s) outsurv=&EM_EXPORT_SCORE;
   where start_date >= '01jan2004'd;
   time tenure*censored(1);
```

One way to fix left truncation is by filtering the data. Simply remove all the customers that started before the truncation date. This method has the virtue of simplicity. However, it has the consequence that survival will not extend far into the future, because customers with long tenures are removed.

In some cases, the effect of removal might change over time. One company that the authors worked with loaded a data warehouse in May 1998. At this time, the data warehouse consisted primarily of active customers. In 2000, there was no extensive history available for survival calculations; there was only a year or two of good data. However, the company continued loading the data warehouse, and now there are years and years of customer data available for such work.

Another way to fix the left truncation problem is by using time windows on the stops. The trick is to get unbiased estimates of the hazards during the period.

Using stops from a time window has several advantages. First, it fixes the problem with left truncation, as long as the time window is after the left truncation date.

Second, you can use the time window to filter out policies and changes in business that might affect the hazard rates. For instance, earlier there was discussion about a wireless phone company that changed its stop policy from “as soon as possible” to “next bill date.” This has a big impact on the hazards. One way to address it was to use stops only after the policy went into effect. In a company with millions of customers, where hundreds of thousands stop every month, there was a sufficient amount of data in a short period of time.

Another possibility is to use different time windows to measure what is happening to the hazards themselves. In a medical study, you can assume that the risk of people dying is not changing over time due to treatment. However, in the business environment, there are competitive effects and other factors that might influence customers to leave. One way to looking at this is by measuring hazards during different time windows to determine what is happening.
Left-truncated data tends to underestimate hazards, as shown in this chart. The upper line is a good estimate of the hazards, by filtering out all starts from before the left-truncation date. The lower line is the estimate with all the data, so it does not take into account an important bias in the data. The result is that the hazards are biased.

This bias is explained as follows. Consider the 30-day hazard, which is the ratio of the number of customers stopped at exactly 30 days to those whose tenure is greater than or equal to 30. For both calculations, the number who stopped is the same. However, the denominator is larger when including all customers, because any customer with a tenure greater than 30 days who started before the left-truncation date is also included. These customers, though, did not have the opportunity to stop at 30 days, or more precisely, the ones who stopped at 30 days are not included in the data set.
If the hazards are underestimated, then the survival is overestimated. This can have a large impact when you use survival to quantify customer behavior.

Survival provides a proportion of how many customers will stop. Often, what is more interesting is the number of customers who will stop or conversely, the number who are going to remain active.

Survival analysis provides the basics to do such forecasting at the customer level. This has the advantage that the forecast can be sliced and diced by customer attributes; product, channel, and geography, for instance. More traditional methods of forecasting rely on aggregations and are difficult to slice and dice.
This chart resembles the chart on a previous slide, except that now you interpret the horizontal axis as days in the future. Assume a group of customers starts today. Then this shows the proportion that is active and stopped at a certain number of days in the future. This is the basis for a forecast.

The area under the survival curve measures the accumulated customer days. This chart can be used to estimate the number of customer days expected for the group in the first year after they start. It can also show the amount of potential revenue lost to stops during this time.

**Forecast Groups**

- People who start today
- People who are already active
- People who will start in the future
The current population consists of customers with different tenures. Developing a forecast is simply a matter of extending the customers' counts into the future. The first hazard, for Tenure 0, is the risk that a customer will stop before Time 1. You can apply this hazard to the number of customers with a tenure of 0 to get the expected number of customers who stop in this group. Repeating this for all the hazards yields an estimate of all the customers who will stop in the first time unit.

This process can be repeated to estimate the number of active customers and the number of stopped customers at any point in the future. This example limits the customers to 10 tenures and forecasts three time periods in the future. However, the process can be repeated as long as there are hazards for the calculation.
Apply the Hazards to New Starts

<table>
<thead>
<tr>
<th>TENURE</th>
<th>HAZARD</th>
<th>POP</th>
<th>STOP</th>
<th>POP</th>
<th>STOP</th>
<th>POP</th>
<th>STOP</th>
<th>POP</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.017%</td>
<td>1,000.0</td>
<td>0.17</td>
<td>100.0</td>
<td>0.02</td>
<td>100.0</td>
<td>0.02</td>
<td>100.0</td>
<td>0.02</td>
</tr>
<tr>
<td>1</td>
<td>0.567%</td>
<td>1,000.0</td>
<td>5.67</td>
<td>999.8</td>
<td>5.67</td>
<td>100.0</td>
<td>0.57</td>
<td>100.0</td>
<td>0.57</td>
</tr>
<tr>
<td>2</td>
<td>0.498%</td>
<td>1,000.0</td>
<td>4.98</td>
<td>994.3</td>
<td>4.95</td>
<td>994.2</td>
<td>4.95</td>
<td>994.4</td>
<td>5.00</td>
</tr>
<tr>
<td>3</td>
<td>0.376%</td>
<td>1,000.0</td>
<td>3.76</td>
<td>995.0</td>
<td>3.74</td>
<td>989.4</td>
<td>3.72</td>
<td>989.2</td>
<td>3.72</td>
</tr>
<tr>
<td>4</td>
<td>0.316%</td>
<td>1,000.0</td>
<td>3.16</td>
<td>996.2</td>
<td>3.15</td>
<td>991.3</td>
<td>3.13</td>
<td>985.7</td>
<td>3.11</td>
</tr>
<tr>
<td>5</td>
<td>0.306%</td>
<td>1,000.0</td>
<td>3.06</td>
<td>996.8</td>
<td>3.05</td>
<td>993.1</td>
<td>3.04</td>
<td>988.1</td>
<td>3.02</td>
</tr>
<tr>
<td>6</td>
<td>0.415%</td>
<td>1,000.0</td>
<td>4.15</td>
<td>996.9</td>
<td>4.14</td>
<td>993.8</td>
<td>4.12</td>
<td>990.1</td>
<td>4.11</td>
</tr>
<tr>
<td>7</td>
<td>0.347%</td>
<td>1,000.0</td>
<td>3.47</td>
<td>995.9</td>
<td>3.46</td>
<td>992.8</td>
<td>3.45</td>
<td>988.7</td>
<td>3.43</td>
</tr>
<tr>
<td>8</td>
<td>0.323%</td>
<td>1,000.0</td>
<td>3.23</td>
<td>996.5</td>
<td>3.22</td>
<td>992.4</td>
<td>3.21</td>
<td>989.4</td>
<td>3.20</td>
</tr>
<tr>
<td>9</td>
<td>0.321%</td>
<td>1,000.0</td>
<td>3.21</td>
<td>996.8</td>
<td>3.20</td>
<td>993.3</td>
<td>3.19</td>
<td>989.2</td>
<td>3.18</td>
</tr>
<tr>
<td>10</td>
<td>0.399%</td>
<td>996.8</td>
<td>3.08</td>
<td>993.6</td>
<td>3.07</td>
<td>990.1</td>
<td>3.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.285%</td>
<td></td>
<td></td>
<td>993.7</td>
<td>2.83</td>
<td>990.5</td>
<td>2.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.291%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>990.9</td>
<td>2.88</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOTAL NEW

<table>
<thead>
<tr>
<th></th>
<th>t=0</th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td>100.0</td>
<td>200.0</td>
<td>299.4</td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td>100.0</td>
<td>300.0</td>
<td>399.4</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>100.0</td>
<td>400.0</td>
<td>499.4</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>100.0</td>
<td>500.0</td>
<td>599.4</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>100.0</td>
<td>600.0</td>
<td>699.4</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>100.0</td>
<td>700.0</td>
<td>799.4</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>100.0</td>
<td>800.0</td>
<td>899.4</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>100.0</td>
<td>900.0</td>
<td>999.4</td>
</tr>
<tr>
<td>8</td>
<td>0.00</td>
<td>100.0</td>
<td>1,000</td>
<td>1,099.4</td>
</tr>
<tr>
<td>9</td>
<td>0.00</td>
<td>100.0</td>
<td>1,100</td>
<td>1,199.4</td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>100.0</td>
<td>1,200</td>
<td>1,299.4</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>100.0</td>
<td>1,300</td>
<td>1,399.4</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>100.0</td>
<td>1,400</td>
<td>1,499.4</td>
</tr>
</tbody>
</table>

TOTAL EXISTING

<table>
<thead>
<tr>
<th></th>
<th>t=0</th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000.0</td>
<td>34.86</td>
<td>9,968.1</td>
<td>37.65</td>
<td>9,927.5</td>
</tr>
<tr>
<td>10,000.0</td>
<td>34.86</td>
<td>10,068.1</td>
<td>37.67</td>
<td>10,127.5</td>
</tr>
</tbody>
</table>

TOTAL STOP RATE

<table>
<thead>
<tr>
<th></th>
<th>t=0</th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35%</td>
<td>0.37%</td>
<td>0.35%</td>
<td>0.33%</td>
<td></td>
</tr>
</tbody>
</table>

The same ideas can be applied for new starts. Here you assume that 100 customers start on each day in the future. You can track these customers as well.

This chart graphically shows what happens when 100 customers start on the first day of the forecast and 50 more start half a year later. Each group follows the survival curve. The overall survival numbers are the sum of the two curves. This is a graphic depiction of the table on the previous slide.
This picture shows an example of a forecast made at a particular client for the number of customer stops during a one-month period. In fact, this is the result from the first time that you used survival for developing a forecast.

The blue lines represent the predicted forecast. After the forecast is developed, you can compare it to the actual values and it compares favorably. The forecast manages to capture the intra-week variability, where some days have a higher number of stops than others. Perhaps more importantly, the forecast does a good job of capturing the average value of stops, so the total for the month should be on target.

There is a big difference on the 22\textsuperscript{nd} of the month. The forecast value is considerably higher than the actual value. The month was November, and this represents Thanksgiving. Because the forecast did not know about holidays, an otherwise impressive forecast is skewed on this day.
The forecast described earlier provides a forecast for the existing base of subscribers. This is only one component of a forecasting system.

The most challenging part of a forecast is determining what happens to customers who start in the future. This requires a new start forecast, which is generally provided as part of the business. The new start forecast is usually divided along various dimensions, such as by region and product. Another key component is comparing the forecasted value to actuals. This is important both for determining whether the forecast is working and also for understanding the performance of the business. The forecast provides an expected value for what should happen and deviations that are worth further investigation.

**Lessons Learned**

Survival analysis is used for understanding time-to-event problems.

Customers have many time-to-event problems, such as customer tenure or time-to-next purchase.

Understanding the why of customer tenure can be important. Techniques such as stratification and proportional hazards regression are quite powerful.

A major use of survival analysis is for customer-centric forecasting.
Follow-On Classes

Applying Survival Analysis to Business Time-to-Event Problems (support.sas.com/edu/schedules.html?id=44)
- Two days
- Level 3

Survival Analysis Using the Proportional Hazards Model (support.sas.com/edu/schedules.html?id=342)
- Two days
- Level 3

Survival Data Mining: Predictive Hazard Modeling for Customer History Data (support.sas.com/edu/schedules.html?id=50)
- Three days
- Level 4

Refer to Exercises for Chapter 9 in the handout.
Chapter 10  Association Rules

10.1  Market Basket Analysis and Association Rules ........................................................ 10-3
10.1 Market Basket Analysis and Association Rules

This Chapter...

…introduces three closely-related ideas:
- market basket analysis
- beer and diapers
- association rules

*Market basket analysis* is the study of who buys what under what circumstances. Association rules are often a big part of market basket analysis. *Association rules* describe which items tend to appear together in the same order.

Sequential rules are a variation on association rules. *Sequential rules* describe the order in which items are purchased.
Market Basket Analysis

*Market basket analysis* is a general term for analyzing the items that customers purchase. *Association rules* specify which items are likely to occur together. *Sequential pattern analysis* looks at the order in which items occur as well.

Association rules are a data-mining technique that determines which things are found together. As a data-mining technique, they focus almost exclusively on categorical data, rather than on numerical data. This gives it a different flavor from other data-mining techniques. The goal of association rules might be the set of related things (associations) or it might be a set of rules derived from the associations.

Association rules can be used for many things. They are a good technique for exploring item-based data to determine which things co-occur. They can also be used in the opposite sense. Instead of looking for the most common rules, look for relatively uncommon rules because these can be indicative of poor data quality.

Market Basket Analysis: Myth and Legend
The Real Story

So what are the facts?

In 1992, Thomas Blischok, manager of a retail consulting group at Teradata, and his staff prepared an analysis of 1.2 million market baskets from about 25 Osco Drug stores. Database queries were developed to identify affinities. The analysis "did discover that between 5:00 and 7:00 p.m. that consumers bought beer and diapers." Osco managers did not exploit the beer-and-diapers relationship by moving the products closer together on the shelves.

This decision support study was conducted using query tools to find an association. The true story is very bland compared to the legend.

Daniel Power at dssresources.com/newsletters/66.php

For some reason, the idea of an association between beer and diapers caught the fancy of marketing people. As the story was told and retold, it gained more embellishments. In one variation, the store manager moved the higher-margin store brand diapers next to the beer but kept the national brands in their original location. The explanation was that when mothers went shopping, they were loyal to a particular brand, but the fathers chose whatever was nearby.

Questions in a Shopping Cart

In this shopping basket, the shopper purchased a quart of orange juice, some bananas, dish detergent, some window cleaner, and a six-pack of soda.

Is soda typically purchased with bananas? Does the brand of soda make a difference?

What should be in the basket but is not?

Are window-cleaning products purchased when detergent and orange juice are bought together?

There are many questions lurking in the shopping cart, such as questions about demographics, customers’ purchase histories, and so on. Perhaps the most important question is what is not in the shopping cart, but should be. For instance, one major grocery store chain said, “We know our customers buy toothpaste, but where do they buy toothbrushes?”
Case Study: Ethnic Marketing

A grocery store chain in Texas wants to extend its advertising by including Spanish-language ads. Should the Spanish-language advertisements be for the same products as the English advertisements?

Approaches to Answering This Question

- Market research
- Guessing
- Doing what you did the last time
- Looking at the data, that is, determining what this case study is about
Available Data

Store summaries of sales by product and by week. Consider the demographics of the catchment area. This is not detailed item-level data for each customer, but it is useful nonetheless.

The catchment area is the area from which stores primarily draw their shoppers.

(This picture is for a metro station.)


The catchment area map is from www.havant.gov.uk/havant-3337.

Summarized by Store-Week-Product

- Used for OLAP system
- Tens of thousands of products in dozens of departments
- No customer information
- Data provided by companies such as IRI
**HISPLVL: A Key Variable**

How Hispanic is the neighborhood?

**HISPLVL** takes on values from 1-15

For values 1-10, it represents the decile of *hispanicity*.
- 1 means 0%-10%.
- 2 means 10%-20%.
- ...

Values 11-15 are special because San Antonio is special.
- 11 means 50-60% in San Antonio.
- 12 means 60-70% in San Antonio.
- ...

(Apparently, no parts of San Antonio are less than 50% Hispanic.)

---

**Little of Interest Correlated with HISP_LVL**

There is a correlation. Areas with high \( \text{afam\_pct} \) have a low Hispanic percent.

This is because they are inversely related. When there is more of one group (as a percentage), there is less of the other.
Hispanics: Similar to Everyone Else

Hispanic individuals spend about the same proportion of their budgets on the following:

- meat
- snacks
- beverages
- fresh vegetables
- other items

Aggregations at the “department level” do not show significant differences.

Process for Answering in More Detail

Change the question:

- Should the Spanish-language advertisements be for the same products as the English advertisements?

becomes

- Do stores in neighborhoods with high HISPLVL sell similar products as stores in neighborhoods with low HISPLVL?

Divide stores into three groups (based on terciles):
High Hispanic, Medium Hispanic, and Low Hispanic.

Compare departments and products in High Hispanic and Low Hispanic areas.

- Ignore the areas in the middle.
- Differences in product preferences are revealed by comparing extremes.

Create a new Hispanic preference measure:

- Popularity in High Hispanic – Popularity in Low Hispanic
Some Departments

Within some departments, Hispanic preference shows high variance.

No overall difference in Hispanic preference exists at the department level. Some departments contain much more variation. That is, some departments consist of products that appeal equally to all groups. Some departments contain products that appeal to Hispanics, and other products that appeal to non-Hispanics.

Spanish-Language Ads

Spanish-language ads highlight different products.

More popular among Hispanic shoppers

More popular among non-Hispanic shoppers
What Are Association Rules?

Association rules reveal what products or events occur together.

- An example of undirected data mining
- LHS $\Rightarrow$ RHS
  - “When the products on the left-hand side are present in a transaction, then the products on the right-hand side are present.”
  - LHS = left-hand side; typically consists of zero or more products
  - RHS = right-hand side; typically consists of one product

What Are Association Rules?

When products simply tend to occur together, that is an item set.

Used for a variety of purposes:

- Embedded in automatic retailing systems to make recommendations online
- Recommending ring tones
- Cross-selling of financial products
- Data cleansing (Exceptions to very common rules suggest data issues.)
This visualization, produced by SAS Enterprise Miner, shows associations between the departments in the catalog data. The size of a node is related to the transaction count. The weight of a link is related to the association’s confidence.

**The Basic Steps for Association Rules**

1. First, determine the right set of items and the right level. Departments might be too high. SKUs might be too low.

2. Next, calculate the probabilities and joint probabilities of items and combinations of interest, perhaps limiting the search using thresholds on support or value.

3. Finally, analyze the probabilities to determine the right rules.
The first step in calculating association rules is putting the data into the proper format. A table with one row per transaction is required. The row consists of a customer ID and the item purchased. In this case, the analysis is at the department level rather than the item level.

The Data

In the Vermont Country Store catalog data, there are 48,356 customers.

Of these customers, 5,266 or about 11% made at least one purchase in the women’s apparel department.

Nearly 14% (6,707 customers) purchased shoes.
The Traditional Tests

**Support** is the proportion of market baskets where the rule is true.

1,572 customers or 3.25% purchased both women’s apparel and women’s footwear.

**Confidence** measures how often the right side of the rule is valid, given the left side.

1,572/5,266 = 29.85% confidence

**Lift** measures how much more likely the right side becomes when the left side is true.

29.85/13.87 = 2.15

A women’s apparel purchaser is twice as likely to purchase women’s footwear.

Rules as a Table

<table>
<thead>
<tr>
<th>Rule</th>
<th>Support (%)</th>
<th>Expected Support (%)</th>
<th>Confidence (%)</th>
<th>Lift</th>
</tr>
</thead>
<tbody>
<tr>
<td>women's apparel === women's sleepwear</td>
<td>2.52</td>
<td>8.24</td>
<td>31.12</td>
<td>2.56</td>
</tr>
<tr>
<td>women's sleepwear === women's apparel</td>
<td>3.43</td>
<td>11.80</td>
<td>36.45</td>
<td>3.44</td>
</tr>
<tr>
<td>women's hosiery === men's hosiery</td>
<td>3.77</td>
<td>8.76</td>
<td>19.82</td>
<td>3.24</td>
</tr>
<tr>
<td>men's hosiery === women's hosiery</td>
<td>2.77</td>
<td>14.13</td>
<td>31.68</td>
<td>2.24</td>
</tr>
<tr>
<td>food === bedding</td>
<td>2.72</td>
<td>28.21</td>
<td>43.48</td>
<td>2.14</td>
</tr>
<tr>
<td>smoking === wine</td>
<td>2.12</td>
<td>9.20</td>
<td>12.06</td>
<td>2.14</td>
</tr>
<tr>
<td>women's sleepwear === women's hosiery</td>
<td>2.43</td>
<td>14.13</td>
<td>28.55</td>
<td>2.09</td>
</tr>
<tr>
<td>women's hosiery === women's sleepwear</td>
<td>2.43</td>
<td>8.26</td>
<td>17.23</td>
<td>2.06</td>
</tr>
<tr>
<td>women's sleepwear === women's footwear</td>
<td>2.81</td>
<td>16.30</td>
<td>31.84</td>
<td>2.06</td>
</tr>
<tr>
<td>women's footwear === women's sleepwear</td>
<td>2.81</td>
<td>8.26</td>
<td>17.15</td>
<td>2.06</td>
</tr>
<tr>
<td>women's apparel === women's hosiery</td>
<td>2.48</td>
<td>16.13</td>
<td>29.31</td>
<td>2.07</td>
</tr>
<tr>
<td>women's hosiery === women's apparel</td>
<td>2.48</td>
<td>11.83</td>
<td>24.67</td>
<td>2.07</td>
</tr>
<tr>
<td>women's food === men's food</td>
<td>1.01</td>
<td>4.04</td>
<td>11.62</td>
<td>1.01</td>
</tr>
<tr>
<td>men's food === women's food</td>
<td>3.02</td>
<td>22.32</td>
<td>45.17</td>
<td>2.02</td>
</tr>
<tr>
<td>women's underwear === women's hosiery</td>
<td>4.42</td>
<td>14.13</td>
<td>39.21</td>
<td>3.06</td>
</tr>
<tr>
<td>women's hosiery === women's underwear</td>
<td>4.42</td>
<td>22.77</td>
<td>45.46</td>
<td>3.06</td>
</tr>
<tr>
<td>women's apparel === women's footwear</td>
<td>4.96</td>
<td>11.39</td>
<td>23.44</td>
<td>1.96</td>
</tr>
<tr>
<td>women's footwear === women's apparel</td>
<td>9.59</td>
<td>11.93</td>
<td>23.44</td>
<td>1.96</td>
</tr>
<tr>
<td>food === bath</td>
<td>3.06</td>
<td>7.89</td>
<td>15.04</td>
<td>1.96</td>
</tr>
<tr>
<td>bath === bedding</td>
<td>3.06</td>
<td>28.21</td>
<td>43.48</td>
<td>1.93</td>
</tr>
<tr>
<td>women's underwear === women's apparel</td>
<td>5.23</td>
<td>22.77</td>
<td>43.05</td>
<td>1.93</td>
</tr>
<tr>
<td>women's food === women's food</td>
<td>5.23</td>
<td>22.77</td>
<td>43.05</td>
<td>1.93</td>
</tr>
<tr>
<td>women's sleepwear === women's underwear</td>
<td>3.02</td>
<td>22.77</td>
<td>43.05</td>
<td>1.93</td>
</tr>
<tr>
<td>women's underwear === women's sleepwear</td>
<td>3.02</td>
<td>3.24</td>
<td>15.91</td>
<td>1.52</td>
</tr>
<tr>
<td>women's hosiery === women's food</td>
<td>3.96</td>
<td>15.23</td>
<td>28.06</td>
<td>1.85</td>
</tr>
<tr>
<td>women's footwear === women's hosiery</td>
<td>3.96</td>
<td>14.13</td>
<td>28.06</td>
<td>1.85</td>
</tr>
<tr>
<td>women's food === women's apparel</td>
<td>4.96</td>
<td>11.83</td>
<td>23.02</td>
<td>1.88</td>
</tr>
<tr>
<td>women's apparel === women's food</td>
<td>4.96</td>
<td>11.83</td>
<td>23.02</td>
<td>1.88</td>
</tr>
<tr>
<td>food === women's sleepwear</td>
<td>3.02</td>
<td>8.26</td>
<td>14.09</td>
<td>1.81</td>
</tr>
<tr>
<td>women's sleepwear === men's food</td>
<td>3.52</td>
<td>28.31</td>
<td>43.76</td>
<td>1.81</td>
</tr>
<tr>
<td>women's sleepwear === women's food</td>
<td>3.52</td>
<td>28.31</td>
<td>43.76</td>
<td>1.81</td>
</tr>
</tbody>
</table>

SAS Enterprise Miner has slightly different figures for support, confidence, and lift because the transaction data set eliminates the several thousand customers in the catalog data who did not shop in any of the 27 departments.
In this chart, the confidence is represented by color and support by size. The arrow is pointing at a rule with high confidence. All rules with household on the right side have high confidence. When the left side is kitchen, there is also high support.

This chart illustrates a problem with the traditional measures for association rules. High support is a good thing. High lift is a good thing. Unfortunately, the rules with the best lift tend to have little support and the rules with the best support tend to have lower lift. In this diagram, the size of a node reflects its support. The color of a node reflects the lift for the corresponding rule. Red nodes tend to be small. The nodes that had high support and high confidence on the previous slide do not have high lift.
The Trouble with Lift, Support, and Confidence

All of these measures say something about how good a rule is, but it is not apparent how to combine them. The usual method is to order rules by lift, but only if they are above a certain threshold of confidence and support. For a given level of support, the highest lift rules are for the rarest products.

Chi-Square Test

The chi-square test picks effective rules that have high support. As discussed in the chapter about decision trees, the chi-square test compares the actual results with the expected results, that is, when the chi-square is higher, it is less likely that the difference is due to chance. Rules that are reliably wrong also have high chi-square scores.

- Negate the right side to get a good rule.
Association Rules: Not Always Useful

Trivial rules tell what you already know or should know.

- Customers who purchase maintenance agreements are very likely to purchase large appliances.
- If a customer has three-way calling, then the customer also has call-waiting.

Results from market basket analysis might simply measure the success of previous marketing campaigns. Inexplicable rules seem to have no explanation and do not suggest a course of action.

- When a new hardware store opens, one of the most commonly sold items is toilet rings.

More exploration might produce an interesting result, but this usually requires going beyond data analysis.

Barbie and Chocolate

Wal-Mart discovered an association rule for Barbie dolls and chocolate. They could not determine what to do with it. Someone seems to have found a solution, however.

Chocolate cake with Barbie frosting

Chocolate coins with Barbie image
This is an example of how association rules can be used to encourage cross-selling on a retail Web site. When a customer looks at one product (Levi’s 501 jeans, in this case), he or she is shown other products that might go well with that one. The products shown are determined by rules. The rules can either be entered by a merchandiser or generated automatically as described.
Sequential Rules and Other Variations

Sequential Rules:
- In the six weeks after someone purchases a new lawn mower, he or she purchases a new garden hose.

Negative rules
- tofu and ~meat ⇒ bulk grains

Sequential rules
- checking followed by savings ⇒ credit line

Incorporating item taxonomy

In addition to finding what sells with what in a single shopping trip, the association rule idea can be extended to include things that do not go together and things that do go together, but not at the same time.

Looking at rules involving purchases over time requires that the purchases be identified as made by the same customer. This is not always possible. In situations where customers are identified through a log in process or where a loyalty card is used, the process of building sequential rules is essentially the same as for market-basket associations. All of the items purchased by a single customer are treated as a single, large market basket. Each item has a timestamp associated with it. Only rules where the left-side timestamps are earlier than the right-side timestamp are considered.

Lessons Learned

Association rules are based on counting co-occurrences.
Rules can be based on items at different levels in a product hierarchy.
Rules that have good support, confidence, and lift are potentially useful.
Even these rules might be trivial, inexplicable, or simply difficult to know what to do with.
Association rules can be used for sequential pattern analysis when purchasers are identified and purchases are time-stamped.
Chapter 11  Link Analysis

11.1  Link Analysis
11.1 Link Analysis

This Chapter...

- introduces link analysis
- provides background on graph theory
- includes a case study using link analysis to generate derived variables
- shows a graph-coloring algorithm.

Graphs Convey Important Information

What is the best way to get from Avenue Wagram (home) to Porte d’Orléans or Arcueil (work)?

A subway map or interstate road network provides good examples of graphs. These show how nodes (subway stations or highway exits) are connected to each other (by subway lines or highways).

As this map of the Paris Metro shows, graphs can contain a wealth of information. You can ask questions about how to get from one point to another.
The Prussian city of Konigsberg (now Kaliningrad, a Russian enclave between Poland and Lithuania) was the inspiration for one of the earliest problems in graph theory. The city is laid out along the banks of the Pregel River. There are two islands in the river and seven bridges that connect them to each other and to the two banks.

In the 17th century, a famous mathematician, Leonhard Euler, was asked whether it was possible to cross each bridge exactly once, without taking a boat or swimming across the river. In order to answer the question, he formulated the foundations of what is known as graph theory. In honor of him, a path through a graph that contains each edge exactly once is called a Eulerian path (pronounced oilerian). (A path through a graph that contains each node is called Hamiltonian to honor another mathematician.)

Euler devised a simple method to determine if such a path exists. Namely, a path exists if the number of edges at all nodes or all nodes but one is an even number. Such a path is not possible in Konigsberg.

Graph theory is still an active branch of mathematics, but you do not actually need to know much graph theory in order to make use of the information derived from simple graphs.
Two Types of Graphs

Regular graphs have nodes connected by edges, which can go either way, in the same way as bridges. The edges could have weights. In a directed graph, the edges have a direction as well, similar to one-way signs on bridges.

Graphs That Show Relationships in Market Baskets

A graph can also be used to represent the items in a market basket. This example from SAS Enterprise Miner shows a sample of items connected by edges. An edge means that the two items were sold to the same customer. When the edge is thicker, more customers purchased both items. In this case, there is an edge between two edges with the weight representing the number of times that both items were sold together.
Who Is Calling Whom?
Directed graph with call durations on the edges

This is an example of a graph that does not represent anything geographic. The nodes represent telephones and the connections are calls between them. Each edge has two properties. First, there is a beginning and an end. (The edge is directed from one node to another.) Such a graph is called a digraph, which is a shortened form of directed graph.

In addition, each edge also has a duration, that is, the length of time that the telephone call lasted.

Unlike the graph of the metro, not all nodes in this graph are connected to each other. One question that immediately arises when looking at graphs is whether they are connected (that is, whether there is a path between any two nodes), and if not, how many components are in the graph.

Case Study
A cellular phone company uses data mining to dramatically improve the outcome of a direct-marketing campaign.
The Problem
A cellular phone company developed a new technology service offering and targeted a test-market for product launch.

Because of budget limitations, mass marketing is not a viable option.

The product is to be offered to a subset of the company’s existing customers.

A predictive model is to be tested against the company’s best effort.

Call Detail Data – Segmentation
The business-as-usual segmentation used data from the one-month billing cycle:
- minutes of usage
- number of incoming calls
- frequency of calls
- voice-mail users
- roamers

The data-mining model used additional information, such as sphere of influence.
Jane and John spend about the same amount of time on the phone. Jane has a greater sphere of influence. 
- If she is unhappy, many people hear about it.
- If her number changes, she must notify many people.

In the days before number portability, the number of distinct callers was a strong predictor of customer loyalty.

A call graph shows the connections between different telephone numbers. You can use a call graph to understand something about telephones and telephone customers.

Sphere of influence is an example of a useful variable that can be derived from a calling graph. It represents the number of distinct numbers that a telephone customer has calls to or from. This variable is predictive of all sorts of things, ranging from frequency of calls to customer service to customer loyalty.

The loyalty result reflected the fact that at the time of the study in question, there was no number portability between telephone companies. If a subscriber left one company to accept an offer from another, it meant changing phone numbers. The graphs show that this would be more of a problem for Jane than for John.
Results

A much bigger piece of the pie…
15% of the candidate population acquired the new service.
3% of the control group acquired the service.

The response rate of the model group was five times larger than the overall response rate.
Finding Fax Machines

What do you know about fax machines? Fax machines call other fax machines.

Fax calls usually last 10 seconds to a few minutes.

Dedicated fax machines do not call information (411).

By their nature, fax machines tend to call other fax machines because they speak the same language. Fax calls have a typical duration. When duration is too short, there is a possibility that a person called a fax machine and hung up quickly. At the same time, fax calls generally last a few minutes, but not much longer. Also, dedicated fax machines do not call known voice numbers, such as information.

Coloring the Graph

The initial call graph has short calls removed and some numbers labeled as Fax or Information. A graph-coloring algorithm walks through the nodes, and assigns values based on the neighbors.

Numbers connected to the initial fax machines are assigned the Fax label.

Those connected to Information are assigned Voice.

Those connected to both are Shared.

This type of algorithm is called a graph-coloring algorithm. Such an algorithm assigns a label to each node based on the neighboring nodes connected to it. You start with the known fax machines and work your way through the graph, repeating until there are no changes to the labels.
In this example, fax machines are labeled F, information (411) is labeled I, and the remainder are labeled U for unknown. Using link analysis, you can change the label of some of the unknown nodes to F. The business reason for this might be to identify residential telephone lines that are used for a business purpose. Where a dedicated fax line exists in a home, it is likely to belong to a home-based business or to a company employee who works at home. Such customers are good prospects for various work-at-home services. Where a single line is used sometimes for voice and sometimes for fax, there is an opportunity to market a second line.
Identify Other Numbers Using Connections

Repeat the Process
The FBI and other law enforcement agencies use link analysis to find connections among eyewitness accounts.

Another example of a graph is the World Wide Web. A hypertext link is a one-way link, so the Web is a directed graph. In this illustration, the home page of site A has many reciprocal links to pages within its own site and a few to other sites that do not link back to it.
The first widely used search engines on the Web relied entirely on the textual content of Web pages in order to find what the user was looking for. Pages were indexed so that the search engines knew what words were on each page. Any page containing the words in a user’s query was likely to be returned.

These early searches ignored the wealth of information contained in the graph of which sites link to what. A number of useful variables can be derived from these connections. For example, a count of the number of pages linking to a page is a measure of its popularity. The insight that Jon Kleinberg at Cornell and Sergey Brin and Lawrence Page at Stanford had was that the choice of what pages to link to reflects the human Webmaster’s opinion of the value of those pages and that furthermore, some people’s opinions are worth more than others.

Certain sites, known as *hubs*, contain links to many authoritative pages. Other sites, known as *authorities*, are linked to by many important hubs.
Identifying Authoritative Sources

Kleinberg's algorithm
- Create the root set.
  - Use a text-based search to find a few hundred close matches.
- Identify the candidates.
  - Enlarge this pool by bringing in documents that link to it or are linked to by it.
- Rank hubs and authorities.
  - A good hub is one that points to strong authorities.
  - A strong authority is one that is pointed to by good hubs.

A good hub points to many authorities. A good authority is pointed to by many hubs. Thus, each is defined in terms of the other. The algorithm for rating the hubs and authorities goes back and forth between the two. Here you concentrate on finding authoritative sources.

Starts With A Seed Set of Pages
The first step is to build a pool of candidates. This can be done by using a text-based search for pages with the right words on them along with other pages that link to them.

Initially, each node in the graph is given a hub score of 1 and an authority score of 1.
Each hub score becomes the sum of the authority scores of the nodes it points to. Each authority score becomes the sum of the hub scores of the nodes that point to it.

The hub and authority scores are normalized and the process is repeated.
Eventually, the scores converge and the pages with the highest authority scores are returned as results of the search.

**Lessons Learned**

Many problems of interest involve relationships and connections.
Connections can be represented as graphs.
There is a branch of mathematics devoted to graph theory.
Variables derived through link analysis can be used with any data-mining technique.
Sometimes the links can be used directly in the mining process. Examples include the graph-coloring algorithm used to find fax machines, and the hubs and authorities algorithm used to find Web pages.
Chapter 12  Genetic Algorithms

12.1  Optimization and Genomes ........................................................................................................12-3
12.1 Optimization and Genomes

This Chapter...
- discusses optimization problems
- describes linear programming problems
- presents the basic idea of genetic algorithms
- applies the case study.
Optimization and Data Mining

- Optimization is the act of searching for minimum and maximum values of known functions.
- Optimization is usually not considered data mining.
- Other optimization techniques include calculus, linear programming, hill-climbing, and simulated annealing.

Unlike these other techniques, genetic algorithms are general purpose and work well with other data-mining techniques such as training neural networks and developing distance functions for memory-based reasoning.

Optimization is one of the data mining tasks. However, it has a different flavor from the other tasks. Many of the data-mining techniques have optimization built into them. For instance, the method of training neural networks is an example of an optimization task. Also, proportional hazards regression used in survival analysis depends on optimization techniques.

Part of the interest in genetic algorithms is no doubt due to their biological origins, similar to neural networks and memory-based reasoning. However, they are a flexible optimization technique that can be applied in many areas. One area where they are commonly applied is for resource allocation problems.

One example is the assignment of medical students to clinics during medical school. Each student has to spend time in a clinic. Each clinic needs a certain number of students, but not too many. Each student has to pass through so many of the clinics, and so on. This type of problem is complicated with all sorts of unusual constraints. However, if an assignment of students to clinics can be represented as a genome, then genetic algorithms can evolve into a solution.
**SAS/OR Software**

The SAS operations research package, SAS/OR, contains the following:
- optimization
- linear programming
- project scheduling (CPM and PERT)
- genetic algorithms

---

**Optimization Alternatives**

- Calculating coefficients for linear regression (using explicit formulas proven mathematically)
- Calculating coefficients for logistic regression (using projection pursuit)
- Calculating coefficients for neural networks (iterative approximation)

All of these involve finding coefficients for complicated functions.
Other Algorithms

Hill-climbing algorithms use a local search.
- Often represented as a graph
- Starting from an initial vertex, looking at all the adjoining ones and moving in the optimal direction
- Easily trapped by local optima

Simulated annealing
- Based on the physics that causes crystals to form
Not All Optimization Problems Are Mathematical Functions

A company communicates to customers through three channels (e-mail, direct mail, telemarketing). The contact strategy limits customers to:
- one e-mail per week
- one call per month
- one direct mail per month
- no more than two contacts per month.

There are two types of marketing contacts:
- Retention
- Upsell

The cost of contacts varies:
- E-mail is cheap.
- Direct mail is more expensive.
- OTM is the most expensive.

Response rates vary by channel.

The value of a response depends on the following:
- offer
- type of marketing campaign

How can the company optimize the value of contacts over the next year?

In the real world, optimization problems are more complicated. This example uses customer marketing contacts as an example. The first things to notice are the limits on the contacts. An approach that contacted every customer by e-mail every week would violate the customer-contact strategy. This is an example of a resource allocation problem, where the resources are limited. The resource might be customer contacts, or it might be something more tangible such as inputs in a factory, ingredients in baked goods, or seeds that a farmer plants.

The second aspect of these types of problems is an optimization function. In this case, the optimization function is the profit over the next year. This is calculated by using the estimated response rates for the different contacts and types of campaigns and calculating the net value. Mathematically speaking, the profit function must be a linear function of the inputs.

Linear Programming Problems

Resources have constraints:
- Resource_j >= 0 (not necessary, but typical)
- Sum(Resources) <= Threshold_{overall}
- Resource_i <= Threshold_i

Profit is linear combination of resources with known coefficients.

Goal is to maximize profit.

Under these circumstances, the problem is easy to solve using the Simplex Method.
Not All Problems Are Linear

Forty medical residents must be assigned to duties at an outpatient clinic, subject to the following constraints:

- The clinic must have appropriate staff on all days.
- The clinic must have the appropriate balance of first-, second-, and third-year residents.
- Third-year residents see eight patients per day, second-year residents see six, and first-year see four.

More complex conditions:

- Third-year residents do not go to a clinic when assigned to the medical intensive-care rotation.
- Second-year residents do not go to a clinic when assigned to cardiac-care rotation.
- No more than two residents from the intensive-care rotation can be assigned to a clinic on the same day.
- No more than three residents from other rotations can be assigned to the clinic on the same day.
- All students must perform all duties at the clinic.

How can the residents be assigned to the clinic?

This is an example of a problem that is defined by the constraints. This particular problem was faced by Dr. Ewen who runs an outpatient clinic in Delaware. To solve this problem, he devised an approach whereby he rated any assignment of students to duties, and calculated the number of rules that the assignment violated. This type of rating is called an objective function.

Dr. Ewen then manually tried to find the best assignment of students to duties by tweaking the assignments. However, he was much more successful using a genetic algorithm package, which performed significantly better than he did.
The Basic Idea of Genetic Algorithms

Represent possible solutions as genomes.
Create a fitness function.
Create an initial generation of genomes.
Apply evolutionary pressure.

- **Selection** — When the genome is more fit, its chance of survival is greater.
- **Crossover** — Create new genomes as in sexual reproduction.
- **Mutation** — Introduce some random changes.

After many generations, the population is very fit.

The basic idea is to represent possible solutions to the problem as a genome. Each genome can be evaluated using something named the fitness function. The goal is to minimize (or maximize) the value of the fitness function. You do this by simulating evolution, using the three basic operations that biologists observe.
Genomes

In biology, a genome is the entire sequence of genetic material that describes an individual organism. The term is also used to describe a whole family of such mappings as in the human genome. In genetic algorithms, a genome is an encoding of a possible solution. Coming up with a good genomic representation of potential solutions is one of the challenges of using genetic algorithms.

Genomes are sequences of codes, which can be interpreted by the fitness function to get a fitness value. Initially, these values are randomly pieced together to create the genomes. This is the first generation, and you can measure the average fitness of the population. More generations are created, using the selection, cross-over, and mutation operations. During each generation, the average fitness generally increases. Over time, an optimal solution evolves.

A good way to think of the coding system is to think of each piece of the genome as being a weight, such as a weight in a regression or a neural network. The genome is then the total set of weights that is applied to a set of data.
Selection of the Fittest

Selection keeps the size of the population constant but increases the fitness of the next generation. Genomes with a higher fitness (darker shading) proliferate, and genomes with lighter shading die.

Each generation has a particular number of genomes. Selection uses the fitness values of genomes to determine which survive. When the fitness value is higher, the probability that the genome will survive from one generation to the next is higher.

Crossover

Crossover mixes the genes from the successful individuals. It is a way of combining two genomes. A crossover position determines where the genomes break and are recombined.

Crossover combines the genes from successful genomes. A crossover position is determined, and then the genes are swapped.
**Mutation Is Rare**

*Mutation* makes an occasional random change to a random position in a genome. This enables features to appear that might not be in the original population. Mutation is rare, and usually occurs once in a generation.

Mutation is the third operation. Mutations are random changes to one of the codes in the genome. The idea is that codes that did not appear in the original population have an opportunity to reappear. However, mutations are rare in nature and they are also rare in genetic algorithms. Most mutations are proven to be harmful.
This is a simple example of using genetic algorithms. The genomes consist of five codes that have the value of 0 or 1. The fitness function operates on each genome by interpreting the genome as an integer and applying the fitness function to it.

The chart shows the behavior of the fitness function. This very simple function is intended to illustrate the example. From the diagram, you can readily see that the maximum value occurs at 01111 and 10000. The values of 15 and 16 both have fitness values of 240. How can you use genetic algorithms to discover this?
The place to start with genetic algorithms is the initial population of randomly selected genomes. In this case, four genomes (sequences of length 5 with 0s and 1s) are generated. Their fitness values are shown.

After selection, one of the least fit genomes disappeared, and the most fit was replicated. This operation increased the average fitness of the second generation.

The second step is to use selection to choose the fittest genomes. You can do this by assigning each genome to a portion of the range from 0 to 1, based on its fitness value; larger fitness values have larger parts of the range. You then generate a random number, and based on where it falls in the range, you choose a genome. You do this until the generation is sufficiently large.
12.1 Optimization and Genomes

Population After Crossover

<table>
<thead>
<tr>
<th>Genome</th>
<th>P</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10010</td>
<td>18</td>
<td>234</td>
</tr>
<tr>
<td>11001</td>
<td>25</td>
<td>150</td>
</tr>
<tr>
<td>00110</td>
<td>6</td>
<td>150</td>
</tr>
<tr>
<td>10110</td>
<td>22</td>
<td>198</td>
</tr>
</tbody>
</table>

Average fitness increased from 122.5 to 183.0.

10110 crossed with 00010
10110 → 10010
00010 → 00110

The crossover operation was applied to two genomes. The result is a further increase in the average fitness. In fact, the fitness of the best genome is approaching the maximum value.

Mutations Are Usually Harmful

<table>
<thead>
<tr>
<th>Genome</th>
<th>P</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10010</td>
<td>18</td>
<td>234</td>
</tr>
<tr>
<td>11101</td>
<td>29</td>
<td>58</td>
</tr>
<tr>
<td>00110</td>
<td>6</td>
<td>150</td>
</tr>
<tr>
<td>10110</td>
<td>22</td>
<td>198</td>
</tr>
</tbody>
</table>

In the example, 11001 → 11101.

Average fitness decreases from 183.0 to 160.0.

A typical mutation rate is one code per generation. In this case, the indicated code changed from 0 to 1, and reduced the fitness value of the indicated genome. This mutation is unlikely to survive into succeeding generations.

These three processes continue until the average fitness does not change from one generation to the next.
Airlines receive comments from customers. These comments generally fall into two categories, complaints and compliments, and different types of comments should be handled by different departments. For instance, comments about food should be directed to the Food Service Department. This is an example of routing. The traditional process is to have people read the comments and decide what to do with them. Then, the comments have to be read again by the appropriate department, which is inefficient. Automatic routing of comments is something that airlines (and other industries) can use to be more efficient.

This is a difficult problem for a variety of reasons.

- First, it is an example of text mining, because most of the information is free text in the messages.
- Second, no one guarantees that the text is grammatical, that the spelling is correct, or that abbreviations are not used.

The next slides show how genetic algorithms were used to address this challenge.
The first step is to translate the customer comments into a comment signature. This is a record that describes the various fields in the comment. Some of these fields are similar to what would be found in a database, that is, the length of the message, its source, and so on. Most of the fields, though, are indicator fields based on the counts of words in the comment. In addition, some common juxtapositions are included as interaction variables.

The result is that the comment signature is very long and very sparse. Most comments contain only a few of the keywords in the signature. In addition, the comment signature has the known classification of the comment. The goal is to predict the classification given the hundreds of head, indicator, and interaction variables. Using a traditional technique such as regression does not work well because the data is so sparse.
From Signature to Genome

The genome consists of a weight corresponding to each element of the signature plus a bias.

The dot product of genome and signature produces a score.

Fitness function is a proportion of the training set correctly classified.

The genetics-algorithm approach assigns a weight to each field in the comment signature. The fitness function works by taking the weighted sum of the genome for every comment signature. For each comment signature, the weighted sum is interpreted as a classification. The actual fitness of the genome is the proportion of the training set that is correctly classified. The fitness function does not have to be simple.

Finding the Optimal Genome

The problem is to find the optimal set of weights in the genome.

Now that the data is reduced to numbers, why not use logistic regression?

- Too many variables
- Too sparse
- Lots of collinearity among the input variables
Evolve a Solution

- The initial genome has many weights set to zero.
  - Concentrate on a few important features.
- The initial non-zero positions correspond to the simplest features of the signature.
- The initial weights are chosen at random.
- Many different initial genomes are created.

The initial population has genomes where almost all the weights are 0, so the genomes can potentially identify key features in the comment signatures. The initial weights within the signature are otherwise chosen and random.

The Best Solution

After tens of thousands of generations, the final model was able to classify 85% of records correctly.

The final model achieved very good results for a problem of this type.
Genetic Algorithms Summary

Genetic algorithms are an optimization technique. They are based on the way natural selection works in natural, biological evolution. There are many ways to encode information as genomes. There are many possible fitness functions. Even problems that do not superficially resemble optimization problems can be solved using genetic algorithms by creative choices of genomes and fitness function.