Machine Learning and Data Mining – Course Notes

Gregory Piatetsky-Shapiro

This course uses the textbook by Witten and Eibe, Data Mining (W&E) and Weka software developed by their group. This course is designed for senior undergraduate or first-year graduate students.

(*) marks more advanced topics (whole modules, as well as slides within modules) that may be skipped for less advanced audiences.

Each module is designed for about 75 minutes.

Modules also contain questions (marked with Q) for discussion with students. The answers are given within the slides using the PowerPoint animation (questions appear first and answers appear after a click, giving the instructor an opportunity to discuss the question with students).

Acknowledgements.

We are grateful to Prof. Witten and Eibe for their generous permission to use many of their viewgraphs that came with their book. We are also grateful to Dr. Weng-Keen Wong for permission to use some of his viewgraphs in Module 16 (section on WSARE). Prof. Georges Grinstein has graciously permitted use of his viewgraph on census visualization. The English translation of the Minard map is used in Module 15 with the permission of Bob Abramms, ODT (www.odt.org). Several slides in Visualization module are used with permission of Ben Bederson at UMD who adapted them from John Stasko at Georgia Tech. Finally, we are grateful to Dr. Eric Bremer for permission to use his microarray data for part of the course.

Syllabus for a 14-week course:

This syllabus assumes that the course is given on Tuesdays and Thursdays, and the first week there is only a Thursday lecture. Other schedules require appropriate adjustments.

Week 1: M1: Introduction: Machine Learning and Data Mining
Assignment 0: Data mining in the news (1 week)

Week 2: M2: Machine Learning and Classification
Assignment 1: Learning to use WEKA (1 week)
M3. Input: Concepts, instances, attributes

Week 3: M4. Output: Knowledge Representation
Assignment 2: Preparing the data and mining it – basic (2 weeks)
M5. Classification - Basic methods

Week 4: M6: Classification: Decision Trees
M7: Classification: C4.5

Week 5: *M8: Classification: CART
Assignment 3: Data cleaning and preparation - intermediate (2 weeks)
*M9: Classification: more methods

Week 6: Quiz
M10: Evaluation and Credibility

Week 7: *M11: Evaluation - Lift and Costs
M12: Data Preparation for Knowledge Discovery
Assignment 4: Feature reduction (2 weeks)

Week 8: M13: Clustering
M14: Associations

Week 9: M15: Visualization
*M16: Summarization and Deviation Detection
*Assignment 5, Use CART to predict treatment outcome (1 week)

Week 10: *M17: Applications: Targeted Marketing and Customer Modeling
*M18: Applications: Genomic Microarray Data Analysis
Final Project: (4 weeks)

Week 11: M19: Data Mining and Society; Future Directions
Final Exam

Weeks 12-14: Lab, work on the final project
Project presentations are given in the last week of the term.

More detailed outline is in Outline.html

The modules are designed to be presented in the order given, from basic concepts to more advanced, and ending with 2 application case studies. The (*) modules can be skipped for a shortened introduction.

Module 1: Machine Learning, Data Mining, and Knowledge Discovery: An Introduction

In this course we will learn about the fields of Machine Learning and Data Mining (which is also sometimes called Knowledge Discovery). We will be using Weka – an excellent open-source Machine Learning Workbench (www.cs.waikato.ac.nz/ml/weka/), [WE99].

We will also be examining case studies in data mining and doing a final project, which will be a competition to predict disease classes on the unlabeled test data, given a similar training data.

1.1 Data Flood

The current technological trends inexorably lead to data flood.

More data is generated from banking, telecom, and other business transactions.
More data is generated from scientific experiments in astronomy, space explorations, biology, high-energy physics, etc.
More data is created on the web, especially in text, image, and other multimedia format.

For example, Europe's Very Long Baseline Interferometry (VLBI) has 16 telescopes, each of which produces **1 Gigabit/second** (yes, per second !) of astronomical data over a 25-day observation session. This truly generates an "astronomical" amount of data.

AT&T handles so many calls per day that it cannot store all of the data – and data analysis has to be done "on the fly".

As of 2003, according to Winter Corp. Survey, (www.eweek.com/article2/0,1759,1377106,00.asp) France Telecom has largest decision-support DB, ~30 TB (terabytes); AT&T was in second place with 26 TB database.

Some of the largest databases on the Web, as of 2003, include

- Alexa (www.alexa.com) internet archive: 7 years of data, 500 TB
- Internet Archive (www.archive.org),~ 300 TB
- Google, over 4 Billion pages (as of April 2004), many TB

UC Berkeley Professors Peter Lyman and Hal R. Varian (see www.sims.berkeley.edu/research/projects/how-much-info-2003/) estimated that 5 exabytes (5 million terabytes) of new data was created in 2002. US produces about 40% of all new stored data worldwide.

According to their analysis, twice as much information was created in 2002 as in 1999 (~30% growth rate). Other estimates give even faster growth rates for data. In any case, it is clear that data growth very rapidly and as a consequence, very little data will ever be looked at by a human

Knowledge Discovery Tools and Algorithms are NEEDED to make sense and use of data

1.2 Data Mining Application Examples

The areas where data mining has been applied recently include:

- Science
    - astronomy,
    - bioinformatics,
    - drug discovery, …
- Business
    - advertising,
    - customer modeling and CRM (Customer Relationship management)
    - e-Commerce,
    - fraud detection
    - health care, …
    - investments,
    - manufacturing,
    - sports/entertainment,
    - telecom (telephone and communications),
    - targeted marketing,
- Web:
    - search engines, bots, …
- Government
    - anti-terrorism efforts (we will discuss controversy over privacy later)
    - law enforcement,
    - profiling tax cheaters

One of the most important and widespread business applications of data mining is Customer Modeling, also called Predictive Analytics. This includes tasks such as

- predicting attrition or churn, i.e. find which customers are likely to terminate service
- targeted marketing:
    - customer acquisition – find which prospects are likely to become customers
    - cross-sell – for given customer and product, find which other product(s) they are likely to buy
- credit-risk – identify the risk that this customer will not pay back the loan or credit card
- fraud detection – is this transaction fraudulent?

The largest users of Customer Analytics are industries such as banking, telecom, retailers, where businesses with large numbers of customers are making extensive use of these technologies.

1.2.1 Customer Attrition: Case Study

Let's consider a case study of mobile phone company. Typical attrition (also called churn) rate at for mobile phone customers is around 25-30% a year!

The task is

- Given customer information for the past N (N can range from 2 to 18 months), predict who is likely to attrite in next month or two.
- Also, estimate customer value and what is the cost-effective offer to be made to this customer.

Verizon Wireless is the largest wireless service provider in the United States with a customer base of 34.6 million subscribers as of 2003 (see http://www.kdnuggets.com/news/2003/n19/22i.html). Verizon built a customer data warehouse that

- Identified potential attriters
- Developed multiple, regional models
- Targeted customers with high propensity to accept the offer
- Reduced attrition rate from over 2%/month to under 1.5%/month (huge impact over 34 million subscribers)

1.2.2 Assessing Credit Risk: Case Study

Let's consider a situation where a person applies for a loan.

Should a bank approve the loan?

Note: People who have the best credit don't need the loans, and people with worst credit are not likely to repay. Bank's best customers are in the middle.

Banks develop credit models using variety of machine learning methods.

Mortgage and credit card proliferation are the results of being able to successfully predict if a person is likely to default on a loan. Credit risk assessment is universally used in the US and widely deployed in most developed countries.

1.2.3 Successful e-commerce – Case Study

Amazon.com is the largest on-line retailer, which started with books and expanded into music, electronics, and other products. Amazon.com has an active data mining group, which focuses on personalization. Why personalization? Consider a person that buys a book (product) at Amazon.com.

Task: Recommend other books (and perhaps products) this person is likely to buy

Amazon initial and quite successful effort was using clustering based on books bought.

For example, customers who bought "Advances in Knowledge Discovery and Data Mining", also bought "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations".

Recommendation program is quite successful and more advanced programs are being developed.

1.2.4 Unsuccessful e-commerce - Case Study (KDD Cup 2000)

Of course application of data mining is no guarantee of success and during the Internet bubble of 1999-2000, we have seen plenty of examples.

Consider the "legwear and legcare" e-tailer Gazelle.com, whose clickstream and purchase data from was the subject of KDD Cup 2000 competition (http://www.ecn.purdue.edu/KDDCUP/), more on KDD Cup later)

One of the questions was: Characterize visitors who spend more than $12 on an average order at the site

The data included a dataset of 3,465 purchases, 1,831 customers

Very interesting and illuminating analysis was done by dozens Cup participants. The total time spend was thousands of hours, which would have been equivalent to millions of dollars in consulting fees.

However, the total sales of Gazelle.com were only a few thousands of dollars and no amount of data mining could help them. Not surprisingly, Gazelle.com went out of business in Aug 2000.

1.2.5 Genomic Microarrays – Case Study

DNA Microarrays are a revolutionary new technology that allows measurement of gene expression levels for many thousands of genes simultaneously (more about Microarrays later). Microarrays have recently become a popular application area for data mining (see, for example, SIGKDD Explorations Special Issue on Microarray Data Mining, [PT03], http://www.acm.org/sigkdd/explorations/issue5-2.htm)

One of the typical problems is, given microarray data for a number of patients (samples), can we

Accurately diagnose the disease?
Predict outcome for given treatment?
Recommend best treatment?

Consider a Leukemia data set [Go99], with 72 samples, and about 7,000 genes. The samples belong to two classes Acute Lymphoblastic (ALL) and Acute Myeloid (AML). The samples look similar under a microscope but have very different genetic expression levels.

We [PKR03] applied a number of different method to the training set (38 samples), took the best model and applied it to a test set (remaining 34 samples). The results were: 33 samples were diagnosed correctly (97% accuracy). Interestingly, the one error (sample 66 from the test set) was consistently misclassified by almost all algorithms. We strongly suspect that the algorithms were correct and it was the pathologist who made the error, but we cannot go back to that sample to find out who was right.

1.2.6 Security and Fraud Detection - Case Study

There are currently numerous applications of data mining for security and fraud detection. One of the most common is Credit Card Fraud Detection. Almost all credit card purchases are scanned by special algorithms that identify suspicious transactions for further action. I have recently received such a call from my bank, when I used a credit card to pay for a journal published in England. This was an unusual transaction for me (first purchase in the UK on this card) and the software flagged it.

Other applications include detection of money laundering – a notable system, called FAIS, was developed by Ted Senator for the US Treasury [Sen95].

National Association of Securities Dealers (NASD) which runs NASDAQ, has developed a system called Sonar that uses data mining for monitoring insider trading and fraud through misrepresentation (http://www.kdnuggets.com/news/2003/n18/13i.html)

Many telecom companies, including AT&T, Bell Atlantic, and British Telecom/MCI have developed systems for catching phone fraud.

Data mining and security was also very much in the headlines in 2003 with US Government efforts on using data mining for terrorism detection, as part of the ill-named and now closed Total Information Awareness Program (TIA). However, the problem of terrorism is unlikely to go away soon, and government efforts are continuing as part of other programs, such as CAPPS II or MATRIX.

Less controversial is use of data mining for bio-terrorism detection, as was done at Salt Lake Olympics 2002 (the only thing that was found was a small outbreak of tropical diseases). The system used there did a very interesting analysis of unusual events – we will return to this topic later in this course.

1.2.7 Problems Suitable for Data Mining

The previous case studies show some of the successful (and unsuccessful) applications of data mining. The areas where data mining applications are likely to be successful have these characteristics:

- require knowledge-based decisions
- have a changing environment
- have sub-optimal current methods
- have accessible, sufficient, and relevant data
- provides high payoff for the right decisions

Also, if the problem involves people, then proper consideration should be given to privacy -- otherwise, as TIA example shows, the result will be a failure, regardless of technical issues.

1.3 Knowledge Discovery

We define **Knowledge Discovery in Data** or KDD [FPSU 96] as the non-trivial process of identifying

- valid
- novel
- potentially useful
- and ultimately understandable patterns in data.

Knowledge Discovery is an interdisciplinary field, which builds upon a foundation provided by databases and statistics and applies methods from machine learning and visualization in order to find the useful patterns. Other related fields include also information retrieval, artificial intelligence, OLAP, etc.

Some people say that data mining is essentially a fancy name for statistics. It is true that data mining has much in common with Statistics and with Machine Learning. However, there are differences.

Statistics provides a solid theory for dealing with randomness and tools for testing hypotheses. It does not study topics such as data preprocessing or results visualization, which are part of data mining.

Machine learning has a more heuristic approach and is focused on improving performance of a learning agent. It also has other subfields such as real-time learning and robotics – which are not part of data mining. Data Mining and Knowledge Discovery field integrates theory and heuristics. It focuses on the entire process of knowledge discovery, including data cleaning, learning, and integration and visualization of results.

1.3.1 Knowledge Discovery Process

The key emphasis in the Knowledge Discovery field is on the process. KDD is not a single step solution of applying a machine learning method to a dataset, but continuous process with many loops and feedbacks. This process has been formalized by an industry group called CRISP-DM, (for CRoss Industry Standard Process for Data Mining). The main steps in the process include:

1. Business (or Problem) Understanding

2. Data Understanding
3. Data Preparation (including all the data cleaning and preprocessing)
4. Modeling (applying machine learning and data mining algorithms)
5. Evaluation (checking the performance of these algorithms
6. Deployment

To this we can add a 7th step – Monitoring, which completes the circle.

See www.crisp-dm.org for more information on CRISP-DM.

1.3.2 Historical Note: Many names of Data Mining

Data Mining and Knowledge Discovery field has been called by many names.

In 1960-s, statisticians have used terms like “Data Fishing” or “Data Dredging” to refer to what they considered a bad practice of analyzing data without a prior hypothesis.

The term “Data Mining” appeared around 1990-s in the database community. Briefly, there was a phrase “database mining”™, but it was trademarked by HNC (now part of Fair, Isaac), and researchers turned to “data mining”. Other terms used include Data Archaeology, Information Harvesting, Information Discovery, Knowledge Extraction, etc.

Gregory Piatetsky-Shapiro coined the term “Knowledge Discovery in Databases” for the first workshop on the same topic (1989) and this term became more popular in AI and Machine Learning Community.

However, the term data mining became more popular in business community and in the press. As of Jan 2004, Google search for "data mining" finds over 2,000,000 pages, while search for “knowledge discovery” finds only 300,000 pages.

In 2003, “data mining” has acquired a bad image because of its association with US government program of TIA (Total information awareness). Headlines such as “Senate Kills Data Mining Program”, ComputerWorld, July 18, 2003, referring to US Senate decision to close down TIA, show how much data mining became associated with TIA.

Currently, Data Mining and Knowledge Discovery are used interchangeably, and we also use these terms as synonyms.

1.4 Data Mining Tasks

Data mining is about many different types of patterns, and there are correspondingly many types of data mining tasks. Some of the most popular are

- **Classification:** predicting an item class
- **Clustering:** finding clusters in data
- **Associations:** e.g. A & B & C occur frequently
- **Visualization:** to facilitate human discovery
- **Summarization:** describing a group
- **Deviation Detection**: finding changes
- **Estimation**: predicting a continuous value
- **Link Analysis**: finding relationships
- …

Classification refers to learn a method for predicting the instance class from pre-labeled (classified) instances. This is the most popular task and there are dozens of approaches including statistics (logistic regression), decision trees, neural networks, etc.

The module examples show difference between classification, where we are looking for method that distinguish pre-classified groups, and clustering, where no classes are given, and we want to find some "natural" grouping of instances.

1.5 Summary

- Technology trends lead to data flood
    - data mining is needed to make sense of data
- Data Mining has many applications, successful and not
- Data Mining and Knowledge Discovery
  Knowledge Discovery Process
- Data Mining Tasks
    - classification, clustering, …

For more information on Data Mining and Knowledge Discovery, including

- News, Publications
- Software, Solutions
- Courses, Meetings, Education
- Publications, Websites, Datasets
- Companies, Jobs

Visit www.KDnuggets.com,

2. Module 2: Machine Learning: Finding Patterns

Study: W&E, Chapter 1

2.1 Machine Learning and Classification

Classification is learning a method for predicting the instance class from pre-labeled (classified) instances.

For example, given a set of points which were pre-classified into blue and green circles, can we guess the class of the new point (white circle)? We will use this very simple 2-dimensional example to illustrate different approaches, which corresponds to data with just 2 attributes. In this example, dividing instances in groups corresponds to drawing lines or other shapes that separate the points.

In the real world, typical data sets used for data mining have many more attributes (we will later study microarray datasets with thousands of attributes), but it is difficult to draw that many dimensions.

There are many approaches to classification, including Regression, Decision Trees, Bayesian, Nearest Neighbor, Neural Networks. In this course we will be studying several of the more popular methods.

2.1.1 Classification with Linear Regression

Linear Regression computes weights w_i for the hyperplane corresponding to an equation of the form

$$w_0 + w_1 x + \ldots + w_i y >= 0$$

which will best separate the classes. In 2-dimensions this corresponds to a straight line with equation

$$w_0 + w_1 x + w_2 y = 0$$

As we see, the straight line is not flexible enough.

2.1.2 Classification with Decision Trees

Decision tree classifiers repeatedly divide the space of examples until the remaining regions are (almost) homogeneous. In our example, we first look for the best dividing line and find

X = 5,

which corresponds to drawing a line X=5.

We then examine the remaining halves, and see that the right half has almost all blue circles, so we can make a rule

if X > 5 then blue.

We examine the left half and find another dividing line Y = 3, etc

We can now classify the white circle as belonging to blue class.

2.1.3 Classification with Neural Nets

- Unlike linear regression or decision trees which divide the space of examples using straight lines, neural nets can be thought of as select more complex regions
- As a result, they can be more accurate
- However, they can also overfit the data – find patterns in random noise

2.2 Example: The Weather Problem

This example comes from J. Ross Quinlan, a pioneer in machine learning, who used it to illustrate his break-through C4.5 program.

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

2.3 Learning as Search

2.4 Bias

2.5 Weka

Introduce students to Weka, including Explorer and Command Language Interface

3. Input: Concepts, Attributes, Instances

3.1 Concepts

W&E Learning Styles are what we call data mining tasks (See Module 1).

3.2 Examples

3.3 Attributes

Each instance is described by a fixed predefined set of features, its “attributes”.

Attribute types:

- Nominal: No inherent order among values, e.g. Outlook = “sunny”, “overcast”, “rainy”.
  - only equality tests
  - important special case: Boolean (True/False)
  - Also called symbolic or categorical attributes e.g. eye color=brown, blue, …

- Ordinal, values are ordered, e.g. Temperature values: “hot” > “mild” > “cool”
- Continuous (numeric), e.g. wind speed
  - interval quantities – integer
  - ratio quantities -- real

Note: Interval and Ratio quantities are usually grouped together as numeric, with values corresponding to numbers. Sometimes it is useful to distinguish between integer and real (continuous) numbers. Additional parameters of numeric attribute values may include precision (e.g. 2 digits after the decimal) and range (minimum and maximum values).

Q: Why does ML learner needs to know about attribute type?

A: To be able to make right comparisons and learn correct concepts, e.g.
Outlook > “sunny”

does not make sense, while
Temperature > “cool”
or
WindSpeed > 10 m/h
does

Additional uses of attribute type: check for valid values, deal with missing, etc.

3.4 Preparing the Data

ARFF format

3.4.1 Missing Values

Value may be missing because it is unrecorded or because it is inapplicable.

Consider Table 1 with an example of a hospital admissions data.

Name	Age	Sex	Pregnant?	..
Mary	25	F	N	
Jane	27	F	-	
Joe	30	M	-	
Anna	2	F	-	

The value for **Pregnant?** is missing for Jane, Joe, and Anna. For Jane, who is a 27-year old woman, we should treat this value as missing, while for Joe or Anna the value should be considered Not applicable (Joe is Male, and Anna is only 2 years old).

3.4.2 Precision Illusion

- Example: gene expression may be reported as X83 = 193.3742, but measurement error may be +/- 20.
- Actual value is in [173, 213] range, so it is appropriate to round the data to 190.
- Don’t assume that every reported digit is significant!

3.5 Summary

- Concept: thing to be learned
- Instance: individual examples of a concept
- Attributes: Measuring aspects of an instance

Note: Don’t confuse learning “Class” and “Instance” with Java “Class” and “instance”

4. Knowledge Representation

4.1 Decision tables

4.2 Decision Trees

4.3 Decision Rules

4.4 Rules involving relations

4.5 Instance-based representation

Prototypes
Representing Clusters

5. Classification: The Basic Methods

Task: Given a set of pre-classified examples, build a model or classifier to classify new cases.

Simple algorithms often work very well!

5.1 OneR: Finding One Attribute Rules

1R: learns a 1-level decision tree. All rules test just **one** attribute.

See W&E, Chapter 4.

5.2 Bayesian Modeling: Use All Attributes

6. Classification: Decision Trees - Introduction

W&E, Chapter 4

7. Classification: C4.5

W&E, Chapter 5

8. Classification: CART

CART Gymtutor Tutorial, CART Documentation, www.salford-systems.com

9. Classification: Rules, Regression, Nearest Neighbor

Study: W&E, Chapter 4

10. Evaluation and Credibility

Study: W&E, Chapter 5

11. Evaluation: Lift and Costs

In the previous chapter we studied evaluation in the typical machine learning situation when we need to classify the entire dataset and the main evaluation metric is the accuracy on the entire dataset. There are common situations, such as direct marketing paradigm, where the appropriate evaluation measure is different. In Direct Marketing, we need to find most likely prospects to contact. Not everybody needs to be contacted. Also, the number of targets is usually much smaller than number of prospects.

Typical applications include retailers, catalogues, direct mail (and email), customer acquisition, cross-sell, and attrition prediction.

Here, accuracy on the entire dataset is not the right measure.

The typical approach is to

1. develop a target model
2. score all prospects and rank them by decreasing score
3. select top P% of prospects for action

How to decide what is the best selection?

11. 1 Lift and Gains

Let us consider a random model, which sorts the prospects in random order, and measure cumulative percent of hits vs. percent of lifts (VG 5).

Definition:

CPH(P,M) = % of all targets in the first P% of the list scored by model M.

CPH (cumulative percent of hits) is also frequently called "gains".

For an "idealized" random model, where the targets will be (on average) uniformly distributed in the list, the first 5% of list will have 5% of targets, next 5% of list will also have 5% of targets, etc.

(We note that an actual random model is likely to have some fluctuations, such as the first 5% of list may have 4.5% of targets, and next 5% may have 5.3, etc, but for our purposes we consider a uniform "idealized" random model , which should more properly be called "uniform ignorant" model, but the name "random" model has stuck historically).

Q: What is expected value for CPH(P, Random) ?

A: Expected value for CPH(P, Random) = P

Consider now a non-random model and lets sort all prospects by the model score (higher score means more likely to be the target). Such score need not be the correct probability. We can plot CPH of this model along with that of the "idealized" random model (VG 6).

We note that while top 5% of random list have 5% of targets,
5% of model ranked list have 21% of targets, i.e. CPH(5%,model)=21%.

Definition: Lift at percentage P of the list sorted according to model M, is

Lift(P,M) = CPH(*P,M*) / *P*

If it is clear which model is discussed, we will omit M and write simply Lift(P).

Example (VG 7) Lift (at 5%) = 21% / 5% = 4.2, that is selection of top 5% according to this model has 4.2 times more targets than a random model.
Note: Some authors (including Witten & Eibe) use "Lift" for what we call CPH.
However, our use of Lift is commonly accepted in data mining literature.

Questions for Class Discussion:

- ***Q: Lift(P, Random) = ?***
  ***A: close to* 1 (1 is the expected value, actual value can vary because of randomness)**

- ***Q: Lift(100%, M) = ?***
  ***A:* 1 (always, and for any model M)**

- ***Q: Can lift be less than 1?***
  A: yes, if the model is very bad or sorts items by decreasing model score.

11.2 *ROC curves

- *ROC curves* are similar to gains charts
  - Stands for "receiver operating characteristic"
  - Used in signal detection to show trade-off between hit rate and false alarm rate over noisy channel
  -
- Differences from gains chart:
  - *y* axis shows percentage of true positives in sample *rather than absolute number*
  - *x* axis shows percentage of false positives in sample *rather than sample size*

(Note: ROC stands for Receiver Operating Characteristic, and was developed many years ago to study performance of radars. This approach was rediscovered by Machine Learning researchers, but the name stuck).

Study: For ROC curves and Convex Hull, see Provost and Fawcett [PF97].

11.3 Taking costs into account

Previously, our model evaluation only considered errors. However, in many (most?) cases, different types of classification errors carry different costs with them. For example, in credit card transactions, a false positive may mean putting a temporary hold on transaction, while verifying it, while a false negative may mean losing a significant amount of money. In medical testing, false positive means making an extra test and unnecessarily worrying a patient, while false negative may mean missing a treatable disease and patient dying.

Most learning schemes do not perform cost-sensitive learning. They generate the same classifier no matter what costs are assigned to the different classes. Example: standard decision tree learner.

There are methods for adapting classifiers for cost-sensitive learning:

- Re-sampling of instances according to costs
- Weighting of instances according to costs
- MetaCost approach by Pedro Domingos [Do99]
- Some schemes are inherently cost-sensitive, e.g. naïve Bayes

11.3.1 KDD-Cup 1998: Cost-sensitive learning case study

While cost-sensitive learning is widely used in the industry, there are not many published cases.

One well known and public case study is KDD Cup 1998
(www.kdnuggets.com/meetings/kdd98/kdd-cup-98.html).

(Note: KDD Cup is the annual Data Mining Competition, where the best researchers and companies compete to do the best data mining job on the same data.)

KDD Cup 1998 used data from a charity called Paralyzed Veterans of America (PVA).
PVA had a list of so-called "lapsed donors" -- people who donated in the past, but not donated recently.
Past data on these donors was provided, but the results of mailing (who actually donated) were withheld until the end of the contest.
The goal of the competition was to select a subset of lapsed donors to contact.

Evaluation: Maximum actual profit from selected list (with mailing cost = $0.68), computed as

> Sum of (actual donation-$0.68) for all records with predicted/ expected donation > $0.68

We will study KDD Cup 1998 in more details in a later lesson.

11.4 Evaluating numeric predictions

- Same strategies: independent test set, cross-validation, significance tests, etc.
- Difference: error measures
- Actual target values: *a1 a2 …an*
- Predicted target values: *p1 p2 … pn*
- Most popular measure: *mean-squared error*

$$\frac{(p_1 - a_1)^2 + ... + (p_n - a_n)^2}{n}$$

Other measures include:

- Root mean-squared error
- Mean absolute error
- Root relative squared error
- Relative absolute error
- Correlation coefficient

It is not always clear which of these measures is the best, so frequently we look at all of them.
Often it doesn't matter, because the rankings of models are frequently the same.

11.5 MDL principle and Occam's razor

W&E, Chapter 5

Discuss Occam's razor – why do we believe that simpler theory is likely to be more accurate?

Common criteria in mathematics and physics – a beautiful or correct theory is more likely to be correct.

However, this is not so in biology. Here is an example. Watson and Creek (discoverers of DNA structure) initial conjecture of how 4 bases (A, T, G, C) encode 20 amino acids. They proposed that 3-base combinations are used, which gives 4x4x4 = 64 combinations. They thought that the same-base combinations (AAA, TTT, GGG, CCC) are used as “STOP” words. The remaining 60 are grouped by threes by grouping combinations that represent a rotational shift (e.g. ATC, TCA, CAT will be grouped together). Each group will correspond to the same amino acid. This method is very elegant and provides error correction. However, Nature uses a much more complex and less elegant mechanism that is more prone to errors like omission or insertion of a single base, which can lead to generating wrong amino acids and diseases.

12. Data Preparation for Knowledge Discovery

The Knowledge Discovery Process (as formalized by CRISP-DM) shows all steps of equal size. However, in practice Data Preparation is estimated to consume 70-80% of the overall effort. In this lesson, we examine the common steps of Data Preparation, beginning with the pre-requisite step of data understanding.

12.1 Data understanding

Typical questions we ask during

- What data is available for the task?
- Is this data relevant?
- Is additional relevant data available?
- How much historical data is available?
- Who is the data expert ?

We also want to understand the data in terms of quantity:

- Number of instances (records)
  - *Rule of thumb: 5,000 or more desired*
  - if less, results are less reliable; use special methods (boosting, …)
- Number of attributes (fields)
  - *Rule of thumb: for each field, 10 or more instances*
  - If more fields, use feature reduction and selection
- Number of targets
  - *Rule of thumb: >100 for each class*
  - if very unbalanced, use stratified sampling

12.2 Data cleaning

Main data cleaning steps include:

- Data acquisition
- Creating metadata
- Dealing with missing values
- Unified date format
- Nominal to numeric
- Discretization
- Data validation and statistics

- *Time series processing (advanced topic, not covered in this course)

12.2.1 Data acquisition

Most tools include methods for reading data from databases (ODBC, JDBC), from flat files (fixed-column or delimited by tab, comma, etc), or other files, like spreadsheets.
Weka uses ARFF format, where data is comma-delimited.

Note: Pay attention to possible field delimiters inside string values, e.g. an address field may contain an embedded comma which may screw things up.

Verify the number of fields before and after conversion.

12.2.2 Creating Metadata

Along with acquiring the data, it is important to acquire the correct metadata, which is computer-readable data description. While there are many parameters that describe data, most important ones are:

- **Field types:**
    - Binary, nominal (categorical), ordinal, numeric (integer or real), …
    - For numeric fields, minimum and maximum value may be specified as part of data checking.
    - For nominal fields that contain codes (e.g. states), get tables that translate codes into full descriptions that are useful for presenting to users.

- **Field role:**
    - input: inputs for modeling
    - target: output
    - id/auxiliary: keep, but not use for modeling
    - ignore: don't use for modeling
    - weight: instance weight
    - …

- **Field descriptions**
  long field descriptions useful for data understanding and output to users

Next step is converting data to a standard format (e.g. arff or csv).
Weka supplies modules that convert CSV data to arff format.

Important considerations include

- Missing values
- Unified date format
- Convert nominal fields whose values have order to numeric.
- Binning of numeric data
- Fix errors and outliers

12.2.3 Missing Values

- Missing data can appear in several forms:
    - <empty field> "0" "." "999" "NA" …
    - Standardize missing value code(s) in converted data (C4.5 and Weka represents missing values by "?".

- Handling missing values:
    - Ignore records with missing values : usually not done, unless many values are missing in the record
    - Treat missing value as a separate value
      If the likely causes of missing data are the same, then this is possible. If not, we may need to have several distinct codes, depending on the reason (e.g. "Not Applicable", "Not reported", etc).
    - Replace with zero, mean, or median values
    - Imputation: try to impute (estimate) the missing value from other field values.

How are they represented in the input data; how to represent them in converted data.

12.2.4 Date Transformation

Some systems accept dates in many formats

- e.g. "Sep 24, 2003" , 9/24/03, 24.09.03, etc
- dates are transformed internally to a standard value

If only comparison is needed, then YYYYMMDD or YYYYMM is sufficient.

(Q: What is a potential problem with YYYYMMDD dates?

- ***A: Looming Year 10,000 crisis ? ☺)***

However, we want modeling algorithms to be able to use interval and look for possible rules of the type

Date > X or Date < Y.

To do that, we need to convert Date field into a format that allows interval comparisons.
Some options include

- Unix system date: Number of seconds since 1970
- Number of days since Jan 1, 1960 (SAS)

Problem: the values are non-obvious and don't facilitate insights that can lead to knowledge discovery (e.g. it is hard to understand the meaning of **Date > 11,270**, without doing translation back and forth)

Compromise option: KSP date format.

$$KSPdate = YYYY + \frac{Ndays - 0.5}{365 + LeapFlag}$$

where Ndays is the number of days starting from Jan 1 (Ndays for Jan 2 will be 2, for Feb 1 will be 32, etc).
LeapFlag is 1 for leap year, 0 otherwise.
This format has several advantages

- Preserves intervals between days, unlike YYYYMMDD format (although intervals are slightly smaller during leap year)
- The year and quarter are obvious
  - Sep 24, 2003 is 2003 + (267-0.5)/365= 2003.7301 (round to 4 digits)
- Consistent with date starting at noon
- Can be extended to include time

Dealing with 2 digit Year

Y2K issues: some data may still have 2-digit dates.

Does the year of "02" correspond to 2002 or 1902?

This is application dependent – if the field is "House construction year", it may be 1902, while for "Year of birth" in a pediatric hospital it is likely to be 2002.

A typical approach is to set a CUTOFF year, e.g. 30, which would be application dependent.

If YY < CUTOFF , then treat the date as 20YY, else 19YY

12.2.5 Conversion: Nominal to Numeric Fields

If a field values are codes that represent ordered values (e.g. Temperature= cool, medium, sunny), then we want to convert the codes to numbers.

Q: Why?
A: To allow learning methods like decision trees to use ranges of values as part of rules

Also, some tools like neural nets, regression, nearest neighbor require only numeric inputs.

Problem: How to use nominal fields, e.g. Gender, Color, State, Grade in a regression?

We use different strategies for binary, ordered, multi-valued fields.

Binary Fields:

E.g. Gender=M, F

Convert to Field_0_1 with 0, 1 values.

e.g. Gender = M → Gender_0_1 = 0
Gender = F → Gender_0_1 = 1

Nominal Ordered

- Ordered attributes (e.g. Grade) can be converted to numbers preserving *natural* order, e.g.
  - A → 4
  - A- → 3.7
  - B+ → 3.3
  - B → 3
  - Etc…

Nominal, Few Values

When converting attributes that are multi-valued, with unordered values and a small (rule of thumb < 20) no. of distinct values, e.g. Color=Red, Orange, Yellow, …, Violet
for each value **v** create a binary "flag" variable **C_v** , which is 1 if Color=v, 0 otherwise

Nominal, Many Values

For examples, US State Code (50 values).

Census Profession Code: may have 7,000 values, but only 50 frequent ones.

Some ID-like fields (e.g. telephone number, employer ID) that are unique for each record should be ignored altogether.

Other fields like **Profession** we should keep, but group values into "natural" groups.
For example, 50 US States can be grouped into regions (e.g. Northeast, South, Midwest, and West)

For **Profession** we can select the most frequent professions and group the rest.
Once we reduced the number of values, we can create binary flag-fields for each value, as above.

12.2.6 Discretization (Binning) of Numeric Fields

On the other hand, for numeric data we may consider binning – to reduce the number of distinct numeric values. A special case of binning is reducing the unnecessary precision. For example, some of genetic expression data is specified with several digits after the period. If we know that the measurement error is more than 10 units, we can round numbers like 293.6743 to 290.

Discretization: Equal-Width

dividing the field range into equal width steps
E.g. Temperature values for the "Golf" game are:

64 65 68 69 70 71 72 72 75 75 80 81 83 85

We can divide them into bins:

[64,67) [67,70) [70,73) [73,76) [76,79) [79,82) [82,85]

Problems with equal width discretization: some fields may have a very unequal distribution of values. Consider salary in a corporation. CEO may have a salary of 2,000,000, while a junior clerk – only 20,000.

If we discretize the field salary into 10 equal-width bins, then almost all values will fall into the first bin (0 to 200,000). This problem can be avoiding by using

Discretization: Equal-Height

The same temperature values can be discretized into 4 bins (each with 4 values, except the last one)

[64 .. .. 69] [70 .. 72] [73 .. .. 81] [83 .. 85]

Generally equal-height is the preferred scheme because it avoids the distorted distribution that can occur with equal height.

In practice, we don't need to have the bin exactly equal and we can improve our intuition by using "almost-equal" height binning with some additional considerations:

- don't split frequent values across bins
- create separate bins for special values (e.g. 0)
- adjust breakpoints a little to make them more readable (e.g. round to nearest integer or nearest 10)

Discretization: Class-dependent

Previously, we have considered discretization that was class-independent. If we have class information, we can also use it to produce discretization. (See W&E, chapter 7)

Discretization Summary:

- Equal Width is simplest, good for many classes
  - can fail miserably for unequal distributions
- Equal Height gives better results
- Class-dependent can be better for classification
  - Note: decision trees build discretization on the fly
  - Naïve Bayes requires initial discretization
- Many other methods exist …

12.2.7 Fix Errors and Outliers.

We want to examine field values that do not fit into the specified data types.
Examine the minimum and maximum field values for possible errors. For example, customer year of birth 2001 may be an error, if we are looking at credit card customers (at least in the year 2004).

We can simply ignore outliers. However, for some numeric method very large values can skew the results. One approach is to truncate the top 1 % of values to the value of 99% percentile.

If we have domain specific bounds, then we can use them for setting upper and lower bounds for field values. For example, for Affymetrix gene expression data, we enforce the floor of 20 and an upper bound of 16,000.

Visual inspection of data and summary statistics is helpful to detect outliers.

12.3 Field Selection

Field selection proceeds in 3 main steps:

- Remove fields with no or little variability
- Remove false predictors
- Select most relevant fields

Remove fields with no or little variability

Examine the number of distinct field values

Rule of thumb: remove a field where almost all values are the same (e.g. null), except possibly in ***minp*** *% or less of all records.*

minp could be 0.5% or more generally less than 5% of the number of targets of the smallest class

12.3.1 False Predictors

False predictors are fields correlated to target behavior, which describe events that happen at the same time or after the target behavior.

If databases don't have the event dates, a false predictor will appear as a good predictor
Example: Service cancellation date is a leaker when predicting attriters.

A manual approach to finding false predictors is:

- Build an initial decision-tree model
- Consider very strongly predictive fields as "suspects"
    - strongly predictive – if a field by itself provides close to 100% accuracy, at the top or a branch below
- Verify "suspects" using domain knowledge or with a domain expert
- Remove false predictors and build a revised model

Automating False Predictor Detection:

- For each field
    - Compute correlation with the target field

    - Or, build 1-field decision trees for each field
- Rank all suspects by strength of correlation or prediction
- Remove suspects whose 1-field prediction strength is close to 100% (however, this is domain dependent threshold -- verify top "suspects" with domain expert)

12.3.2 Selecting Most Relevant Fields

- If there are too many fields, select a subset that is most relevant.
- Can select top N fields using 1-field predictive accuracy as computed earlier.
- What is good N?
    - Rule of thumb -- keep top 50 fields

Why reduce the number of fields?

- most learning algorithms look for non-linear combinations of fields -- can easily find many spurious combinations given small # of records and large # of fields
- Classification accuracy improves if we first reduce number of of fields
- Multi-class heuristic: select equal # of fields from each class

12.4 Derived Variables

- Better to have a fair modeling method and good variables, than to have the best modeling method and poor variables.
- Insurance Example: People are eligible for pension withdrawal at age 59 ½. Create it as a separate Boolean variable!
- *Advanced methods exist for automatically examining variable combinations, but it is very computationally expensive!

12.5 Unbalanced Target Distribution

- Sometimes, classes have very unequal frequency
    - Attrition prediction: 97% stay, 3% attrite (in a month)
    - medical diagnosis: 90% healthy, 10% disease
    - eCommerce: 99% don't buy, 1% buy
    - Security: >99.99% of Americans are not terrorists
- Similar situation with multiple classes
- Majority class classifier can be 97% correct, but useless

Handling unbalanced data:

- With two classes: let positive targets be a minority
- Separate raw Held set (e.g. 30% of data) and raw train
    - put aside raw held and don't use it till the final model
- Select remaining positive targets (e.g. 70% of all targets) from raw train
- Join with equal number of negative targets from raw train, and randomly sort it.
- Separate randomized balanced set into balanced train and balanced test

Learning with Unbalanced Data

- Build models on balanced train/test sets
- Estimate the final results (lift curve) on the raw held set

- Can generalize “balancing” to multiple classes
    - stratified sampling
    - Ensure that each class is represented with approximately equal proportions in train and test

12.6 Data Preparation Key Ideas

- Use meta-data
- Inspect data for anomalies and errors
- Eliminate “false positives”
- Develop small, reusable software components
- Plan for verification - verify the results after each step

Good data preparation is the key to producing valid and reliable models

13. Clustering

We have earlier studied classification task or supervised learning. There we have data with instances assigned to several predefined classes (groups) and the goal is to find a method for classifying a new instance into one of these classes.

Clustering (unsupervised learning) is a related but different task – given unclassified, un-labeled data, find what are the natural groupings of instances.

There are very many different clustering methods and algorithms

- For numeric and/or symbolic data
- Deterministic vs. probabilistic
- Exclusive vs. overlapping (see example on a viewgraph)
- Hierarchical vs. flat
- Top-down vs. bottom-up

The main issue in clustering is how to evaluate the quality of potential grouping. There are many methods, ranging from manual, visual inspection to a variety of mathematical measures that minimize the similarity of items within the cluster and maximize the difference between the clusters.

We will study two basic clustering algorithms – K-means clustering, appropriate for numeric data and COBWEB, appropriate for symbolic data.

13.1 K-means Clustering

This algorithm works with numeric data only.

Algorithm:

1) Pick a number (K) of cluster centers (at random)
2) Assign every item to its nearest cluster center (e.g. using Euclidean distance)
3) Move each cluster center to the mean of its assigned items
4) Repeat steps 2,3 until convergence (change in cluster assignments less than a threshold)

The viewgraphs give an example of K-means clustering.

Advantages of K-means clustering are that it is simple, quick, and understandable.
The items are automatically assigned to clusters.

The disadvantages of the basic K-means are

- the number of clusters has to be picked in advance (possible solution: try K-means with a different K; choose clustering that has a better quality, according to some metric)
- Results can vary significantly depending on initial choice of seeds. Also the method cab get trapped in local minimum (possible solution: restart clustering a number of times with different random seeds)
- All items are assigned to a cluster
- Too sensitive to outliers – very large values can skew the means (possible solution: use medians instead of means. The resulting algorithm is called K-medoids)
- Can be slow for large databases (possible solution: use sampling)

13.2 Hierarchical Clustering

W&E, Chapter 6

13.3 Discussion

We can interpret clusters by using supervised learning – learn a classifier based on clusters.

Another issue is dependence between attributes. Multiple correlated attributes tend to increase the weight of clustering based on those attributes. Such attributes should be removed to get a more objective picture of data. This could be done in pre-processing, by looking at inter-attribute correlation or principal component analysis.

Some of the clustering applications are:

- **Marketing:** discover customer groups and use them for targeted marketing and re-organization
- **Astronomy:** find groups of similar stars and galaxies
- **Earth-quake studies:** Observed earth quake epicenters should be clustered along continent faults
- **Genomics:** finding groups of gene with similar expressions

14. Associations and Frequent Item Analysis

(Study – see also W&E, 4.5)

What items people buy together when they shop?

This question is an example of association analysis, and it is one of the most popular research topics, with many interesting algorithms. One problem, however, is that it is almost too easy to find frequent associations and it is much more difficult to figure out how to use them.

14.1 Transactions

Consider the following list of transactions

TID	Produce
1	MILK, BREAD, EGGS
2	BREAD, SUGAR
3	BREAD, CEREAL
4	MILK, BREAD, SUGAR
5	MILK, CEREAL
6	BREAD, CEREAL
7	MILK, CEREAL
8	MILK, BREAD, CEREAL, EGGS
9	MILK, BREAD, CEREAL

What pairs (triples) of items are bought together frequently?
To save space in the following analysis, we replace the actual names with letters, as below.

A = milk, B= bread, C= cereal, D= sugar, E= eggs.

We can rewrite the example as

TID	Products
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

We will be referring to this example throughout this lecture.

Note – if we need to store transaction in a relational database or to arff file, we need to convert them to a fixed-column-number format. For that, we introduce for each item a binary flag (1/0) attribute, which will be 1 if this item was in the transaction, and 0 otherwise. The data can be represented as the following table.

TID	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

We define

- **Item**: *attribute=value* pair or simply *value*
  - usually attributes are converted to binary *flags* for each value, e.g. **product="A"** is written as **"A"**
- **Itemset *I*** : a subset of possible items
  - Example: ***I*** = {A,B,E} (order is unimportant)
- **Transaction**: (TID, itemset)

  Note: TID is transaction ID

14.2 Frequent itemsets

We define: **Support of an itemset**

- sup(***I***) = no. of transactions ***t*** that support (i.e. contain) ***I***

In the example database:

sup ({A,B,E}) = 2
sup ({B,C}) = 4

Frequent itemset ***I*** is one with at least the minimum support count

sup(***I***) >= minsup

14.2.1 Subset Property

Theorem: Every subset of a frequent set is frequent!

(this is the core idea that underlies much of the association analysis)

Why?

- Example: Suppose {A,B} is frequent. Since each occurrence of A, B includes both A and B, then both A and B must also be frequent
- Similar argument holds for larger itemsets.

14.3 Association rules

Once we found frequent item sets, like A,B,C , we can extract from them association rules.

Definition: Association rule *R*: *Itemset1 => Itemset2*

- ***Itemset1, 2*** are disjoint and ***Itemset2*** is non-empty
- meaning: if transaction includes ***Itemset1*** then it also has ***Itemset2***

Example: From the frequent set {A,B,E}, we can extract these 7 association rules

- A => B, E
- A, B => E
- A, E => B
- B => A, E
- B, E => A
- E => A, B
- __ => A,B,E (empty rule), which can also be written as true => A,B,E

Note: A,B,E → (empty set) is not considered an association rule, because RHS has to be non-empty.

Classification Rules are different from Association Rules.

Classification Rules

- Focus on one target field
- Specify class in all cases
- Key measure: Accuracy

Association Rules

- Have many target fields
- Applicable in some cases
- Key measures: Support, Confidence, Lift

We define Association Rule Support and Confidence as follows.

- Suppose *R : I => J* is an association rule
  - sup (R) = sup (I ∪ J) is the *support count*
    - support of itemset I ∪ J (that is I or J)
  - conf (R) = sup(J) / sup(R) is the *confidence* of R
    - fraction of transactions with I ∪ J that have J

- Association rules with minimum support and count are sometimes called "***strong***" rules

Q: Given frequent set {A,B,E}, and the current transaction example, what association rules have *minsup* = 2 and *minconf*= 50% ?

A:

A, B => E : conf=2/4 = 50%
A, E => B : conf=2/2 = 100%
B, E => A : conf=2/2 = 100%
E => A, B : conf=2/2 = 100%

Don't qualify

A =>B, E : conf=2/6 =33%< 50%
B => A, E : conf=2/7 = 28% < 50%
__ => A,B,E : conf: 2/9 = 22% < 50%

14.3.1 Finding Strong Association Rules

A rule has the parameters *minsup* and *minconf*:
sup(R) >= *minsup* and conf (R) >= *minconf*

Problem: Find all association rules with given *minsup* and *minconf*.

First, we find all frequent itemsets. Then we generate association rules from the itemsets.

14.3.2 Finding Frequent Itemsets

We start by finding one-item sets, which can be done by a single pass thru the data and counting frequent values.

The idea for the next step (Apriori algorithm, developed by Agrawal and Srikant [AS94])

Use one-item sets to generate two-item sets, two-item sets to generate three-item sets, …

From the subset property we know that if (A B) is a frequent item set, then (A) and (B) have to be frequent item sets as well!

In general: if X is frequent k-item set, then all (k-1)-item subsets of X are also frequent

Therefore, we can compute k-item set by considering and merging only (k-1) frequent item sets

For example, given: five three-item sets
(A B C), (A B D), (A C D), (A C E), (B C D)

- (note: before merging, we want to sort all item set in lexicographic order to improve efficiency)

Consider candidate four-item sets:
(A B C D) **is frequent 4-item-set**, because all its 3-item subsets are frequent

(A C D E) is **not frequent 4-item-set,** because (C D E) is not frequent

14.3.3 Generating Association Rules

This is a two stage process:

First, determine frequent itemsets e.g. with the Apriori algorithm.

```
Then, for each frequent item set I
    for each subset J of I
        determine all association rules of the form: I-J => J
```

Main idea used in both stages: subset property

Problem: Too many rules. How to compute interesting ones? We will discuss later measures for filtering rules.

Example: we can analyze weather data and find this item-set

Humidity = Normal, Windy = False, Play = Yes (Support=4)

We can generate 7 rules from an item set

In general, from K-itemset we can generate 2^K -1 potential rules

14.3.4 Using Weka to Generate Association Rules

Demonstrate using Weka

Use file: weather.nominal.arff
Use **Associate** tab
Specify MinSupport: 0.2

See what rules you find.

14.3.5 Filtering Association Rules

Problem: any large dataset can lead to very large number of association rules, even with reasonable Min Confidence and Support

Also, Confidence by itself is not sufficient

For example, if all transactions include Z, then any rule I → Z will have confidence 100%.

There are additional measures we can use to filter association rules

14.3.6 Association Rule Lift

The ***lift*** of an association rule I → J is defined as:

***lift* = P(J|I) / P(J)**

Note, P(I) = (support of I) / (no. of transactions)
ratio of confidence to expected confidence

Lift of more than 1 indicates that I and J are positively correlated, while lift < 1 indicates negative correlation. Lift=1 means that I and J are independent.

14.3.7 Other Issues

- ARFF format very inefficient for typical *market basket data*
- Attributes represent items in a basket and most items are usually missing
- More efficient formats are used

So far we talked only about analyzing simple binary data.

More complex types of analysis are also possible, and they include:

Using hierarchies:

- drink → milk → low-fat milk → Stop&Shop low-fat milk …
- this would allow us to find associations on any level

Analyze associations (sequences) over time

14.4 Applications

Most typical application of association rules is market basket analysis, which can have implications for store layout, client offers, etc.

We will also talk in a later module about another interesting application –
WSARE – “What is Strange About Recent Events”, for finding unusual events in health care data.

However, the number of reported applications of association rules is not as large because just finding the rule does not mean that we know how to use it.

Consider this example:

- Wal-Mart knows that customers who buy Barbie dolls have a 60% likelihood of buying one of three types of candy bars. (represent this as an association rules)
- What does Wal-Mart do with information like that? 'I don't have a clue,' says Wal-Mart's chief of merchandising, Lee Scott

See KDnuggets 98:01 www.kdnuggets.com/news/98/n01.html for discussion of this example and many ideas on how to use this information.

15. Visualization

Visualization has many roles in the knowledge discovery process. It can support interactive exploration, and help in effective result presentation. The human eye is a wonderful instrument, trained by millions of years of evolution to quickly see a number of patterns, especially movement, boundaries, and natural shapes, such as human faces. A picture is frequently worth a thousand words.

However, there is a big disadvantage to relying on human eyes for pattern discovery – there is too much data sometimes to visualize (it may take thousands of years to see all astronomical data generated by a Sloan Sky Survey). Also, the visualization can be misleading.

For people interested in visualization, I highly recommend the work of Edward Tufte, Professor Emeritus of Yale University, who has written a number of landmark and very beautiful books, including *The Visual Display of Quantitative Information,* **and** *Envisioning Information.*

Visualization is a huge subject and we will only give a brief introduction and overview of visualization methods, as they relate to data mining.

15.1 Examples of excellent and not so graphics

Our first example is the map of Napoleon Campaign in Russia, 1812-1813.

This classic map is a wonderful example of effectively presenting several parameters at once.
You can see a copy of the original graphic and a version translated in English (from http://www.odt.org/Pictures/minard.jpg)

The map represents the declining strength and losses of Napoleon's army during the Russian campaign of 1812-1813. It was made by Charles Joseph Minard, Inspector General of Public Works (retired) in 1869.

The number of men in the army is represented by the width of the grey line (going to Russia) and black line (coming back). One millimeter indicates ten thousand men. We can also see the geography, the town names, the river crossings (notice the correlation between those and decreases in width of the black lines – the reason is that many French soldiers drowned during the crossings).

The chart below also shows temperature (in the translated graph in Celsius and Fahrenheit) at key points.

Note how the accurate representation of the army strength by the line width conveys the dramatic decline in the army, from over 400,000 going into Russia to only 10,000 coming back.

Compare that with the New York Times 1978 chart for Fuel Economy Standards for Autos [Tu01]. Compare the line representing 18 miles per gallon (1978 standard) which is 0.6 inches long, with line representing 1985 standard of 27.5 MPG, which is 5.3 inches long.

Tufte defines

$$Lie\ Factor = \frac{size\ of\ \ effect\ shown\ in\ graphic}{size\ of\ \ effect\ in\ data} =$$

which for this example comes to

$$= \frac{\dfrac{(5.3-0.6)}{0.6}}{\dfrac{(27.5-18.0)}{18}} = \frac{7.833}{0.528} = 14.8$$

Tufte wants the lie factor to be in the range 0.95<Lie Factor<1.05 .

Other Tufte principles of graphical excellence [Tu01] include

- Give the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.
- Tell the truth about the data.

Let’s keep these principles in mind as we examine different visualization methods.

15.2 Visualization in 1-dimension

There are relatively few options for one-dimensional representation.

We can represent raw data (if numeric) as points.

We can summarize the density using a histogram, which is a discretization of a numeric value (recall our earlier discussion about equi-width vs. equi-depth discretization).

We can also present key summary statistics using Tukey box plot.
The center of the box is the median. The box edges show the first quartile (q1) and the third quartile (q3). The distance from q3 to q1 is called the inter-quartile range. The thick line is the mean (note that the mean is frequently different from the median). There are additional details and variations on the box plot that we will not go into here.

15.3 Visualization in 2 and 3 dimensions

A typical 2-D visualization is a scatter plot.

We can show a 2-D projection that will look like a 3-D visualization.

Using special red and blue glasses and a special camera we can actually create a 3-D effect (popular in some Disney attractions).

15.4 Visualization in 4 or more dimensions

We discussed a number of well-known visualization techniques for data sets of 1-3 dimensions, including line graphs, bar graphs, and scatter plots.

How can we represent higher-dimensional data?
One way is to add time, which can show movement effectively.
We can represent each variable separately, which is useful, but does not show interactions between them.

Scatterplot matrix is a method which shows all pair-wise scatterplots.

It is useful for showing linear correlations (e.g. in the Car Data example, we can see a correlation between horsepower & weight,

Q: Misses what?
A: multivariate effects

There are a number of interesting techniques for higher dimensional data.

15.4.1 Parallel Coordinates

One method is Parallel Coordinates. The concept of the parallel coordinates, as originally conceived by Alfred Inselberg [Is87], can be described as follows: In parallel coordinates, the principle coordinate axes are parallel and equidistant to each other. For each variable, we create a separate axis, and represent each multi-dimensional point as a line which connects the corresponding values along all dimensions.

The viewgraphs show an example of representing the Iris data.

Important properties of parallel visualization:

- Each data point is a line
- Similar points correspond to similar lines
- Lines crossing over correspond to negatively correlated attributes
- Parallel Visualization is a way for interactive exploration and clustering

Problems: order of axes is important (we can see negatively correlated attributes only if their corresponding axes are adjacent). Also, the graph becomes hard to see when the number of dimensions is large. A practically useful number of dimensions is 20 or less.

15.4.2 Chernoff Faces

This method was invented by Chernoff [Ch73]. The idea was to leverage human capabilities in recognizing faces and to encode different variables' values in different facial feature. For example, X can be associated with the mouth width, Y with nose length, Z with eye size. About 10 variables can be represented effectively in this way (although as a curiosity we found a SAS Macro that represents 18 facial parameters).

See also some cute interactive applets available on the web.

A major drawback of Chernoff faces is that the assignment of facial expressions to variables is difficult to optimize and strongly affects recognition of facial pattern. This is an interesting method, but rarely used for serious data analysis.

15.4.3 Stick Figures

Another idea is to visualize multi-dimensional data using stick figures (Pickett and Grinstein [PG88], and many more recent references).

Two of the variables are mapped to X, Y axes.
Other variables are mapped to limb lengths and angles
Texture patterns can show data characteristics.

15.5 Visualization Software

There is a large number of commercial for multi-dimensional visualization. Among free packages, we can mention

- Ggobi
- Xmdv

See also www.kdnuggets.com/software/visualization.html

16. Summarization and Deviation Detection

In this module we will talk about the task of summarization. Unlike predictive modeling, which tries to predict the future, the goal here is to look back at historical data and summarize it concisely. Usually, the focus of summarization is on what is new and different, unexpected.

Values can be new and different in several ways, for example with respect to previous values and/or with respect to norms or average values.

We will illustrate the issues on two examples: one is KEFIR (Key Finding Reporter), developed by G. Piatetsky-Shapiro and C. Matheus at GTE Laboratories, 1994-96, and another is WSARE (What is Strange about Recent Events), developed by Andrew Moore, Jeff Schneider, and Weng-Keen Wong at CMU (2001-3).

16.1 GTE Key Findings Reporter (KEFIR)

Study material: KEFIR Chapter from [FPSU96].

As a background, consider the issue of healthcare costs in US. Around 1995, healthcare costs consumed 12-13 % of GDP – Gross Domestic Products and were rising faster than GDP. Recent GAO projections predict increase to over 16% of GDP by 2010.

Some of the costs are due to potential problems such as fraud or misuse. Understanding where the problems are is first step to fixing them. GTE, which was a large telephone company (now part of Verizon), was self insured for medical costs. The total GTE healthcare costs in mid-90s were in hundreds of millions of dollars.

The task for our project in support of GTE Health Care Management was to analyze employee health care data and generate a report that describes the major problems.

The KEFIR approach was:

- Analyze all possible deviations
- Select interesting findings
- Augment key findings with:
  - Explanations of plausible causes
  - Recommendations of appropriate actions
- Convert findings to a user-friendly report with text and graphics

16.1.1 KEFIR Search Space

To find all possible deviations, KEFIR examined its search space, which had two main hierarchies.

First, the data was broken down by population group, which included

- by organization (e.g. GTE Local , GTE Directories, GTE Labs, …)
- by region (Florida, Texas, CA, …)
- age / sex group

Then, the data was also broken down by the medical problem area. The top levels of the hierarchy included

- Inpatient
  - Surgical

    - Maternity
    - Mental Health
    - Chemical Dependency
    - Medical
- Outpatient

and so on.

Finally, for each population group and for each medical problem area, KEFIR computed several measures.

Some of the measures were generic, such as **Total_Payments**, and other were specific to particular problem areas, such as **Average Length Of Stay**, which was relevant to Inpatient but not to Outpatient admissions.

Measures were related by formulas to other measures, e.g.

Total_Payments = **Payments_Per_Case** * **Num_Cases**

The total number of measures computed for all population groups and problem areas was on the order of several thousands.

The search "drills down", going from total payments overall to more specific measures like payment per day for specific problem areas (e.g. Surgical Inpatient Admissions) for specific groups, like Group 2 in Region X. (Actual group names and dollar amounts not used in examples to protect confidentiality).

16.1.2 KEFIR Deviation Detection

As KEFIR drills down through the search space, it generates a finding for each measure.

There main idea that helps us to determine which findings are interesting among thousands of possibilities is this: What is actionable**?**

Let us consider an example.

For example, if **Payments_Per_Case** for surgical admissions in the west region increased from $14,818 to $23,187 between 1992 and 1993, what is the financial impact on the bottom line?

Since
Total Payments = Payments_Per_Case * Num_Cases

and number of Surgical Admissions cases was 149, the impact on the bottom line was
($23,187 - $14,818) * 149 = $1,246,981.

However change from the previous value is not as important, since we cannot go back in time.
What is important is what will be the value in the next period and what can we do about it. In health-care, we can compare a value to the norm, and if the value is significantly above the norm, then there may be corrective actions to take.

(Note: The medical norms were obtained from a separate database, or computed as average values across all population groups.)

To summarize, the important comparison is the deviation of the expected measure value versus the expected norm for this medical problem.

(Note: the example shows only one past data point, but obviously more can be used).

Next, we determine the interestingness of the deviation by multiplying it by impact factor (for example, for payments per case the impact factor is the number of cases), and we get the projected impact on the bottom line.

Finally, we look if there are actions or recommendations that can be taken to reduce the impact on the bottom line.   For example, we have this recommendation:

If measure = admission_rate_per_1000 &
study_area = Inpatient_ admissions &
percent_change > 0.10

Then Utilization review is needed in the area of admission certification.

Expected Savings: 20%

We use the saving percentage to determine the projected savings, which the measure we use for determining the interestingness of findings.

16.1.3 Explanation and Report Generation

A change in measure can be explained by finding the path of related measures with the highest impact.

Example:

> The increase in **average length of stay** in this study area is related to the fact that **average length of stay** in **High-cost Admissions** rose 93.6 percent, from 16.8 days to 32.5 days. The observed deviation from the expected value is related to the fact that **average length of stay** in **Mental Health Admissions** was 544.8% above the expected value (72.0 days versus 11.2 days).

KEFIR automatically generated business-user-oriented reports, converting the findings into simple natural language sentences like one above, generated with templates. The report was augmented with appropriate bar charts and pie graphs.

16.1.4 KEFIR Overall Architecture

The data was extracted from an Informix database using SQL interface and it went thru the steps of

Deviation Detection
Key Findings and Ordering
Explanation Generator
Report Generator

The final report was delivered to the user via a browser (a very early version of Netscape).

The report was hierarchical – starting with a high-level overview which summarized key findings, there were hyperlinks that allowed user to drill down to lower level reports and explanations.

In addition to HTML report, LaTeX was used to generate a printable report in postscript.

Two pages from KEFIR output are included – the overview for Study Group 2 and Inpatient Admissions report.

KEFIR prototype implemented in GTE Laboratories in mid 1990s and it received GTE’s highest award for technical achievement in 1995. Unfortunately, Dwight McNeill, the key champion and user of KEFIR who was in charge of health care analysis left GTE in 1996 and system was no longer used.

However, the design of the system very nicely integrated several innovative ideas and is worth studying.

See book chapter on KEFIR [MPSM96].

16.2 WSARE: What’s Strange About Recent Events

WSARE system was developed at CMU and U. Pittsburgh [Wo02]. It was designed to detect the possible instances of suspicious health-care events such as bio-terrorism.

See [Wo02] paper for the detailed explanation of the system.

17. Application: Targeted Marketing and KDD Cup 1997

This module will focus on targeted or direct marketing and will examine the landmark KDD Cup 1997 competition. As we have discussed earlier, direct marketing paradigm is different from machine learning.

In direct marketing we typically have a large number of prospects and among them there is a (usually much smaller) number of targets, e.g. customer likely to buy a product or to attrite. We need to find most likely prospects and contact them.
Note that it is usually undesirable to contact everybody. The total number of contacts can be limited by profitability (a contact has a cost), capacity (limited number of offers), or other of factors.

Data mining helps to select the best list of prospects to contact.

How do we evaluate direct marketing models?

In Machine Learning we can evaluate the model by looking at the average error rate over all cases.
Here, this is not the right measure because we do NOT need to evaluate all cases.

Instead, the approach usually taken is

- develop a target model
- score all prospects and rank them by decreasing score
- select top P% of prospects for action

Typical evaluation measures that we studied earlier, such as Gains (Cum Pct Hits) and Lift are measures that can be used to measure the performance on the top P%.

We will examine KDD Cup 1997 as a case study in model building for direct marketing.

17.1 KDD Cup 1997

KDD Cup is a "world championship" in data mining, which has been held since annually since 1997, in conjunction with the International Annual Conference on Knowledge Discovery and Data Mining (KDD).

The first KDD Cup, chaired by Ismail Parsa, had the data on about 750,000 people who have previously donated to a charity. Given a lot of demographic data on past responders, predict most likely responders for the new campaign. The new campaign has been run and about 10,000 people responded to the mailing (1.4% response rate).

The training file included a stratified (non-random) sample of 10K responders and 26K non-responders (28.7% response rate). Of these, 75% used for learning; 25% used for validation
Target variable was removed from the validation data set. The goal was to predict the response in the validation data set.

Data contained 321 fields, with 'sanitized' names and labels (individually identifying information was removed). Data covered demographic information, credit history, and promotion history.

Significant part of KDD Cup dealt with data preprocessing issues. There were several leakers (false predictors) in the original data, which had to be removed over several iterations.

45 companies/institutions participated (signed the forms and downloaded the data), of which 23 used research software and 22 used commercial tools. Only 16 contestants actually completed the entire process and turned in their results. Of these, 9 used research software and 7 commercial tools.

The results show that a number of different methods were used by the contestants, with best results obtained by Rules, k-NN, and Bayes methods. The worst results were obtained by decision-tree methods.

Ismail Parsa tried a number of methods for evaluating the contestants and finally ended with Gains at 10% and at 40% of the list. According to those, first place was shared by Urban Science team with their software gain, and Charles Elkan, from UCSD, with his software BNB (Boosted Naïve Bayes).

The top 2 finishers were very close according to their gains chart.

Interestingly, the worst two finishers (code named C6 and C16) had results that were close to random and in some cases worse.

17.2 Model Evaluation Measures

One problem in KDD Cup evaluation is that comparing Gains at 10% and 40% is ad-hoc.

A more principled measure is called "Area Under the Curve" (AUC), and it is simply the area between the CPH (Gains) curve for the model and line for the random model.

Another measure is Lift Quality – which is relative to random and perfect models [PS00].

Ultimately, models should be evaluated by benefits.

17.3 Estimating Campaign Profits

Consider a Direct Mail example, based on KDD Cup 1997. We have the following parameters:

- *N* -- number of prospects, e.g. 750,000
- *T* -- fraction of targets, e.g. 0.014
- *B* -- benefit of hitting a target, e.g. $20
  - *Note: this is simplification – actual benefit will vary for different targets*
- *C* -- cost of contacting a prospect, e.g. $0.68
- *P* -- percentage selected for contact, e.g. 10%
- Lift(*P*) -- model lift at *P* , e.g. 3

The question we want to ask: What is the benefit (profit) of selecting and making an offer to the top P percent of model-score-sorted list?

Let selection be ***P*** = 10% and **Lift*(P)* = 3.** Then

- Selection size = ***N P* ,** e.g. 75,000
- Random has ***N P T*** targets in first P list, e.g. 1,050
- Model has more targets by a factor **Lift*(P)*** or ***N P T*** **Lift*(P)*** targets in the selection, e.g. 3,150
- Benefit of contacting the selection is ***N P T*** **Lift*(P) B*** , e.g. $63,000
- Cost of contacting ***N P*** is ***N P C* ,** e.g. $51,000
- The profit of contacting the top P selection is

$$Profit(P) = Benefit(P) - Cost(P) = N \bullet P \bullet T \bullet \mathrm{Lift}(P) \bullet B - N \bullet P \bullet C =$$

$$N \bullet P \bullet (T \bullet \mathrm{Lift}(P) \bullet B - C) \text{ e.g. \$12,000}$$

We can examine this equation and ask – when is the profit positive? We note that this does not depend on N or P. The profit will be positive when

$\boldsymbol{T} \bullet \textbf{Lift}(\boldsymbol{P}) \bullet \boldsymbol{B} > \boldsymbol{C}$ or $\textbf{Lift}(\boldsymbol{P}) > \boldsymbol{C} / (\boldsymbol{T} \bullet \boldsymbol{B})$, which in our example is 2.4

We can use formula for Profit (P) to evaluate the profit for every P and find the maximum value.

*17.4 Feasibility Assessment

This section is based on G. Piatetsky-Shapiro, B. Masand, Estimating Campaign Benefits and Modeling Lift, Proc. KDD-99, ACM Press, available from www.KDnuggets.com/gpspubs/

17.5 KDD Cup 1998

Paralyzed Veterans of America (PVA), a not-for-profit organization that provides programs and services for US veterans with spinal cord injuries or disease, generously provided the data set

- PVA's June 97 fund raising mailing, sent to 3.5 million donors, was selected as the competition data
- Within this universe, a group of 200K "Lapsed" donors was of particular interest to PVA. "Lapsed" donors are individuals who made their last donation to PVA 13 to 24 months prior to the mailing

Evaluation: Expected profit maximization with a mailing cost of $0.68

- Sum of (actual donation-$0.68) for all records with predicted/ expected donation > $0.68
- Participant with the highest actual sum wins

Winners: Urban Science, SAS, Quadstone.

See full results and winner's presentations at www.kdnuggets.com/meetings/kdd98

18. Application: DNA Microarray Data Mining

Study: Capturing Best Practice for Microarray Gene Expression Data Analysis, Gregory Piatetsky-Shapiro, Tom Khabaza, Sridhar Ramaswamy, Proceedings of KDD-03 Conference [PKR03].

Lawrence Hunter, Molecular Biology for Computer Scientists, Chapter 1
www.aaai.org/Library/Books/Hunter/01-Hunter.pdf or
www.cs.princeton.edu/~mona/IntroMaterials/ hunter-bio-for-CS.pdf

18.1 A Very Brief Introduction to Molecular Biology

DNA Microarray are creating a revolution in molecular biology and have the potential of helping to create personalized and customized new drugs, faster and more accurate diagnostics, and better treatments. They also generate large amounts of special data that needs to be analyzed carefully. We will study the process of microarray data analysis in this module, but first a brief introduction to molecular biology.

All living organisms consist of cells. Humans have trillions of cells. Yeast has just one cell.

Cells are of many different types (blood, skin, nerve), but all arose from a single cell (the fertilized egg). Each* cell (with very few exceptions that we will not cover here) contains a complete copy of the genome (the “program” for making the organism), encoded in DNA.

DNA molecules are long double-stranded chains; 4 types of bases are attached to the backbone: adenine (A) pairs with thymine (T), and guanine (G) with cytosine (C). A gene is a segment of DNA that specifies how to make a protein. DNA has coding and non-coding segments. Coding segments are called **exons** – they are translated into a protein. They constitute only about 2% of human genome. **Introns** are non-coding DNA, which provide structural integrity, regulatory (control), and possibly other, still unknown functions.

We as computer scientists can think of **exons** as program data (to make proteins), while introns provide the program logic.

Proteins are large molecules are essential to the structure, function, and regulation of the body. For example hormones, enzymes, and antibodies are all proteins. Human DNA has been practically fully sequenced and has about 30-35,000 genes. Interestingly rice has about 50-60,000 genes. However human genes are much longer and we think that we have much more control structure than rice (but nobody asked rice).

Cells in the same organism normally have the same genes, but these genes can be expressed differently, i.e. manufacture different messenger RNA or mRNA, which in turn manufacture different proteins, allowing creation of a huge variety of different cells. Virtually all differences in cell state or type are correlated with changes in the mRNA levels of many genes. Detection and cure of many diseases can be assisted by understanding gene expression in human and animal tissues and cells.

18.2 DNA Microarrays

The amazing thing about microarray chips is that they measure the expression levels of thousands of genes simultaneously. There are several different types of microarrays, including

- Short oligonucleotide arrays (made by Affymetrix);
- cDNA or spotted arrays (originated by Pat Brown lab at Stanford);
- Long oligonucleotide arrays (Agilent Inkjet);
- Fiber-optic arrays.

Different types of microarray use different technologies for measuring RNA expression levels;

In the Affymetrix microarrays, mRNA is isolated and labeled with fluorescent protein. Then mRNA is hybridized to the target; level of hybridization corresponds to light emission which is measured with a laser. Hybridization events are detected using a fluorescent dye and a scanner that can detect fluorescence intensities. The scanners and associated software perform various forms of image analysis to measure and report raw gene expression values. This allows for a quantitative readout of gene expression on a gene-by-gene basis. As of 2003, Affymetrix ands other companies have produce microarrays that can measure expression of over 30,000 genes, nearly all expressed human genome.

Detailed description of these technologies is beyond the scope of this course. Here we will focus on the analysis of data from Affymetrix arrays, which are currently the most popular commercial arrays. However, the methodology for analysis of data from other arrays would be similar, but would use different technology-specific data preparation and cleaning steps.

Potential applications of microarrays include:

- earlier and more accurate diagnostics
- new molecular targets for therapy
- improved and individualized treatments
- fundamental biological discovery (e.g. finding and refining biological pathways)

Some of the recent examples are

- molecular diagnosis of leukemia, breast cancer, ...
- discovery that the genetic signature strongly predicts disease outcome
- a few new drugs (Gleevac), many new promising drug targets

18.3 Microarray Data Analysis

Microarray data analysis types include:

- Gene Selection
  - Find genes for therapeutic targets (new drugs)
- Classification (Supervised)
  - Identify disease
  - Predict outcome / select best treatment
- Clustering (Unsupervised)
  - Find new biological classes / refine existing ones
  - Exploration

Here we will primarily focus on classification.

Comparing microarray data analysis with the more widespread applications of data mining, such as analytical CRM using customer data, we can see that it presents quite different challenges, for two reasons. First, the typical number of records (or samples) for microarray analysis is small – usually less than 100. This is likely to remain so for many areas of investigation, especially for human data, because of the difficulty of collecting and processing samples.

Second, the number of columns (or genes) in microarray data is normally large – typically many thousands. When building predictive models, having so many columns relative to the number of rows is likely to lead to "false positives" – gene combinations what correlate with a target variable purely by chance. This is particularly true for learning algorithms such as decision trees and neural networks, which find complex non-linear combinations of features, and therefore have a large model space in which to find spurious models.

The model also has to be explainable to a biologist and in particular, indicate which genes contribute to the model and what is the probability of a particular classification.

To get best results, we need to follow a correct methodology, which we describe in the following sections.

18.3.1 Data Preparation

The first step is data preparation.

The Affymetrix arrays measure gene expression with two types of probes: PM – perfect matches and MM – mismatches. Typically there are 10 to 20 probes per gene, depending on the chip. For each probe, the scanner measures the fluorescence level and obtains a number, which corresponds to the expression level for this probe. The quantitative value of gene expression is some function of PM and MM probes. For older arrays processed with MAS-4 software it was simply the difference sum(PM) – sum(MM) and the results could be negative (if there are more mismatches than matches).

For these datasets, we need to threshold the data to some minimum value (usually 20 is used) and some maximum (usually 16,000). Newer chips and processing software may require different thresholds.

Next step is filtering – removing genes with insufficient variation (after the thresholding), for reasons of efficiency – faster processing with fewer genes, and for biological reasons – it is desirable to look for genes with high variation.

Typically, for each gene we compute MaxVal and MinVal – maximum and minimum across all samples. Then typical filter is to EXCLUDE genes with

- MaxVal - MinVal < 500 and MaxVal / MinVal < 5 (strong filter)
- or MaxVal - MinVal < 200 and MaxVal/MinVal < 2 (weak filter)
- or Std. Dev across samples in the bottom 1/3

18.3.2 Feature (Gene) reduction

After the previous step, we typically still have several thousand of genes. We want to further reduce the genes to only 30-50 most relevant ones.

Most learning algorithms look for non-linear combinations of features and can easily find many spurious combinations given small number of records and large number of genes. To deal with this problem, we reduce number of genes by a linear method, such as T-values for mean difference between classes:

- T-test for Mean Difference= $$\frac{(Avg_1 - Avg_2)}{\sqrt{(\sigma_1^2 / N_1 + \sigma_2^2 / N_2)}}$$

Signal to Noise (S2N) = $$\frac{(Avg_1 - Avg_2)}{(\sigma_1 + \sigma_2)}$$

These measures are defined for 2 classes.
If there are multiple classes, then we compute these measures for each class vs. the rest.

Then we typically select the top 30 or 50 genes with the highest T-values for each class.

18.3.3 Measuring False Positives with Randomization

Because of the large number of genes compared to a relatively small number of samples, we are likely to get false positives, i.e. genes that appear to be correlated with the target classes but are not.

We can measure the likelihood of false positives in several ways. First, if we use T-value, we can compute the significance value for the T-value. However, the analytic approach assumes that gene values have a well-behaved distribution and are independent, which is usually not the case.

A more general and non-parametric approach is to test for randomness effect by using randomization, that is randomly permuting the class column a number of times (e.g. 1000 times). For each permutation, we then compute the T-value value for the mean difference for each randomization and for each gene, and compute for each gene the maximum and minimum T-value obtained.

If a T-value of 3.5 or higher (for any gene) occurred in only 1% of tries, we can then assign to 3.5 a 1% likelihood of occurring at random.
We can use this approach to filter genes that do not have a T-value with the desired level of significance. However, this strict filtering approach is not optimal. First, in some cases no single gene may pass the significance test. Second, this approach may remove genes that are not significant by themselves but would be significant in combination.

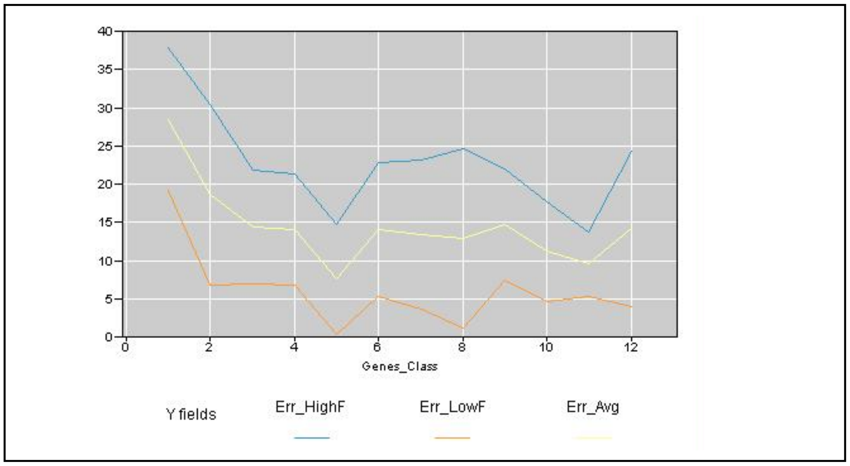
A better approach for filtering is to use the wrapper approach.

18.3.4 Wrapper approach to best gene set selection

We observe that the model that uses the top 50 genes with the highest T-values is usually not optimal.
A good way to find a better model is to test the all the subsets – e.g. top 1, 2, 3, …, 49, 50 genes (each time taking that number of genes from each class).
If that is too time consuming, we can usually get similar results by testing only some of the subsets, more initially, and less for larger number of genes, e.g. test subset of 1,2,3,4,6,8,10,12,15,20,25,30,40,50 genes.

For each subset we evaluate the model accuracy using cross-validation. If we are using an algorithm like neural networks, which uses a random number seed and whose results are not exactly deterministic, then we should repeat each run 10 times.

Finally, we get a curve that may look like this (in this figure the yellow curve is the average error, while blue and orange are 1 standard error up and down). We see that initially the average error declines as the number of genes increases. Then it reaches an “optimal plateau” where the error fluctuates up and down, and then further increase in number of genes corresponds to increasing error (because of overfitting).



The simplest approach is to select the gene set that minimizes the overall error.

g

A possible refinement is to select the smallest number of genes that is within a small margin of the lowest error. E.g. if 5 gene set has 7% error and 10 gene set has 6% error, then it is better to select a 5 gene set. (Compare with 1-SE rule for CART).

(*Advanced topic: For multi-class data sets, another possible refinement is to examine an error separately for each class and select a separate optimal gene set for each class.)

18.3.5 Feature selection should inside X-validation

Another important point is how to do feature selection. In many papers you can find an approach of global feature selection using all data, and then splitting data into train and test set (or X-validation). This approach is wrong because it potentially "leaks" information from training set to validation set. While the bias may be small for a small number of features, it becomes significant for microarray data sets with thousands of features [Si03]. As a result, classification accuracy obtained with such method is "optimistic" – higher than it should be. R. Simon reports an experiment that showed that one can achieve good classification results using this incorrect approach even on completely random data.

A proper approach is to first split the data into train and test and select genes ONLY using the train data. Once the gene set is selected, same genes should be used on test data to evaluate it. Then the process should be repeated in a cross-validation loop.

18.4 Case Studies

We have applied this methodology to several microarray data sets and obtained excellent results.
One well-known and well-studied dataset is Leukemia: Acute Lymphoblastic (ALL) vs Acute Myeloid (AML), [Go99]. It has 72 samples (38 train, 34 test) and about 7,000 genes.
It was used as part of CAMDA-2000 competition and hundreds of researchers have analyzed it.

We have done genes selection and model training on Train set only and used Neural Nets.
Best Neural Net with 10 top genes per class (20 overall) has zero errors on the train set. When it was applied to the test data (34 samples): we got 33 out 34 correct predictions (97% accuracy).

The single error was on sample 66. Interestingly, many papers and other methods also report error on the same sample, and there is strong suspicion that the neural network was correct and this sample was misclassified by the original pathologist. In this case, the classifier we got is essentially optimal – zero errors.

In another example, we applied this method to a much more complex brain tumor data [PKR03], which had 92 examples, about 7,000 genes, and 5 classes. The best model has only about 3 errors when doing proper multi-level cross-validation.

In summary, Gene Expression Microarrays have tremendous potential in biology and medicine.

Microarray Data Analysis is difficult and poses unique challenges.

Capturing the entire Microarray Data Analysis Process is critical for good, reliable results.

19. Machine Learning and Data Mining Summary

19.1 Data Mining and Society: Privacy and Security issues

Data mining offers the promise of being able to predict (sometimes) people's behavior.
For example, based on historical data, we may be able to come up with a profile that says that terrorists are likely to be males between 18 and 35 from certain countries that have produced terrorists in the past.

Does it mean that all such people should not be allowed to board a plane?

What if a bank finds that people of particular race are less likely to repay mortgage?
Should they be automatically denied mortgage?

Women in developed world live longer than men – should they pay lower insurance rates?

What about people who smoke and die younger – should they pay higher insurance rates?

Questions like these shows conflicts between patterns that data mining (or our common experience) can find, and the desires of society to avoid discrimination and to keep privacy. These conflicts involve a host of privacy, ethical, legal and political issues that are mostly beyond the scope of this course. We will discuss some of these issues from data mining point of view.

19.1.1 Data Mining and Discrimination

Can we find patterns based on sex, age, national origin, and other fields that we do not want to discriminate on?

For some activities, such as employment or issuing mortgages, there are laws that prohibit certain types of discrimination. Even if a bank finds a rule that says people with **Feature1 = X** are more likely to default on mortgages, the bank cannot use this rule to deny mortgages if **Feature1** is one of the protected features. If there is another **Feature2** closely correlated features to **Feature1** (e.g. a housing area may be strong indicator of race), then Feature2 also cannot be used. In fact, data mining can be used to find features that can be used as strong predictors of Feature1 and exclude them from decision making.

For other applications, such as medicine, it may be very important to know person's age, sex, national origin, and other parameters, because there are different risk factors for people with different parameters.

For example, people of African descent are more susceptible to sickle cell anemia. People of Eastern European Jewish descent and French Canadians are more susceptible to Tay-Sachs disease. People of Mediterranean or Asian descent experience a high incidence of Thalassemia.

(Source: http://www.childbirthsolutions.com/articles/preconception/geneticscreening/index.php)

On the other hand, there are also positive discoveries, such as discovery of a group of Indians that does not suffer from increased blood pressure (due to a very high level of cocoa consumption) or that some Latino populations in California are healthier than other populations of similar socio-economic status (the cause is also probably something in the diet), Such discoveries would be impossible if we eliminate race or national origin from medical data analysis.

Thus, when mining people data, we should be aware of the potential applications of the work and the legal and societal requirements and act accordingly.

19.1.2 Data Mining and Privacy

Can information about a person be collected for one purpose be used for mining data for another purpose?

In EU, the answer is no, not without an explicit consent of the person. See a somewhat dated, but still useful discussion on data mining and privacy at [PS95].

In US, the answer is generally yes, although the laws on this subject are in flux.

Companies routinely collect information about customers and use it for marketing, etc.
People may be willing to give up some of their privacy in exchange for some benefits.
For example, if you use frequent shopper card (such as Stop and Shop), you give up some of your privacy – your market basket transactions are recorded, but in exchange you get

The US experience is that a majority of US public is rather apathetic to issues of privacy. However, there is a strong and vocal group that is very sensitive to privacy violations and can generate significant amount of protest.

Examples: Real Networks flap over RealPlayer collection of personal information; a company that planned to sell a CD for marketing purposes with data on 100 millions US households – there were thousands of protests. However, same information is maintained today by Acxiom.

However, Data Mining looks for patterns, not people!

There are a variety of technical solutions can limit privacy invasion, while still allowing discovery of useful patterns. Data mining with privacy has been an active area of research recently.
Some of the possible solutions include:

- Replacing sensitive personal data with anon. ID
- Give randomized outputs to query, for example, instead of salary return salary + random(). This will preserve on the global patterns of salary for larger groups, but will keep the exact salary hidden
- Many interesting technical issues

For more information, see [BS03].

19.1.3 Data Mining and Security

This topic was very much in the news in 2003, due to the badly named "Total Information Awareness" (TIA) program, which was a DARPA research program designed to combine data from government and commercial databases and mine them for terrorist threat patterns. This program was closed by Congress in Sep 2003 because of the outcry from privacy advocates. However, neither the terrorist threat nor data mining capabilities went away, and so it is not surprising that many of its functions were transferred to other (mainly intelligence) agencies.

Another controversial program was CAPPS II, which was designed to use data mining methods to screen airline passengers and assign Red, Yellow, or Green labels.

There are valid reasons to criticize such programs, but first let's see if such programs can actually be effective. One line of criticism said that since databases have many errors (e.g. 5%), analyzing 100 million suspects will generate 5 million false positives, and therefore data mining will be ineffective. This line of reasoning is wrong because even if there is high likelihood of error in one individual data item, combining many data items reduces the error rate.

Let's consider a simple example. We have one thousand coins and one of them is biased (with likelihood of heads at least 60%). After throwing each coin once (which corresponds to one data item per coin), we cannot identify the biased coin. However, if we throw each coin 30 times, the biased coin will stand out with high probability.

One can conduct a similar analysis and determine that to identify 19 biased coins from about 100 million we need several hundred throws per coin.

Furthermore, there are different approaches such as link analysis. Starting with a known terrorist X, and given contact data (e.g. who calls or meets whom), the system can trace links (such as who called X) and then follow those links and identify potential suspects. It is also possible to analyze the network structure and find unusual patterns or suspicious cells.

Combining multiple models and link analysis can reduce further reduce false positives. Today there are millions of false positives with manual analysis (e.g. all millions of airline passengers who had removed their shoes in 2002-2003 were false positives). Data mining should never be considered as the ultimate answer, perfectly predicting the future, "Minority report" style. Data mining is just one additional tool to help analysts, and if used properly, it has the potential to reduce the current high rate of false positives and may actually help to catch true positives.

The recent hearings on Sep 11, 2003 have revealed that the CAPPS passenger screening system actually "flagged" nine of the 19 Sep 11 hijackers (Washington Post, Jan 28 2004). Unfortunately, the only action that was considered at the time for "flagged" passengers the system was to examine their checked luggage. Several of the hijackers did not have any luggage, and others did not have anything suspicious in the luggage, so they all went on board.

19.2 The Hype Curve

Most new technologies go thru a hype curve, as shown on the viewgraph.

At the beginning, when technology just begins to show promise, the expectations begin to rise. Then, as expectations become inflated by marketing they reach the "peak of hype", where expectations greatly exceed the actual performance of the technology. This inevitably leads to the fall into the "valley" of disappointment. However, if the technology is solid and real, then eventually performance catches up with expectations and technology becomes the mainstream.

We have seen this with the Internet, which has reached the peak of internet "bubble" in early 2000.

Data mining also went thru thus curve, as evidenced by acquisitions of data mining technology companies at the peak of the bubble at highly inflated prices. In November of 1999, E.piphany bought RightPoint for $393 million (http://www.kdnuggets.com/news/99/n24/i2.html) and in January of 2000 Vignette bought DataSage for $555 million (http://www.kdnuggets.com/news/2000/n01/i1.html).

This was followed by the drop in expectations and stock prices of data mining technology companies.

However, we are now in the post-bubble phase, with more modest but realistic expectations. The data mining technology is finally able to deliver, as evidenced by the success of companies like Amazon and Google.

19.3 Future Directions

Currently, most data mining is done on flat tables.

One of the important future directions is to develop effective methods for mining richer data sources, such as text, web with all its rich structure, images and other multimedia data.
A particularly interesting are is the analysis of links and networks, including social networks. This has wide applications, from library science, to bioinformatics, to crime detection and security.

Another important problem is "stream mining", which deals with very large data streams, so large that data cannot be effectively stored and needs to be analyzed on the fly.

Among promising applications for data mining we can see many areas of business, science, and government where there is a lot of data, including the web, ecommerce, customer modeling, and bioinformatics.

There are types of challenges that need to be overcome for further progress in data mining:

- Technical
    - dealing with tera-bytes and peta-bytes
    - dealing with complex, multi-media, structured data
    - integration with domain knowledge
    - better algorithms
    - a fundamental theory of knowledge discovery
- Business
    - finding good application area that can deliver ROI
- Societal
    - addressing privacy issues

19.4 Summary

The central quest of data mining and knowledge discovery is

Find true patterns
and avoid *overfitting*
(false patterns due
to randomness)

The key point about Knowledge Discovery is that it is a process, starting with business (problem) understanding and going thru steps of data understanding, data preparation, modeling, evaluation, deployment, and monitoring. This process is not linear, but iterative, frequently going back to previous steps.

Data Preparation is usually the most time consuming part of the process, and has been estimated to take 70-80% of the overall effort.

However, data mining and knowledge discovery is essential for making sense of the flood of data that we are generating. If done correctly, data mining and knowledge discovery has a tremendous potential to improve our lives.

Happy Discoveries!

References

[AS94] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules (1994) Proc. 20th Int. Conf. Very Large Data Bases, VLDB-94.

[BS03] Bayardo, R. and R. Srikant, Technological Solutions for Protecting Privacy, IEEE Computer, Sep 2003

[Ch73] Chernoff, H. (1973). Using faces to represent points in k-dimensional space graphically. *Journal of American Statistical Association*, *68*, 361-368.

[Do01] P. Domingos, MetaCost: a general method for making classifiers cost-sensitive, KDD-99, Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, 1999.

[FPSU96] Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy, Advances in Knowledge Discovery and Data Mining, , (Chapter 1), AAAI/MIT Press 1996.

[Go99] Golub, T. et al, Science (v. 286), Oct 1999.

[Is87] Inselberg, A., and Dimsdale, B. 1987. Parallel Coordinates for Visualizing Multi-Dimensional Geometry. In Proceedings of the Computer Graphics Intl. Conf.

[MPSM96] C. Matheus, G. Piatetsky-Shapiro, and D. McNeill, Selecting and Reporting What is Interesting: The KEFIR Application to Healthcare Data, in Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996

[PF97] Foster Provost and Tom Fawcett, Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distribution, KDD-97 (Third International Conference on Knowledge Discovery and Data Mining (http://www.hpl.hp.com/personal/Tom_Fawcett/ROCCH/)

[PG88] Pickett and Grinstein, 1988, Iconographic displays for visualizing multidimensional data. In Proc. IEEE Conf. Systems, Man, and Cybernetics, 1988.

[PKR03] Piatetsky-Shapiro, G., T. Khabaza, S. Ramaswamy, Capturing Best Practices for Microarray Data Analysis, Proceedings of ACM KDD-2003 Conference, ACM Press, 2003.

[PM99] Piatetsky-Shapiro, G., B. Masand, Estimating Campaign Benefits and Modeling Lift, Proc. KDD-99, ACM Press, available from www.KDnuggets.com/gpspubs

[PS95] Piatetsky-Shapiro, G., Guest Editor, Knowledge Discovery in Databases vs. Personal Privacy – A discussion, IEEE Expert, April 95, http://www.kdnuggets.com/gpspubs/ieee-expert-9504-priv.html

[PS00] Piatetsky-Shapiro, G. and S. Steingold, Measuring Lift Quality in Database Marketing, *SIGKDD Explorations,* December 2000.

[PT03] Piatetsky-Shapiro, G. and P. Tamayo, Guest Editors, Special Issue on Microarray Data Mining, SIGKDD Explorations 5(2), Dec 2003, http://www.acm.org/sigkdd/explorations/issue5-2.htm

[Se95] Senator, T.E., et al The Financial Crimes Enforcement Network AI System (FAIS): Identifying potential money laundering from reports of large cash transactions. AI Magazine 16, 4 (Winter 1995), 21-39.

[Si03] R. Simon, Supervised Analysis When the Number of Candidate Features (p) Greatly Exceeds The Number of Cases (n), SIGKDD Explorations Special Issue on Microarray Data Mining, Dec 2003, www.acm.org/sigkdd/explorations/issue5-2.htm

[Tu01] E. R. Tufte, The Visual Display of Quantitative Information", 2nd Edition, Graphics Press, 2001.

[Qu93] Quinlan, J. Ross, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[WE99] Witten, Ian and Eibe Frank, Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 1999.

[Wo02] Weng-Keen Wong, Andrew Moore, Gregory Cooper, Michael Wagner, Rule-based Anomaly Pattern Detection for Detecting Disease Outbreaks, Proc. of the 18th National Conference on Artificial Intelligence, 2002. (http://www.autonlab.org/autonweb/showPaper.jsp?ID=wong-rule)