

DATA MINING

LECTURE 1

Introduction



What is data mining?

- After years of data mining there is still no unique answer to this question.
- A tentative definition:



(KDD)

Data mining is the use of **efficient** techniques for the analysis of **very large** collections of data and the extraction of **useful** and possibly **unexpected** patterns in data.



Why do we need data mining?

- Really, really huge amounts of raw data!!
 - In the digital age, TB of data is generated by the second
 - Mobile devices, digital photographs, web documents.
 - Facebook updates, Tweets, Blogs, User-generated content
 - Transactions, sensor data, surveillance data
 - Queries, clicks, browsing
 - Cheap storage has made possible to maintain this data
- Need to analyze the raw data to extract knowledge

Why do we need data mining?

- “The data is the computer”
 - Large amounts of **data** can be more **powerful** than complex **algorithms** and models
 - Google has solved many Natural Language Processing problems, simply by looking at the data
 - Example: misspellings, synonyms
 - Data is power!
 - Today, the collected data is one of the biggest **assets** of an online company
 - Query logs of Google
 - The friendship and updates of Facebook
 - Tweets and follows of Twitter
 - Amazon transactions
 - We need a way to harness the **collective intelligence**

The data is also very **complex**

- Multiple **types** of data: tables, time series, images, graphs, etc
- **Spatial** and **temporal** aspects
- **Interconnected** data of different types:
 - From the mobile phone we can collect, location of the user, friendship information, check-ins to venues, opinions through twitter, images though cameras, queries to search engines

Example: transaction data

- Billions of real-life customers:
 - WALMART: 20M transactions per day
 - AT&T 300 M calls per day
 - Credit card companies: billions of transactions per day.
- The point cards allow companies to collect information about specific users

Example: document data

- Web as a document repository: estimated 50 billions of web pages
- Wikipedia: 4 million articles (and counting)
- Online news portals: steady stream of 100's of new articles every day
- Twitter: ~300 million tweets every day

Example: network data

- Web: 50 billion pages linked via hyperlinks
- Facebook: 500 million users
- Twitter: 300 million users
- Instant messenger: ~1billion users
- Blogs: 250 million blogs worldwide, presidential candidates run blogs

Example: genomic sequences

- <http://www.1000genomes.org/page.php>
- Full sequence of 1000 individuals
- 3×10^9 nucleotides per person $\rightarrow 3 \times 10^{12}$ nucleotides
- Lots more data in fact: medical history of the persons, gene expression data

Example: environmental data

- Climate data (just an example)

<http://www.ncdc.gov/oa/climate/ghcn-monthly/index.php>

- “a database of temperature, precipitation and pressure records managed by the National Climatic Data Center, Arizona State University and the Carbon Dioxide Information Analysis Center”
- “6000 temperature stations, 7500 precipitation stations, 2000 pressure stations”
 - Spatiotemporal data

Behavioral data

- Mobile phones today record a large amount of information about the user behavior
 - GPS records position
 - Camera produces images
 - Communication via phone and SMS
 - Text via facebook updates
 - Association with entities via check-ins
- Amazon collects all the items that you browsed, placed into your basket, read reviews about, purchased.
- Google and Bing record all your browsing activity via toolbar plugins. They also record the queries you asked, the pages you saw and the clicks you did.
- Data collected for millions of users on a daily basis

So, what is Data?

- Collection of data **objects** and their **attributes**
- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as **variable**, **field**, **characteristic**, or **feature**
- A collection of attributes describe an object
 - Object is also known as **record**, **point**, **case**, **sample**, **entity**, or **instance**

Objects

Attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Size: Number of objects

Dimensionality: Number of attributes

Sparsity: Number of populated object-attribute pairs

Types of Attributes

- There are different types of attributes
 - Categorical *black, brown, green, blue*
 - Examples: eye color, zip codes, words, rankings (e.g, good, fair, bad), height in {tall, medium, short}
 - Nominal (no order or comparison) vs Ordinal (order but not comparable)
 - Numeric
 - Examples: dates, temperature, time, length, value, count.
 - Discrete (counts) vs Continuous (temperature)
 - Special case: Binary attributes (yes/no, exists/not exists)

0 ✗ ✓ /

Numeric Record Data

- If data objects have the same **fixed set of numeric attributes**, then the data objects can be thought of as **points** in a multi-dimensional space, where each **dimension** represents a distinct attribute
- Such data set can be represented by an **n-by-d data matrix**, where there are **n** rows, one for each object, and **d** columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Categorical Data

- Data that consists of a collection of records, each of which consists of a **fixed set of categorical attributes**

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes ↗	Single ↘	High ↗	No
2	No ↙	Married ↗	Medium ↘	No
3	No ↙	Single	Low ↗	No
4	Yes	Married	High	No
5	No	Divorced ↗	Medium	Yes
6	No	Married	Low	No
7	Yes	Divorced	High	No
8	No	Single	Medium	Yes
9	No	Married	Medium	No
10	No	Single	Medium	Yes

Document Data

- Each document becomes a `term' vector,
 - each term is a component (attribute) of the vector,
 - the value of each component is the number of times the corresponding term occurs in the document.
 - **Bag-of-words** representation – no ordering

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

posts
documents

Transaction Data

- Each record (transaction) is a **set of items**.

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

- A set of items can also be represented as a **binary vector**, where each attribute is an item.
- A document can also be represented as a **set of words** (no counts)

Sparsity: average number of products bought by a customer

Ordered Data

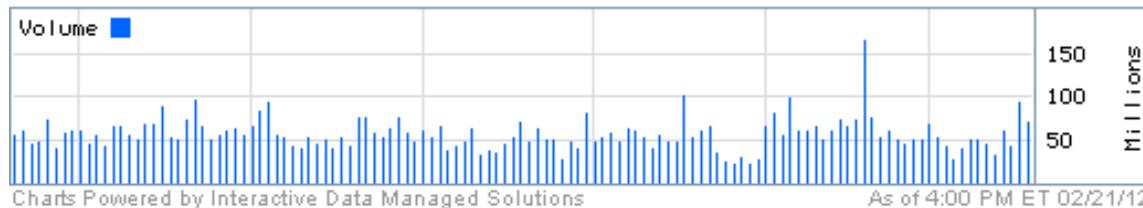
- Genomic **sequence** data

```
GGTTCCGCCTTCAGCCCCGCCGC  
CGCAGGGCCCGCCCCGCGCCGTC  
GAGAAGGGCCCCTGGCGGGCG  
GGGGGAGGCAGGGCCGCCGAGC  
CCAACCGAGTCCGACCAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGCAGGGACAG  
GCCAAGTAGAACACGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

- Data is a long **ordered** string

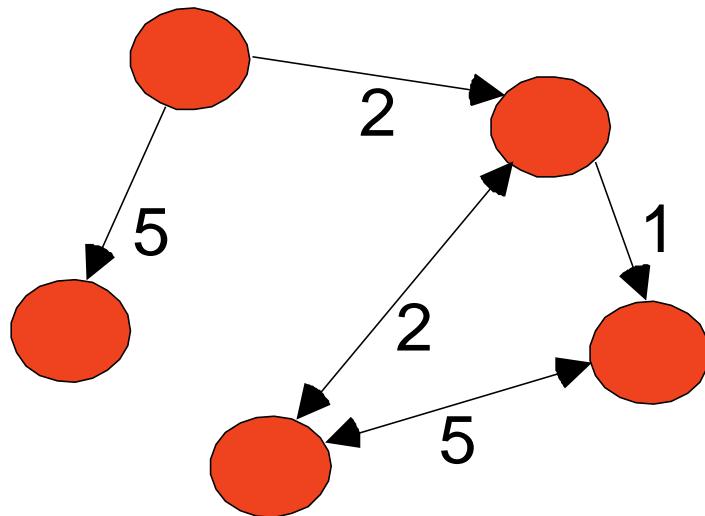
Ordered Data

- Time series
 - Sequence of ordered (over “time”) numeric values.



Graph Data

- Examples: Web graph and HTML Links



```
<a href="papers/papers.html#bbbb">  
Data Mining </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Graph Partitioning </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Parallel Solution of Sparse Linear System of Equations </a>  
<li>  
<a href="papers/papers.html#ffff">  
N-Body Computation and Dense Linear System Solvers
```

Types of data

- **Numeric data**: Each object is a point in a multidimensional space
- **Categorical data**: Each object is a vector of categorical values
- **Set data**: Each object is a set of values (with or without counts)
 - Sets can also be represented as binary vectors, or vectors of counts
- **Ordered sequences**: Each object is an ordered sequence of values.
- **Graph data**

What can you do with the data?

- Suppose that you are the owner of a supermarket and you have collected billions of **market basket** data. What information would you extract from it and how would you use it?

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Product placement

Catalog creation

Recommendations

- What if this was an online store?

What can you do with the data?

- Suppose you are a search engine and you have a **toolbar log** consisting of
 - pages browsed,
 - queries,
 - pages clicked,
 - ads clicked

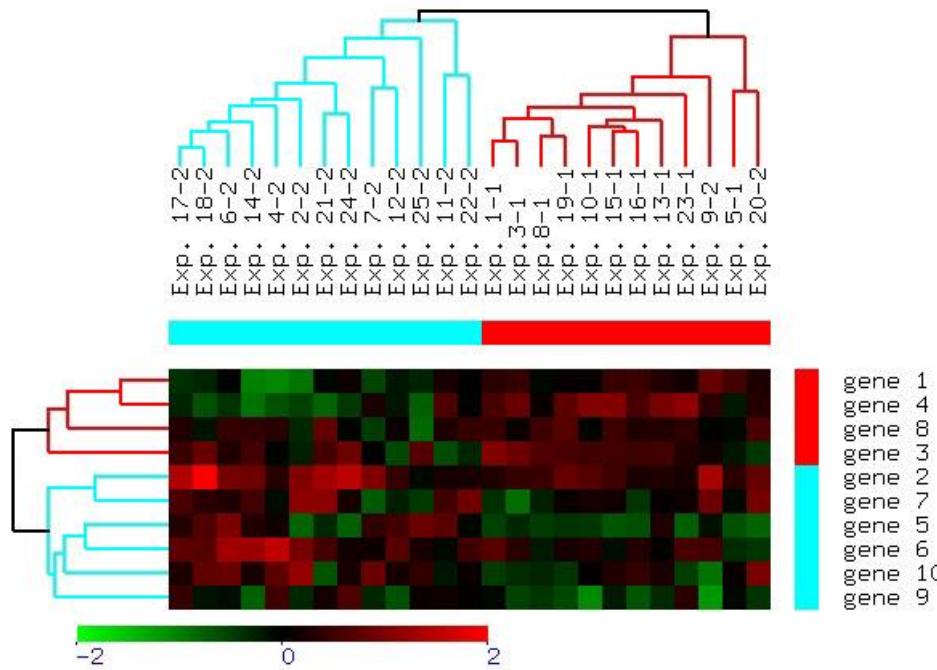
Ad click prediction

Query reformulations

each with a **user id** and a **timestamp**. What information would you like to get our of the data?

What can you do with the data?

- Suppose you are biologist who has **microarray expression data**: thousands of genes, and their expression values over thousands of different settings (e.g. tissues). What information would you like to get out of your data?



What can you do with the data?

- Suppose you are a stock broker and you observe the fluctuations of multiple stocks over time. What information would you like to get out of your data?



What can you do with the data?

- You are the owner of a social network, and you have full access to the social graph, what kind of information do you want to get out of your graph?

- Who is the most important node in the graph?
- What is the shortest path between two nodes?
- How many friends two nodes have in common?
- How does information spread on the network?

Why data mining?

- **Commercial** point of view
 - Data has become the key competitive advantage of companies
 - Examples: Facebook, Google, Amazon
 - Being able to extract useful information out of the data is key for exploiting them commercially.
- **Scientific** point of view
 - Scientists are at an unprecedented position where they can collect TB of information
 - Examples: Sensor data, astronomy data, social network data, gene data
 - We need the tools to analyze such data to get a better understanding of the world and advance science
- **Scale** (in data **size** and feature **dimension**)
 - Why not use traditional analytic methods?
 - Enormity of data, **curse of dimensionality**
 - The amount and the complexity of data does not allow for manual processing of the data. We need automated techniques.

What is Data Mining again?

- “Data mining is the analysis of (often large) observational data sets to find **unsuspected relationships** and to **summarize** the data in novel ways that are both **understandable** and **useful** to the data analyst” (Hand, Mannila, Smyth)
- “Data mining is the discovery of **models** for data” (Rajaraman, Ullman)
 - We can have the following types of models
 - Models that **explain** the data (e.g., a single function)
 - Models that **predict** the future data instances.
 - Models that **summarize** the data
 - Models that **extract** the most prominent **features** of the data.

What can we do with data mining?

- Some examples:
 - Frequent itemsets and Association Rules extraction
 - Coverage
 - Clustering
 - Classification
 - Ranking
 - Exploratory analysis

Frequent Itemsets and Association Rules

- Given a set of records each of which contain some number of items from a given collection;
 - Identify sets of items (**itemsets**) occurring frequently together
 - Produce **dependency rules** which will predict occurrence of an item based on occurrences of other items.

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Itemsets Discovered:

{Milk,Coke}

{Diaper, Milk}

Rules Discovered:

{Milk} --> {Coke}

{Diaper, Milk} --> {Beer}

Frequent Itemsets: Applications

- Text mining: finding associated phrases in text
 - There are lots of documents that contain the phrases “association rules”, “data mining” and “efficient algorithm”
- Recommendations:
 - Users who buy this item often buy this item as well
 - Users who watched James Bond movies, also watched Jason Bourne movies.
 - Recommendations make use of item and user similarity

Association Rule Discovery: Application

- Supermarket **shelf management**.
 - Goal: To identify items that are bought together by sufficiently many customers.
 - Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
 - A classic rule --
 - If a customer buys diaper and milk, then he is very likely to buy beer.
 - So, don't be surprised if you find six-packs stacked next to diapers!

Clustering Definition

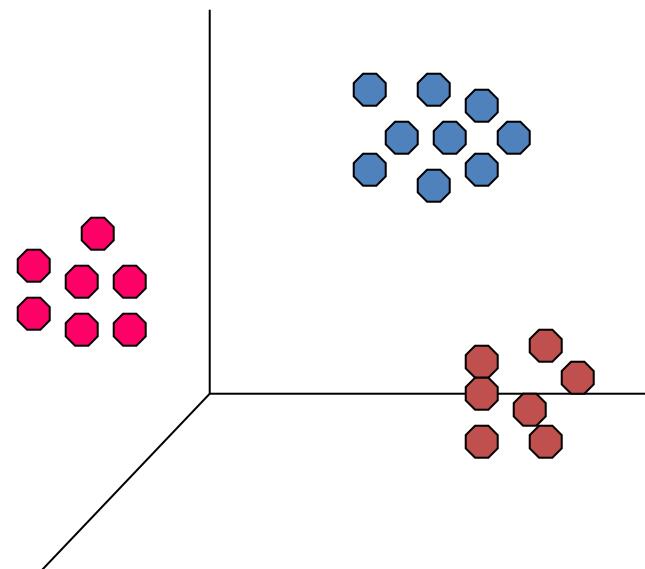
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
 - Data points in one cluster are more similar to one another.
 - Data points in separate clusters are less similar to one another.
- Similarity Measures?
 - Euclidean Distance if attributes are continuous.
 - Other Problem-specific Measures.

Illustrating Clustering

Euclidean Distance Based Clustering in 3-D space.

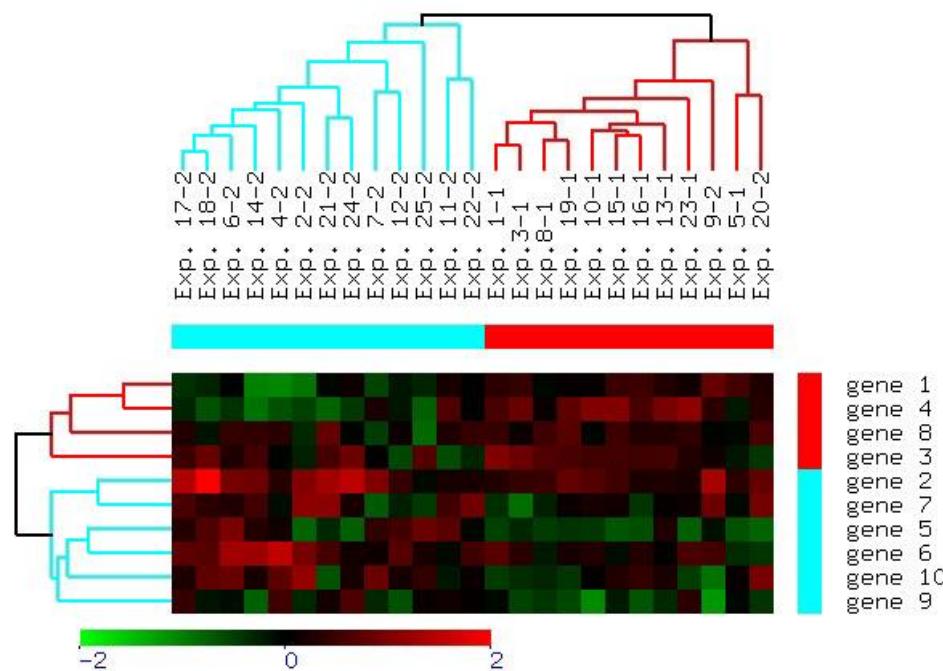
Intracluster distances
are minimized

Intercluster distances
are maximized



Clustering: Application 1

- Bioinformatics applications:
 - Goal: Group genes and tissues together such that genes are coexpressed on the same tissues



Clustering: Application 2

- Document Clustering:
 - Goal: To find groups of documents that are similar to each other based on the important terms appearing in them.
 - Approach: To identify frequently occurring terms in each document. Form a similarity measure based on the frequencies of different terms. Use it to cluster.
 - Gain: Information Retrieval can utilize the clusters to relate a new document or search term to clustered documents.

Clustering of S&P 500 Stock Data

- Observe Stock Movements every day.
- Cluster stocks if they change similarly over time.

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matl-DOWN, Bay-Network-Down, 3-COM-DOWN, Cabletron-Sys-DOWN, CISCO-DOWN, HP-DOWN, DSC-Comm-DOWN, INTEL-DOWN, LSI-Logic-DOWN, Micron-Tech-DOWN, Texas-Inst-Down, Tellabs-Inc-Down, Natl-Semiconduct-DOWN, Oracle-DOWN, SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN, Autodesk-DOWN, DEC-DOWN, ADV-Micro-Device-DOWN, Andrew-Corp-DOWN, Computer-Assoc-DOWN, Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN, Microsoft-DOWN, Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN, Fed-Home-Loan-DOWN, MBNA-Corp-DOWN, Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP, Dresser-Inds-UP, Halliburton-HLD-UP, Louisiana-Land-UP, Phillips-Petro-UP, Unocal-UP, Schlumberger-UP	Oil-UP

Coverage

- Given a set of customers and items and the transaction relationship between the two, select a small set of items that “**covers**” all users.
 - For each user there is at least one item in the set that the user has bought.
- Application:
 - Create a catalog to send out that has at least one item of interest for every customer.

Classification: Definition

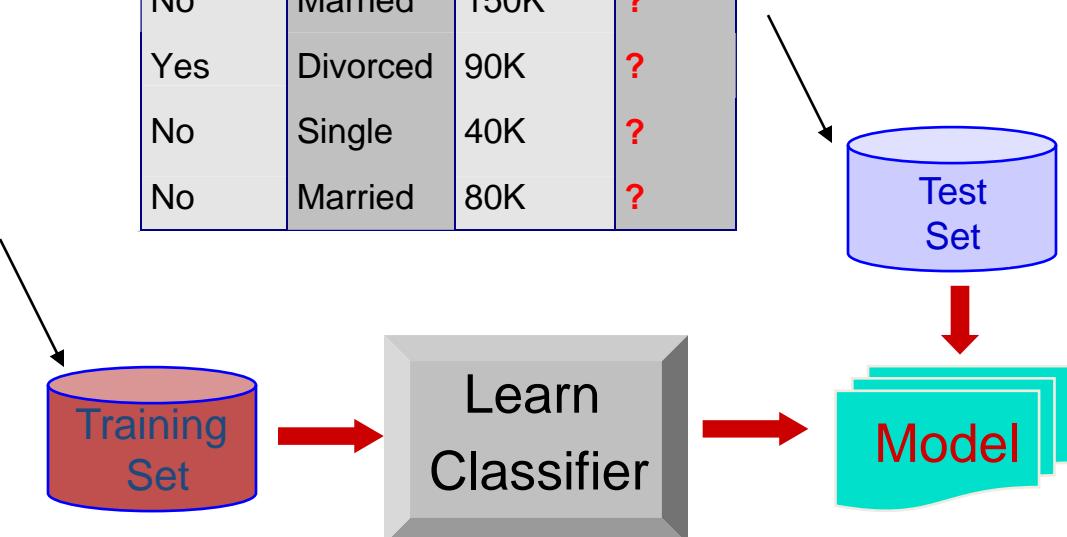
- Given a collection of records (*training set*)
 - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
 - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Example

categorical
categorical
continuous
class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Classification: Application 1

- Ad Click Prediction
 - Goal: Predict if a user that visits a web page will click on a displayed ad. Use it to target users with high click probability.
 - Approach:
 - Collect data for users over a period of time and record who clicks and who does not. The {click, no click} information forms the **class attribute**.
 - Use the history of the user (web pages browsed, queries issued) as the features.
 - Learn a classifier model and test on new users.

Classification: Application 2

- Fraud Detection
 - Goal: Predict fraudulent cases in credit card transactions.
 - Approach:
 - Use credit card transactions and the information on its account-holder as attributes.
 - When does a customer buy, what does he buy, how often he pays on time, etc
 - **Label** past transactions as fraud or fair transactions. This forms the class attribute.
 - Learn a model for the class of the transactions.
 - Use this model to detect fraud by observing credit card transactions on an account.

Link Analysis Ranking

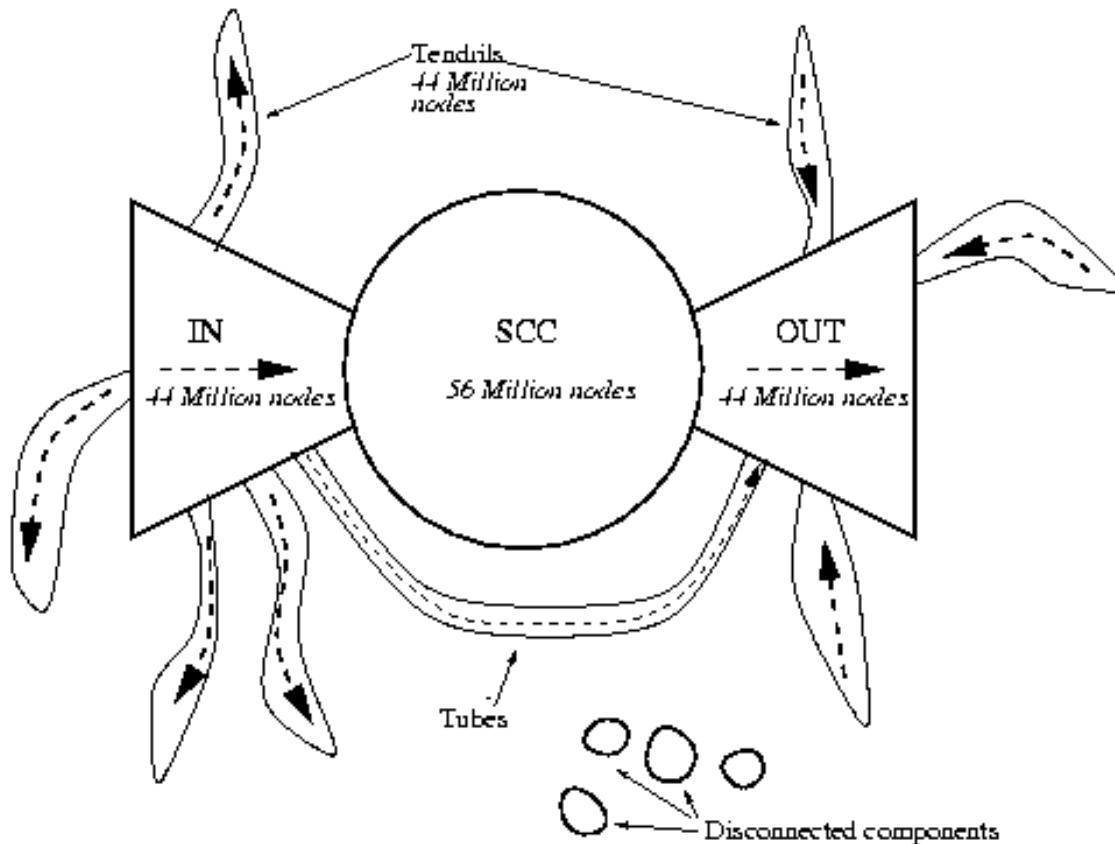
- Given a collection of web pages that are linked to each other, rank the pages according to importance (**authoritiveness**) in the graph
 - Intuition: A page gains authority if it is linked to by another page.
- Application: When retrieving pages, the authoritiveness is factored in the ranking.

Exploratory Analysis

- Trying to understand the data as a **physical phenomenon**, and describe them with simple metrics
 - What does the web graph look like?
 - How often do people repeat the same query?
 - Are friends in facebook also friends in twitter?
- The important thing is to find the right **metrics** and ask the right **questions**
- It helps our understanding of the world, and can lead to **models** of the phenomena we observe.

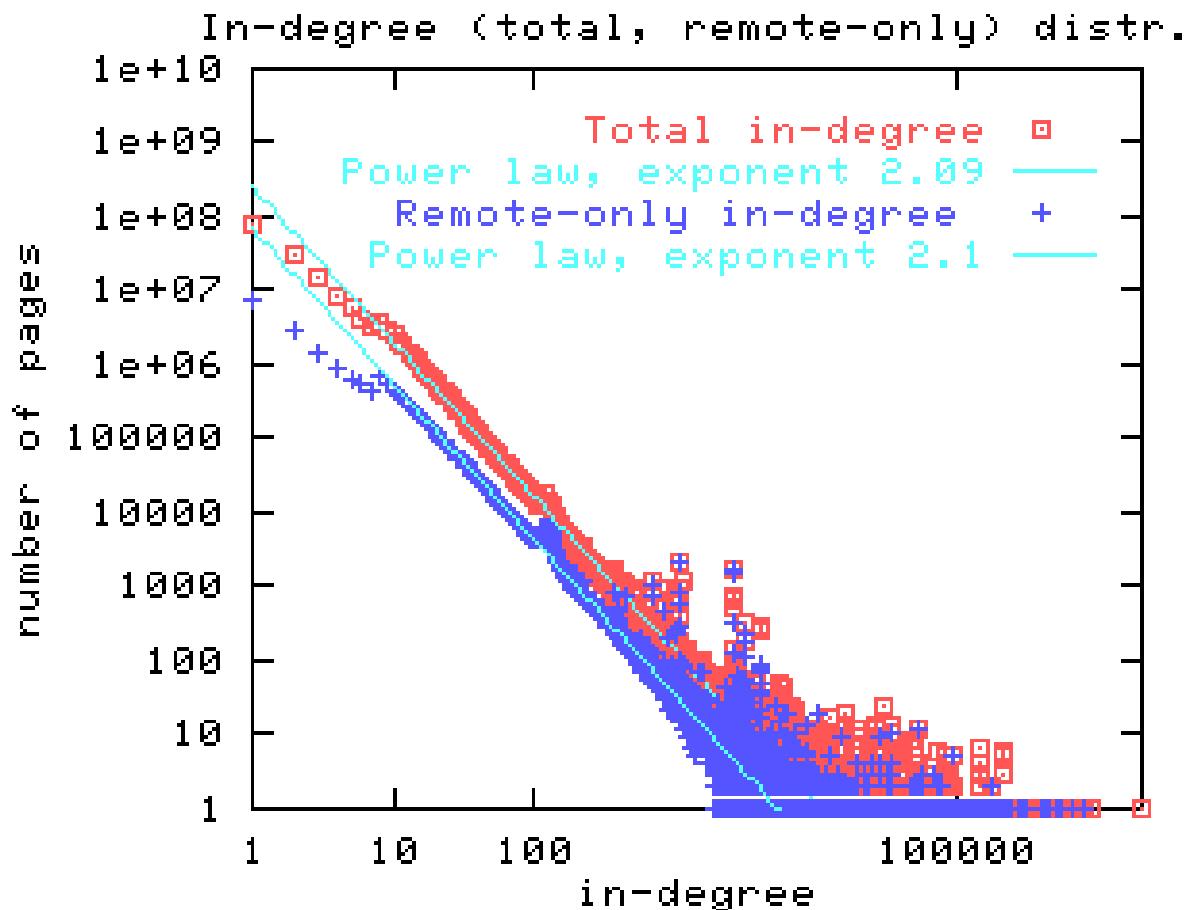
Exploratory Analysis: The Web

- What is the structure and the properties of the web?



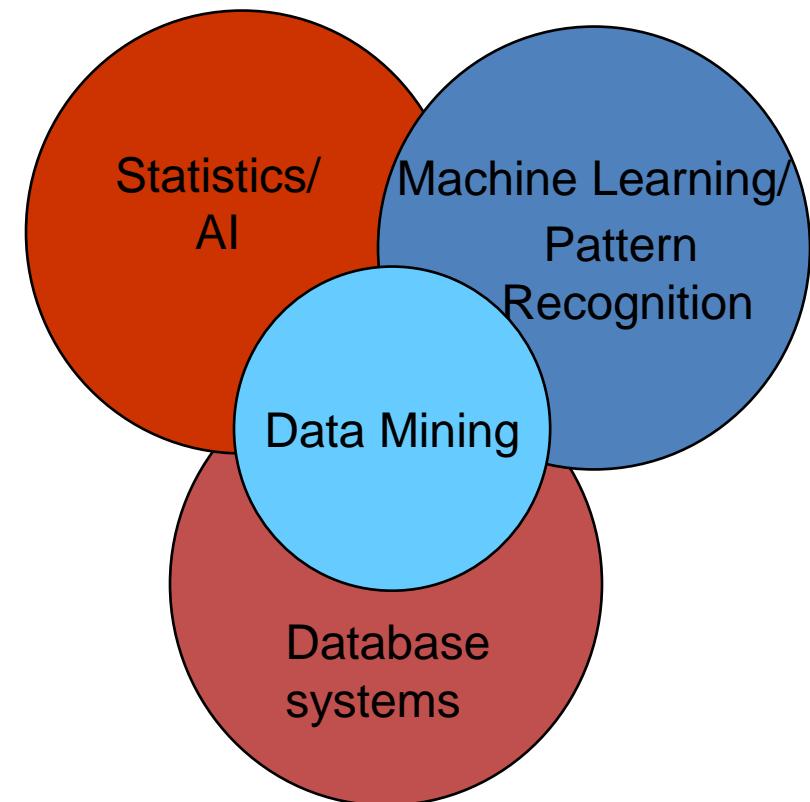
Exploratory Analysis: The Web

- What is the distribution of the incoming links?



Connections of Data Mining with other areas

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional Techniques may be unsuitable due to
 - Enormity of data
 - High dimensionality of data
 - Heterogeneous, distributed nature of data
 - Emphasis on the use of data



Cultures

- **Databases**: concentrate on large-scale (non-main-memory) data.
- **AI** (machine-learning): concentrate on complex methods, small data.
 - In today's world data is more important than algorithms
- **Statistics**: concentrate on models.

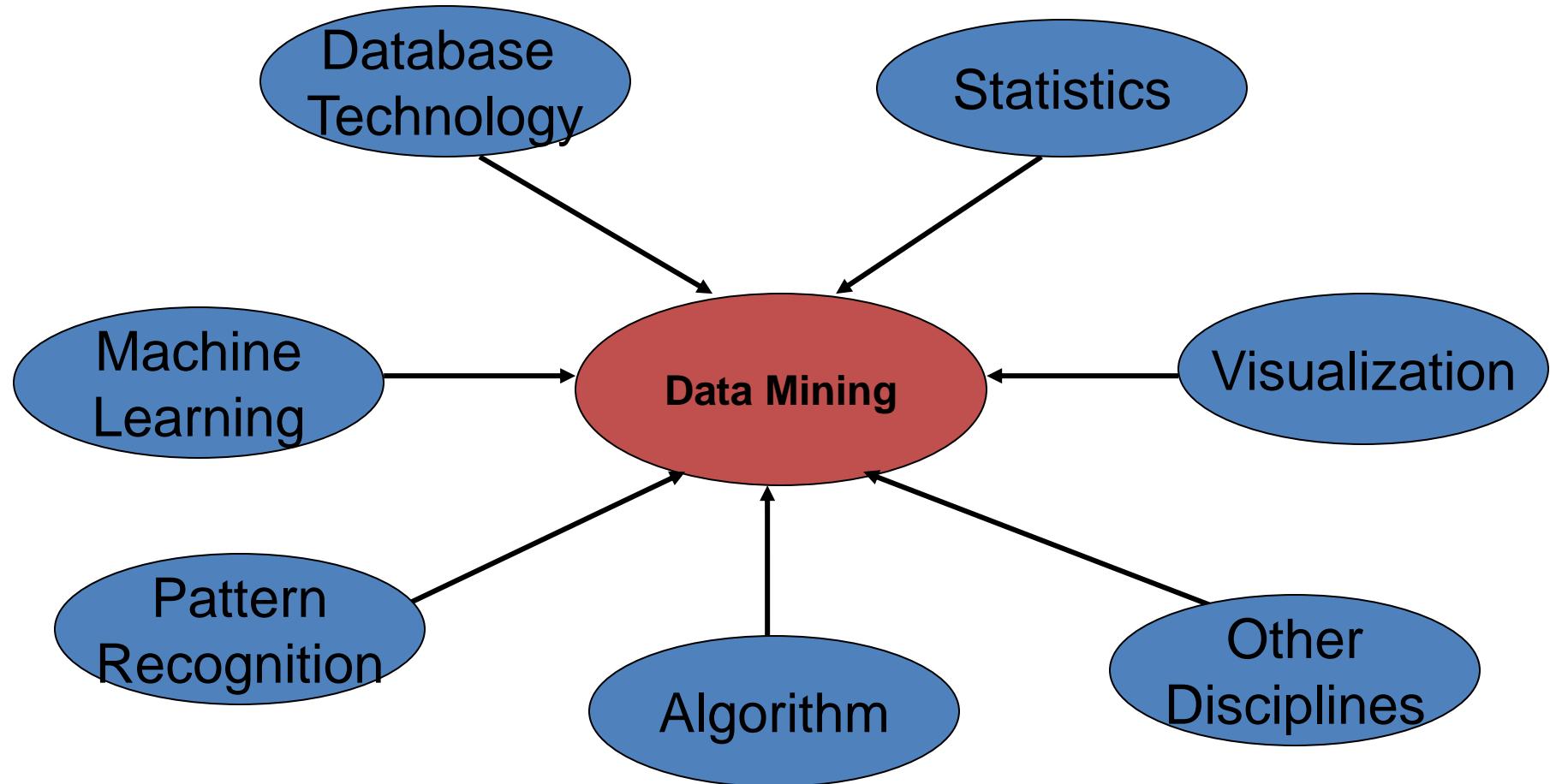
Models vs. Analytic Processing

- To a database person, data-mining is an extreme form of **analytic processing** – queries that examine large amounts of data.
 - Result is the query answer.
- To a statistician, data-mining is the inference of models.
 - Result is the parameters of the model.

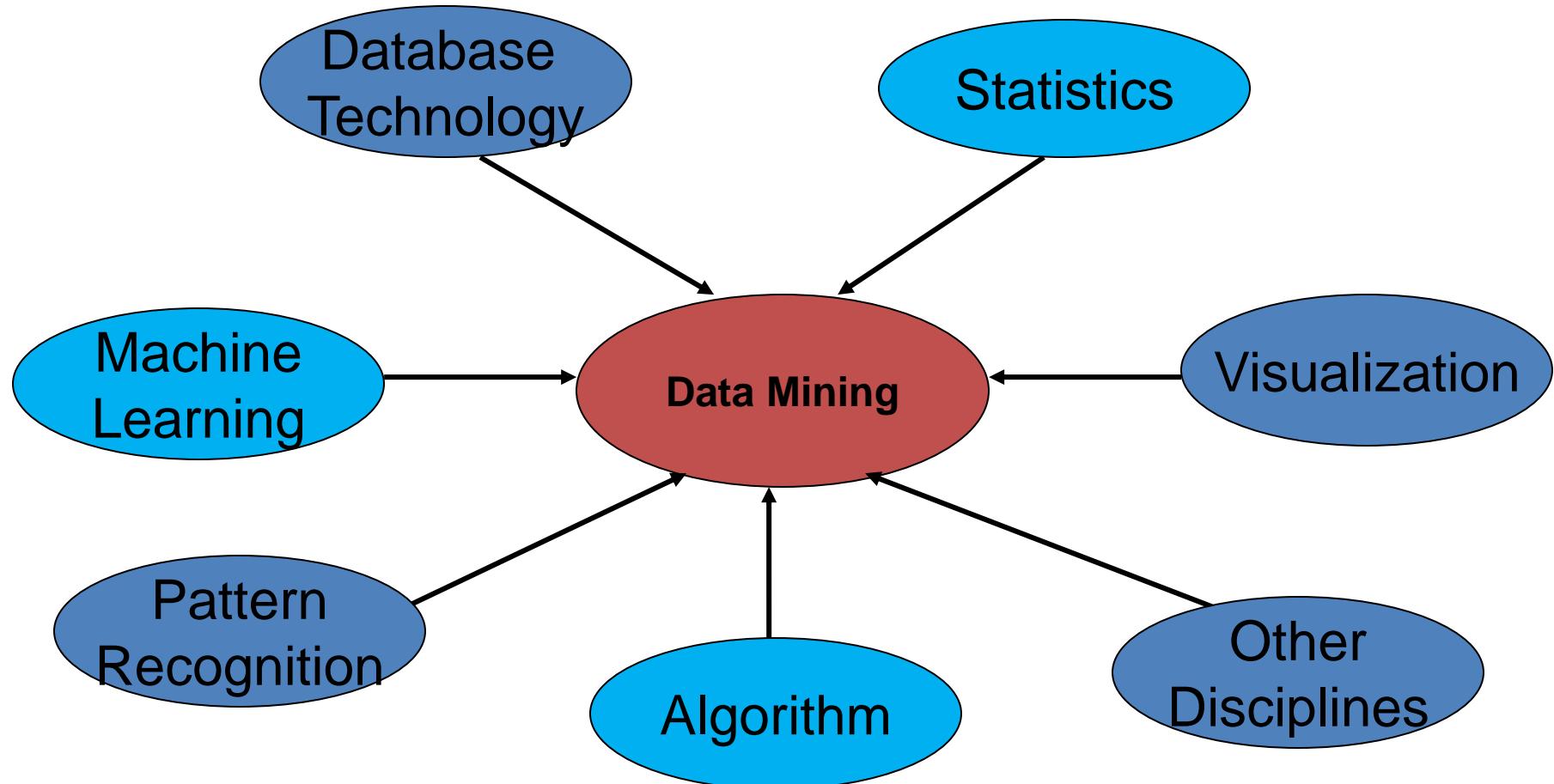
(Way too Simple) Example

- Given a billion numbers, a DB person would compute their average and standard deviation.
- A statistician might fit the billion points to the best Gaussian distribution and report the mean and standard deviation *of that distribution*.

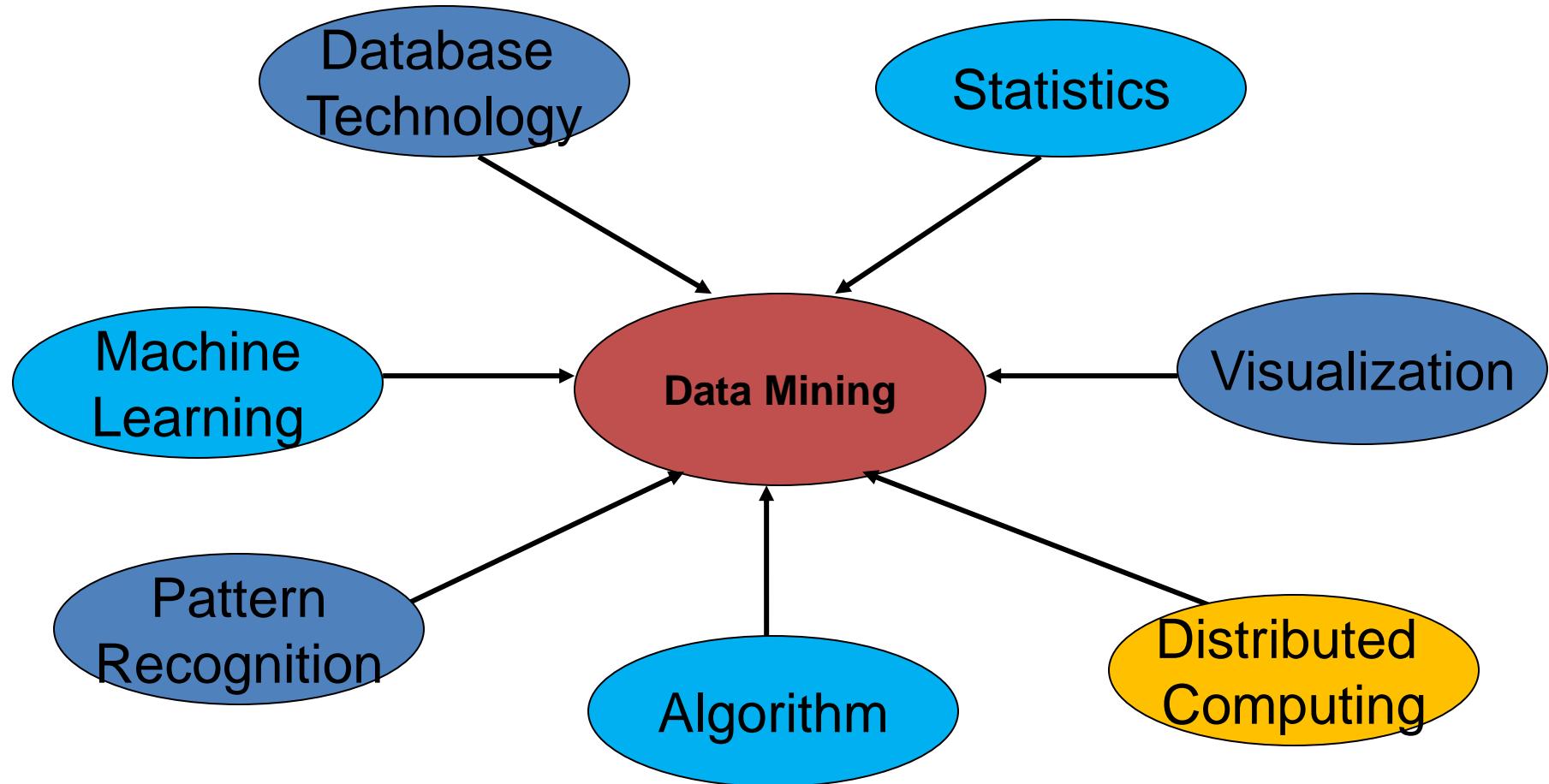
Data Mining: Confluence of Multiple Disciplines



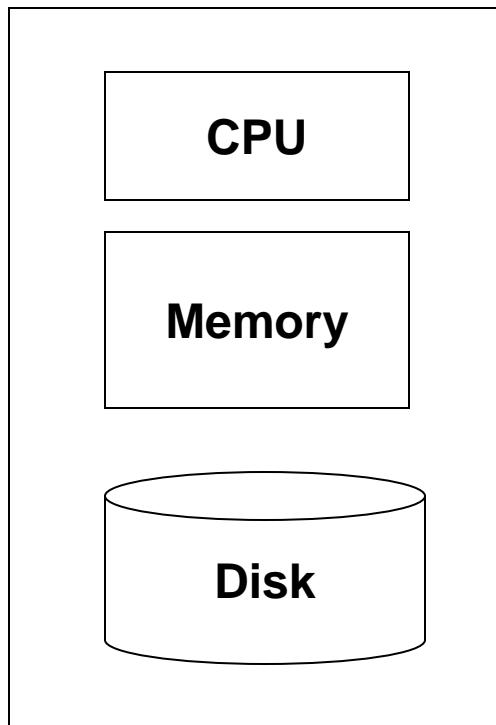
Data Mining: Confluence of Multiple Disciplines



Data Mining: Confluence of Multiple Disciplines



Single-node architecture



Machine Learning, Statistics

“Classical” Data Mining

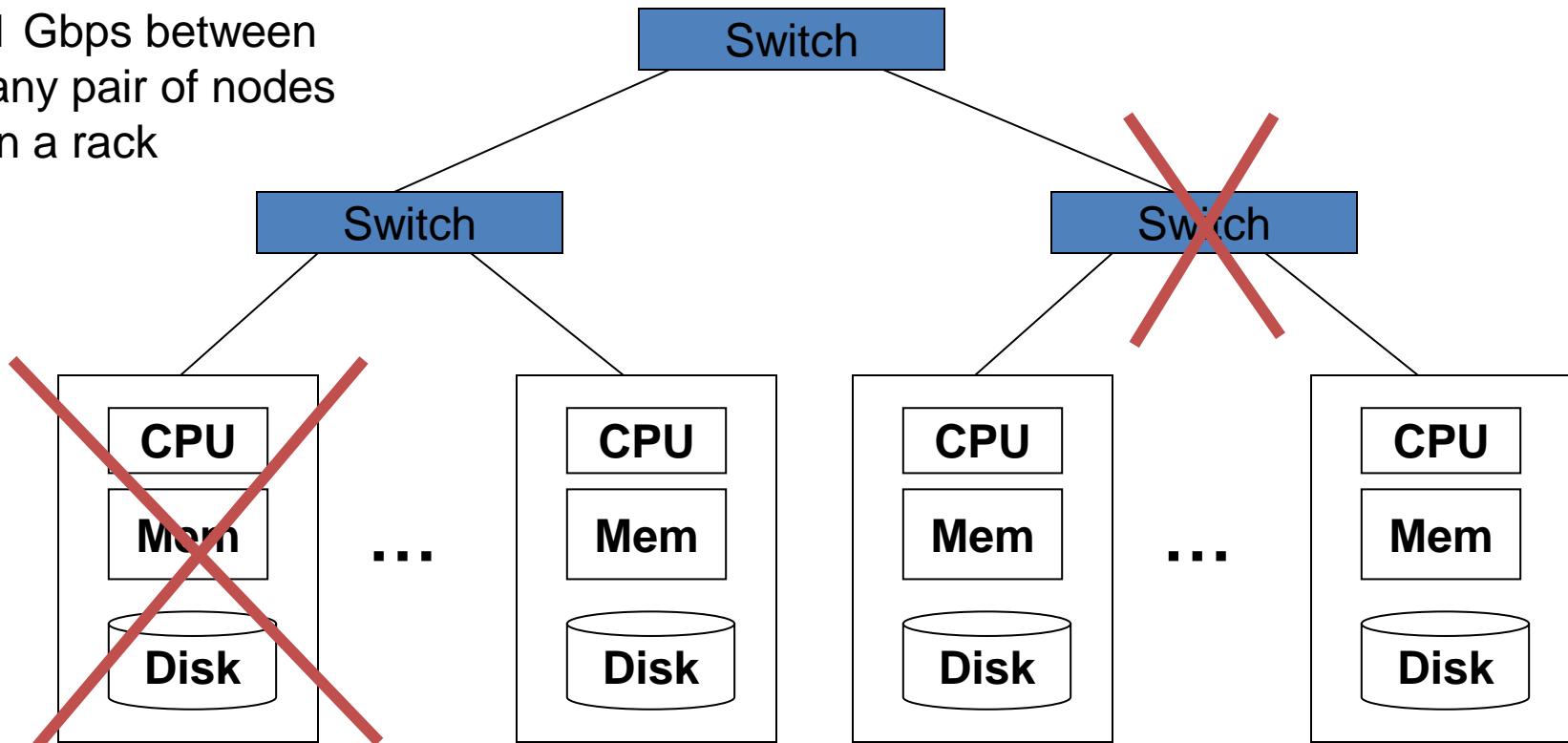
Commodity Clusters

- Web data sets can be very large
 - Tens to hundreds of terabytes
 - Cannot mine on a single server
- Standard architecture emerging:
 - Cluster of commodity Linux nodes, Gigabit ethernet interconnect
 - Google GFS; Hadoop HDFS; Kosmix KFS
- Typical usage pattern
 - Huge files (100s of GB to TB)
 - Data is rarely updated in place
 - Reads and appends are common
- How to organize computations on this architecture?
 - **Map-Reduce** paradigm

Cluster Architecture

1 Gbps between
any pair of nodes
in a rack

2-10 Gbps backbone between racks



Each rack contains 16-64 nodes

Map-Reduce paradigm

- Map the data into key-value pairs
 - E.g., map a document to word-count pairs
- Group by key
 - Group all pairs of the same word, with lists of counts
- Reduce by aggregating
 - E.g. sum all the counts to produce the total count.

The data analysis pipeline

- Mining is not the only step in the analysis process



- Preprocessing:** real data is noisy, incomplete and inconsistent. **Data cleaning** is required to make sense of the data
 - Techniques: Sampling, Dimensionality Reduction, Feature selection.
 - A dirty work, but it is often the most important step for the analysis.
- Post-Processing:** Make the data actionable and useful to the user
 - Statistical analysis of importance
 - Visualization.
- Pre- and Post-processing are often data mining tasks as well

Data Quality

- Examples of data quality problems:
 - Noise and outliers
 - missing values
 - duplicate data

Sampling

- Sampling is the main technique employed for data selection.
 - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because **obtaining** the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because **processing** the entire set of data of interest is too expensive or time consuming.

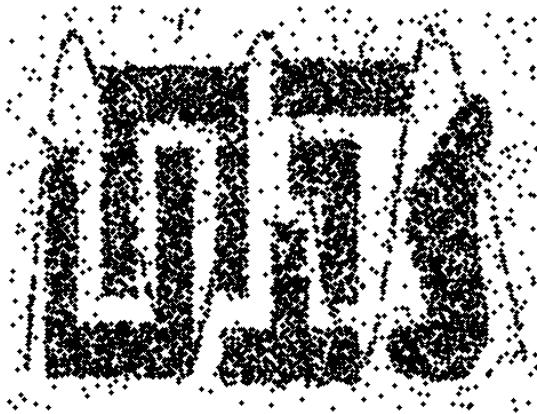
Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is representative
 - A sample is representative if it has approximately the same property (of interest) as the original set of data

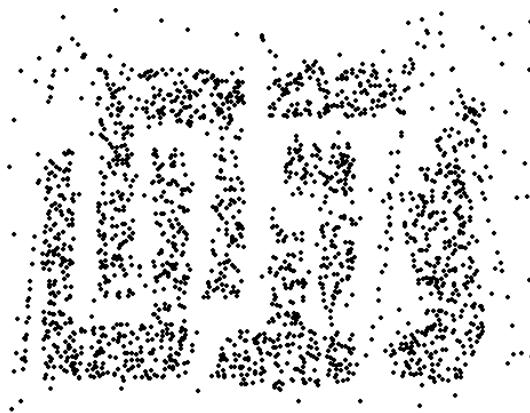
Types of Sampling

- Simple Random Sampling
 - There is an equal probability of selecting any particular item
- Sampling without replacement
 - As each item is selected, it is removed from the population
- Sampling with replacement
 - Objects are not removed from the population as they are selected for the sample.
 - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
 - Split the data into several partitions; then draw random samples from each partition

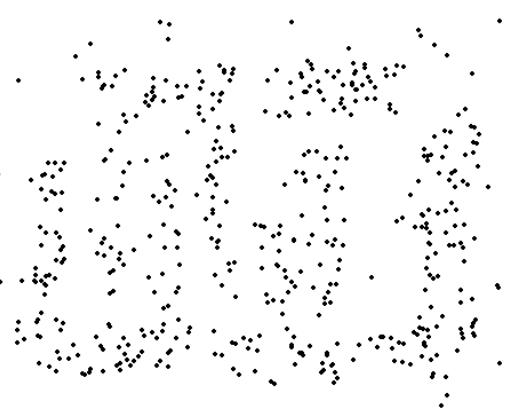
Sample Size



8000 points



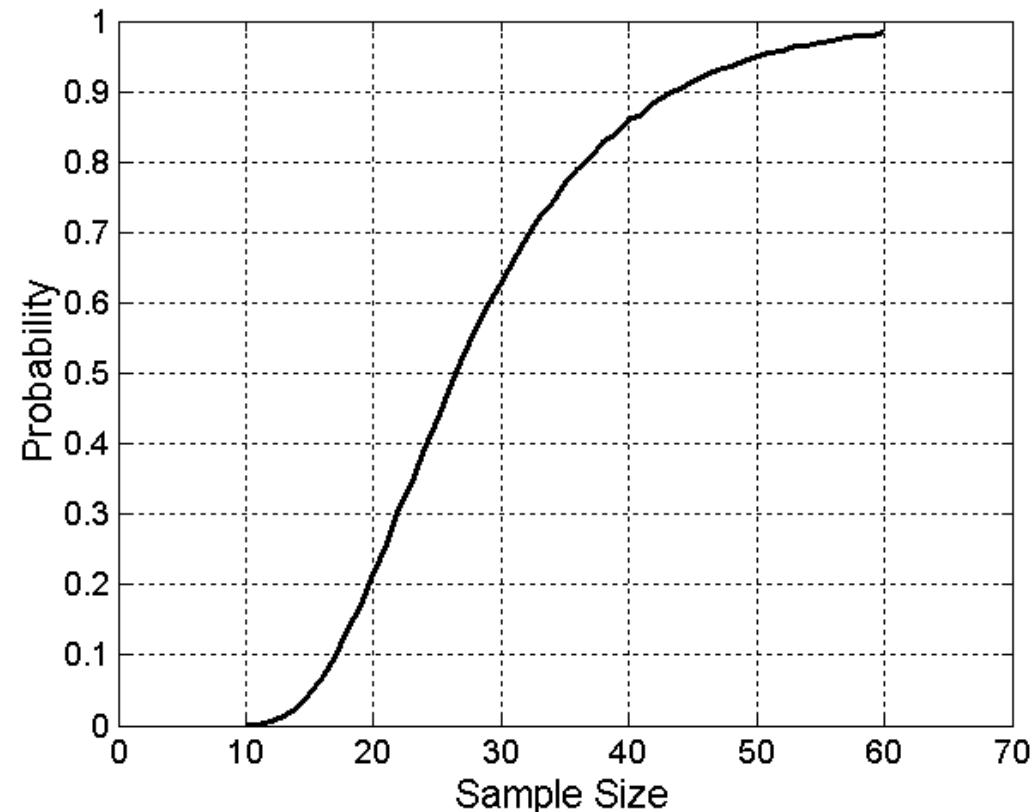
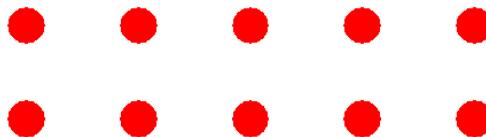
2000 Points



500 Points

Sample Size

- What sample size is necessary to get at least one object from each of 10 groups.



A data mining challenge

- You are reading a stream of integers, and you want to sample one integer uniformly at random but you do not know the size (N) of the stream in advance. You can only keep a constant amount of integers in memory
- How do you sample?
 - Hint: the last integer in the stream should have probability $1/N$ to be selected.
- Reservoir Sampling:
 - Standard interview question

Meaningfulness of Answers

- A big data-mining risk is that you will “discover” patterns that are meaningless.
- Statisticians call it **Bonferroni’s principle**: (roughly) if you look in more places for interesting patterns than your amount of data will support, you are bound to find crap.
- The **Rhine Paradox**: a great example of how not to conduct scientific research.

Rhine Paradox – (1)

- Joseph Rhine was a parapsychologist in the 1950's who hypothesized that some people had Extra-Sensory Perception.
- He devised (something like) an experiment where subjects were asked to guess 10 hidden cards – red or blue.
- He discovered that almost 1 in 1000 had ESP – they were able to get all 10 right!

Rhine Paradox – (2)

- He told these people they had ESP and called them in for another test of the same type.
- Alas, he discovered that almost all of them had lost their ESP.
- What did he conclude?
 - Answer on next slide.

Rhine Paradox – (3)

- He concluded that you shouldn't tell people they have ESP; it causes them to lose it.

DATA MINING

LECTURE 2

Data Preprocessing
Exploratory Analysis
Post-processing



What is Data Mining?

- Data mining is the use of **efficient** techniques for the analysis of **very large** collections of data and the extraction of **useful** and possibly **unexpected** patterns in data.
- “Data mining is the analysis of (often large) observational data sets to find **unsuspected relationships** and to **summarize** the data in novel ways that are both **understandable** and **useful** to the data analyst” (Hand, Mannila, Smyth)
- “Data mining is the discovery of **models** for data” (Rajaraman, Ullman)
 - We can have the following types of models
 - Models that **explain** the data (e.g., a single function)
 - Models that **predict** the future data instances.
 - Models that **summarize** the data
 - Models that ~~extract~~ the most prominent **features** of the data.

Why do we need data mining?

- Really **huge** amounts of **complex** data generated from multiple sources and **interconnected** in different ways
 - **Scientific** data from different disciplines
 - Weather, astronomy, physics, biological microarrays, genomics
 - Huge **text** collections
 - The Web, scientific articles, news, tweets, facebook postings.
 - **Transaction** data
 - Retail store records, credit card records
 - **Behavioral** data
 - Mobile phone data, query logs, browsing behavior, ad clicks
 - **Networked** data
 - The Web, Social Networks, IM networks, email network, biological networks.
 - All these types of data can be **combined** in many ways
 - Facebook has a network, text, images, user behavior, ad transactions.
- We need to **analyze** this data to **extract knowledge**
 - Knowledge can be used for **commercial** or **scientific** purposes.
 - Our solutions should **scale** to the size of the data

The data analysis pipeline

- Mining is not the only step in the analysis process



- **Preprocessing:** real data is noisy, incomplete and inconsistent. **Data cleaning** is required to make sense of the data
 - Techniques: Sampling, Dimensionality Reduction, Feature selection.
 - A dirty work, but it is often the most important step for the analysis.
- **Post-Processing:** Make the data actionable and useful to the user
 - Statistical analysis of importance
 - Visualization.
- Pre- and Post-processing are often data mining tasks as well

Data Quality

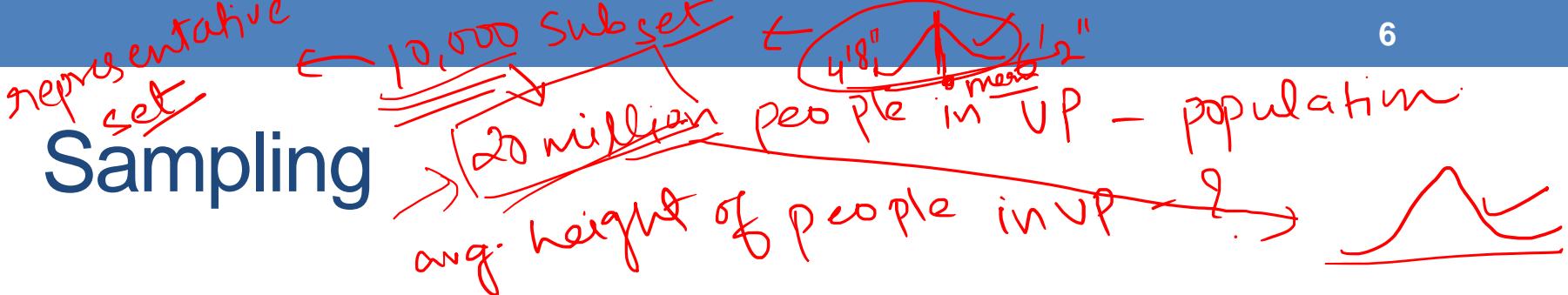
- Examples of data quality problems:
 - Noise and outliers
 - Missing values
 - Duplicate data

A mistake or a millionaire?

Missing values

Inconsistent duplicate entries

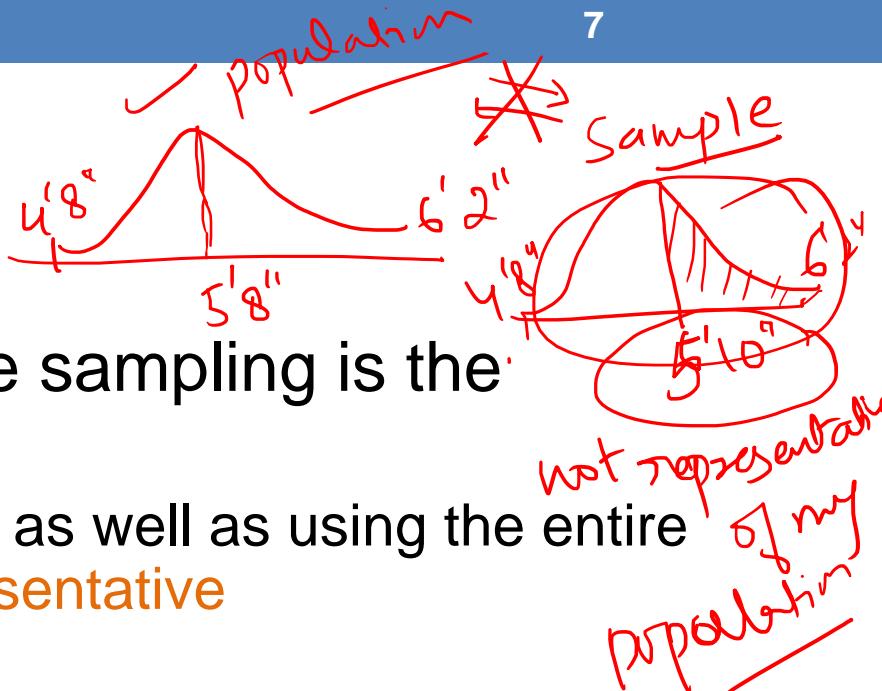
<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat	
1	Yes	Single	125K	No	
2	No	Married	100K	No	
3	No	Single	70K	No	
4	Yes	Married	120K	No	
5	No	Divorced	10000K	Yes	
6	No	NULL	60K	No	
7	Yes	Divorced	220K	NULL	
8	No	Single	85K	Yes	
9	No	Married	90K	No	
9	No	Single	90K	No	



- Sampling is the main technique employed for data selection.
 - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
 - Example: What is the average height of a person in Ioannina? UP-
We cannot measure the height of everybody
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.
 - Example: We have 1M documents. What fraction has at least 100 words in common?
 - Computing number of common words for all pairs requires 10^{12} comparisons
 - Example: What fraction of tweets in a year contain the word "Greece"?
 - 300M tweets per day, if 100 characters on average, 86.5TB to store all tweets

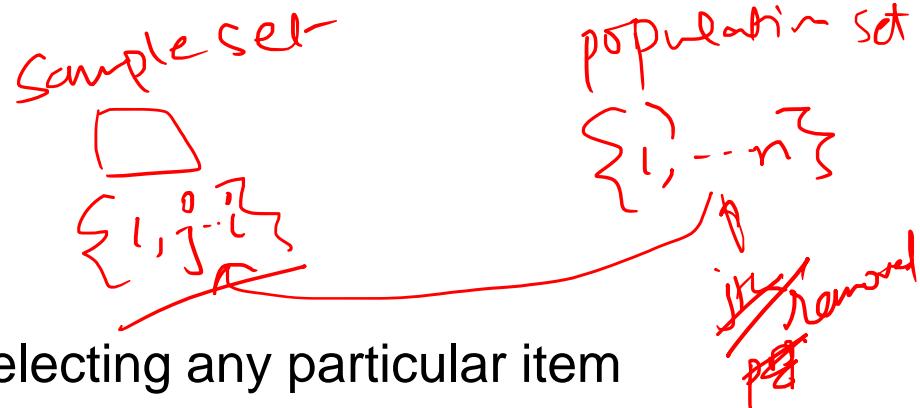
Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is **representative**
 - A sample is representative if it has approximately the same property (of interest) as the original set of data
 - Otherwise we say that the sample introduces some **bias**
 - What happens if we take a sample from the university campus to compute the average height of a person at Ioannina?
- ~~Ioannina~~



Types of Sampling

- Simple Random Sampling
 - There is an equal probability of selecting any particular item



- Sampling **without replacement**
 - As each item is selected, it is removed from the population

- Sampling **with replacement**
 - Objects are not removed from the population as they are selected for the sample.

- In sampling with replacement, the same object can be picked up more than once. This makes analytical computation of probabilities easier
- E.g., we have 100 people, 51 are women $P(W) = 0.51$, 49 men $P(M) = 0.49$. If I pick two persons what is the probability $P(W,W)$ that both are women?

• Sampling with replacement: $P(W,W) = 0.51^2$

$$P(W) * P(W)$$

• Sampling without replacement: $P(W,W) = \underline{51/100} * \underline{50/99}$

$$\underline{51} - 99$$

Types of Sampling

- **Stratified sampling**

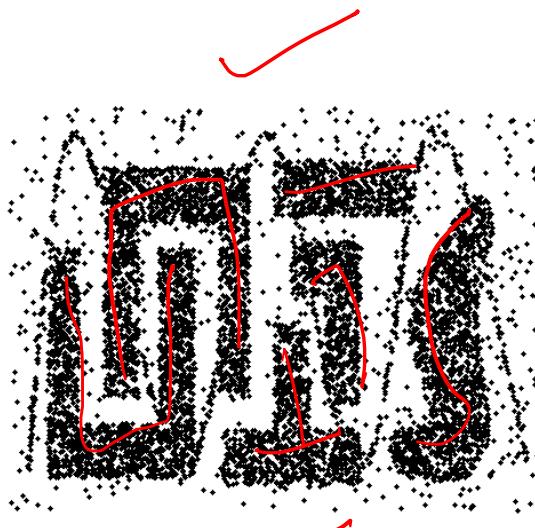
- Split the data into several **groups**; then draw random samples from each group.
 - Ensures that both groups are represented.
- **Example 1.** I want to understand the differences between legitimate and fraudulent credit card transactions. **0.1%** of transactions are fraudulent. What happens if I select **1000** transactions at random?
 - I get **1** fraudulent transaction (in expectation). Not enough to draw any conclusions.
Solution: sample **1000** legitimate and **1000** fraudulent transactions

Probability Reminder: If an event has probability p of happening and I do N trials, the expected number of times the event occurs is pN

- **Example 2.** I want to answer the question: Do web pages that are linked have on average more words in common than those that are not? I have **1M** pages, and **1M** links, what happens if I select **10K** pairs of pages at random?
 - Most likely I will not get any links. Solution: sample **10K** random pairs, and **10K** links



Sample Size



8000 points



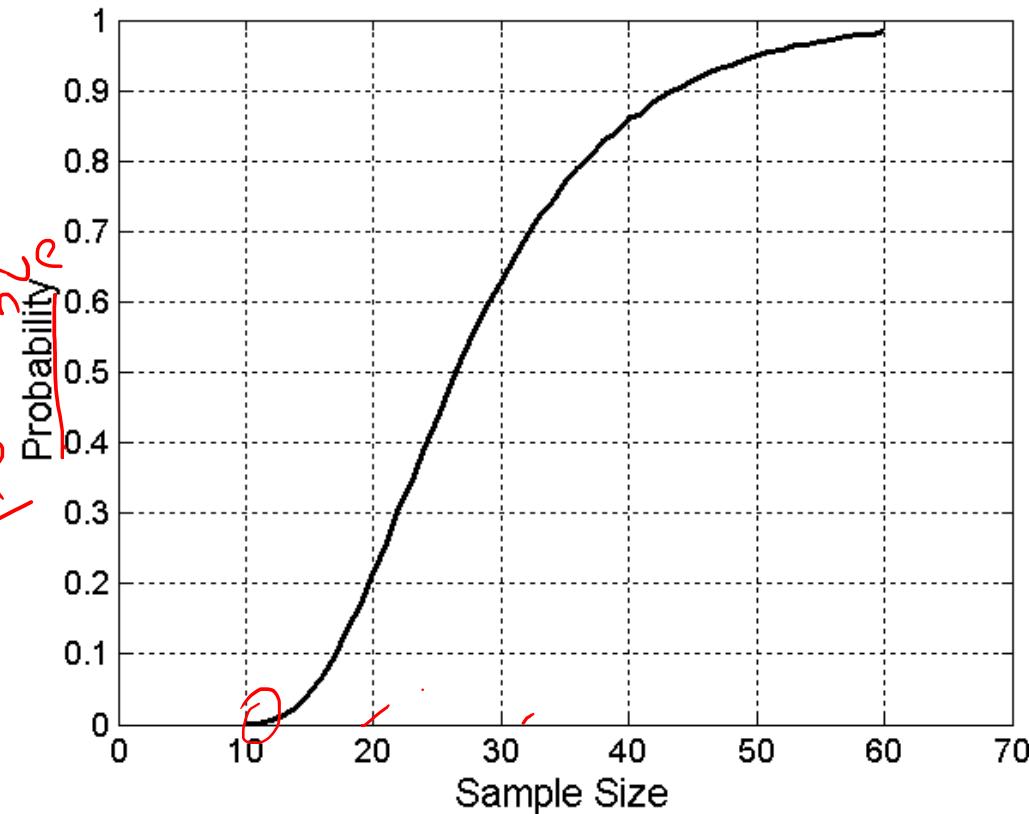
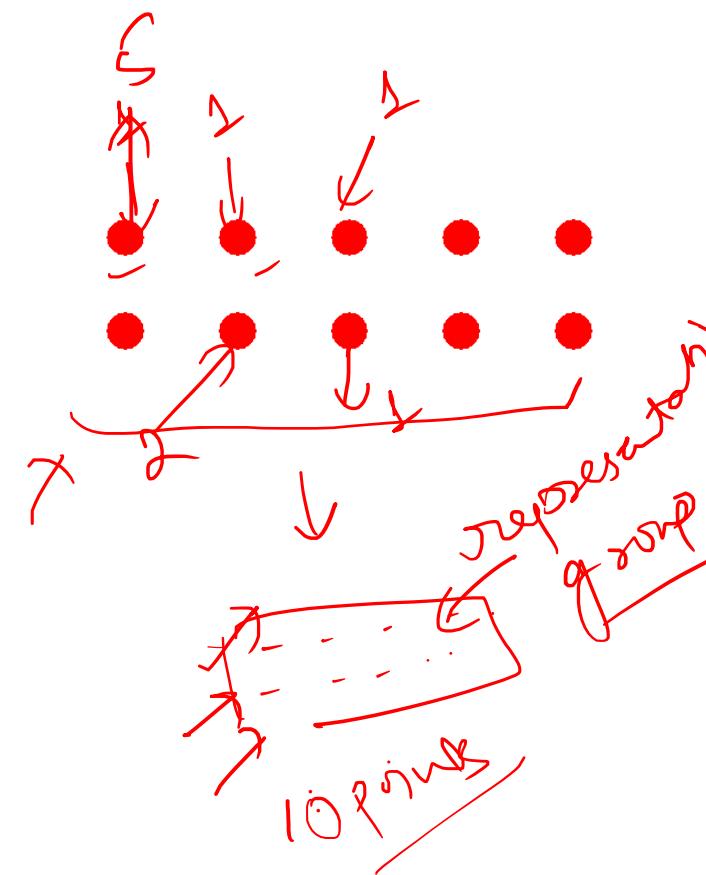
2000 Points



500 Points

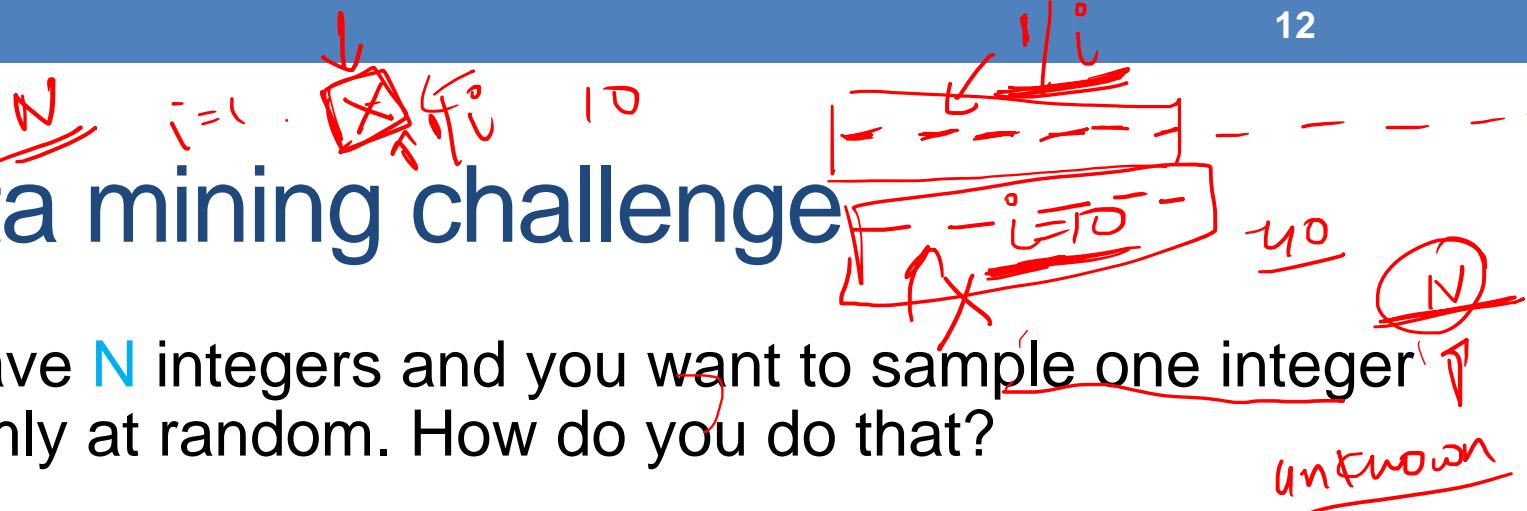
Sample Size

- What sample size is necessary to get at least one object from each of 10 groups.



A data mining challenge

- You have N integers and you want to sample one integer uniformly at random. How do you do that?
- The integers are coming in a stream: you do not know the size of the stream in advance, and there is not enough memory to store the stream in memory. You can only keep a constant amount of integers in memory
- How do you sample?
 - Hint: if the stream ends after reading n integers the last integer in the stream should have probability $1/n$ to be selected.
- Reservoir Sampling:
 - Standard interview question for many companies



~~1/i~~

~~N = 50~~

~~1/n~~

Reservoir sampling

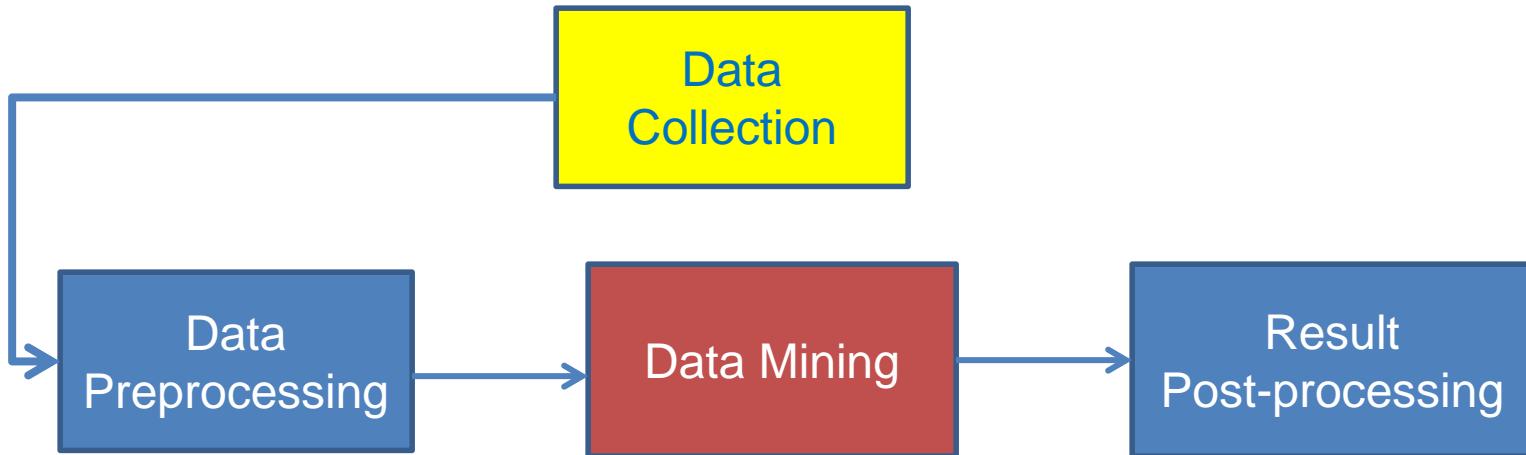
- Algorithm: With probability $1/n$ select the n -th item of the stream and replace the previous choice.
- Claim: Every item has probability $1/N$ to be selected after N items have been read.
- Proof
 - What is the probability of the n -th item to be selected?
 - $\frac{1}{n}$
 - What is the probability of the n -th items to survive for $N-n$ rounds?
 - $\left(1 - \frac{1}{n+1}\right) \left(1 - \frac{1}{n+2}\right) \dots \left(1 - \frac{1}{N}\right)$

A (detailed) data preprocessing example

- Suppose we want to mine the comments/reviews of people on [Yelp](#) and [Foursquare](#).



Data Collection



- Today there is an abundance of data online
 - Facebook, Twitter, Wikipedia, Web, etc...
- We can extract interesting information from this data, but first we need to collect it
 - Customized crawlers, use of public APIs
 - Additional cleaning/processing to parse out the useful parts
 - Respect of crawling etiquette

Mining Task

- Collect all reviews for the top-10 most reviewed restaurants in NY in Yelp
 - (thanks to Hady Law)
- Find few terms that best describe the restaurants.
- Algorithm?

Example data

- I heard so many good things about this place so I was pretty juiced to try it. I'm from Cali and I heard Shake Shack is comparable to IN-N-OUT and I gotta say, Shake Shack wins hands down. Surprisingly, the line was short and we waited about 10 MIN. to order. I ordered a regular cheeseburger, fries and a black/white shake. So yummerz. I love the location too! It's in the middle of the city and the view is breathtaking. Definitely one of my favorite places to eat in NYC.
- I'm from California and I must say, Shake Shack is better than IN-N-OUT, all day, err'day.
- Would I pay \$15+ for a burger here? No. But for the price point they are asking for, this is a definite bang for your buck (though for some, the opportunity cost of waiting in line might outweigh the cost savings) Thankfully, I came in before the lunch swarm descended and I ordered a shake shack (the special burger with the patty + fried cheese & portabella topping) and a coffee milk shake. The beef patty was very juicy and snugly packed within a soft potato roll. On the downside, I could do without the fried portabella-thingy, as the crispy taste conflicted with the juicy, tender burger. How does shake shack compare with in-and-out or 5-guys? I say a very close tie, and I think it comes down to personal affiliations. On the shake side, true to its name, the shake was well churned and very thick and luscious. The coffee flavor added a tangy taste and complemented the vanilla shake well. Situated in an open space in NYC, the open air sitting allows you to munch on your burger while watching people zoom by around the city. It's an oddly calming experience, or perhaps it was the food coma I was slowly falling into. Great place with food at a great price.

First cut

- Do simple processing to “normalize” the data (remove punctuation, make into lower case, clear white spaces, other?)
- Break into words, keep the most popular words

the 27514	the 16710	the 16010	the 14241
and 14508	and 9139	and 9504	and 8237
i 13088	a 8583	i 7966	a 8182
a 12152	i 8415	to 6524	i 7001
to 10672	to 7003	a 6370	to 6727
of 8702	in 5363	it 5169	of 4874
ramen 8518	it 4606	of 5159	you 4515
was 8274	of 4365	is 4519	it 4308
is 6835	is 4340	sauce 4020	is 4016
it 6802	burger 432	in 3951	was 3791
in 6402	was 4070	this 3519	pastrami 3748
for 6145	for 3441	was 3453	in 3508
but 5254	but 3284	for 3327	for 3424
that 4540	shack 3278	you 3220	sandwich 2928
you 4366	shake 3172	that 2769	that 2728
with 4181	that 3005	but 2590	but 2715
pork 4115	you 2985	food 2497	on 2247
my 3841	my 2514	on 2350	this 2099
this 3487	line 2389	my 2311	my 2064
wait 3184	this 2242	cart 2236	with 2040
not 3016	fries 2240	chicken 2220	not 1655
we 2984	on 2204	with 2195	your 1622
at 2980	are 2142	rice 2049	so 1610
on 2922	with 2095	so 1825	have 1585

First cut

- Do simple processing to “normalize” the data (remove punctuation, make into lower case, clear white spaces, other?)
- Break into words, keep the most popular words

the 27514
and 14508
i 13088
a 12152
to 10672
of 8702
ramen 8518
was 8274
is 6835
it 6802
in 6402
for 6145
but 5254
that 4540
you 4366
with 4181
pork 4115
my 3841
this 3487
wait 3184
not 3016
we 2984
at 2980
on 2922

the 16710
and 9139
a 8583
i 8415
to 7003
in 5363
it 4606
of 4365
is 4340
burger 432
was 4070
for 3441
but 3284
shack 3278
shake 3172
that 3005
you 2985
my 2514
line 2389
this 2242
fries 2240
on 2204
are 2142
with 2095

the 16010
and 9504
i 7966
to 6524
a 6370
it 5169
of 5159
is 4519
sauce 4020
in 3951
this 3519
was 3453
for 3327
you 3220
that 2769
but 2590
food 2497

the 14241
and 8237
a 8182
i 7001
to 6727
of 4874
you 4515
it 4308
is 4016
was 3791
pastrami 3748
in 3508
for 3424
sandwich 2928
that 2728
but 2715
on 2247

Most frequent words are **stop words**

cart 2236
chicken 2220
with 2195
rice 2049
so 1825

not 1655
your 1622
so 1610
have 1585

Second cut

- Remove stop words
 - Stop-word lists can be found online.

a, about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, before, being, below, between, both, but, by, can't, cannot, could, could n't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, he re, here's, hers, herself, him, himself, his, how, how's, i, i'd, i'll, i'm, i've, if, in, into, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, shan't, she, she'd, she'll, she's, should, shouldn't, so, some, such, than, tha t, that's, the, their, theirs, them, themselves, then, there, there's, these, they, th ey'd, they'll, they're, they've, this, those, through, to, too, under, until, up, very , was, wasn't, we, we'd, we'll, we're, we've, were, weren't, what, what's, when, when's , where, where's, which, while, who, who's, whom, why, why's, with, won't, would, would n't, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves,

Second cut

- Remove stop words
 - Stop-word lists can be found online.

ramen 8572

pork 4152

wait 3195

good 2867

place 2361

noodles 2279

ippudo 2261

buns 2251

broth 2041

like 1902

just 1896

get 1641

time 1613

one 1460

really 1437

go 1366

food 1296

bowl 1272

can 1256

great 1172

best 1167

burger 4340

shack 3291

shake 3221

line 2397

fries 2260

good 1920

burgers 1643

wait 1508

just 1412

cheese 1307

like 1204

food 1175

get 1162

place 1159

one 1118

long 1013

go 995

time 951

park 887

can 860

best 849

sauce 4023

food 2507

cart 2239

chicken 2238

rice 2052

hot 1835

white 1782

line 1755

good 1629

lamb 1422

halal 1343

just 1338

get 1332

one 1222

like 1096

place 1052

go 965

can 878

night 832

time 794

long 792

people 790

pastrami 3782

sandwich 2934

place 1480

good 1341

get 1251

katz's 1223

just 1214

like 1207

meat 1168

one 1071

deli 984

best 965

go 961

ticket 955

food 896

sandwiches 813

can 812

beef 768

order 720

pickles 699

time 662

Second cut

- Remove stop words
 - Stop-word lists can be found online.

ramen 8572
 pork 4152
 wait 3195
 good 2867
 place 2361
 noodles 2279
 ippudo 2261
 buns 2251
 broth 2041
like 1902
 just 1896
get 1641
 time 1613
 one 1460
 really 1437
 go 1366
 food 1296
 bowl 1272
 can 1256
 great 1172
 best 1167

burger 4340
 shack 3291
 shake 3221
 line 2397
 fries 2260
 good 1920
 burgers 1643
 wait 1508
 just 1412
 cheese 1307
like 1204
 food 1175
get 1162

sauce 4023
 food 2507
 cart 2239
 chicken 2238
 rice 2052
 hot 1835
 white 1782
 line 1755
 good 1629
 lamb 1422
 halal 1343
 just 1338
get 1332

pastrami 3782
 sandwich 2934
 place 1480
 good 1341
get 1251
 katz's 1223
 just 1214
like 1207
 meat 1168
 one 1071
 deli 984
 best 965
 go 961

Commonly used words in reviews, not so interesting

long 1013
 go 995
 time 951
 park 887
 can 860
 best 849

place 1052
 go 965
 can 878
 night 832
 time 794
 long 792
 people 790

sandwiches 813
 can 812
 beef 768
 order 720
 pickles 699
 time 662

IDF

- Important words are the ones that are unique to the document (differentiating) compared to the rest of the collection
 - All reviews use the word “like”. This is not interesting
 - We want the words that characterize the specific restaurant
- Document Frequency $DF(w)$:** fraction of documents that contain word w .

$$DF(w) = \frac{D(w)}{D}$$

$D(w)$: num of docs that contain word w
 D : total number of documents

- Inverse Document Frequency $IDF(w)$:**

$$IDF(w) = \log\left(\frac{1}{DF(w)}\right)$$

- Maximum when unique to one document : $IDF(w) = \log(D)$
- Minimum when the word is common to all documents: $IDF(w) = 0$

TF-IDF

- The words that are best for describing a document are the ones that are **important for the document**, but also **unique to the document**.
- **TF(w,d)**: term frequency of word w in document d
 - Number of times that the word appears in the document
 - Natural measure of **importance** of the word for the document
- **IDF(w)**: inverse document frequency
 - Natural measure of the **uniqueness** of the word w
- **TF-IDF(w,d) = TF(w,d) × IDF(w)**

Third cut

- Ordered by TF-IDF

ramen 3057.4176194	fries 806.08537330	lamb 985.655290756243	pastrami 1931.94250908298
akamaru 2353.24196	custard 729.607519	halal 686.038812717726	katz's 1120.62356508209
noodles 1579.68242	shakes 628.4738038	53rd 375.685771863491	rye 1004.28925735888
broth 1414.7133955	shroom 515.7790608	gyro 305.809092298788	corned 906.113544700399
miso 1252.60629058	burger 457.2646379	pita 304.984759446376	pickles 640.487221580035
hirata 709.1962086	crinkle 398.347221	cart 235.902194557873	reuben 515.779060830666
hakata 591.7643688	burgers 366.624854	platter 139.45990308004	matzo 430.583412389887
shiromaru 587.1591	madison 350.939350	chicken/lamb 135.852520	sally 428.110484707471
noodle 581.8446147	shackburger 292.42	carts 120.274374158359	harry 226.323810772916
tonkotsu 529.59457	'shroom 287.823136	hilton 84.2987473324223	mustard 216.079238853014
ippudo 504.5275695	portobello 239.806	lamb/chicken 82.8930633	cutter 209.535243462458
buns 502.296134008	custards 211.83782	yogurt 70.0078652365545	carnegie 198.655512713779
ippudo's 453.60926	concrete 195.16992	52nd 67.5963923222322	katz 194.387844446609
modern 394.8391629	bun 186.9621782983	6th 60.7930175345658	knish 184.206807439524
egg 367.3680056967	milkshakes 174.996	4am 55.4517744447956	sandwiches 181.415707218
shoyu 352.29551922	concretes 165.7861	yellow 54.4470265206673	brisket 131.945865389878
chashu 347.6903490	portabelllo 163.483	tzatziki 52.95945713886	fries 131.613054313392
karaka 336.1774235	shack's 159.334353	lettuce 51.323016802268	salami 127.621117258549
kakuni 276.3102111	patty 152.22603588	sammy's 50.656872045869	knishes 124.339595021678
ramens 262.4947006	ss 149.66803104461	sw 50.5668577816893	delicatessen 117.488967607
bun 236.5122638036	patties 148.068287	platters 49.90659700031	deli's 117.431839742696
wasabi 232.3667512	cam 105.9496067806	falafel 49.479699521204	carver 115.129254649702
dama 221.048168927	milkshake 103.9720	sober 49.2211422635451	brown's 109.441778045519
brulee 201.1797390	lamps 99.011158998	moma 48.1589121730374	matzoh 108.22149937072

Third cut

- TF-IDF takes care of stop words as well
- We do not need to remove the stopwords since they will get $IDF(w) = 0$

Decisions, decisions...

- When mining real data you often need to make some
 - What data should we collect? How much? For how long?
 - Should we throw out some data that does not seem to be useful?

An actual review

AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAA AAA

- Too frequent data (stop words), too infrequent (errors?), erroneous data, missing data, outliers
 - How should we weight the different pieces of data?
- Most decisions are application dependent. Some information may be lost but we can usually live with it (most of the times)
- We should make our decisions clear since they affect our findings.
- Dealing with real data is hard...

Exploratory analysis of data

- **Summary statistics:** numbers that summarize properties of the data
 - Summarized properties include **frequency**, **location** and **spread**
 - Examples: location - mean
spread - standard deviation
 - Most summary statistics can be calculated in a single pass through the data

Frequency and Mode

- The **frequency** of an attribute value is the percentage of time the value occurs in the data set
 - For example, given the attribute ‘gender’ and a representative population of people, the gender ‘female’ occurs about 50% of the time.
- The **mode** of a attribute is the most frequent attribute value
- The notions of frequency and mode are typically used with categorical data

Percentiles

- For continuous data, the notion of a **percentile** is more useful.

Given an ordinal or continuous attribute x and a number p between 0 and 100, the p^{th} percentile is a value x_p of x such that $p\%$ of the observed values of x are less than x_p .

- For instance, the 50th percentile is the value $x_{50\%}$ such that 50% of all values of x are less than $x_{50\%}$.

Measures of Location: Mean and Median

- The **mean** is the most common measure of the location of a set of points.
- However, the mean is very sensitive to outliers.
- Thus, the **median** or a trimmed mean is also commonly used.

$$\text{mean}(x) = \bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r + 1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

Example

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	10000K	Yes
6	No	NULL	60K	No
7	Yes	Divorced	220K	NULL
8	No	Single	85K	Yes
9	No	Married	90K	No
10	No	Single	90K	No

Mean: 1090K

Trimmed mean (remove min, max): 105K

Median: $(90+100)/2 = 95K$

Measures of Spread: Range and Variance

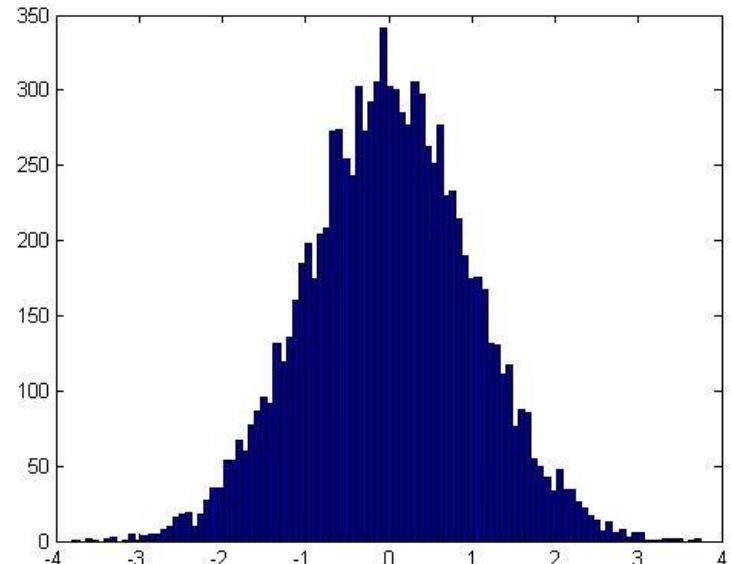
- Range is the difference between the max and min
- The variance or standard deviation is the most common measure of the spread of a set of points.

$$var(x) = \frac{1}{m} \sum_{i=1}^m (x - \bar{x})^2$$

$$\sigma(x) = \sqrt{var(x)}$$

Normal Distribution

- $\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

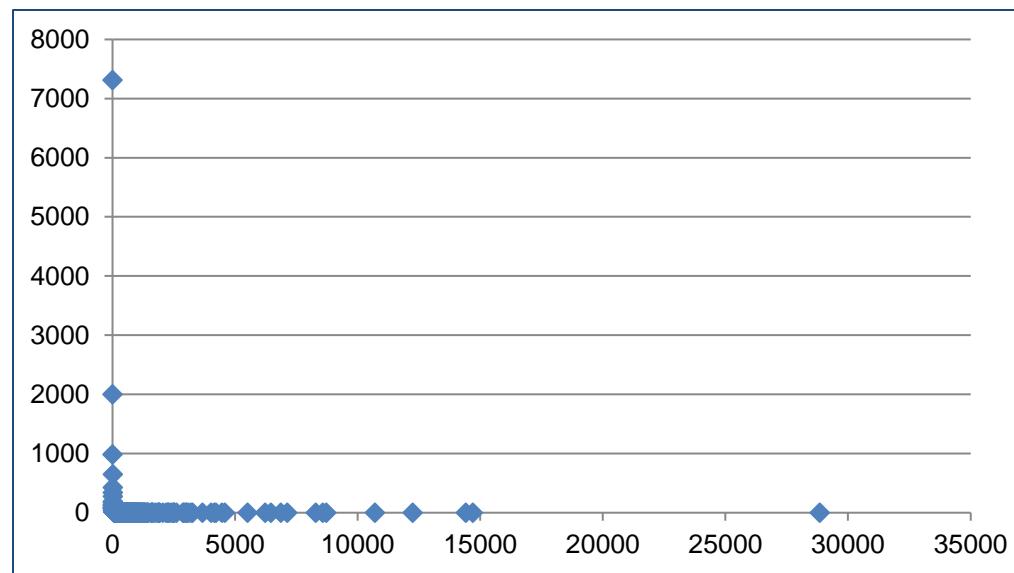


This is a value histogram

- An important distribution that characterizes many quantities and has a central role in probabilities and statistics.
 - Appears also in the central limit theorem
- Fully characterized by the mean μ and standard deviation σ

Not everything is normally distributed

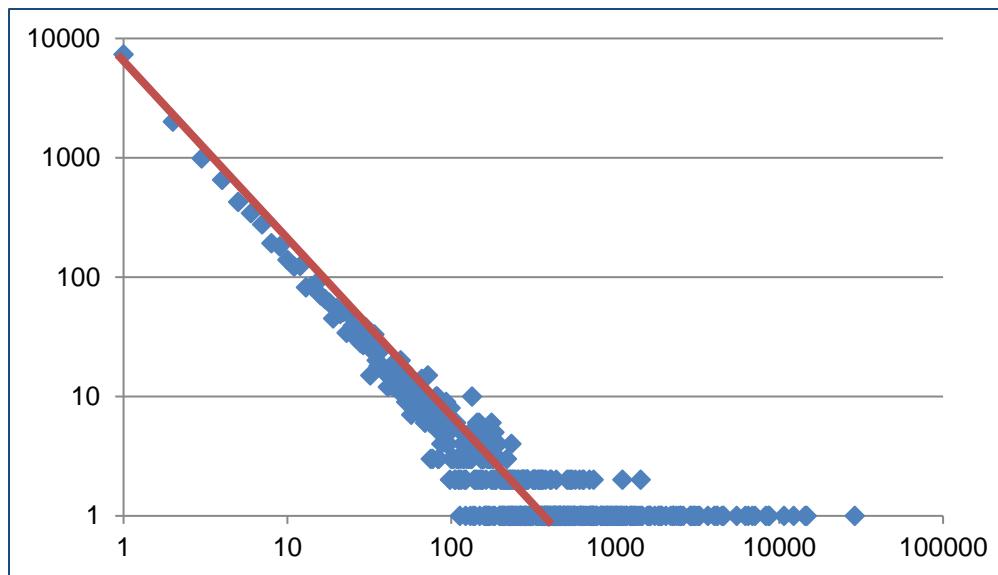
- Plot of number of words with x number of occurrences



- If this was a normal distribution we would not have a frequency as large as 28K

Power-law distribution

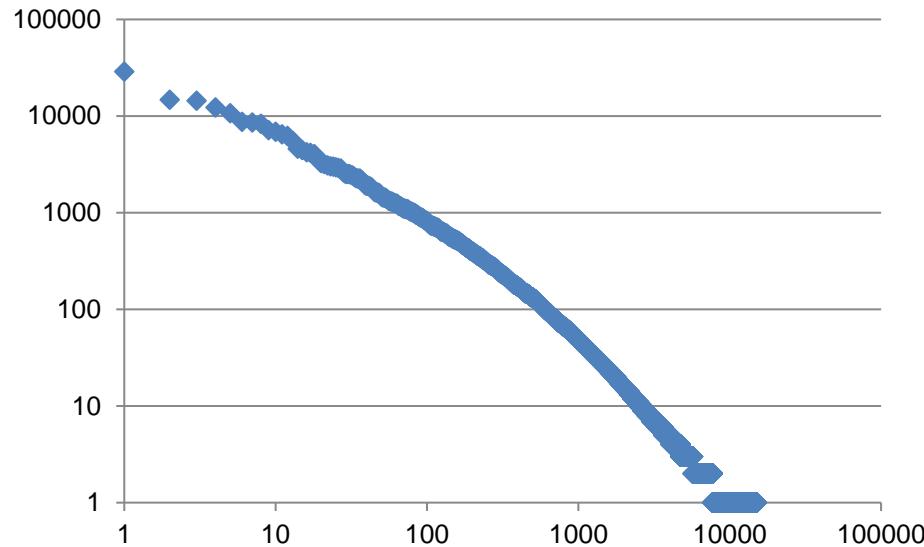
- We can understand the distribution of words if we take the **log-log** plot



- Linear relationship in the log-log space
- $$p(x = k) = k^{-\alpha}$$

Zipf's law

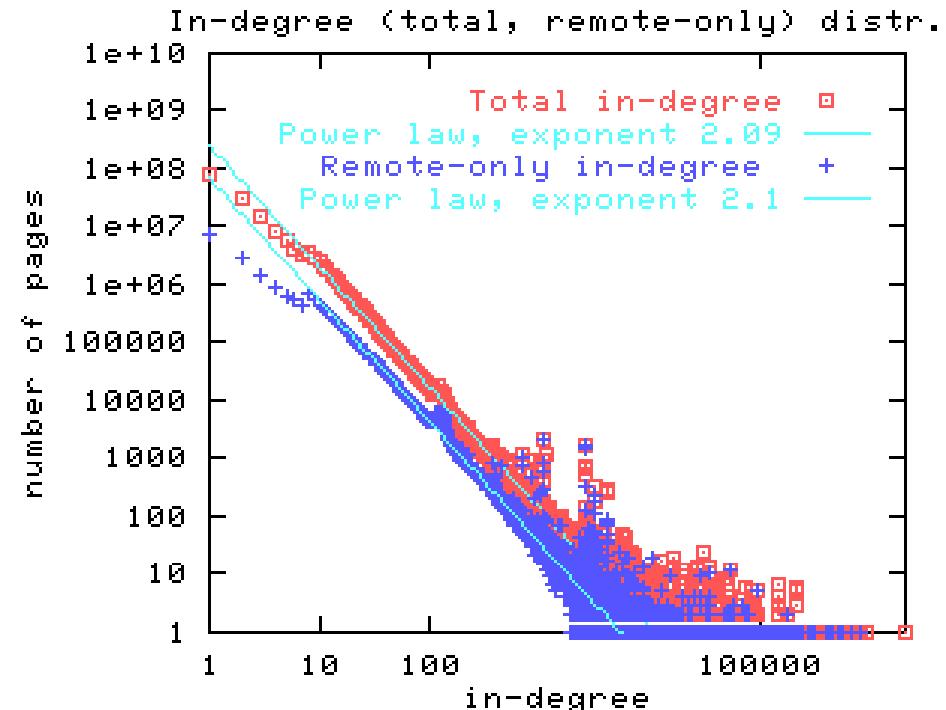
- Power laws can be detected by a linear relationship in the log-log space for the **rank-frequency** plot



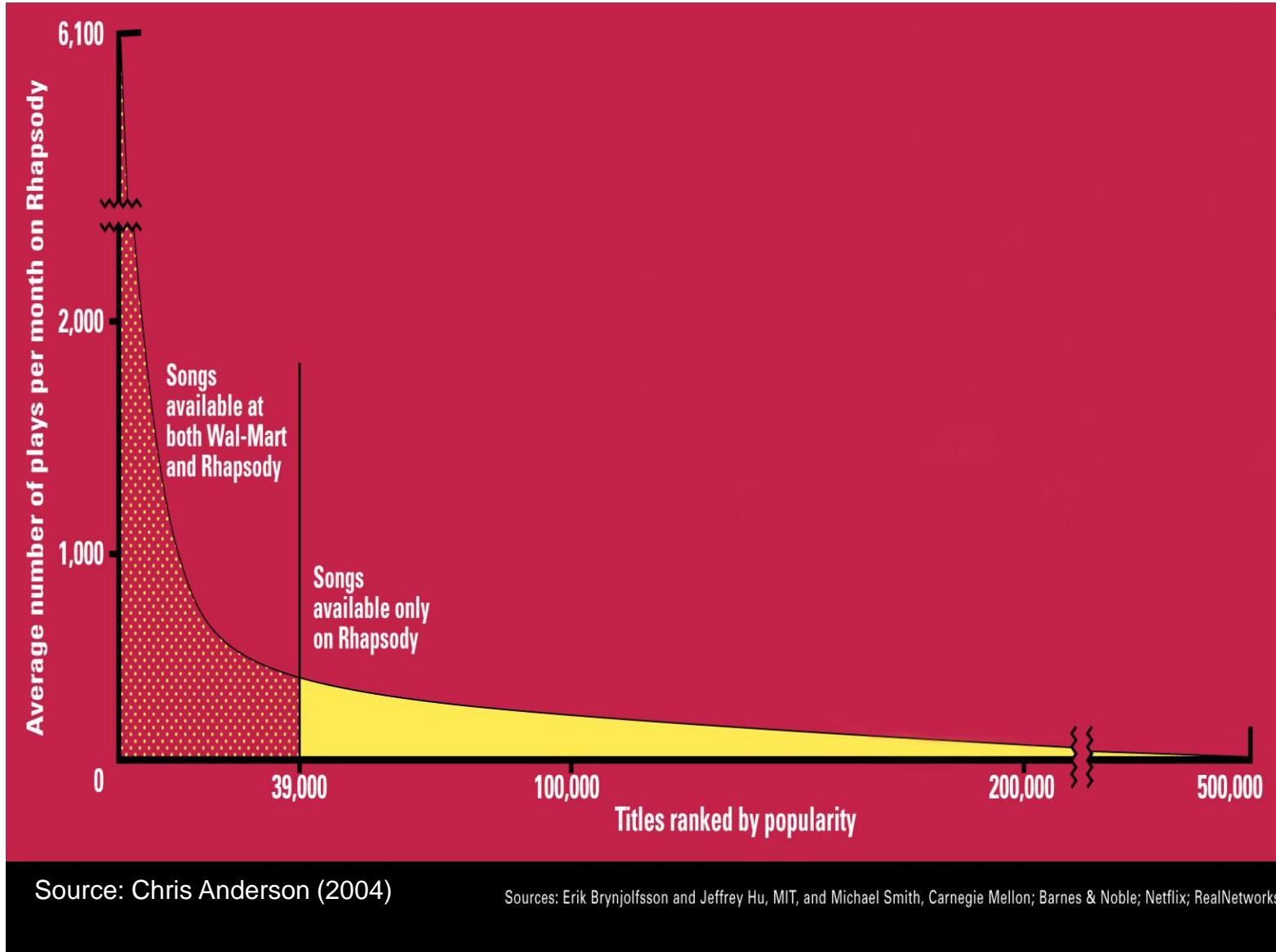
- $f(r)$: Frequency of the **r-th** most frequent word
$$f(r) = r^{-\beta}$$

Power-laws are everywhere

- Incoming and outgoing links of web pages, number of friends in social networks, number of occurrences of words, file sizes, city sizes, income distribution, popularity of products and movies
 - Signature of human activity?
 - A mechanism that explains everything?
 - Rich get richer process



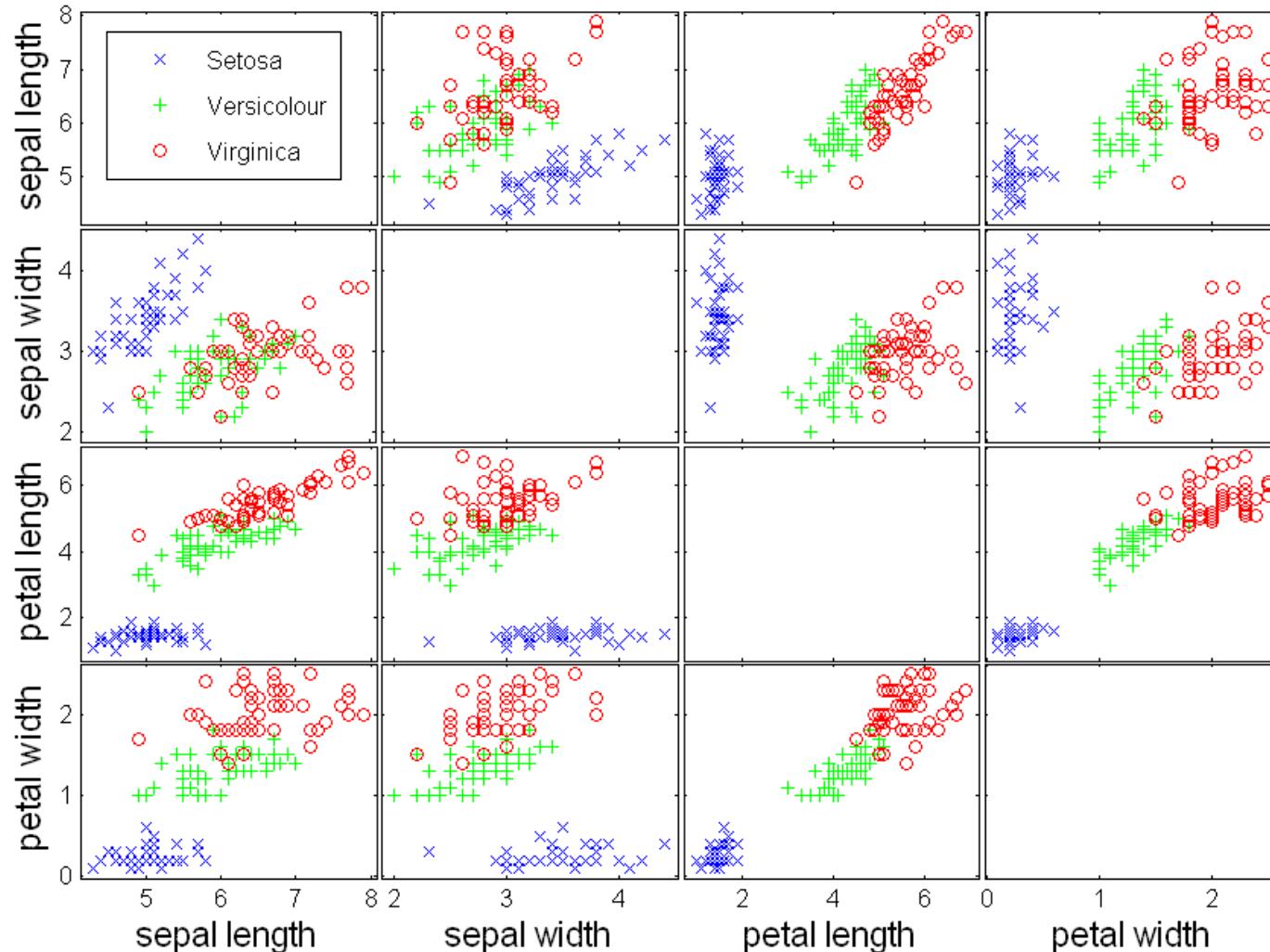
The Long Tail



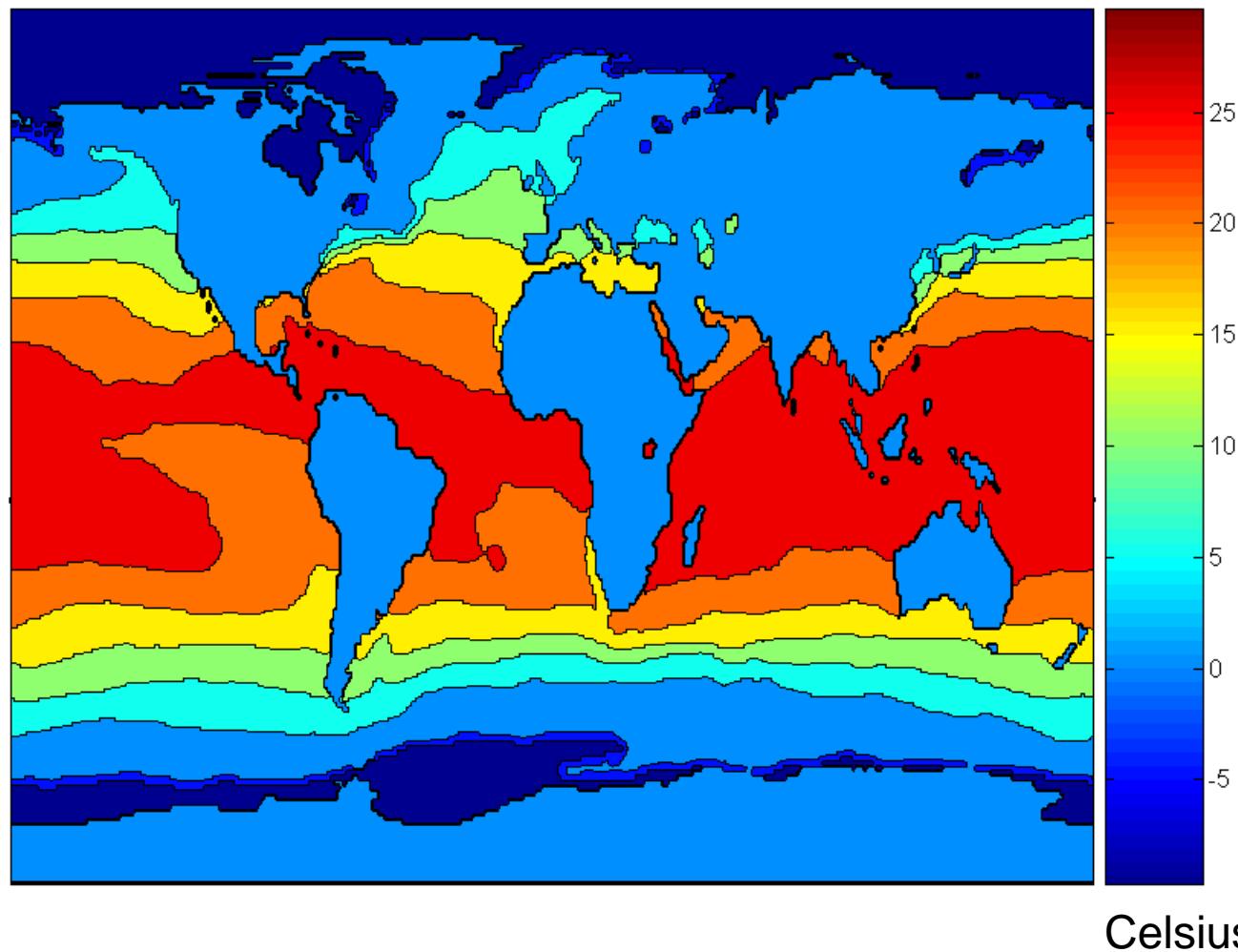
Post-processing

- Visualization
 - The human eye is a powerful analytical tool
 - If we visualize the data properly, we can discover patterns
 - Visualization is the way to present the data so that patterns can be seen
 - E.g., histograms and plots are a form of visualization
 - There are multiple techniques (a field on its own)

Scatter Plot Array of Iris Attributes



Contour Plot Example: SST Dec, 1998



Meaningfulness of Answers

- A big data-mining risk is that you will “discover” patterns that are meaningless.
- Statisticians call it **Bonferroni’s principle**: (roughly) if you look in more places for interesting patterns than your amount of data will support, you are bound to find crap.
- The **Rhine Paradox**: a great example of how not to conduct scientific research.

Rhine Paradox – (1)

- Joseph Rhine was a parapsychologist in the 1950's who hypothesized that some people had Extra-Sensory Perception.
- He devised (something like) an experiment where subjects were asked to guess 10 hidden cards – red or blue.
- He discovered that almost 1 in 1000 had ESP – they were able to get all 10 right!

Rhine Paradox – (2)

- He told these people they had ESP and called them in for another test of the same type.
- Alas, he discovered that almost all of them had lost their ESP.
- What did he conclude?
 - Answer on next slide.

Rhine Paradox – (3)

- He concluded that you shouldn't tell people they have ESP; it causes them to lose it.

Data Mining: Data

What is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
 - Object is also known as record, point, case, sample, entity, or instance

Objects

Attributes

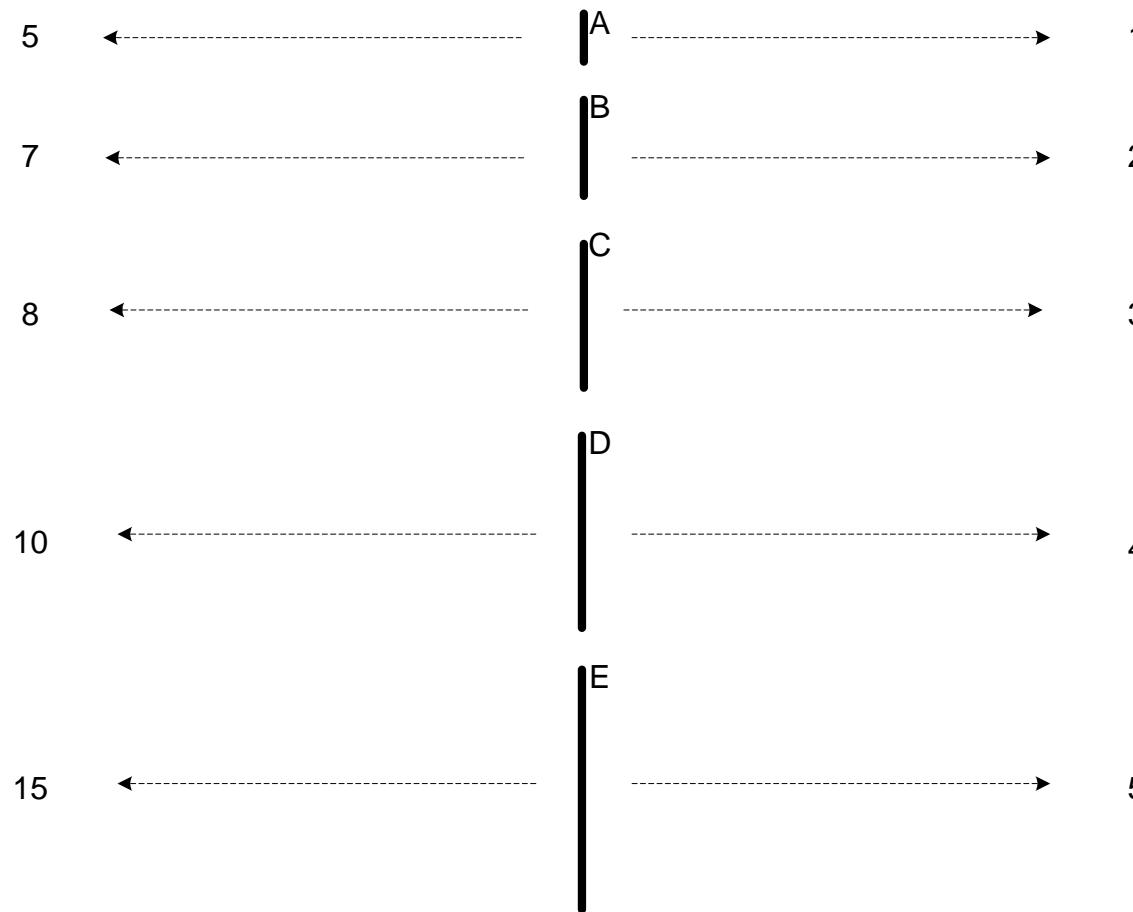
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Attribute Values

- Attribute values are numbers or symbols assigned to an attribute
- Distinction between attributes and attribute values
 - Same attribute can be mapped to different attribute values
 - Example: height can be measured in feet or meters
 - Different attributes can be mapped to the same set of values
 - Example: Attribute values for ID and age are integers
 - But properties of attribute values can be different
 - ID has no limit but age has a maximum and minimum value

Measurement of Length

- The way you measure an attribute is somewhat may not match the attributes properties.



Types of Attributes

- There are different types of attributes
 - **Nominal:** Examples: ID numbers, eye color, zip codes
 - **Ordinal:** Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
 - **Interval:** Examples: calendar dates, temperatures in Celsius or Fahrenheit.
 - **Ratio:** Examples: temperature in Kelvin, length, time, counts

Properties of Attribute Values

- The type of an attribute depends on which of the following properties it possesses:
 - Distinctness: $= \neq$
 - Order: $< >$
 - Addition: $+ -$
 - Multiplication: $* /$
 - Nominal attribute: distinctness
 - Ordinal attribute: distinctness & order
 - Interval attribute: distinctness, order & addition
 - Ratio attribute: all 4 properties

Attribute Type	Description	Examples	Operations
Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. ($=, \neq$)	zip codes, employee ID numbers, eye color, sex: $\{male, female\}$	mode, entropy, contingency correlation, χ^2 test
Ordinal	The values of an ordinal attribute provide enough information to order objects. ($<, >$)	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. ($+, -$)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
Ratio	For ratio variables, both differences and ratios are meaningful. ($*, /$)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Attribute Level	Transformation	Comments
Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
Ordinal	An order preserving change of values, i.e., $\text{new_value} = f(\text{old_value})$ where f is a monotonic function.	An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by { 0.5, 1, 10}.
Interval	$\text{new_value} = a * \text{old_value} + b$ where a and b are constants	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
Ratio	$\text{new_value} = a * \text{old_value}$	Length can be measured in meters or feet.

Discrete and Continuous Attributes

- Discrete Attribute
 - Has only a finite or countably infinite set of values
 - Examples: zip codes, counts, or the set of words in a collection of documents
 - Often represented as integer variables.
 - Note: binary attributes are a special case of discrete attributes
- Continuous Attribute
 - Has real numbers as attribute values
 - Examples: temperature, height, or weight.
 - Practically, real values can only be measured and represented using a finite number of digits.
 - Continuous attributes are typically represented as floating-point variables.

Types of data sets

- **Record**
 - Data Matrix
 - Document Data
 - Transaction Data
- **Graph**
 - World Wide Web
 - Molecular Structures
- **Ordered**
 - Spatial Data
 - Temporal Data
 - Sequential Data
 - Genetic Sequence Data

Important Characteristics of Structured Data

- **Dimensionality**
 - Curse of Dimensionality
- **Sparsity**
 - Only presence counts
- **Resolution**
 - Patterns depend on the scale

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Data Matrix

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute
- Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Document Data

- Each document becomes a 'term' vector,
 - each term is a component (attribute) of the vector,
 - the value of each component is the number of times the corresponding term occurs in the document.

	team	coach	play	ball	score	game	winn	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

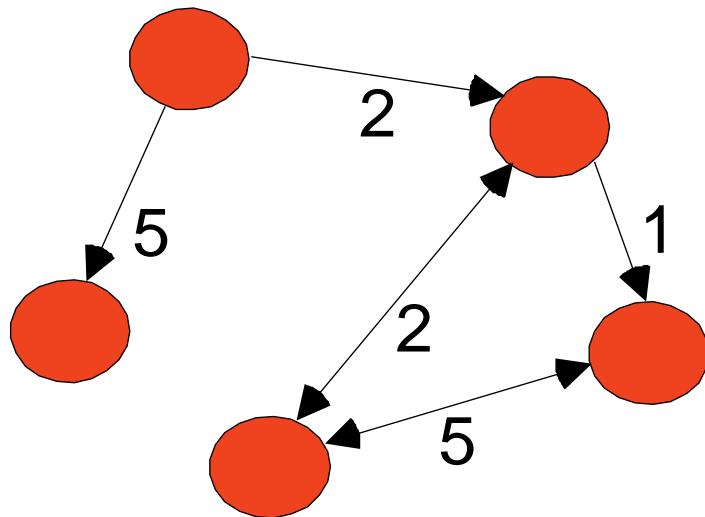
Transaction Data

- A special type of record data, where
 - each record (transaction) involves a set of items.
 - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph Data

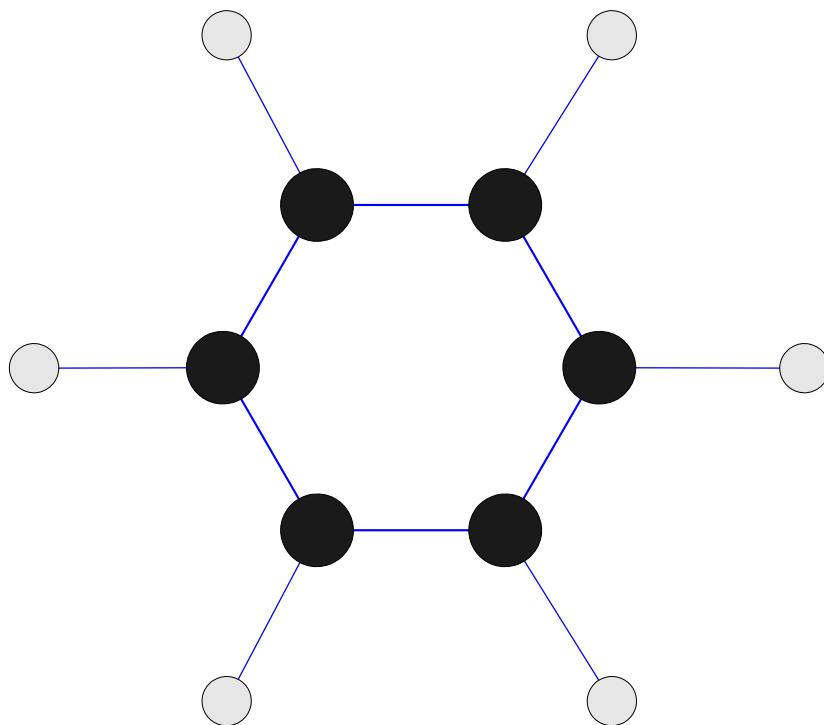
- Examples: Generic graph and HTML Links



```
<a href="papers/papers.html#bbbb">  
Data Mining </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Graph Partitioning </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Parallel Solution of Sparse Linear System of Equations </a>  
<li>  
<a href="papers/papers.html#ffff">  
N-Body Computation and Dense Linear System Solvers
```

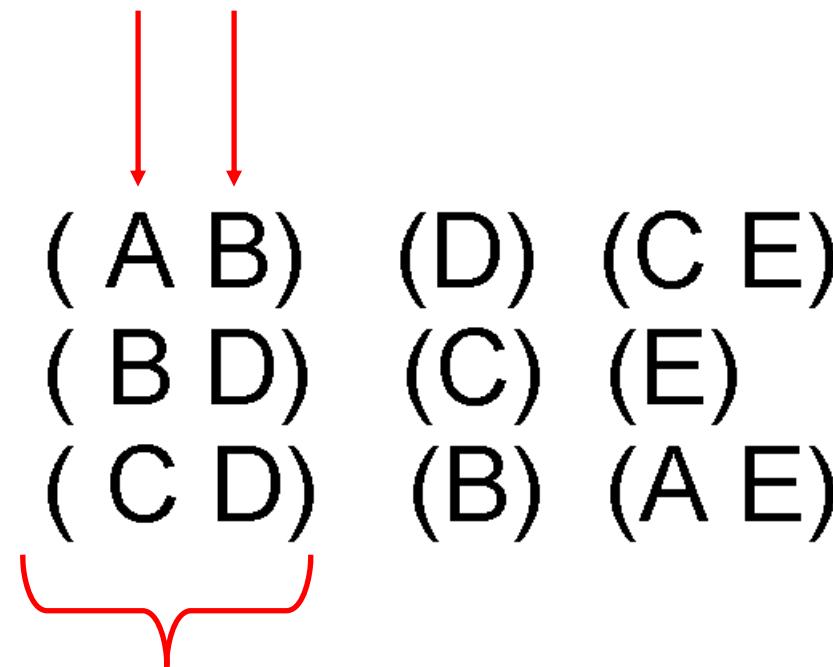
Chemical Data

- Benzene Molecule: C_6H_6



Ordered Data

- Sequences of transactions
Items/Events



An element of
the sequence

Ordered Data

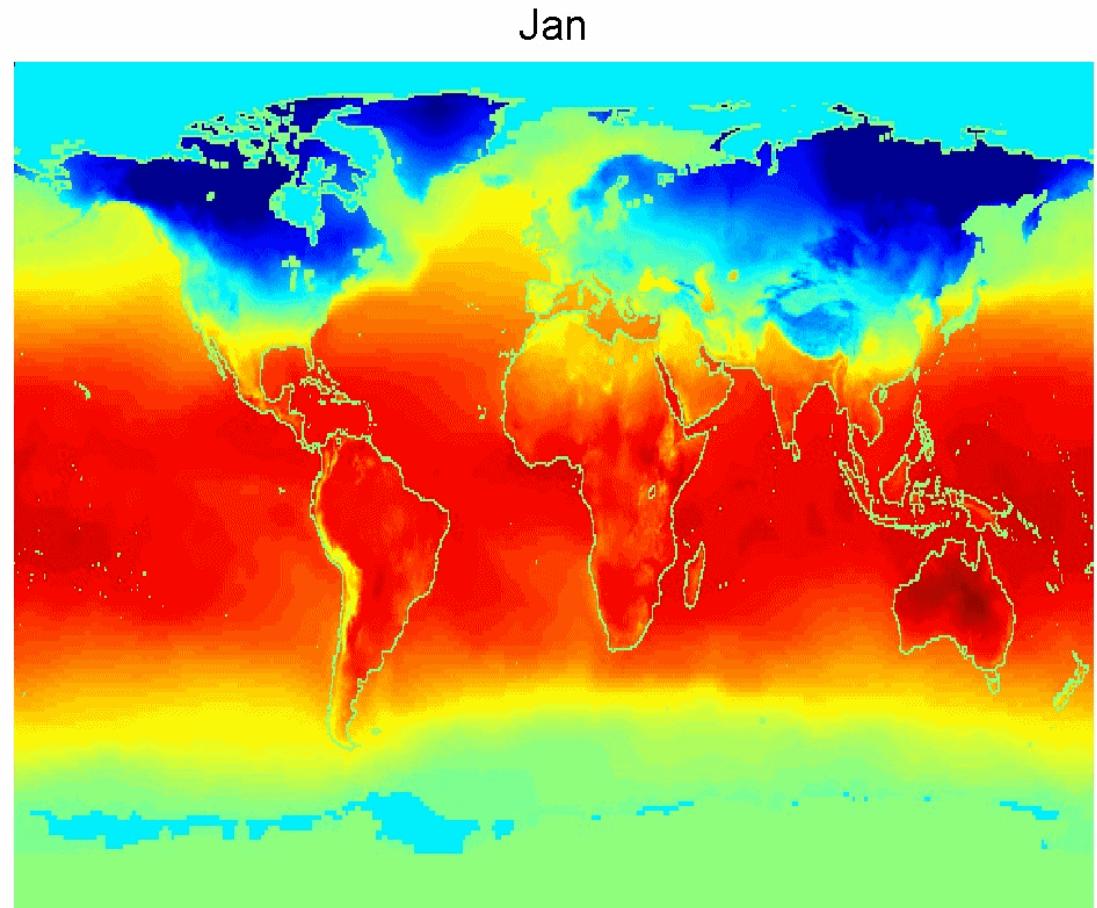
- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCGCCCCGGCGCCGTC  
GAGAAGGGCCCAGCTGGCGGGCG  
GGGGGAGGCAGGGCCGCCGAGC  
CAAACCGAGTCCGACCAAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGGGACAG  
GCCAAGTAGAACACGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

Ordered Data

- Spatio-Temporal Data

Average Monthly
Temperature of
land and ocean

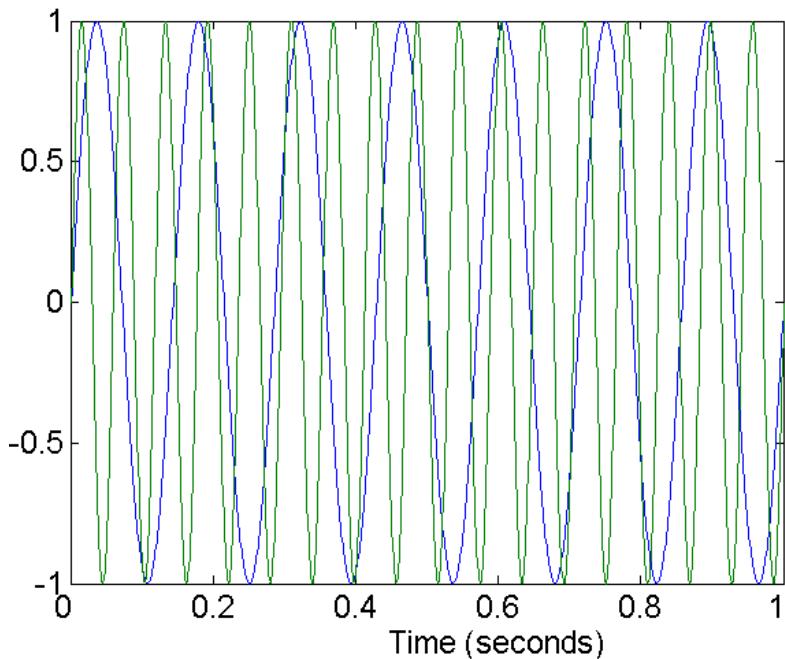


Data Quality

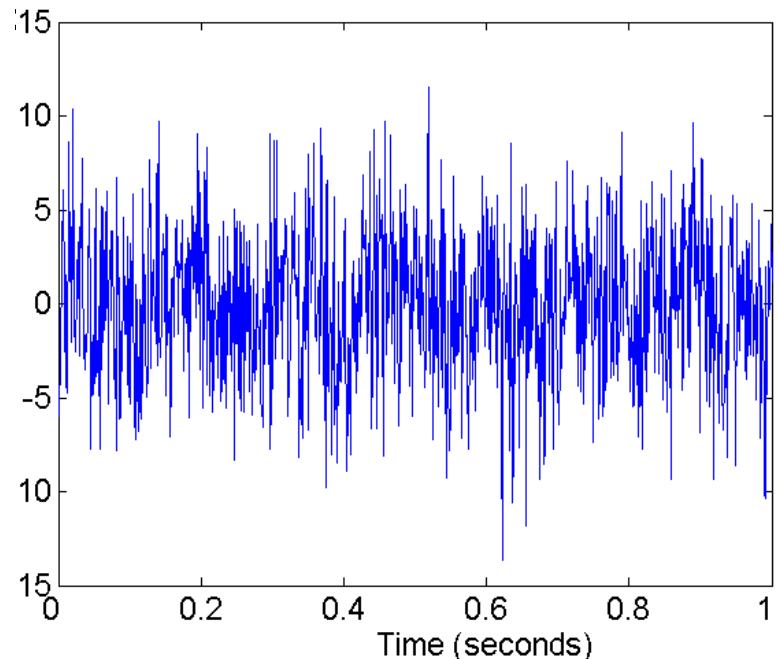
- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?
- Examples of data quality problems:
 - Noise and outliers
 - missing values
 - duplicate data

Noise

- Noise refers to modification of original values
 - Examples: distortion of a person's voice when **talking on a poor**



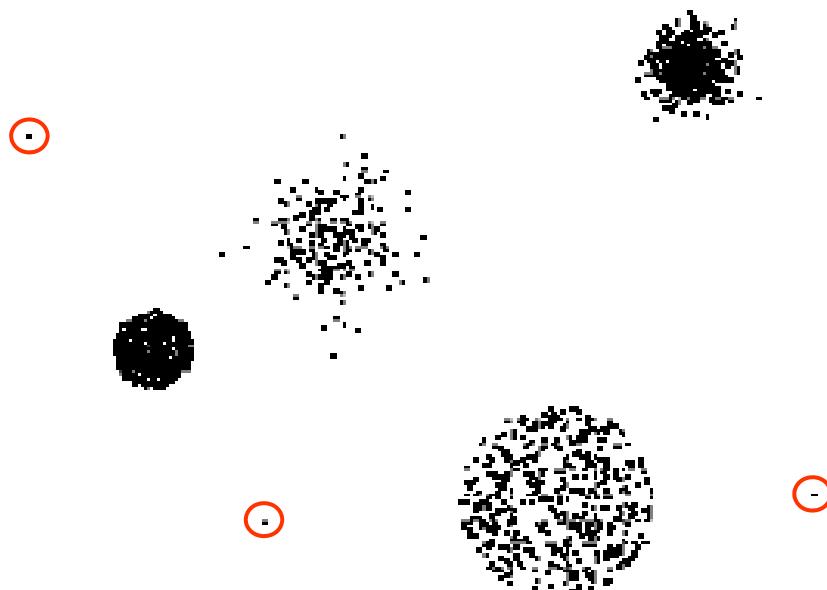
Two Sine Waves



Two Sine Waves +
Noise

Outliers

- Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set



Missing Values

- Reasons for missing values
 - Information is not collected
(e.g., people decline to give their age and weight)
 - Attributes may not be applicable to all cases
(e.g., annual income is not applicable to children)
- Handling missing values
 - Eliminate Data Objects
 - Estimate Missing Values
 - Ignore the Missing Value During Analysis
 - Replace with all possible values (weighted by their probabilities)

Duplicate Data

- Data set may include data objects that are duplicates, or almost duplicates of one another
 - Major issue when merging data from heterogeneous sources
- Examples:
 - Same person with multiple email addresses
- Data cleaning
 - Process of dealing with duplicate data issues

Data Preprocessing

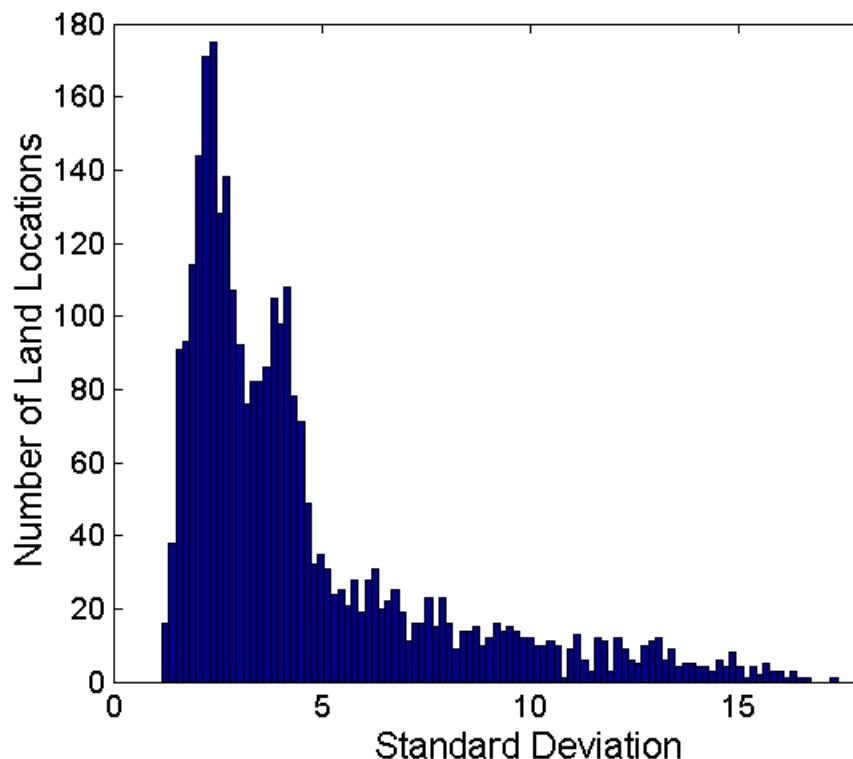
- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

Aggregation

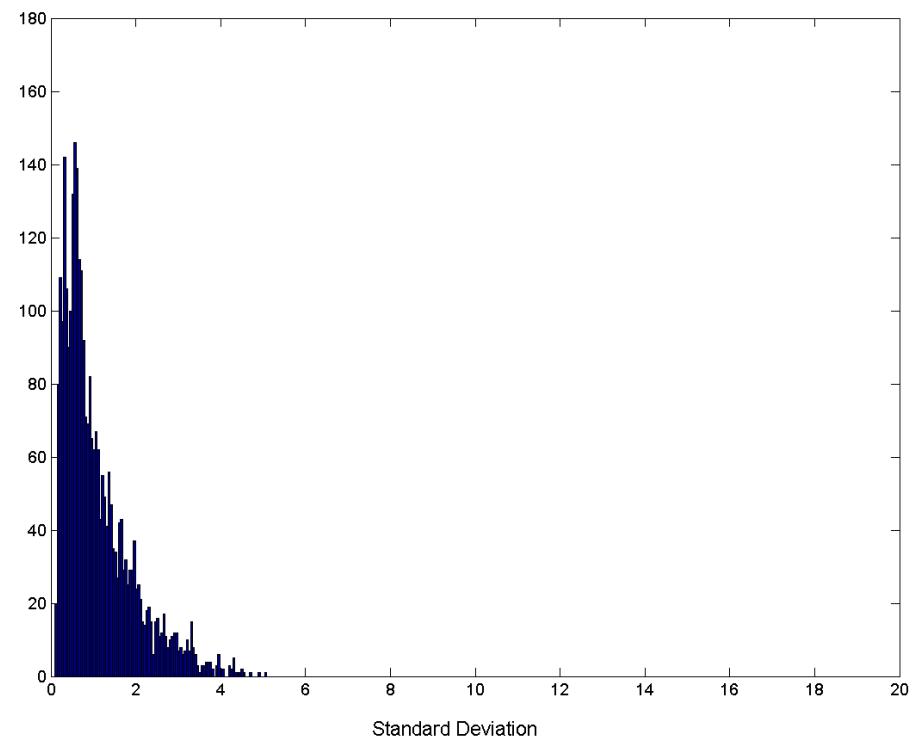
- Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
 - Data reduction
 - Reduce the number of attributes or objects
 - Change of scale
 - Cities aggregated into regions, states, countries, etc
 - More “stable” data
 - Aggregated data tends to have less variability

Aggregation

Variation of Precipitation



Standard Deviation of
Average Monthly
Precipitation



Standard Deviation of
Average Yearly
Precipitation

Sampling

- Sampling is the main technique employed for data selection.
 - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.

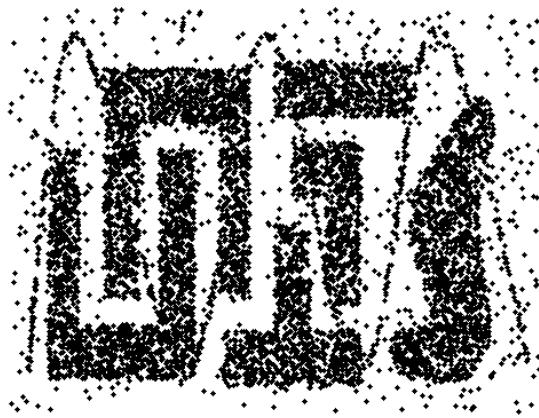
Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is representative
 - A sample is representative if it has approximately the same property (of interest) as the original set of data

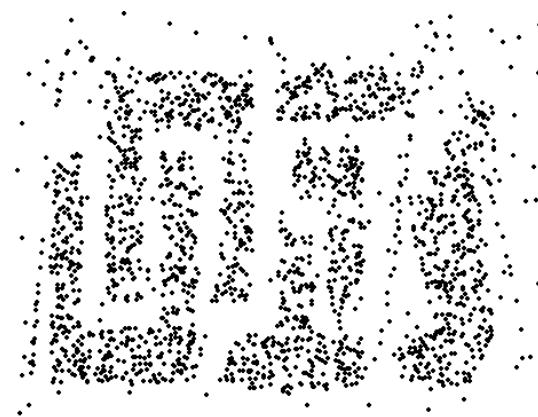
Types of Sampling

- Simple Random Sampling
 - There is an equal probability of selecting any particular item
- Sampling without replacement
 - As each item is selected, it is removed from the population
- Sampling with replacement
 - Objects are not removed from the population as they are selected for the sample.
 - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
 - Split the data into several partitions; then draw random samples from each partition

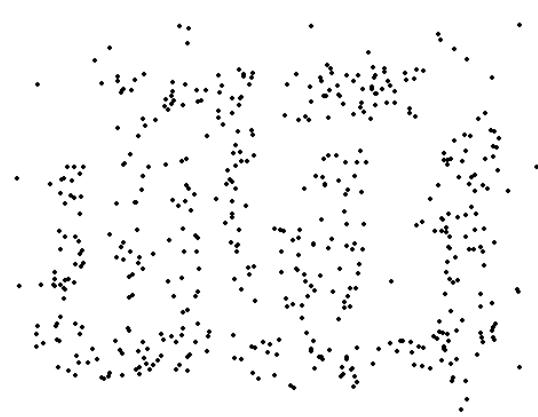
Sample Size



8000 points



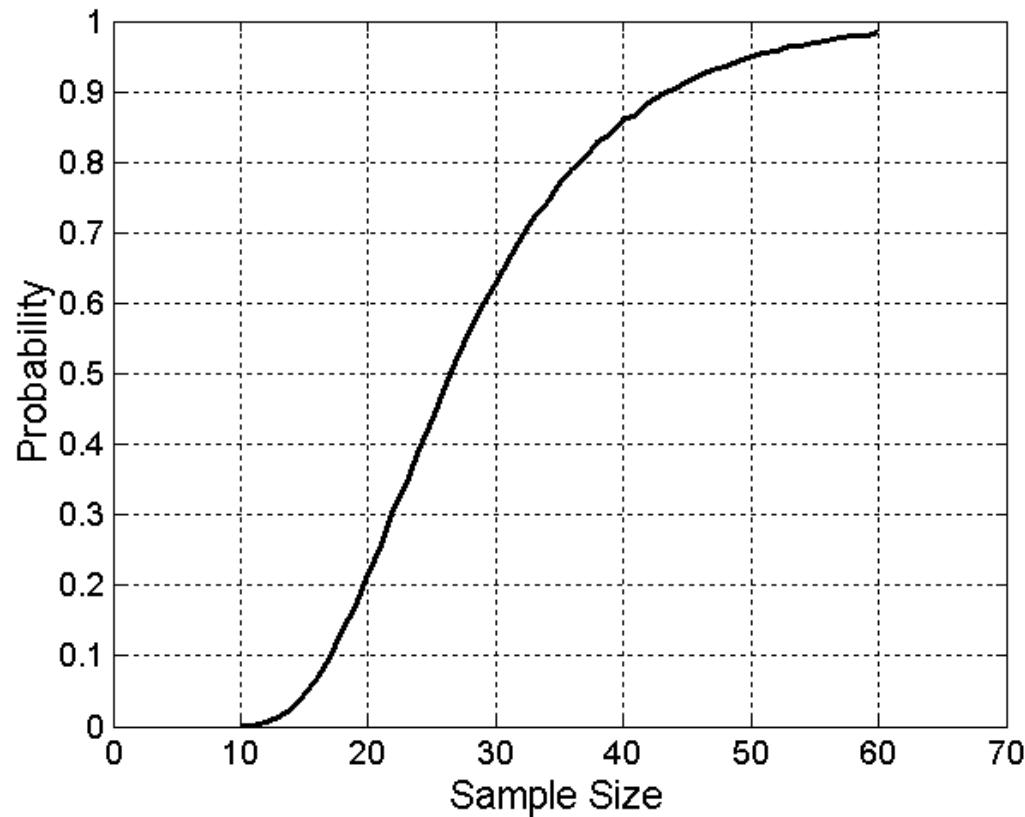
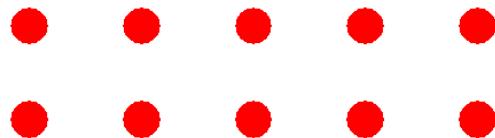
2000 Points



500 Points

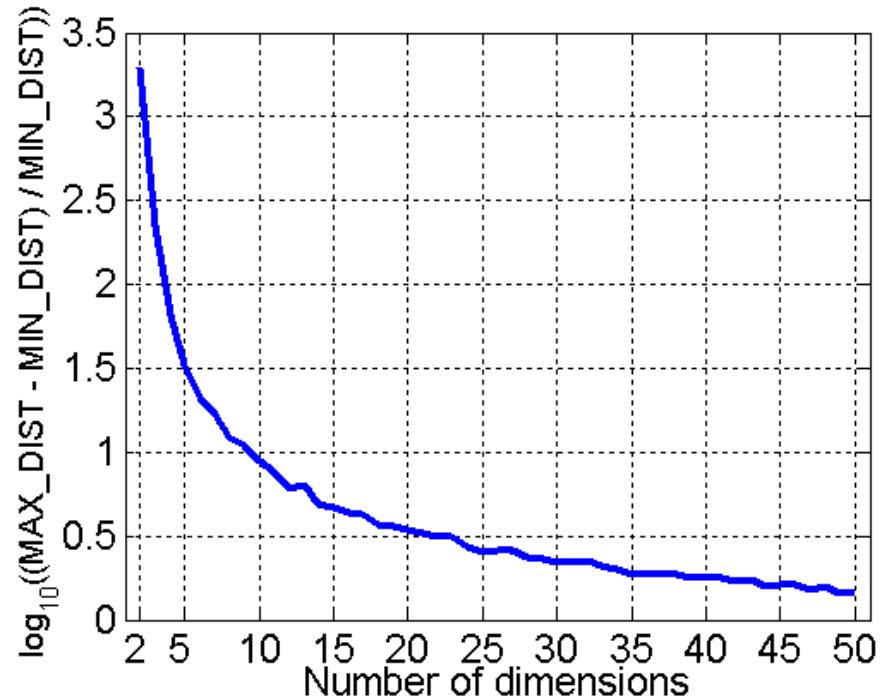
Sample Size

- What sample size is necessary to get at least one object from each of 10 groups.



Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



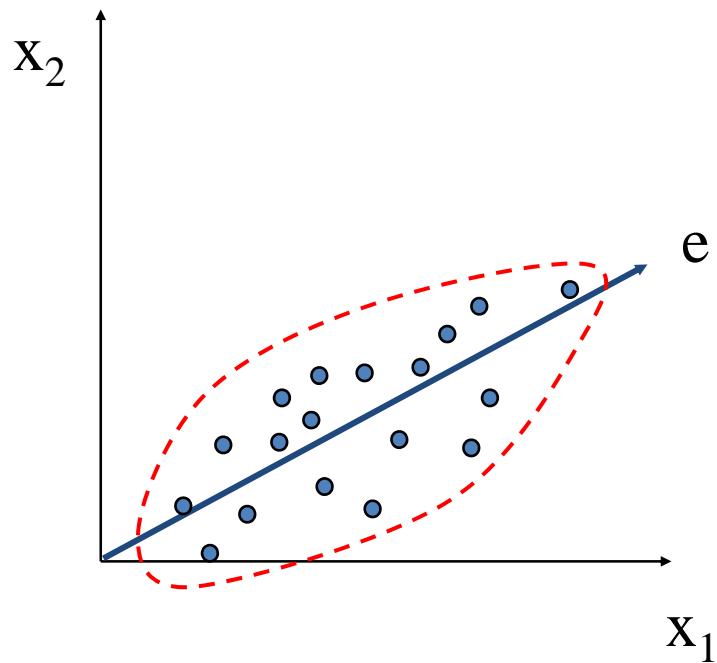
- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

Dimensionality Reduction

- Purpose:
 - Avoid curse of dimensionality
 - Reduce amount of time and memory required by data mining algorithms
 - Allow data to be more easily visualized
 - May help to eliminate irrelevant features or reduce noise
- Techniques
 - Principal Component Analysis
 - Singular Value Decomposition
 - Others: supervised and non-linear techniques

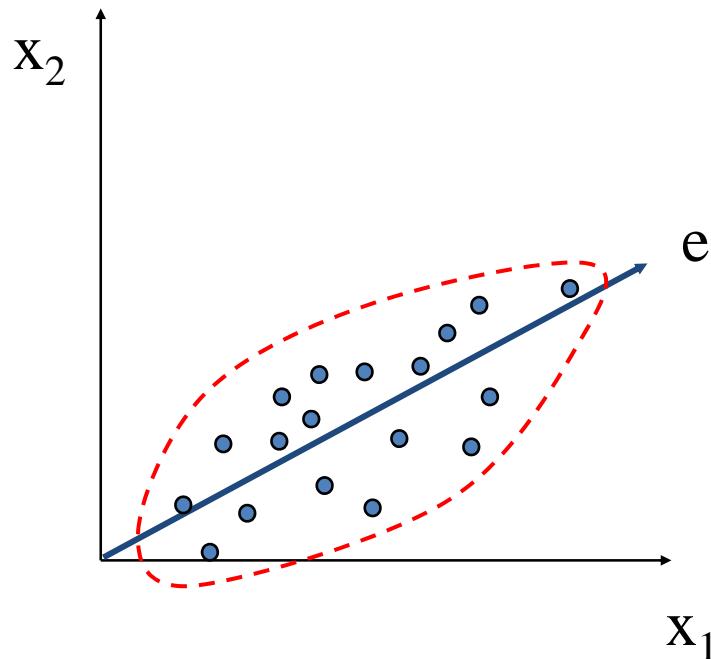
Dimensionality Reduction: PCA

- Goal is to find a projection that captures the largest amount of variation in data



Dimensionality Reduction: PCA

- Find the eigenvectors of the covariance matrix
- The eigenvectors define the new space



Feature Subset Selection

- Another way to reduce dimensionality of data
- Redundant features
 - duplicate much or all of the information contained in one or more other attributes
 - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
 - contain no information that is useful for the data mining task at hand
 - Example: students' ID is often irrelevant to the

Feature Subset Selection

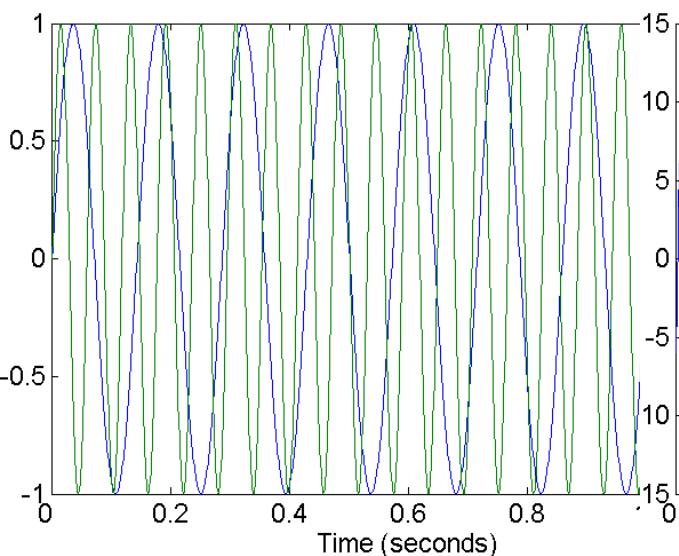
- Techniques:
 - Brute-force approach:
 - Try all possible feature subsets as input to data mining algorithm
 - Embedded approaches:
 - Feature selection occurs naturally as part of the data mining algorithm
 - Filter approaches:
 - Features are selected before data mining algorithm is run
 - Wrapper approaches:
 - Use the data mining algorithm as a black box to find best subset of attributes

Feature Creation

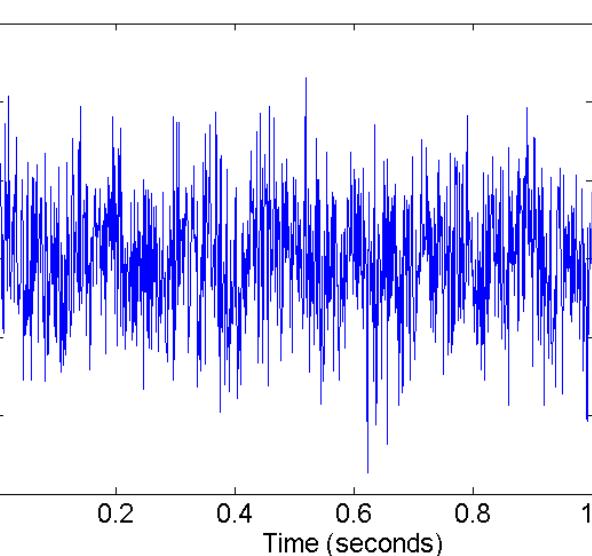
- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes
- Three general methodologies:
 - Feature Extraction
 - domain-specific
 - Mapping Data to New Space
 - Feature Construction
 - combining features

Mapping Data to a New Space

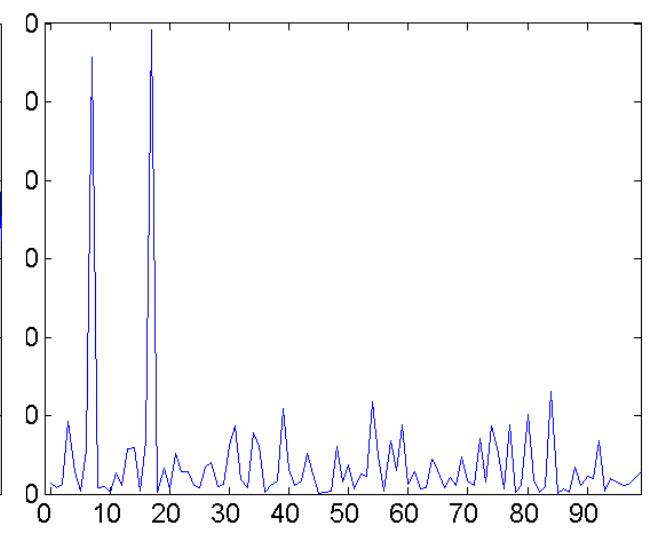
- Fourier transform
- Wavelet transform



Two Sine Waves



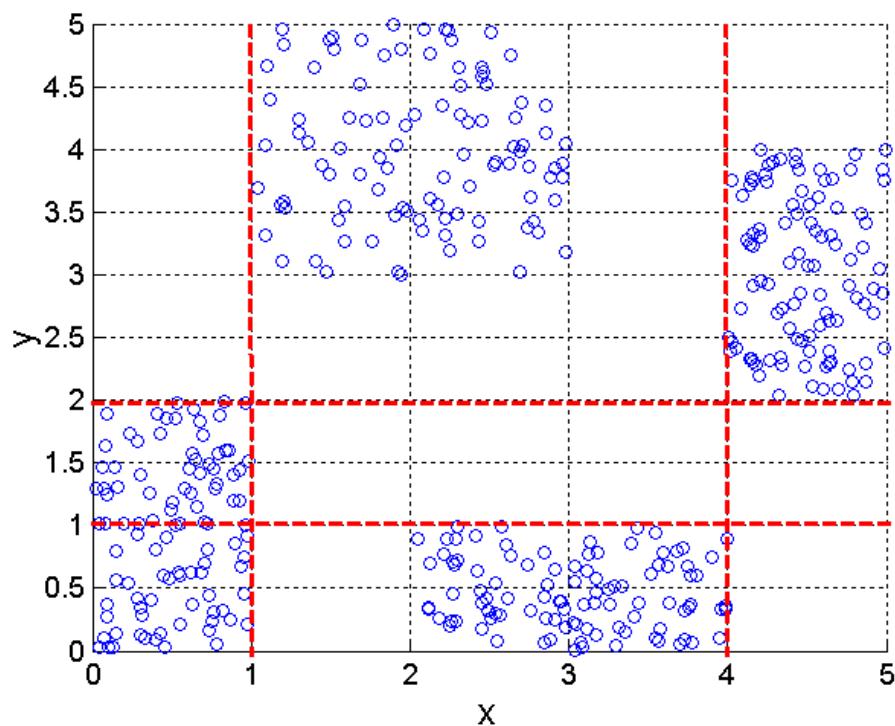
Two Sine Waves +
Noise



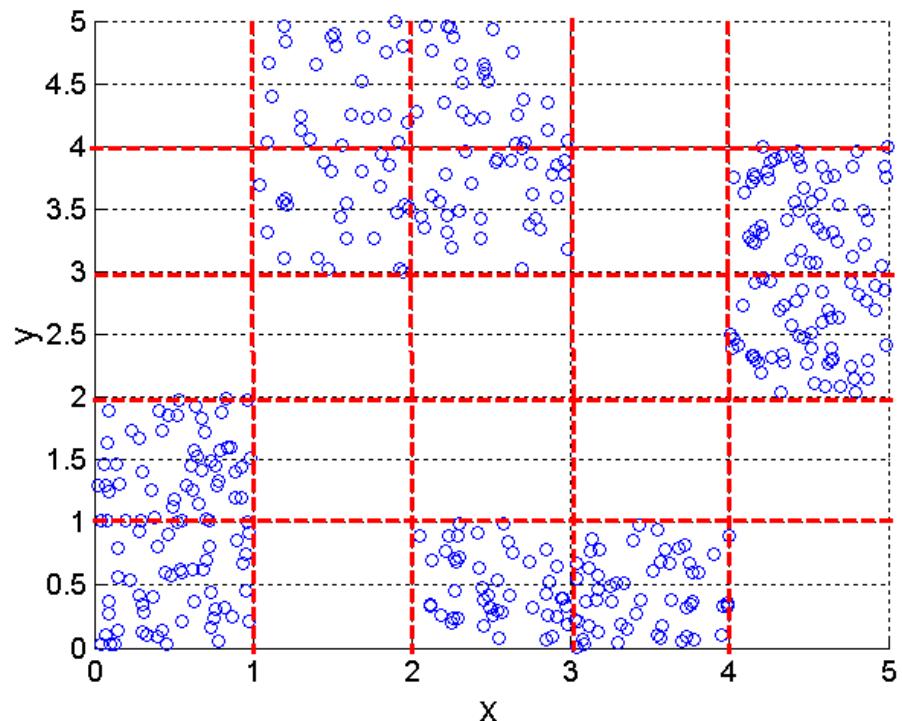
Frequency

Discretization Using Class Labels

- Entropy based approach

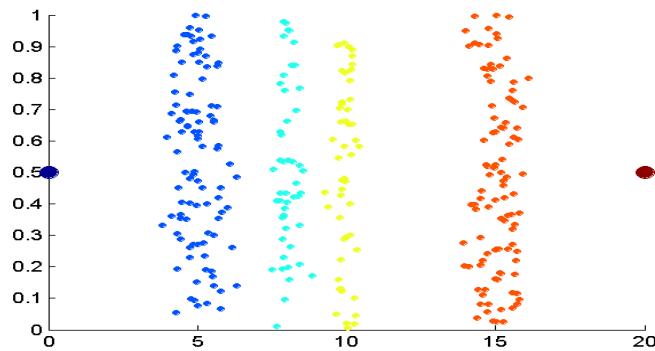


3 categories for both x and
y

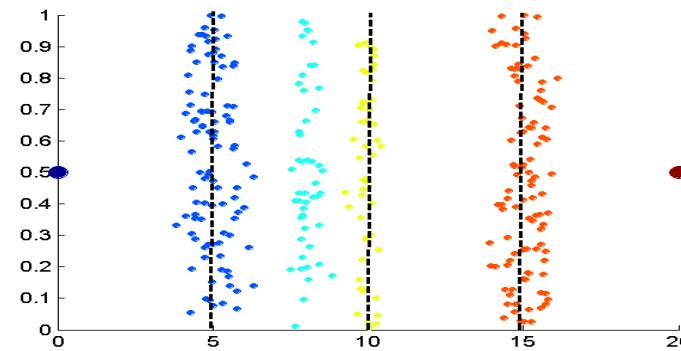


5 categories for both x and
y

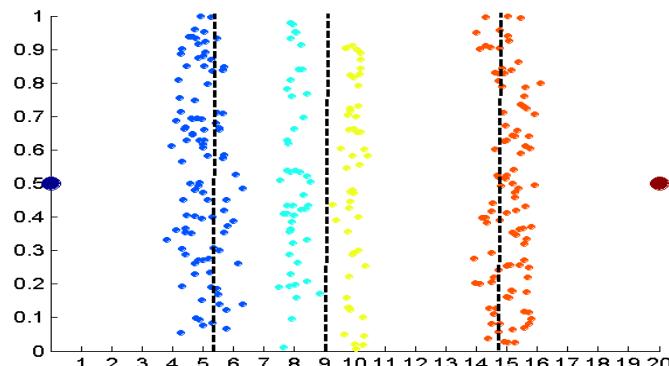
Discretization Without Using Class Labels



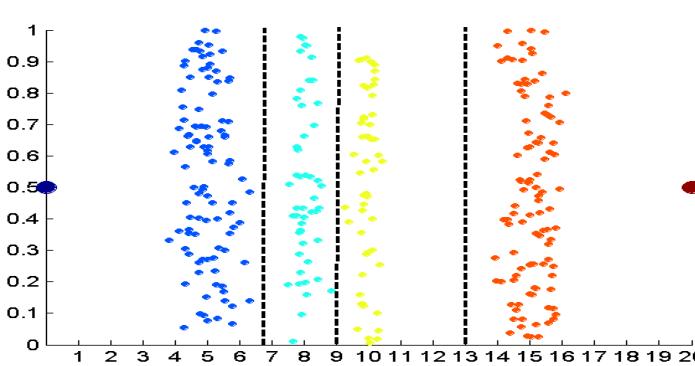
Data



Equal interval



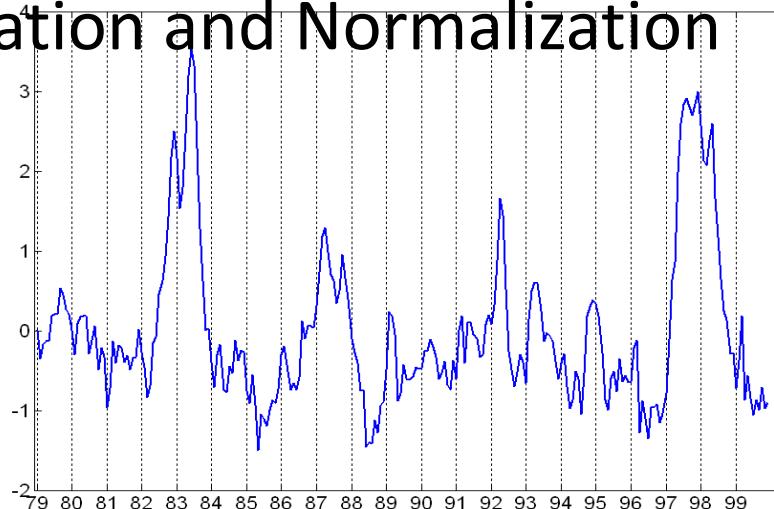
Equal frequency



K-means

Attribute Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
 - Simple functions: x^k , $\log(x)$, e^x , $|x|$
 - Standardization and Normalization



Similarity and Dissimilarity

- Similarity
 - Numerical measure of how alike two data objects are.
 - Is higher when objects are more alike.
 - Often falls in the range [0,1]
- Dissimilarity
 - Numerical measure of how different are two data objects
 - Lower when objects are more alike
 - Minimum dissimilarity is often 0
 - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Similarity/Dissimilarity for Simple Attributes

p and q are the attribute values for two data objects.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{ p-q }{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{ p-q }{n-1}$
Interval or Ratio	$d = p - q $	$s = -d, s = \frac{1}{1+d}$ or $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Euclidean Distance

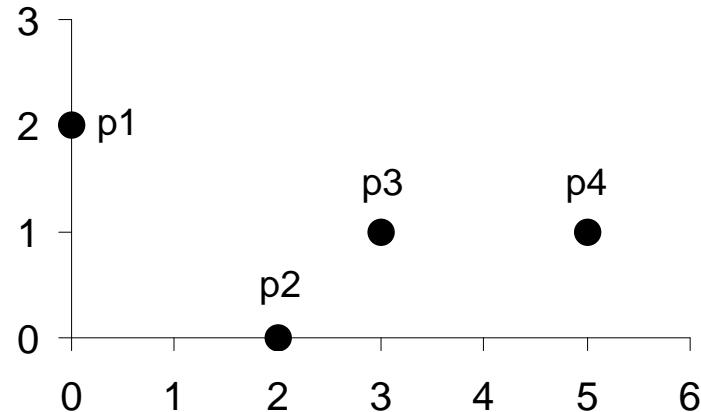
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects p and q .

- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^n |p_k - q_k|^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k th attributes (components) or data objects p and q .

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
 - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$. Euclidean distance
- $r \rightarrow \infty$. “supremum” (L_{\max} norm, L_∞ norm) distance.
 - This is the maximum difference between any component of the vectors
- Do not confuse r with n , i.e., all these distances are defined for all numbers of dimensions.

Minkowski Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

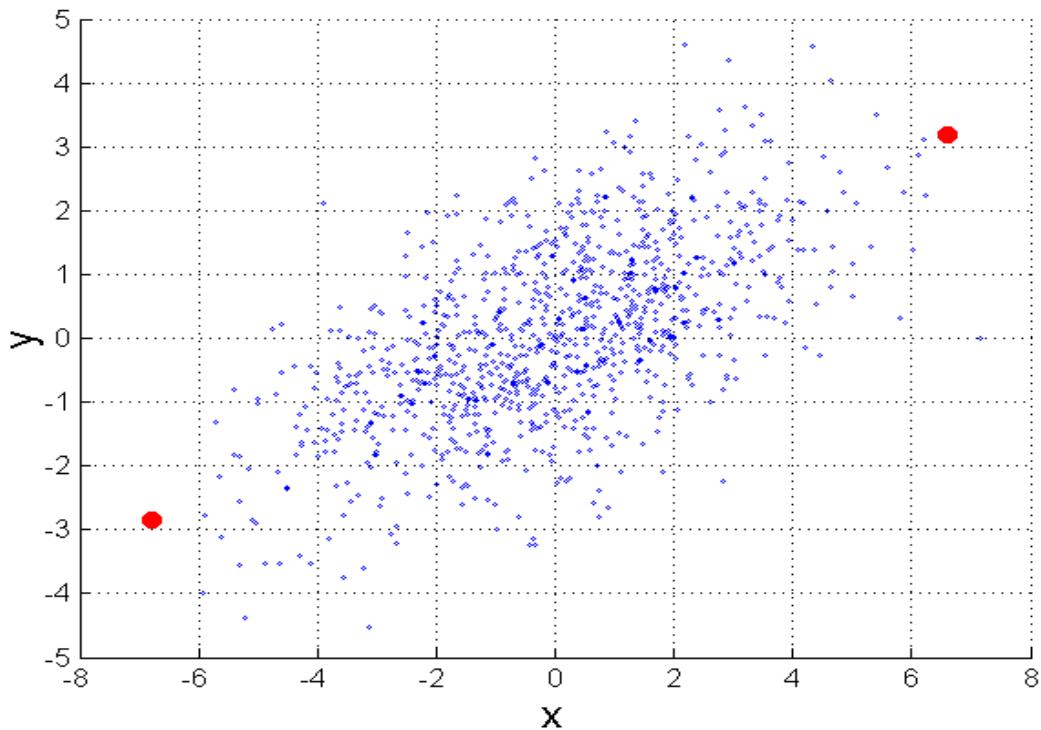
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L ∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Mahalanobis Distance

$$mahalanobi\ s(p, q) = (p - q) \Sigma^{-1} (p - q)^T$$

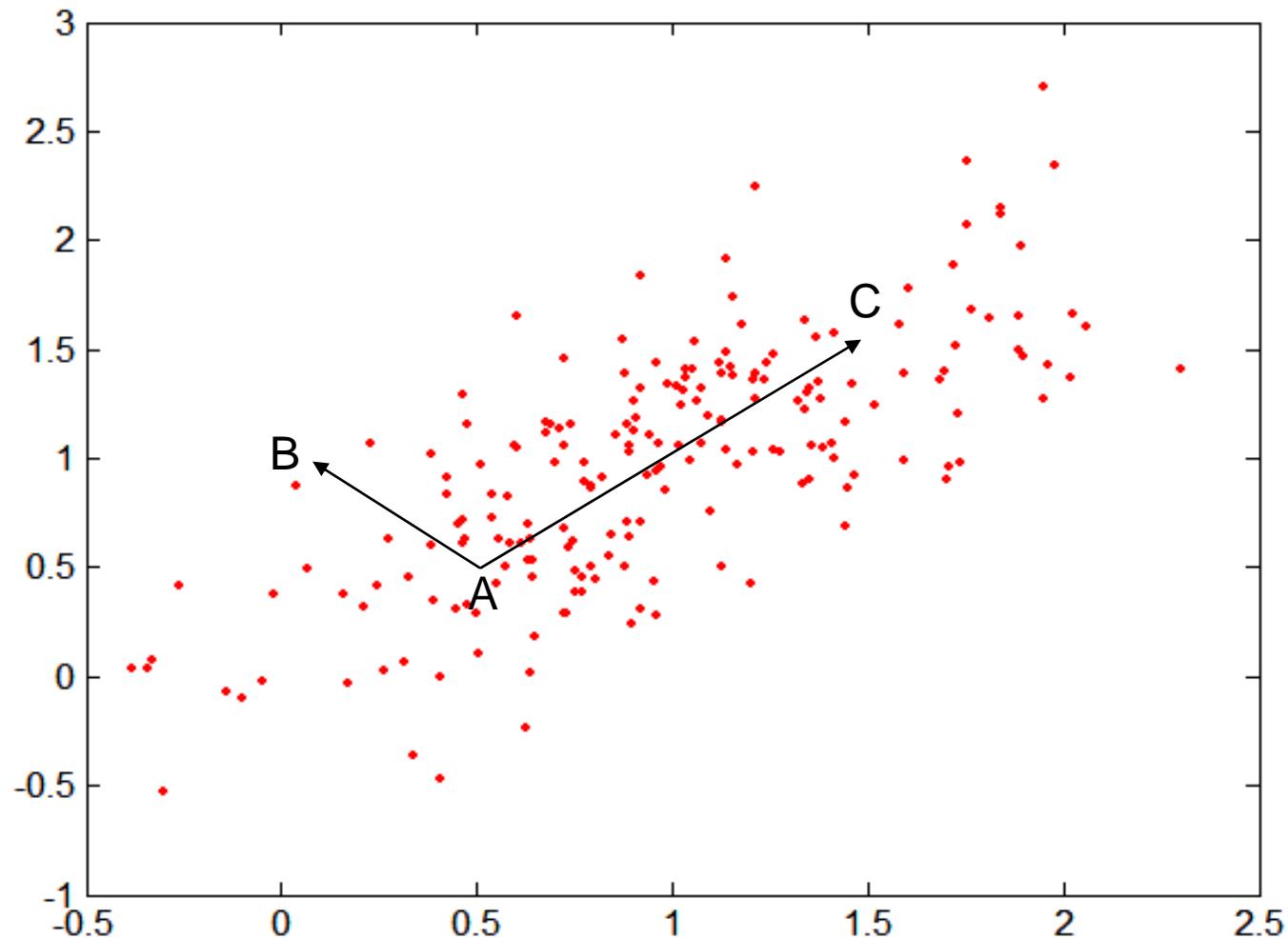


Σ is the covariance matrix of the input data X

$$\Sigma_{j,k} = \frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)(X_{ik} - \bar{X}_k)$$

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

Mahalanobis Distance



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

$\text{Mahal}(A,B) = 5$

$\text{Mahal}(A,C) = 4$

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.
 1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
 2. $d(p, q) = d(q, p)$ for all p and q . (Symmetry)
 3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p , q , and r . (Triangle Inequality)where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q .
- A distance that satisfies these properties is a **metric**

Common Properties of a Similarity

- Similarities, also have some well known properties.
 1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.
 2. $s(p, q) = s(q, p)$ for all p and q . (Symmetry)

where $s(p, q)$ is the similarity between points (data objects), p and q .

Similarity Between Binary Vectors

- Common situation is that objects, p and q , have only binary attributes
- Compute similarities using the following quantities

M_{01} = the number of attributes where p was 0 and q was 1

M_{10} = the number of attributes where p was 1 and q was 0

M_{00} = the number of attributes where p was 0 and q was 0

M_{11} = the number of attributes where p was 1 and q was 1

- Simple Matching and Jaccard Coefficients

$SMC = \text{number of matches} / \text{number of attributes}$

$$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$$

$J = \text{number of } 11 \text{ matches} / \text{number of not-both-zero attributes values}$

$$= (M_{11}) / (M_{01} + M_{10} + M_{11})$$

SMC versus Jaccard: Example

$p = 1000000000$

$q = 000001001$

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

$$SMC = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$$

$$J = (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$$

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \|d_2\|,$$

where • indicates vector dot product and $\|d\|$ is the length of vector d .

- Example:

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

$$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|d_1\| = (3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$$

$$\|d_2\| = (1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2)^{0.5} = (6)^{0.5} = 2.245$$

$$\cos(d_1, d_2) = .3150$$

Extended Jaccard Coefficient (Tanimoto)

- Variation of Jaccard for continuous or count attributes
 - Reduces to Jaccard for binary attributes

$$T(p, q) = \frac{p \bullet q}{\|p\|^2 + \|q\|^2 - p \bullet q}$$

Correlation

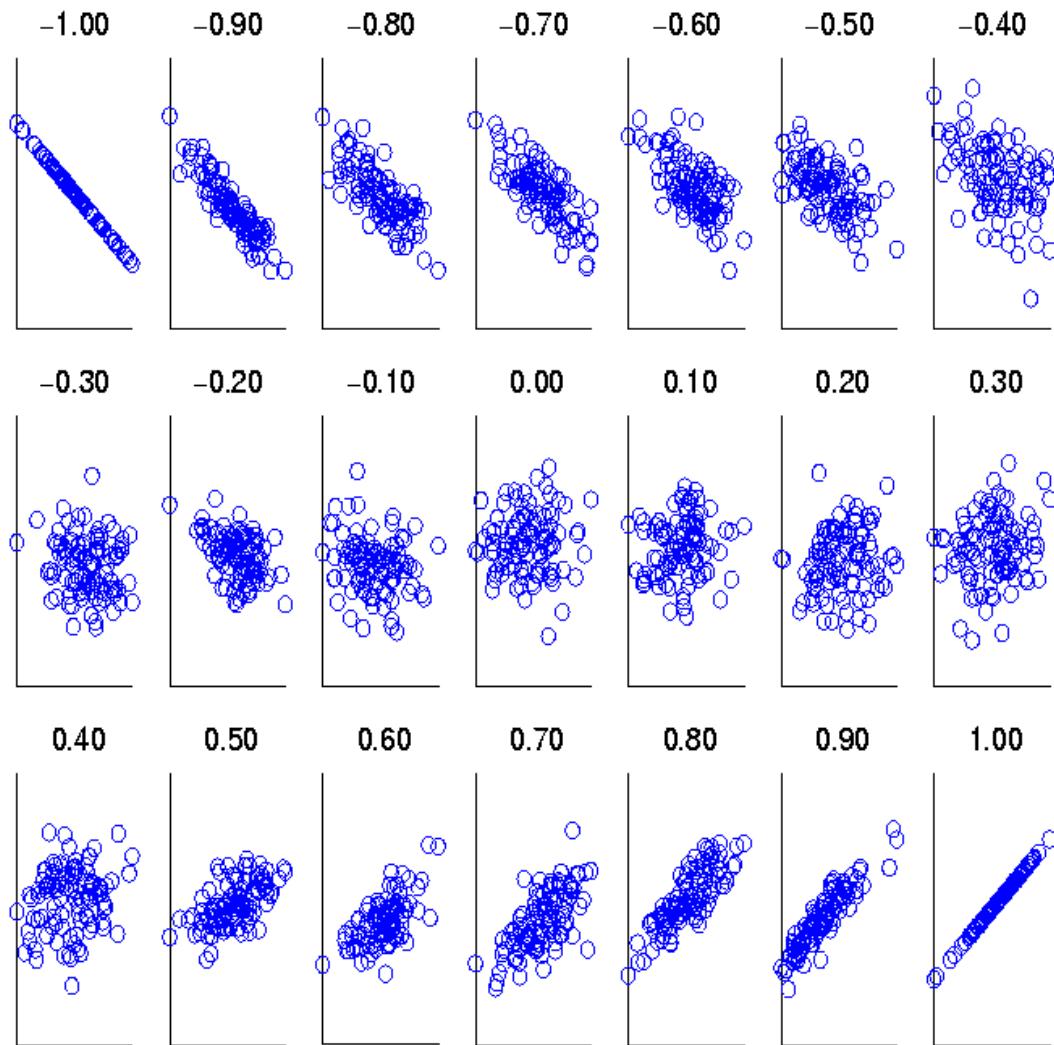
- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q, and then take their dot product

$$p'_k = (p_k - \text{mean}(p)) / \text{std}(p)$$

$$q'_k = (q_k - \text{mean}(q)) / \text{std}(q)$$

$$\text{correlation}(p, q) = p' \bullet q'$$

Visually Evaluating Correlation



Scatter plots
showing the
similarity from
–1 to 1.

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k , in the range $[0, 1]$.
2. Define an indicator variable, δ_k , for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^n \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
 - Use weights w_k which are between 0 and 1 and

s

$$similarity(p, q) = \frac{\sum_{k=1}^n w_k \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

$$distance(p, q) = \left(\sum_{k=1}^n w_k |p_k - q_k|^r \right)^{1/r}$$

Density

- Density-based clustering require a notion of density
- Examples:
 - Euclidean density
 - Euclidean density = number of points per unit volume
 - Probability density
 - Graph-based density

Euclidean Density – Cell-based

- Simplest approach is to divide region into a number of rectangular cells of equal volume

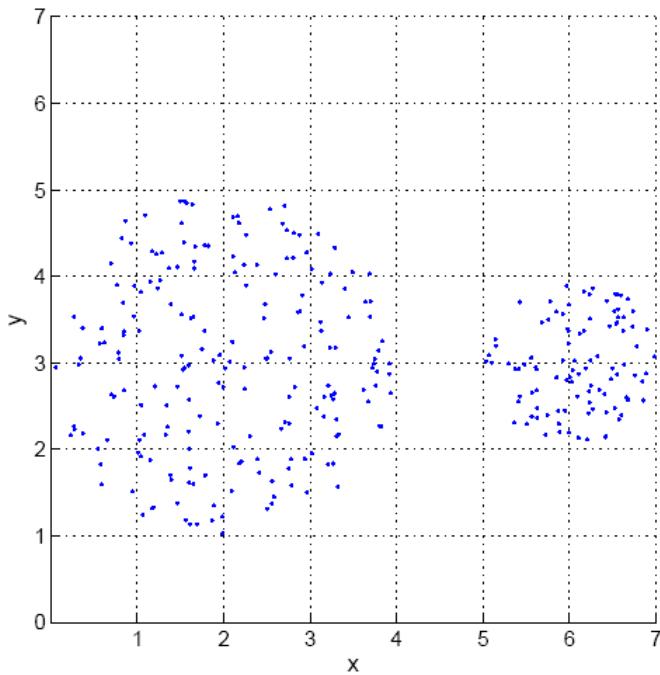


Figure 7.13. Cell-based density.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

Table 7.6. Point counts for each grid cell.

Euclidean Density – Center-based

- Euclidean density is the number of points within a specified radius of the point

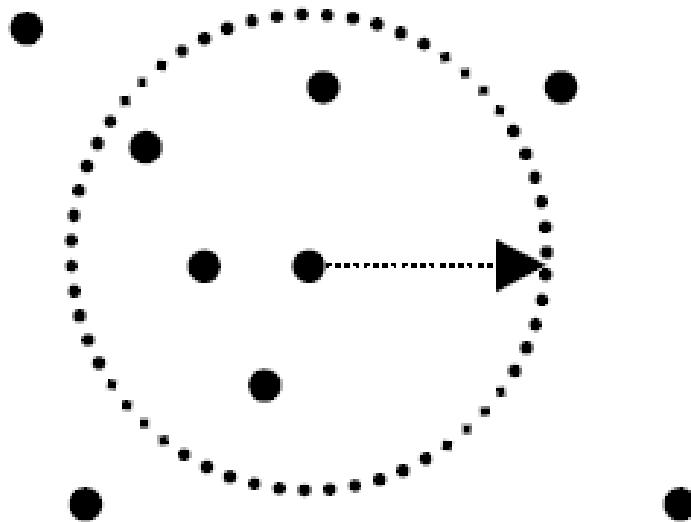


Figure 7.14. Illustration of center-based density.

DATA MINING

LECTURE 10

Classification

Basic Concepts

Decision Trees

Catching tax-evasion

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tax-return data for year 2011

A new tax return for 2012
Is this a cheating tax return?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

An instance of the classification problem: learn a method for discriminating between records of different **classes** (**cheaters** vs **non-cheaters**)

What is classification?

$x \rightarrow \boxed{f} \rightarrow y$, $y = \{ \text{yes, No} \}$

- **Classification** is the task of *learning* a target function f that maps attribute set x to one of the predefined class labels y

$x = \{ \text{Refund, Marital Status} \}$

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

One of the attributes is the **class attribute**
In this case: Cheat

Two **class labels** (or **classes**): Yes (1), No (0)

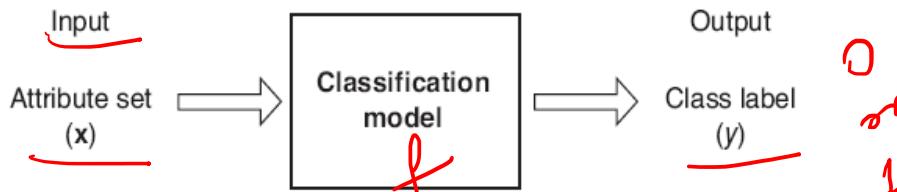


Figure 4.2. Classification as the task of mapping an input attribute set x into its class label y .

Why classification?

- The target function f is known as a **classification model**
- **Descriptive modeling:** Explanatory tool to distinguish between objects of different classes (e.g., understand why people cheat on their taxes)
- **Predictive modeling:** Predict a class of a previously unseen record

Examples of Classification Tasks

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Identifying spam email, spam web pages, adult content
- Understanding if a web query has commercial intent or not

General approach to classification

train
test

- Training set consists of records with known class labels
- Training set is used to build a classification model
- A labeled test set of previously unseen data records is used to evaluate the quality of the model.
- The classification model is applied to new records with unknown class labels

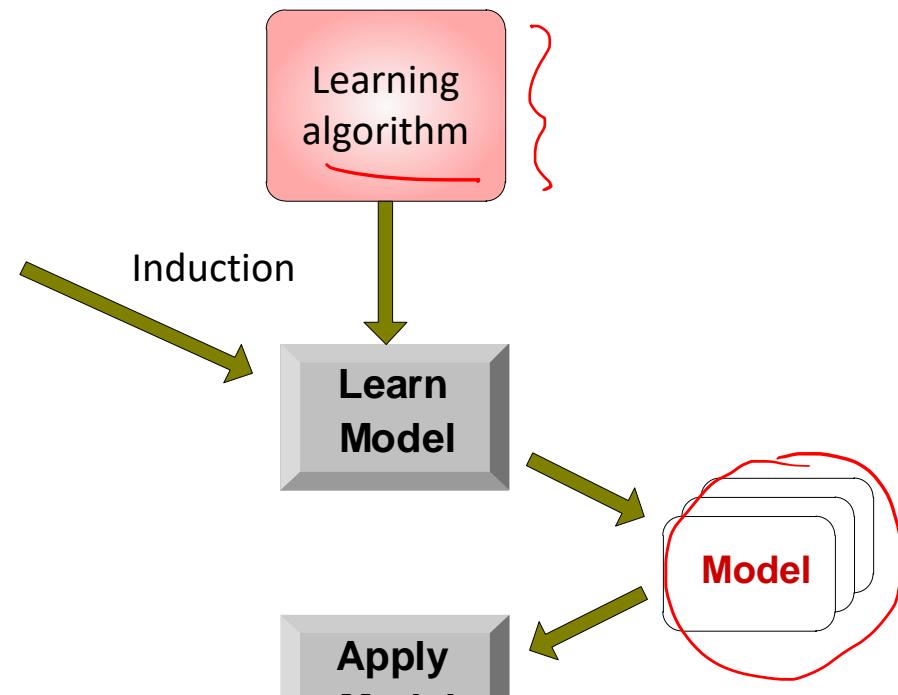
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Deduction

~~Total no. of xve records.~~ FN → are the no. of xve records missed by the model to predict correctly.

Evaluation of classification models

- Counts of test records that are correctly (or incorrectly) predicted by the classification model

Confusion matrix

Precision
Recall
F1 score

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

$$P = \frac{TP + FN}{TP + FP + FN}$$

$$R = \frac{TP + TN}{TP + FN}$$

$$F_1 = \frac{2 * P * R}{P + R}$$

$$\checkmark \text{Accuracy} = \frac{\# \text{ correct predictions}}{\text{total } \# \text{ of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$\checkmark \text{Error rate} = \frac{\# \text{ wrong predictions}}{\text{total } \# \text{ of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

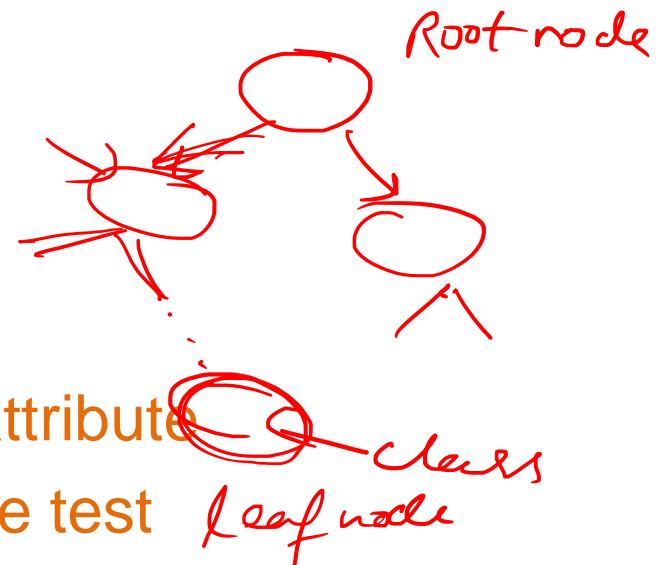
Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Decision Trees

Decision tree

- A flow-chart-like tree structure
- Internal node denotes a test on an attribute
- Branch represents an outcome of the test
- Leaf nodes represent class labels or class distribution



Root node →

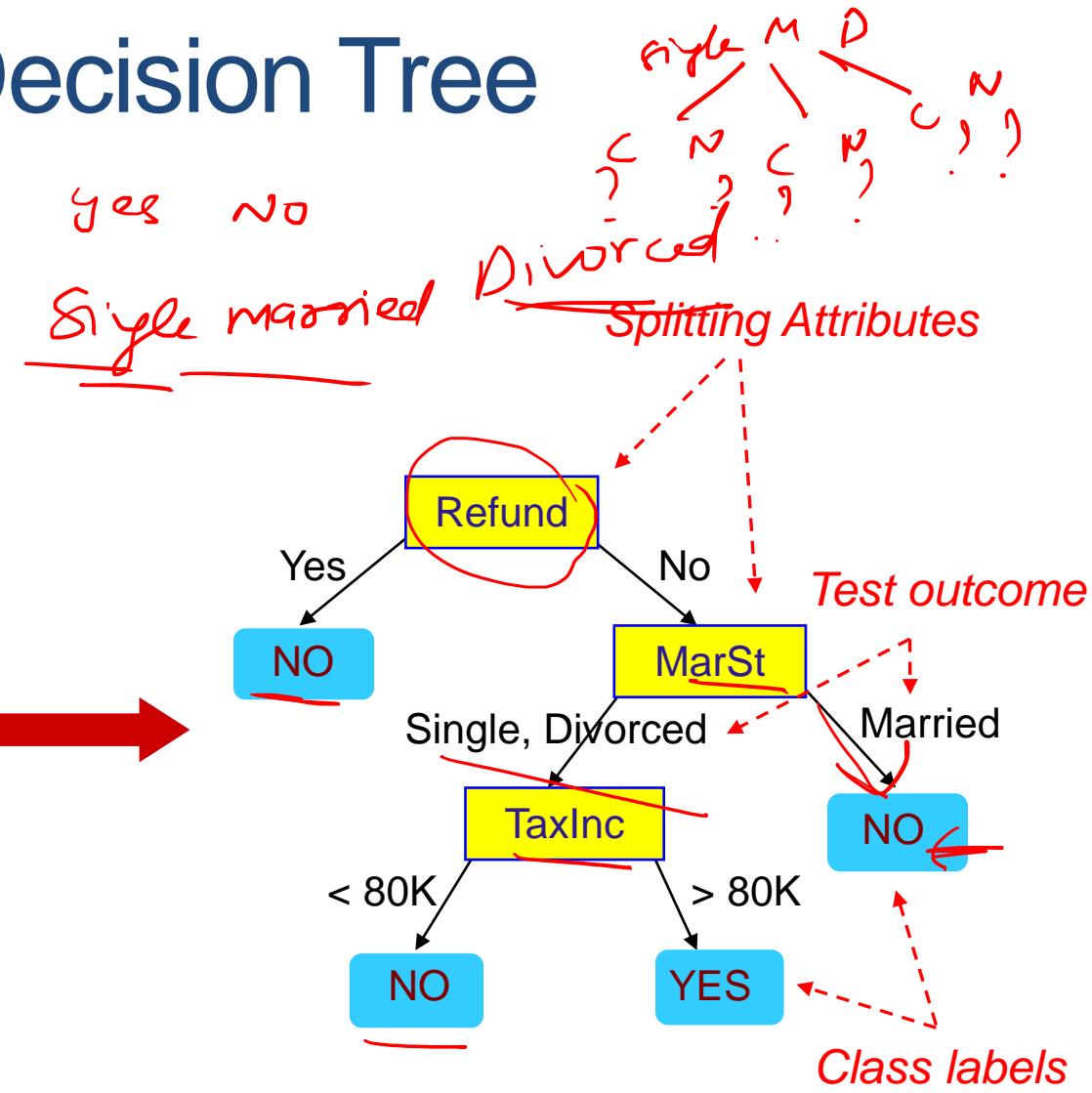
Internal nodes

Leaf or terminal nodes

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

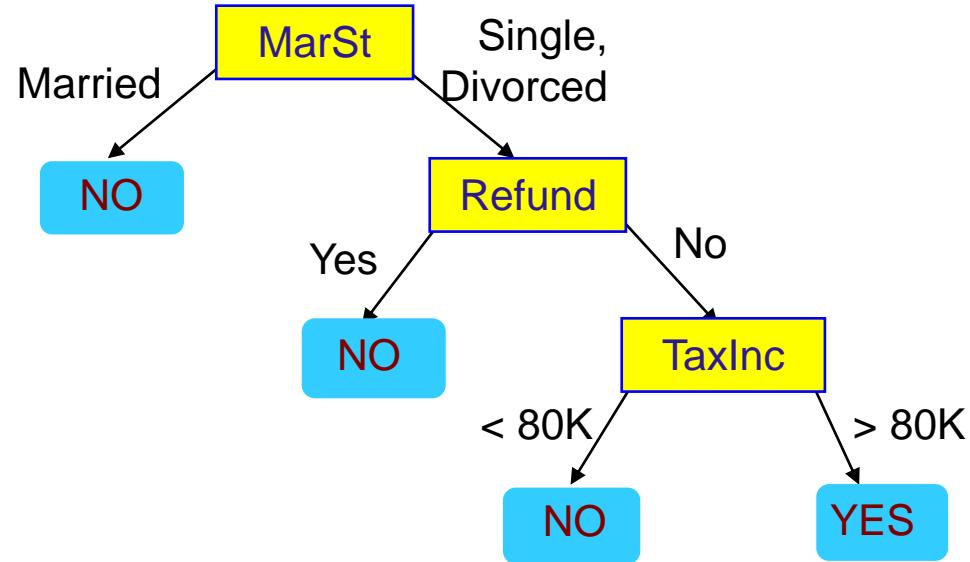
Training Data



Model: Decision Tree

Another Example of Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat	class
				categorical	
1	Yes	Single	125K	No	
2	No	Married	100K	No	
3	No	Single	70K	No	
4	Yes	Married	120K	No	
5	No	Divorced	95K	Yes	
6	No	Married	60K	No	
7	Yes	Divorced	220K	No	
8	No	Single	85K	Yes	
9	No	Married	75K	No	
10	No	Single	90K	Yes	



There could be more than one tree that fits the same data!

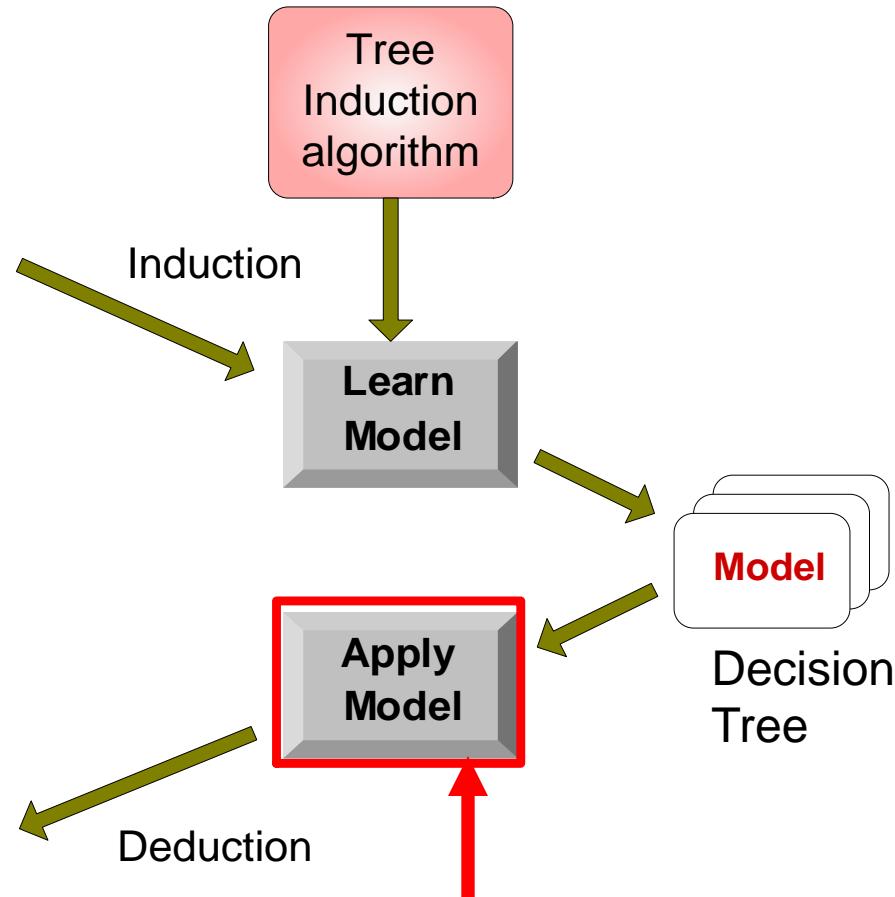
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

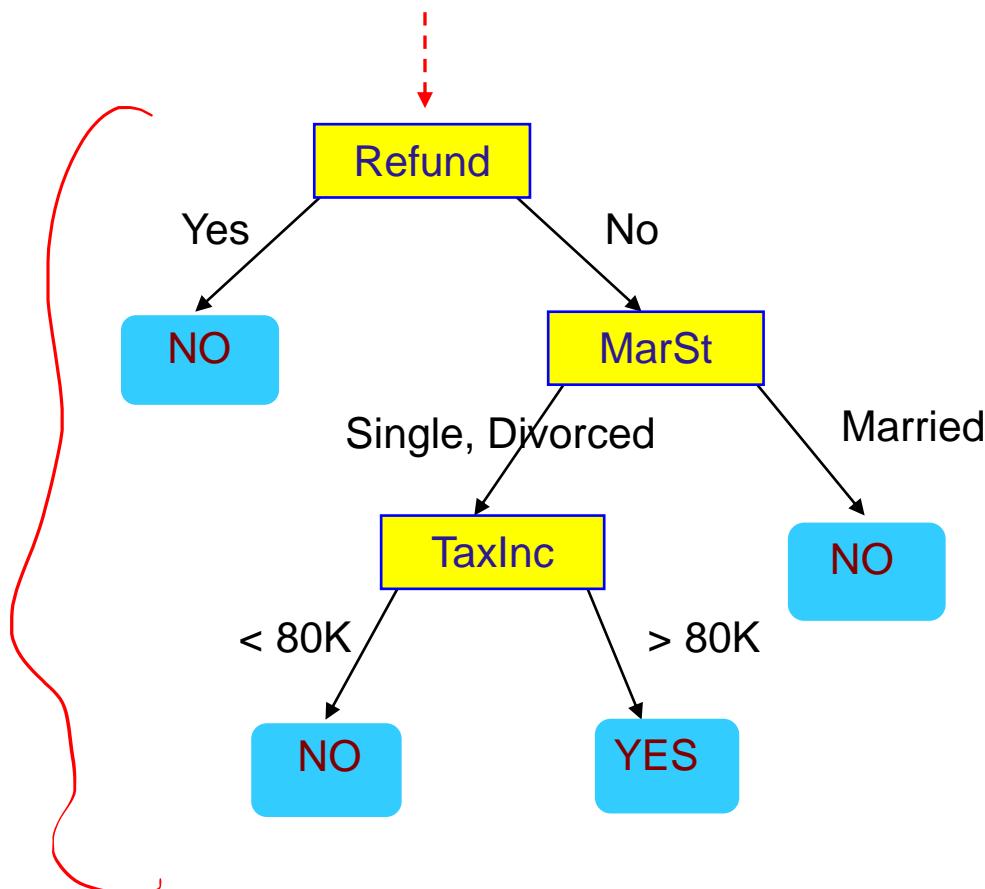
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply Model to Test Data

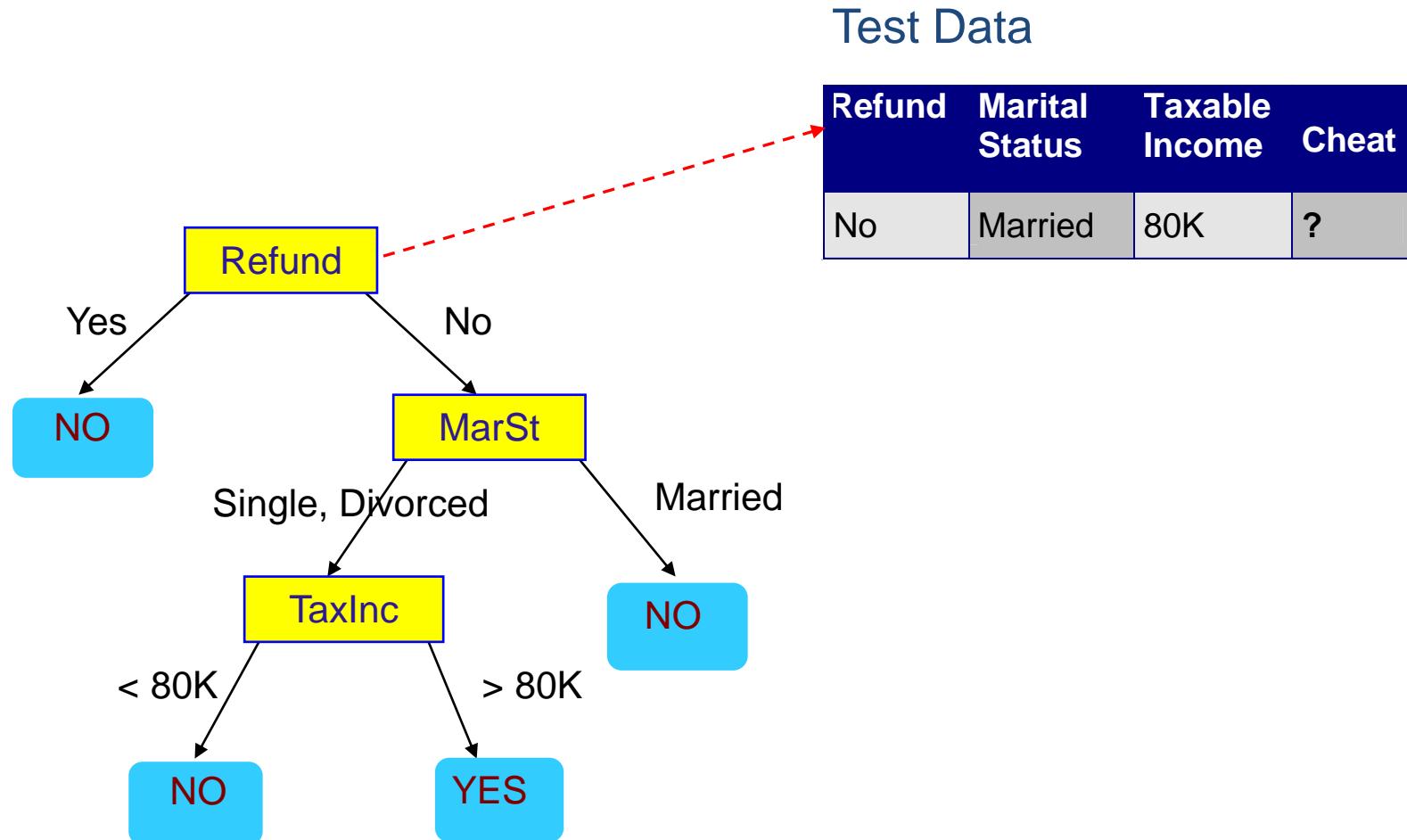
Start from the root of tree.



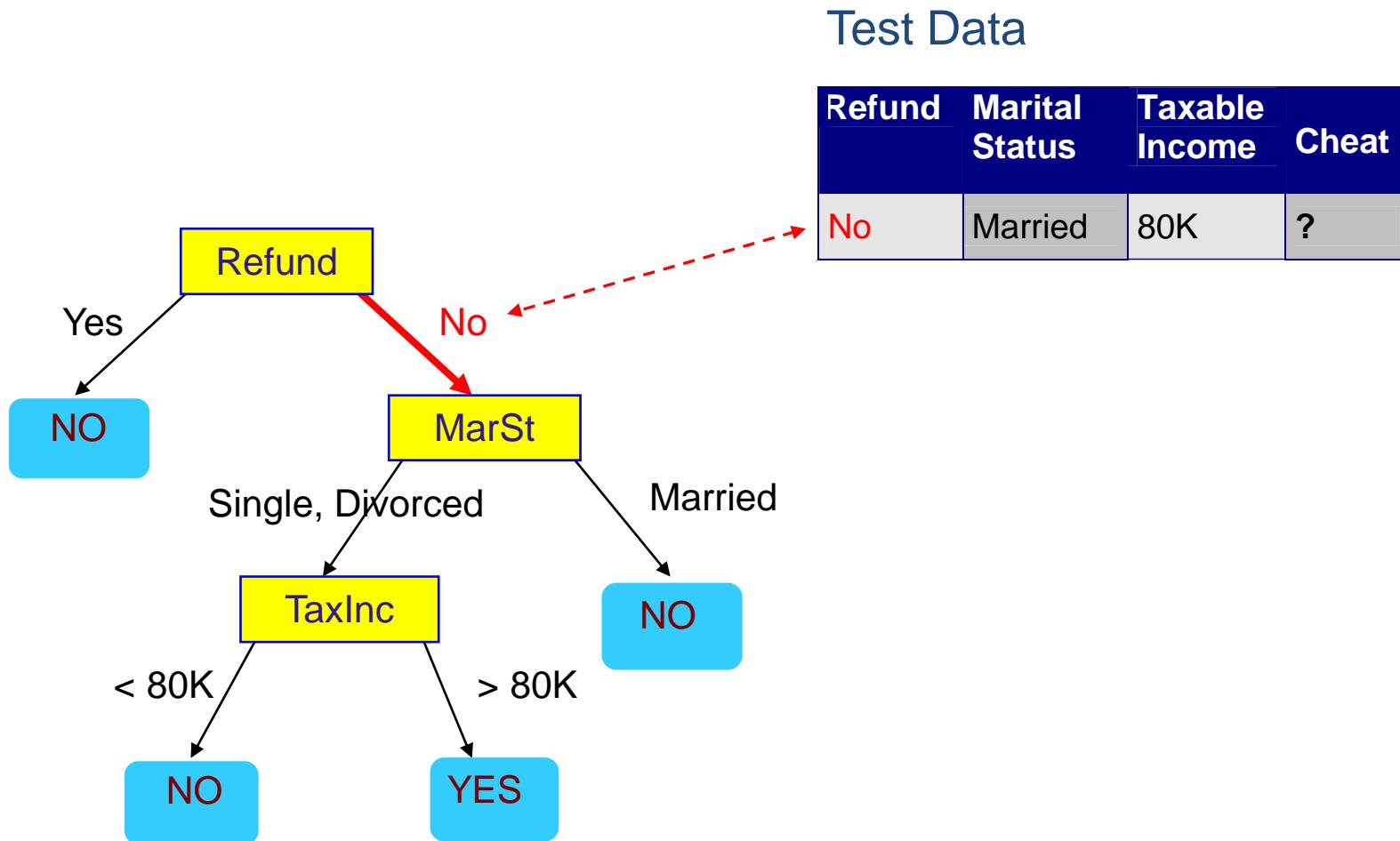
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



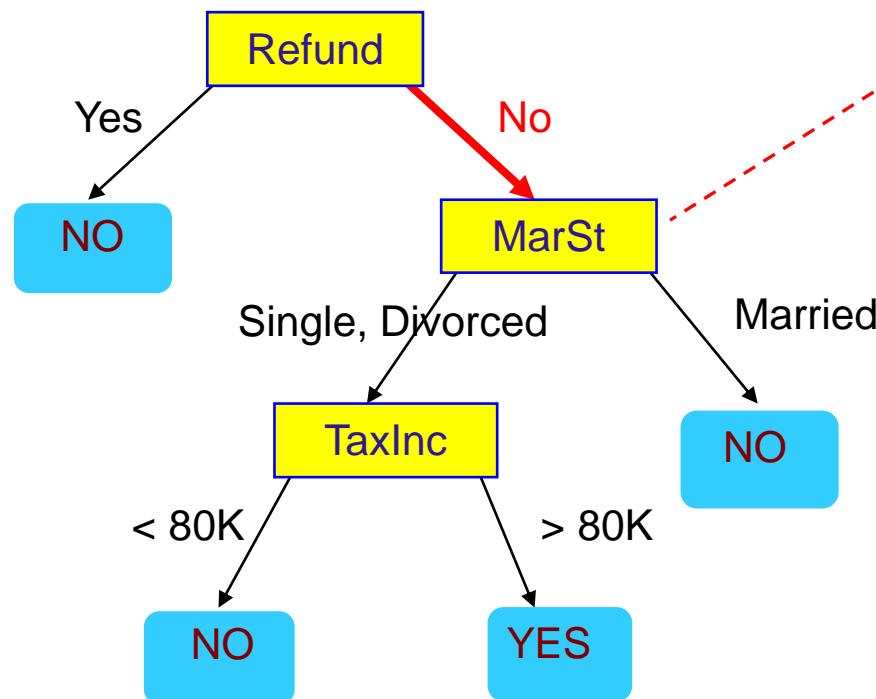
Apply Model to Test Data



Apply Model to Test Data

Test Data

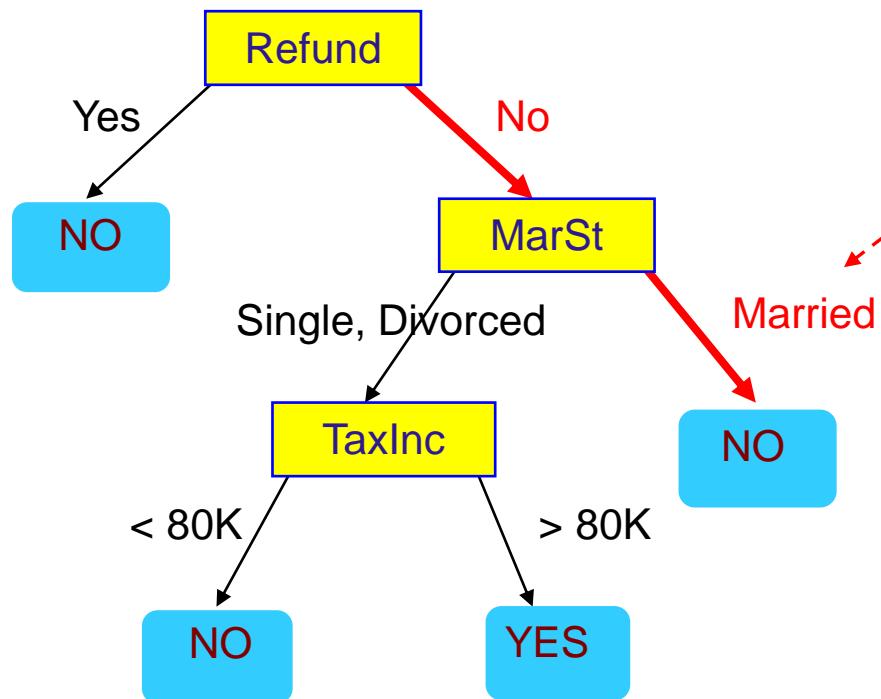
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

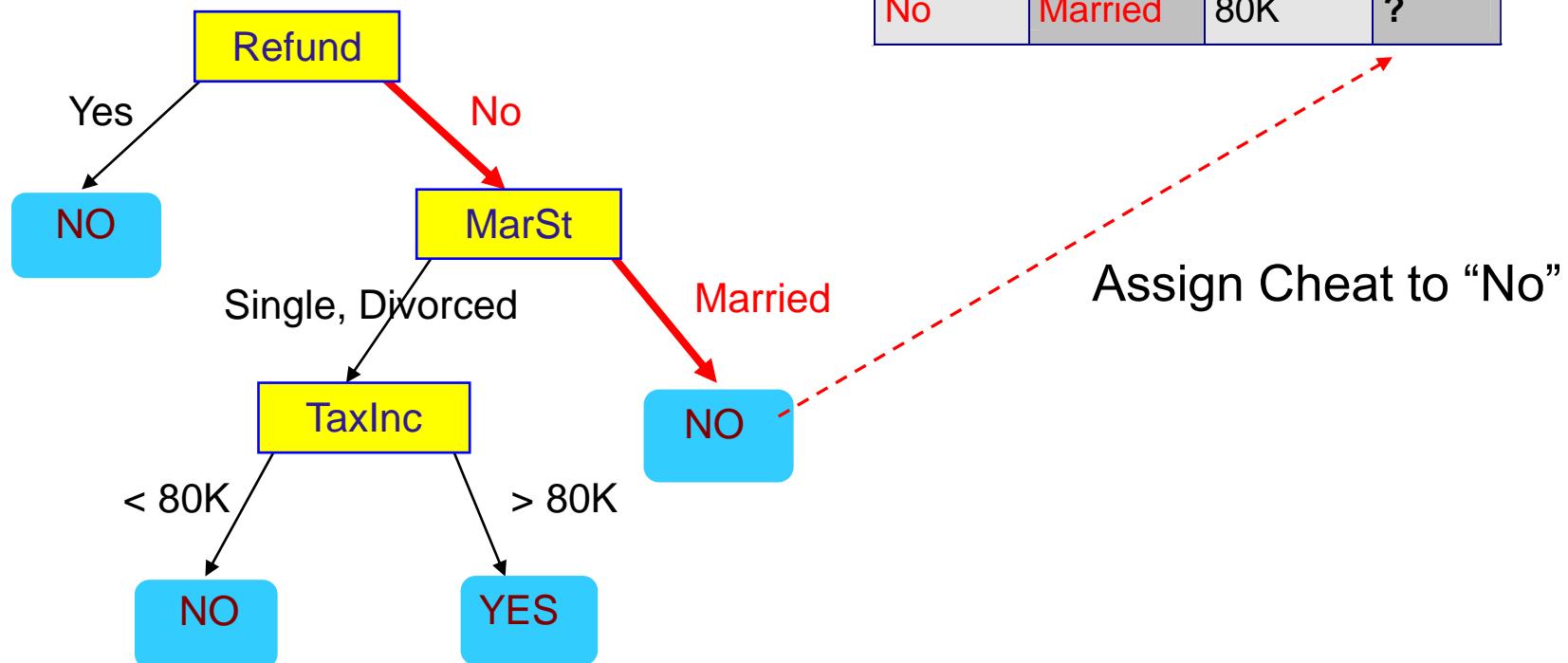
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



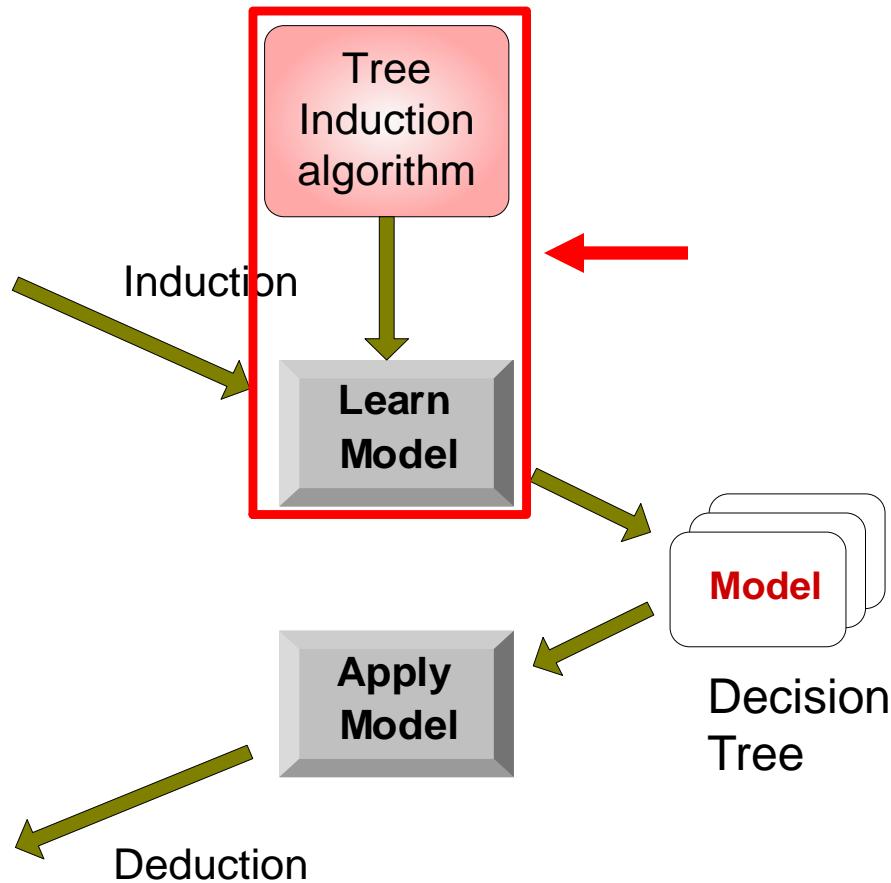
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Tree Induction

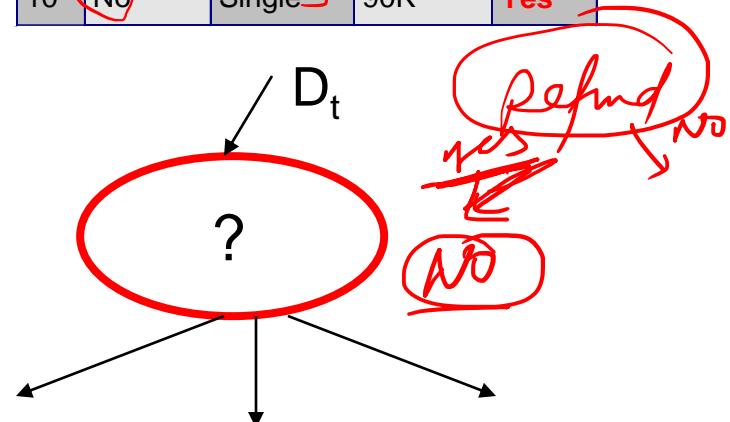
- Finding the best decision tree is NP-hard
- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Many Algorithms:
 - Hunt's Algorithm (one of the earliest)
 - CART
 - ID3, C4.5
 - SLIQ, SPRINT

$D \rightarrow \text{Dataset (table)}, D_t \rightarrow \text{Column 1}$ $D_{\text{Refund}} \rightarrow \text{Column 1}$

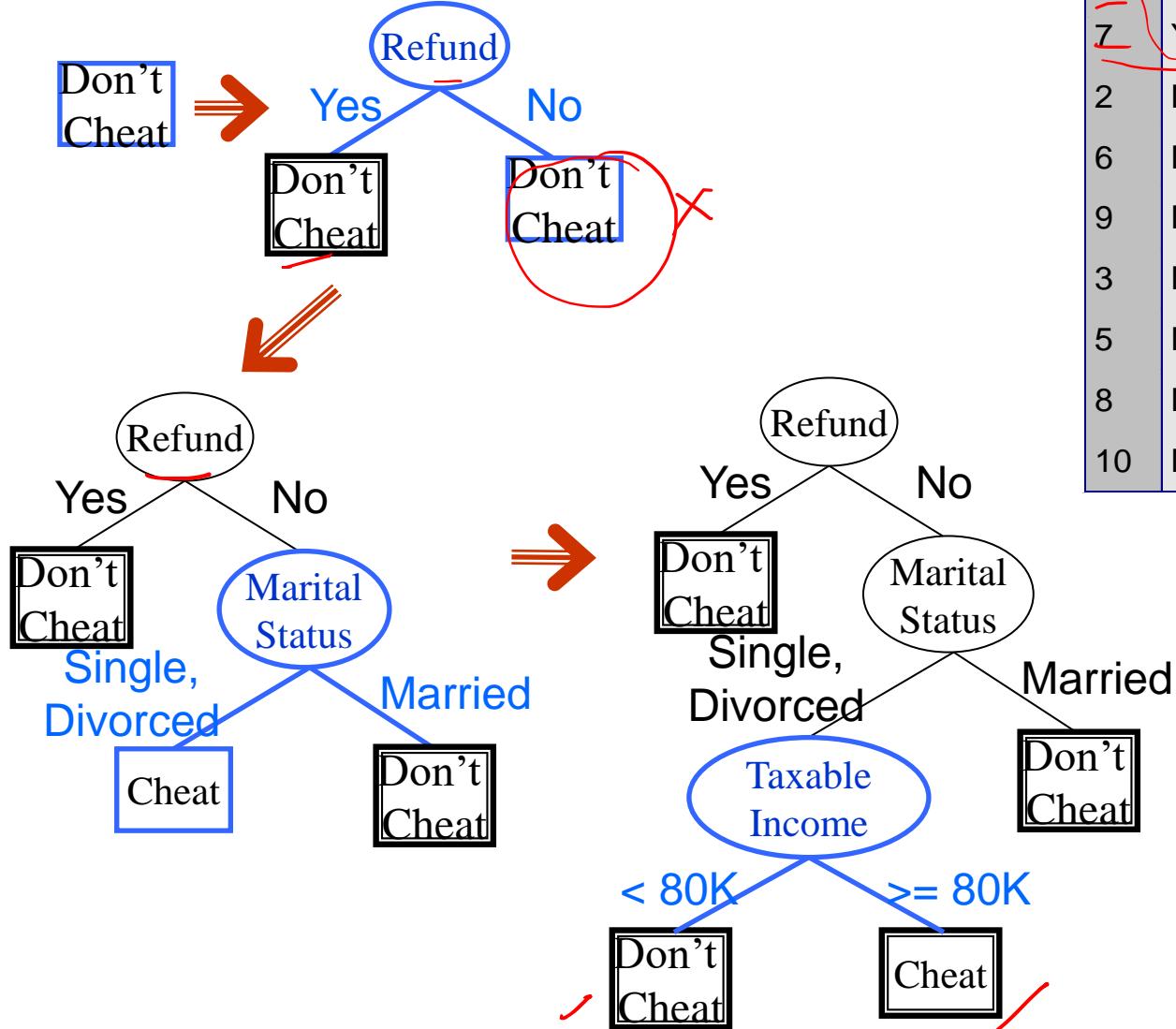
General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
 $D_t \rightarrow D_{\text{Refund}}$.
- General Procedure:
 - If D_t contains records that belong to the same class y_t , then t is a leaf node labeled as y_t
 - If D_t contains records with the same attribute values, then t is a leaf node labeled with the majority class y_t
 - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
 - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.
 - Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No
2	No	Married	100K	No
6	No	Married	60K	No
9	No	Married	75K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
8	No	Single	85K	Yes
10	No	Single	90K	Yes

Constructing decision-trees (pseudocode)

GenDecTree(Sample S , Features F)

1. If **stopping_condition(S, F)** = true then
 - a. **leaf** = **createNode()**
 - b. **leaf.label**= **Classify(S)**
 - c. **return leaf**
2. **root** = **createNode()**
3. **root.test_condition** = **findBestSplit(S, F)**
4. $V = \{v | v \text{ a possible outcome of } \text{root.test_condition}\}$
5. **for each** value $v \in V$:
 - a. $S_v := \{s | \text{root.test_condition}(s) = v \text{ and } s \in S\};$
 - b. **child** = **GenDecTree(S_v, F)** ;
 - c. Add **child** as a descent of **root** and label the edge **(root}→child**) as v
6. **return root**

Tree Induction

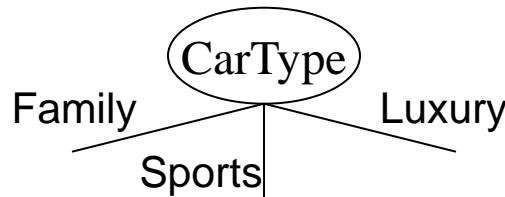
- Issues
 - How to **Classify** a leaf node
 - Assign the **majority class**
 - If leaf is empty, assign the **default class** – the class that has the highest popularity.
 - Determine how to split the records
 - **How to specify the attribute test condition?**
 - **How to determine the best split?**
 - Determine when to stop splitting

How to Specify Test Condition?

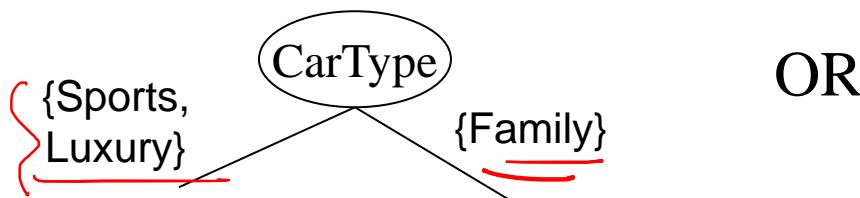
- Depends on attribute types
 - Nominal
 - Ordinal
 - Continuous
- Depends on number of ways to split
 - 2-way split
 - Multi-way split

Splitting Based on Nominal Attributes

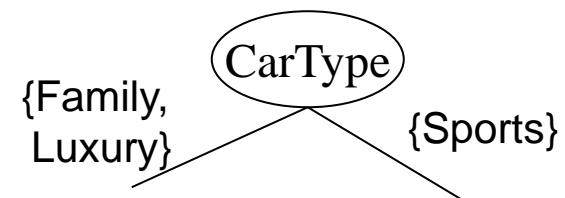
- **Multi-way split:** Use as many partitions as distinct values.



- **Binary split:** Divides values into two subsets.
Need to find optimal partitioning.

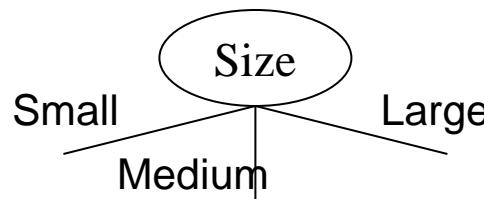


OR

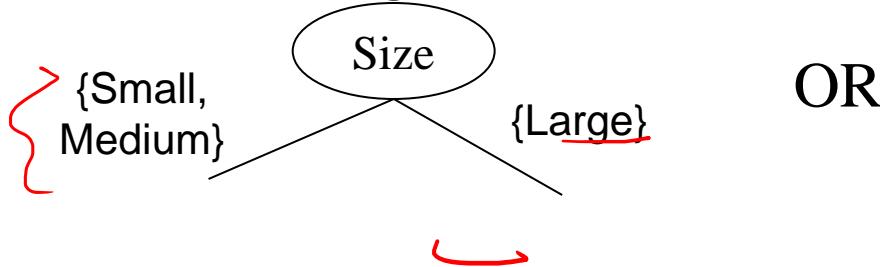


Splitting Based on Ordinal Attributes

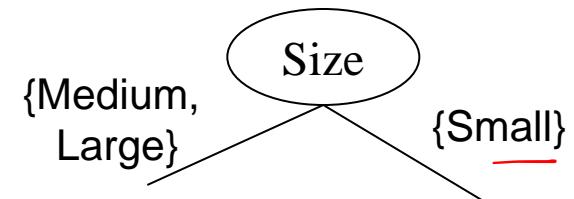
- **Multi-way split:** Use as many partitions as distinct values.



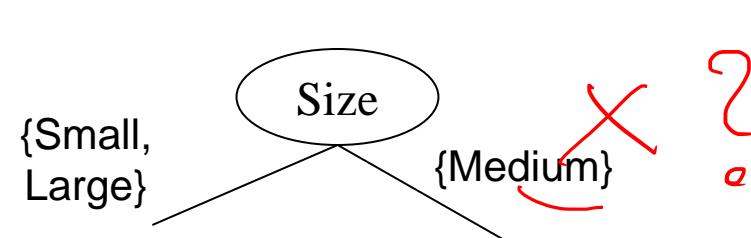
- **Binary split:** Divides values into two subsets – respects the order. Need to find optimal partitioning.



OR



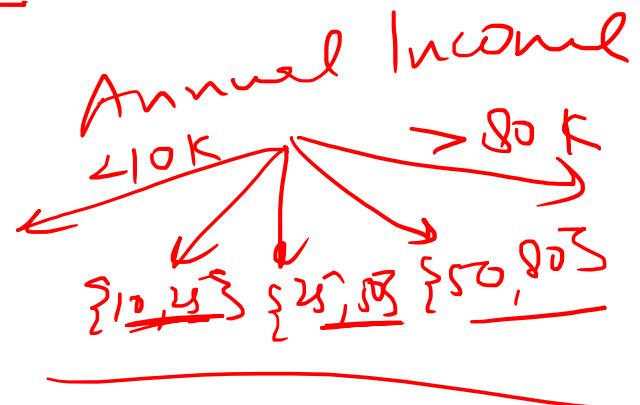
- What about this split?



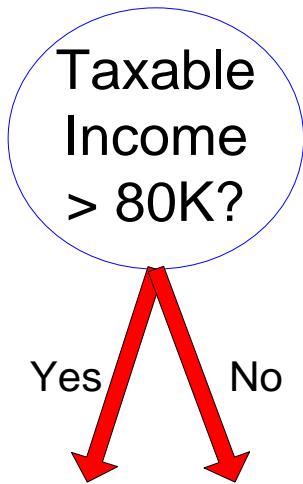
Splitting Based on Continuous Attributes

- Different ways of handling
 - Discretization to form an ordinal categorical attribute
 - Static – discretize once at the beginning
 - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
 - Binary Decision: $(A < v)$ or $(A \geq v)$
 - consider all possible splits and finds the best cut
 - can be more compute intensive

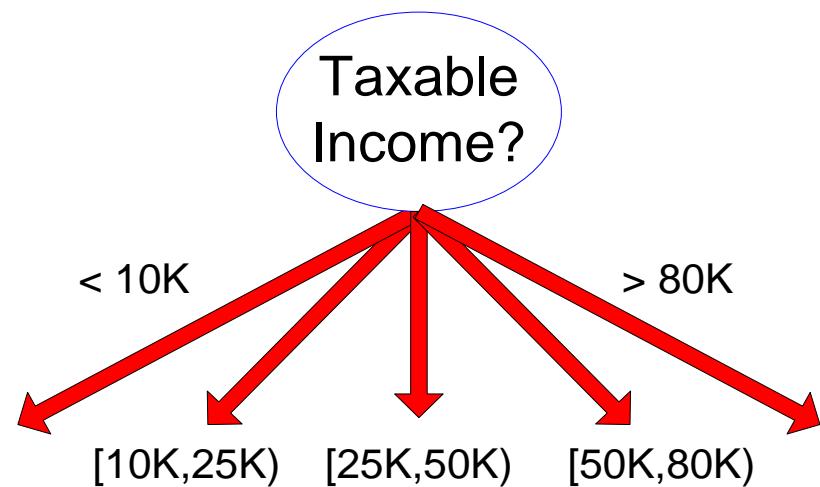
Annual income
yes → no



Splitting Based on Continuous Attributes



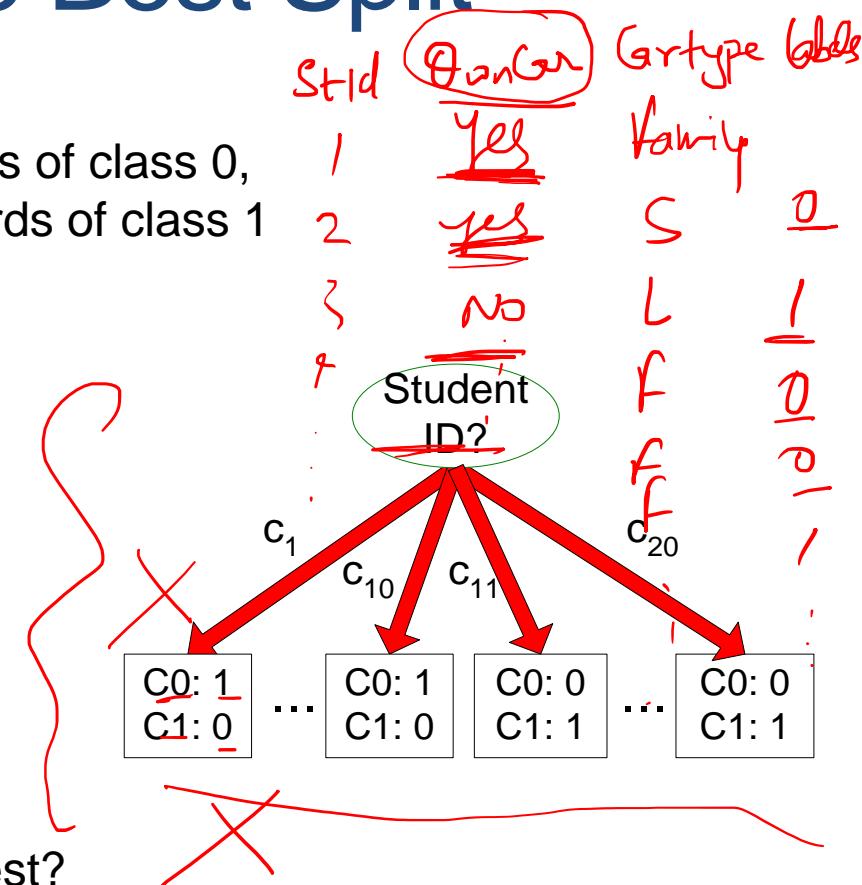
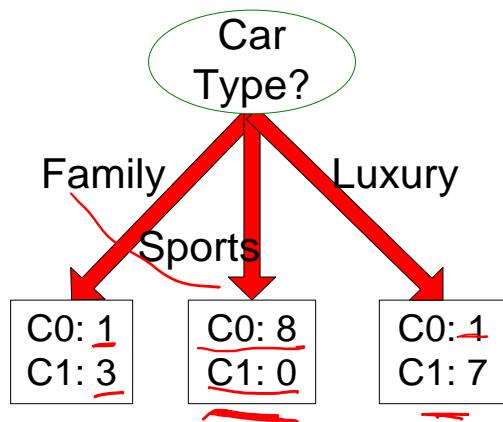
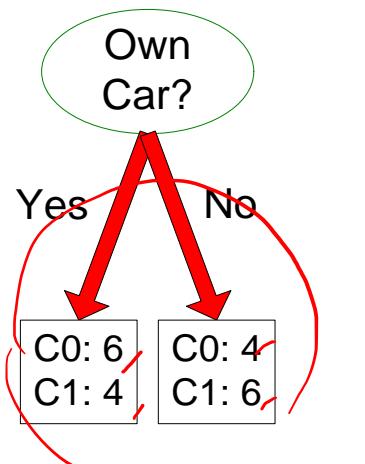
(i) Binary split



(ii) Multi-way split

How to determine the Best Split

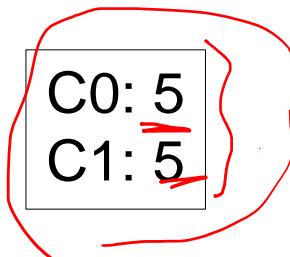
Before Splitting: 10 records of class 0,
10 records of class 1



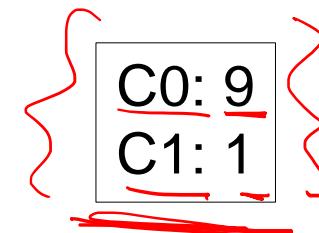
Which test condition is the best?

How to determine the Best Split

- Greedy approach:
 - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:



Non-homogeneous,
High degree of impurity



Homogeneous,
Low degree of impurity

C0: 10
C1: 0

- Ideas?

Measuring Node Impurity

- $p(i|t)$: fraction of records associated with node t belonging to class i $\xrightarrow{\text{yes}) \wedge \text{no}}$ $\xrightarrow{\text{Refund}}$

✓ Entropy(t) = $-\sum_{i=1}^c p(i|t) \log p(i|t)$ $\xrightarrow{0 \log 0 = 0}$ $\xrightarrow{P(O|t) = 0}$

- Used in ID3 and C4.5

✓ Gini(t) = $1 - \sum_{i=1}^c [p(i|t)]^2$

- Used in CART, SLIQ, SPRINT.

✓ Classification error(t) = $1 - \max_i [p(i|t)]$

Gain

$I(\cdot)$ is the impurity measure of the given node
 N is the total no. of records at the parent node | k is the no. of att. values

- **Gain of an attribute split:** compare the impurity of the parent node with the average impurity of the child nodes

$$\Delta = \underline{I(\text{parent})} - \sum_{j=1}^k \frac{\underline{N(v_j)}}{\underline{N}} I(v_j)$$

no. of records associated with the child node v_j

- Maximizing the gain \Leftrightarrow Minimizing the weighted average impurity measure of children nodes
- If $I()$ = Entropy(), then Δ_{info} is called information gain

Example

$$\left. \begin{array}{l} C1 \quad 3 \\ C2 \quad 3 \end{array} \right\} Gini = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$Entropy = -\frac{3}{6} \log_2(3/6) - \frac{3}{6} \log_2(3/6)$$

$$Error = 1 - \max\{3/6, 3/6\} = 0.5 = 1$$

C1	0	1
C2	6	

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

$$Entropy = -0 \log_2 0 - 1 \log_2 1 = -0 - 0 = 0$$

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

$$1 - \sum_{i=1}^C P(i)$$

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\left. \begin{array}{l} Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278 \end{array} \right\}$$

$$Entropy = -(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$$

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

C1	2	4
C2	4	

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

$$Entropy = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

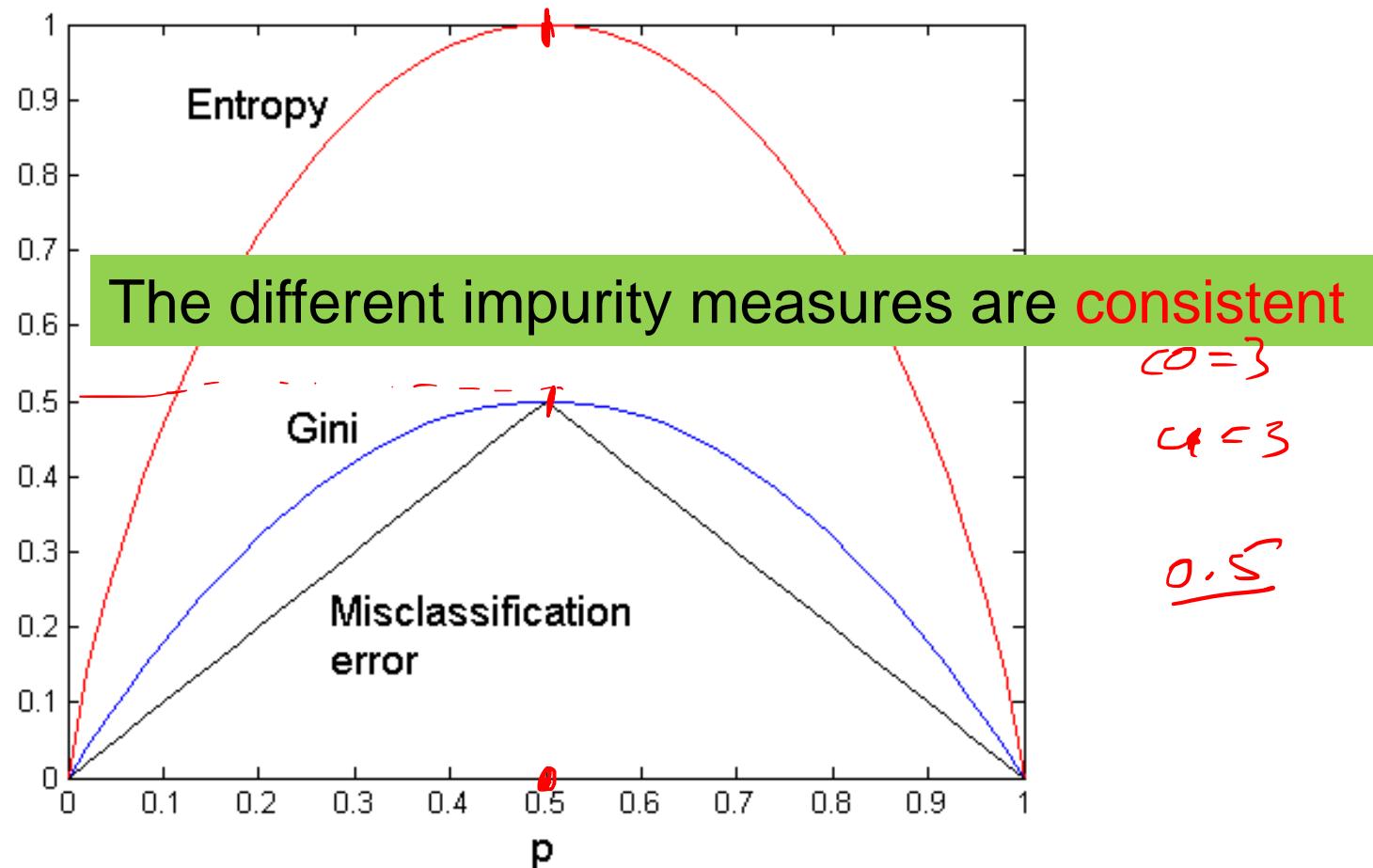
$$Error = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

Impurity measures

- All of the impurity measures take value zero (**minimum**) for the case of a pure node where a single value has probability 1
- All of the impurity measures take **maximum** value when the class distribution in a node is **uniform**.

Comparison among Splitting Criteria

For a 2-class problem:



Categorical Attributes

- For **binary** values split in two
- For **multivalued** attributes, for each distinct value, gather counts for each class in the dataset
 - Use the **count matrix** to make decisions

Multi-way split

CarType			
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

CarType		
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini		0.400

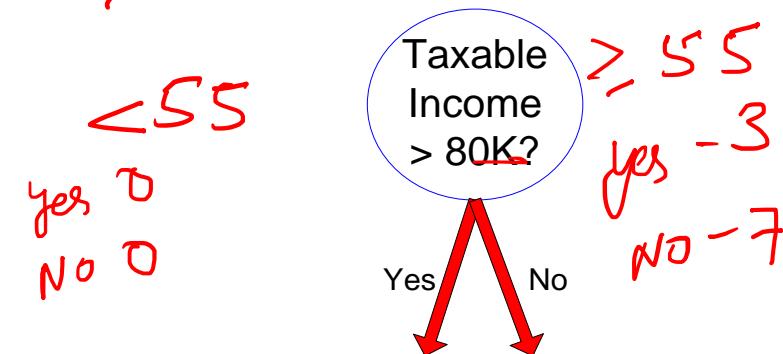
CarType		
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini		0.419

Continuous Attributes



- Use Binary Decisions based on one value
- Choices for the **splitting value**
 - Number of possible splitting values = Number of **distinct values**
- Each **splitting value** has a **count matrix** associated with it
 - Class counts in each of the partitions, $A < v$ and $A \geq v$
- Exhaustive method to choose best v
 - For each v , scan the database to gather count matrix and compute the impurity index
 - Computationally Inefficient! Repetition of work.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Continuous Attributes

$$Gini(\leq 55) = 1 - (0 + 0) = 1$$

$$Gini(\geq 55) = 1 - (3/10)^2 - (7/10)^2 = 0.42$$

- For efficient computation: for each attribute, $Gini = 0/10 \times 1 + 10/10 \times 0.42 = 0 + 0.42 = 0.42$
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing impurity
 - Choose the split position that has the least impurity

$P(\text{just two}) \times Gini(\leq)$
 total no. of records
 + $\times Gini(\geq)$
 =

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
	60	70	75	85	90	95	100	120	125	220
Sorted Values →	55	65	72	80	87	92	97	110	122	172
Split Positions →	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400

Splitting based on impurity

- Impurity measures favor attributes with large number of values
- A test condition with large number of outcomes may not be desirable
 - # of records in each partition is too small to make predictions

Splitting based on INFO

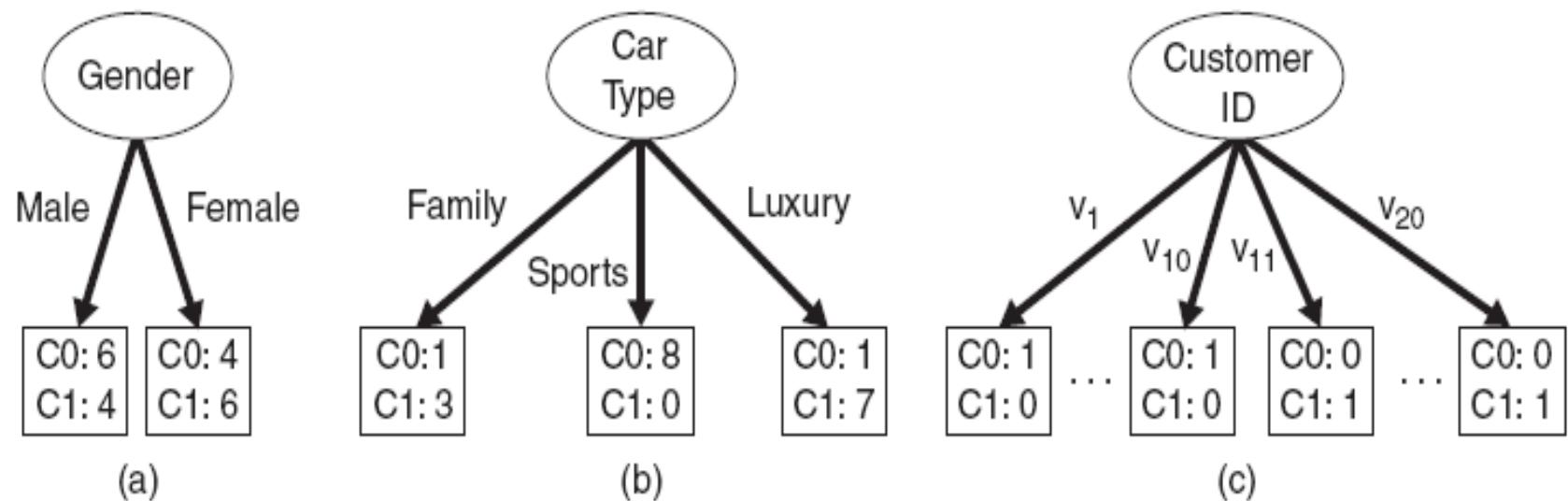


Figure 4.12. Multiway versus binary splits.

Gain Ratio

- Splitting using information gain

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of impurity

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

Decision Tree Based Classification

- Advantages:
 - Inexpensive to construct
 - Extremely fast at classifying unknown records
 - Easy to interpret for small-sized trees
 - Accuracy is comparable to other classification techniques for many simple data sets

Example: C4.5

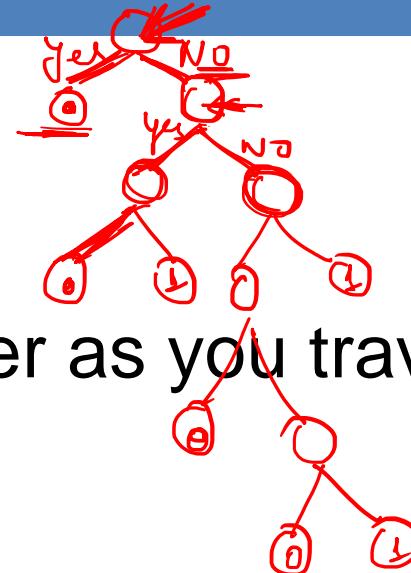
- Simple depth-first construction.
- Uses Information Gain
- Sorts Continuous Attributes at each node.
- Needs entire data to fit in memory.
- Unsuitable for Large Datasets.
 - Needs out-of-core sorting.
- You can download the software from:
<http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz>

Other Issues

- Data Fragmentation
- Expressiveness

Data Fragmentation

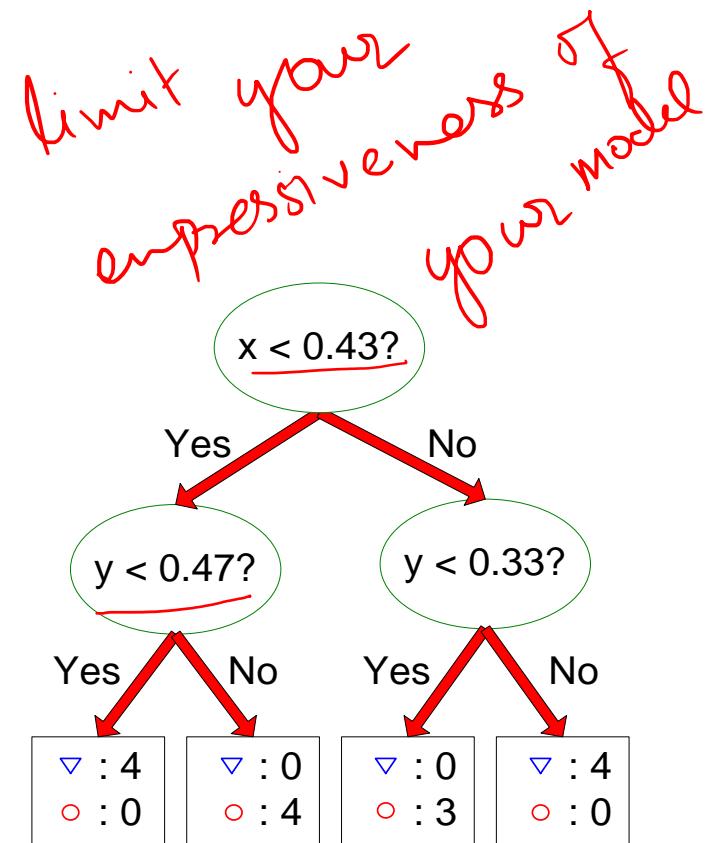
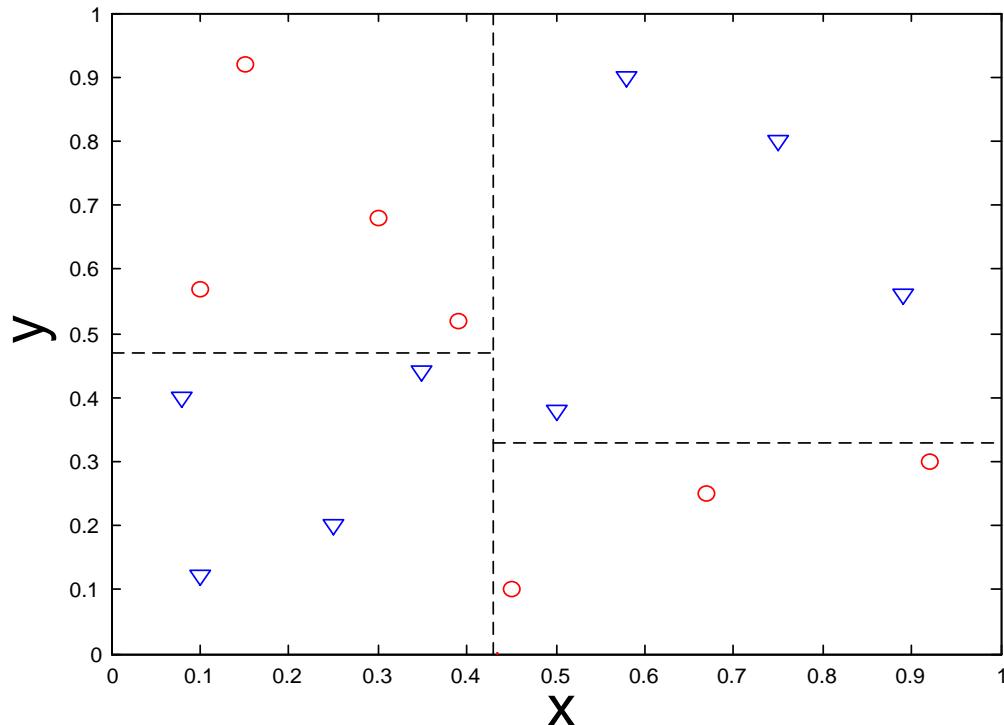
- Number of instances gets smaller as you traverse down the tree
- Number of instances at the leaf nodes could be too small to make any statistically significant decision
- You can introduce a lower bound on the number of items per leaf node in the stopping criterion.



Expressiveness

- A classifier defines a **function** that discriminates between two (or more) classes.
- The **expressiveness** of a classifier is the **class of functions** that it can model, and the kind of data that it can **separate**
 - When we have **discrete** (or binary) values, we are interested in the class of **boolean functions** that can be modeled
 - If the data-points are real vectors we talk about the **decision boundary** that the classifier can model

Decision Boundary

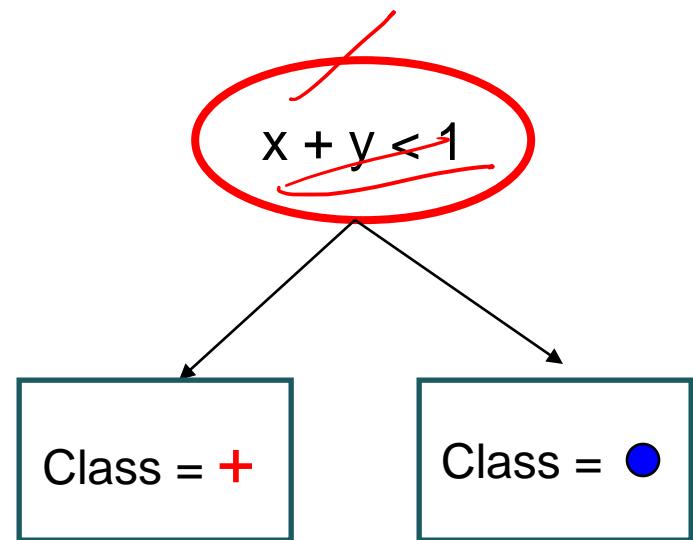
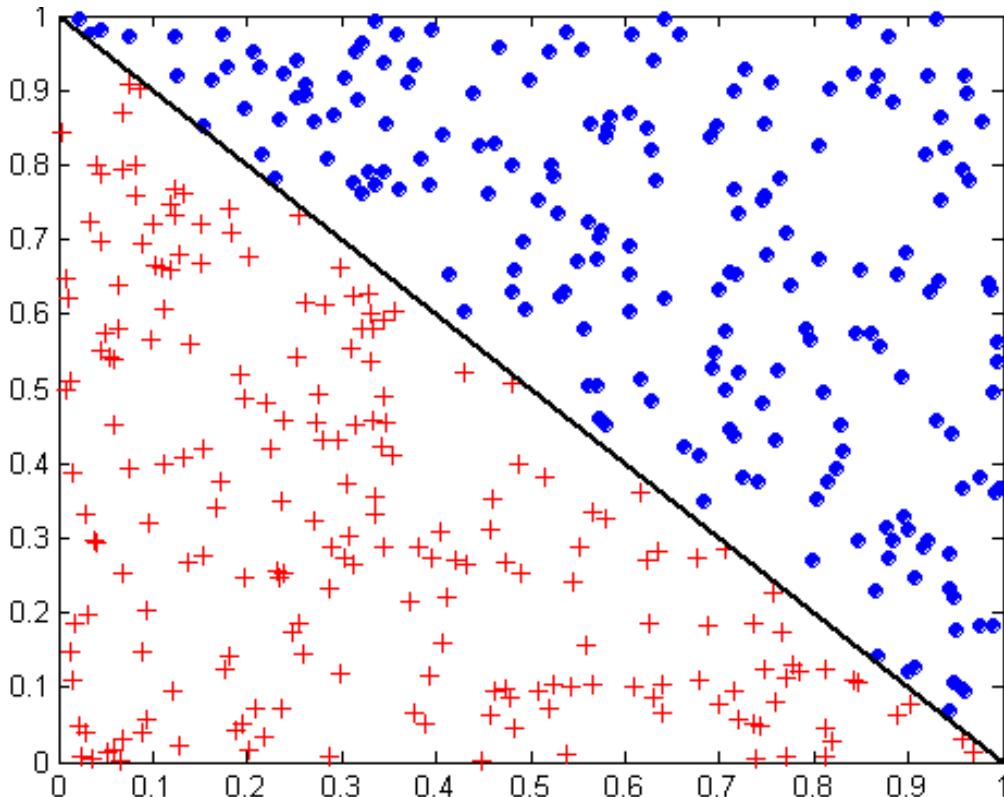


- Border line between two neighboring regions of different classes is known as **decision boundary**
- Decision boundary is **parallel to axes** because test condition involves a single attribute at-a-time

Expressiveness

- Decision tree provides **expressive** representation for learning discrete-valued function
 - But they do not generalize well to certain types of Boolean functions
 - Example: **parity function**:
 - Class = 1 if there is an **even** number of Boolean attributes with truth value = True
 - Class = 0 if there is an **odd** number of Boolean attributes with truth value = True
 - For accurate modeling, must have a complete tree
- Less expressive for modeling continuous variables
 - Particularly when test condition involves only a single attribute at-a-time

Oblique Decision Trees

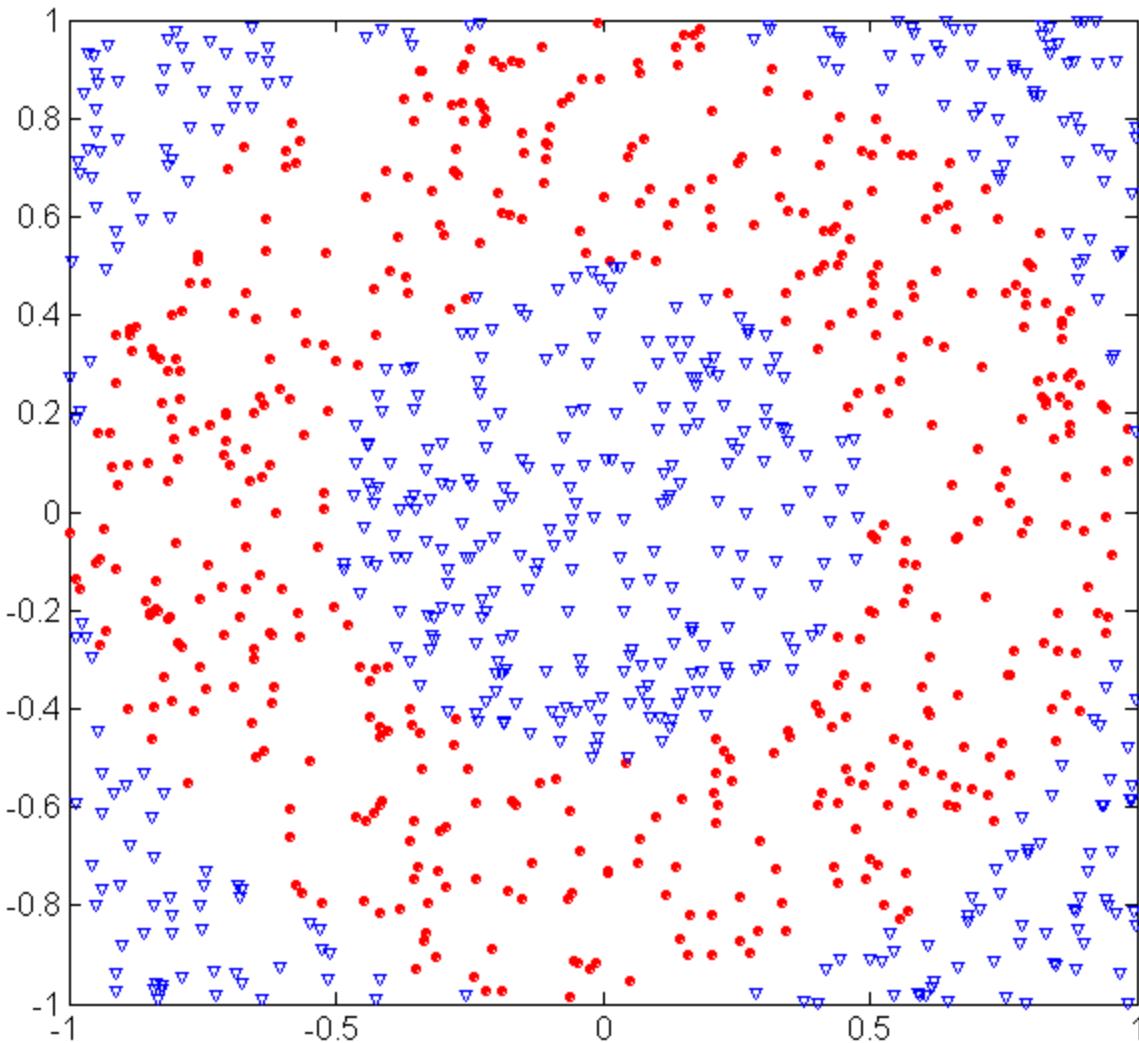


- Test condition may involve multiple attributes
- More expressive representation
- Finding optimal test condition is computationally expensive

Practical Issues of Classification

- Underfitting and Overfitting
- Evaluation

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

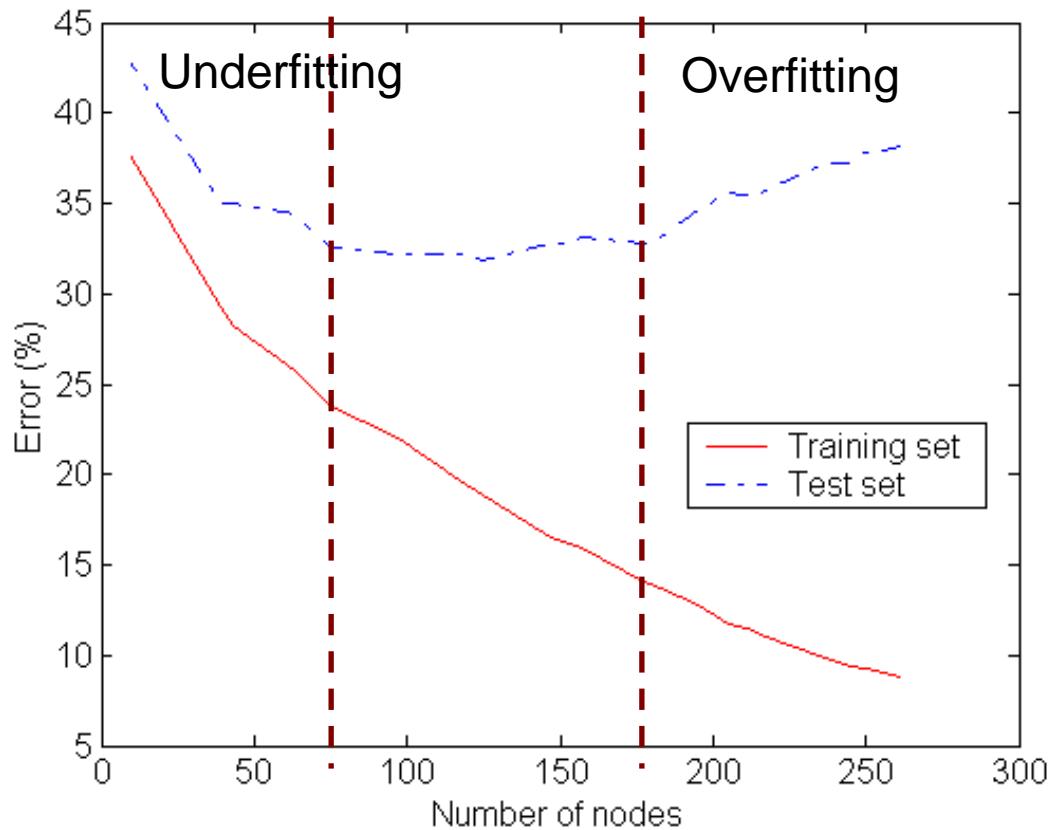
$$0.5 \leq \sqrt{x_1^2+x_2^2} \leq 1$$

Triangular points:

$$\sqrt{x_1^2+x_2^2} > 0.5 \text{ or}$$

$$\sqrt{x_1^2+x_2^2} < 1$$

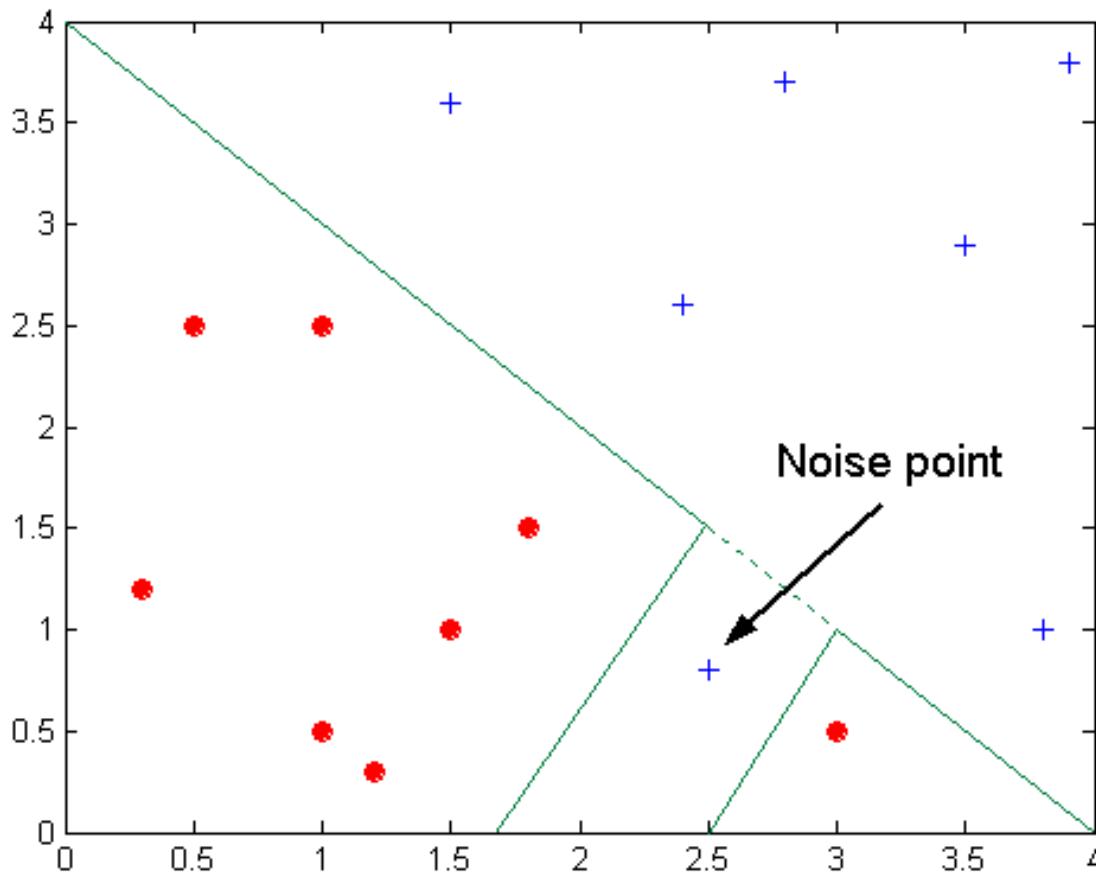
Underfitting and Overfitting



Underfitting: when model is **too simple**, both training and test errors are large

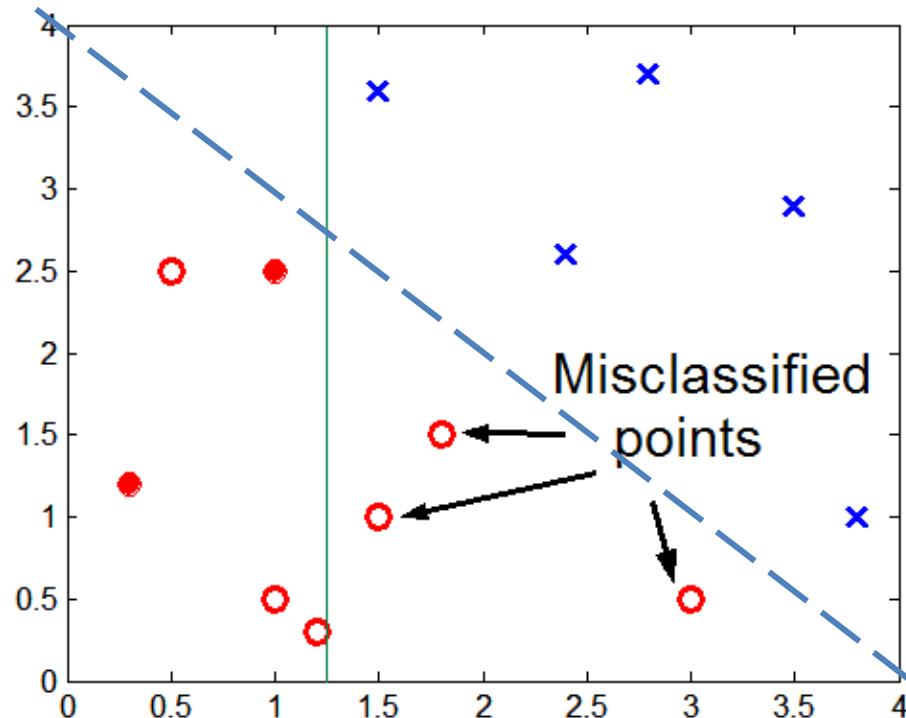
Overfitting: when model is **too complex** it models the details of the training set and fails on the test set

Overfitting due to Noise



Decision boundary is distorted by noise point

Overfitting due to Insufficient Examples



Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region

- Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
 - The model does not **generalize** well
- Need new ways for estimating errors

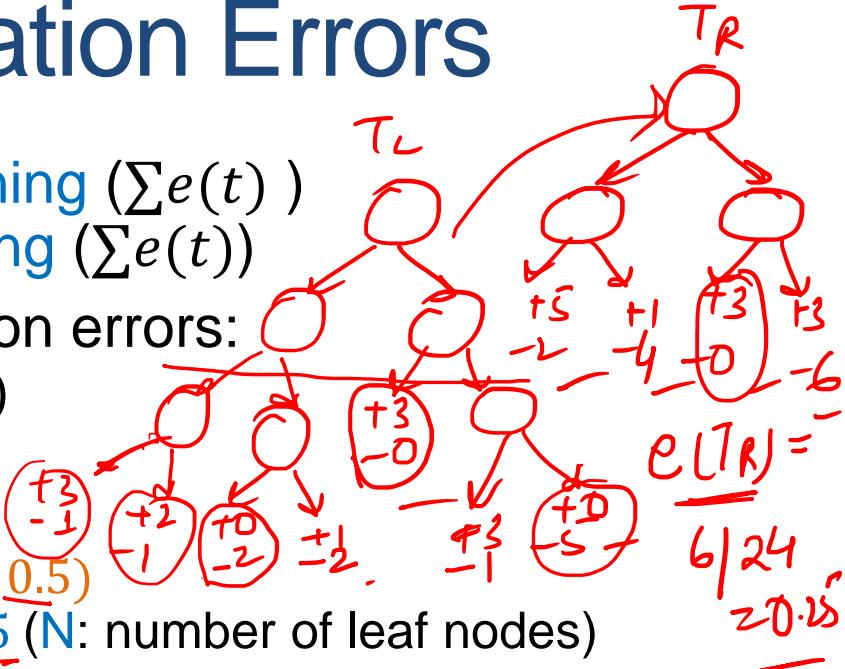
$$\Omega(t) = 0.5; \quad \Omega(t') = 1$$

Estimating Generalization Errors

- Re-substitution errors: error on training ($\sum e(t)$)
- Generalization errors: error on testing ($\sum e(t')$)
- Methods for estimating generalization errors:
 - Optimistic approach: $e'(t) = e(t)$

- Pessimistic approach:

- For each leaf node: $e'(t) = (\underline{e(t)} + \underline{0.5})$
- Total errors: $e'(T) = \underline{e(T)} + \underline{N} \times 0.5$ (N : number of leaf nodes)
 - Penalize large trees
- For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances)
 - Training error = $10/1000 = 1\%$
 - Generalization error = $(10 + 30 \times 0.5)/1000 = 2.5\%$



- Using validation set:

- Split data into training, validation, test
- Use validation dataset to estimate generalization error
- Drawback: less data for training.

$$e_g(T_R) = \underline{10} = 0.417$$

$$e_g(T_L) = \underline{\underline{4}} = 0.167$$

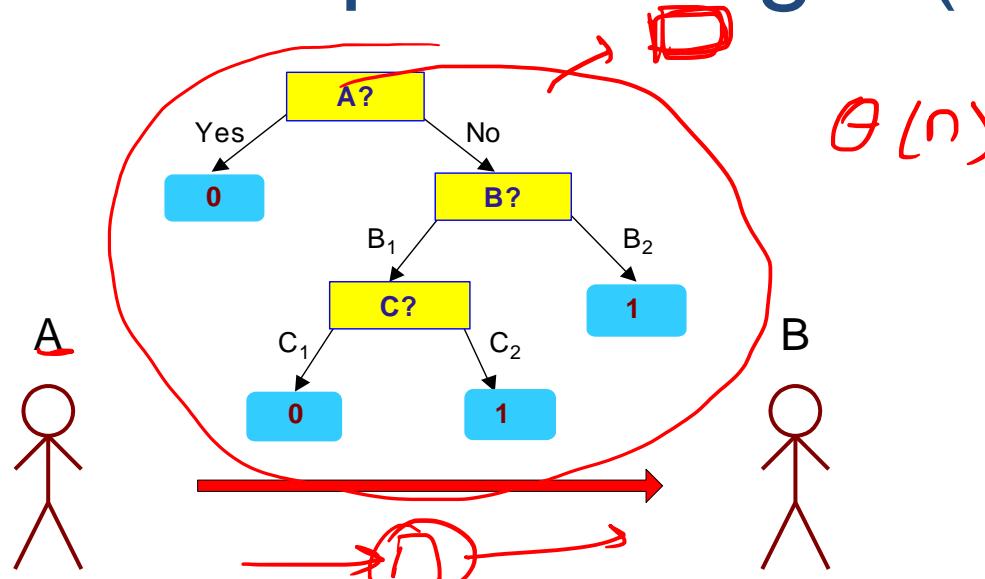
$$e_g(T_R) = \frac{24}{6+4} = 0.375$$

Occam's Razor *(principle of parsimony)*

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

Minimum Description Length (MDL)

X	y
X_1	1
X_2	0
X_3	0
X_4	1
...	...
X_n	1



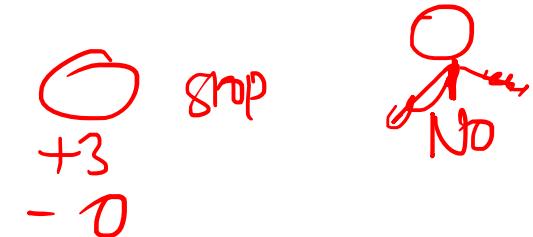
X	y
X_1	?
X_2	?
X_3	?
X_4	?
...	...
X_n	?

- $\text{Cost}(\text{Model}, \text{Data}) = \text{Cost}(\text{Data}|\text{Model}) + \text{Cost}(\text{Model})$
 - Search for the least costly model.
 - $\text{Cost}(\text{Data}|\text{Model})$ encodes the misclassification errors.
 - $\text{Cost}(\text{Model})$ encodes the decision tree
 - node encoding (number of children) plus splitting condition encoding.
- \nearrow less complex models

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)

- Stop the algorithm before it becomes a fully-grown tree
- Typical stopping conditions for a node:
 - Stop if all instances belong to the same class
 - Stop if all the attribute values are the same



- More **restrictive** conditions:

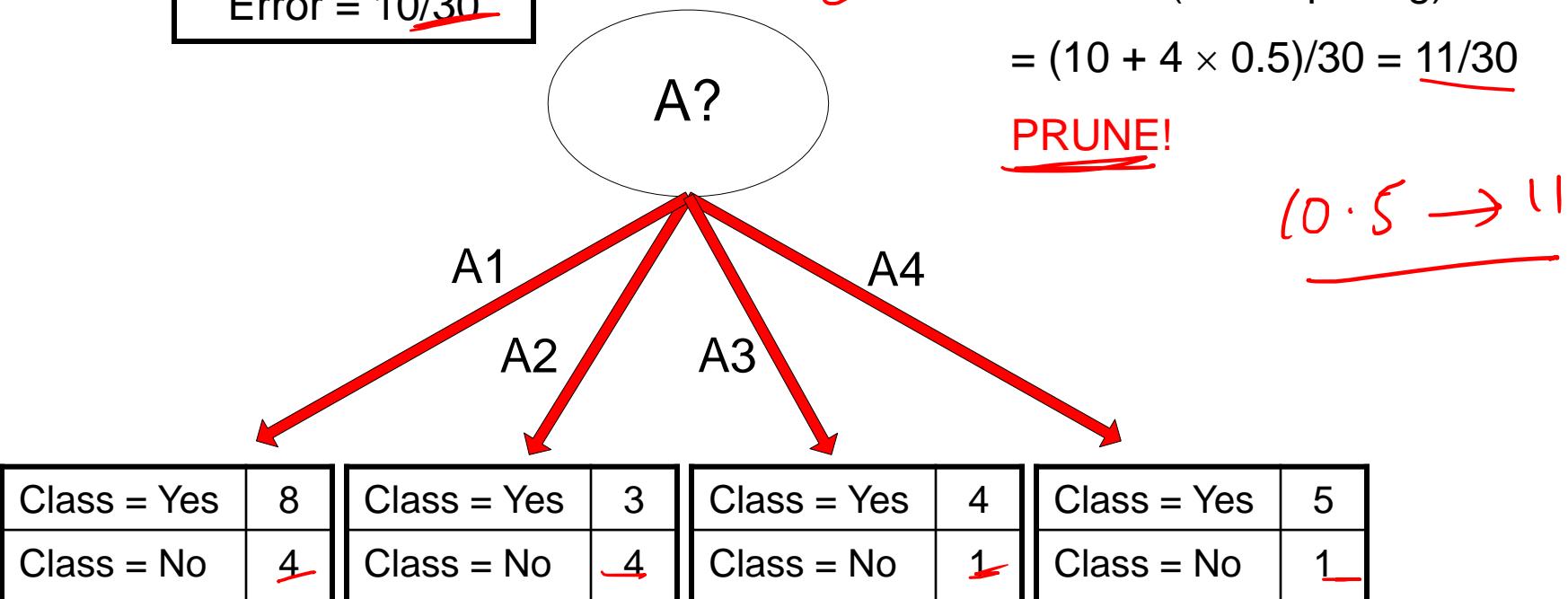
- Stop if **number of instances** is less than some user-specified threshold
- Stop if class distribution of instances are **independent** of the available features (e.g., using χ^2 test)
- Stop if expanding the current node **does not improve impurity** measures (e.g., Gini or information gain).

How to Address Overfitting...

- Post-pruning
 - Grow decision tree to its entirety
 - Trim the nodes of the decision tree in a bottom-up fashion
 - If generalization error improves after trimming, replace sub-tree by a leaf node.
 - Class label of leaf node is determined from majority class of instances in the sub-tree
- Can use MDL for post-pruning

Example of Post-Pruning

Class = Yes	20
Class = No	10
Error = 10/30	



Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?
- Methods for Model Comparison
 - How to compare the relative performance among competing models?

Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?
- Methods for Model Comparison
 - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the **predictive capability** of a model
 - Rather than how fast it takes to classify or build models, scalability, etc.
- **Confusion Matrix:**

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a	b
	Class>No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)
—

Metrics for Performance Evaluation...

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Limitation of Accuracy

- Consider a 2-class problem
 - Number of Class 0 examples = 9990 ← w_1 lower weight
 - Number of Class 1 examples = 10 ← w_2 higher weight
- If model predicts everything to be class 0, accuracy is $9990/10000 = \underline{99.9\%}$
 - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

		PREDICTED CLASS	
		C(i j)	Class=Yes
ACTUAL CLASS	C(i j)	a	Class=No
	Class=Yes	$C(\text{Yes} \text{Yes})$ $P(\text{Yes} \text{Yes})$	$C(\text{No} \text{Yes})$ $P(\text{No} \text{Yes})$
	Class=No	$C(\text{Yes} \text{No})$ $P(\text{Yes} \text{No})$	$C(\text{No} \text{No})$ $P(\text{No} \text{No})$

C(i|j): Cost of classifying class j example as class i

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

?

Computing Cost of Classification

Cost Matrix		PREDICTED CLASS	
ACTUAL CLASS	$C(i j)$	+	-
	+	-1	100
	-	1	0

Model M ₁	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%



Cost = 3910

Model M ₂	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%



Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	a	b
	Class>No	c	d

Accuracy is proportional to cost if
 1. $C(Yes|No)=C(No|Yes) = q$
 2. $C(Yes|Yes)=C(No|No) = p$

$$N = a + b + c + d$$

$$\text{Accuracy} = (a + d)/N$$

Cost	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	p	q
	Class>No	q	p

$$\begin{aligned}
 \text{Cost} &= p (a + d) + q (b + c) \\
 &= p (a + d) + q (N - a - d) \\
 &= q N - (q - p)(a + d) \\
 &= N [q - (q-p) \times \text{Accuracy}]
 \end{aligned}$$

Precision-Recall

$$\text{Precision (p)} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

$$\text{Recall (r)} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

$$\text{F-measure (F)} = \frac{1}{\left(\frac{1/r + 1/p}{2}\right)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} = \frac{2TP}{2TP+FP+FN}$$

$$\frac{2 \times P \times R}{P+R}$$

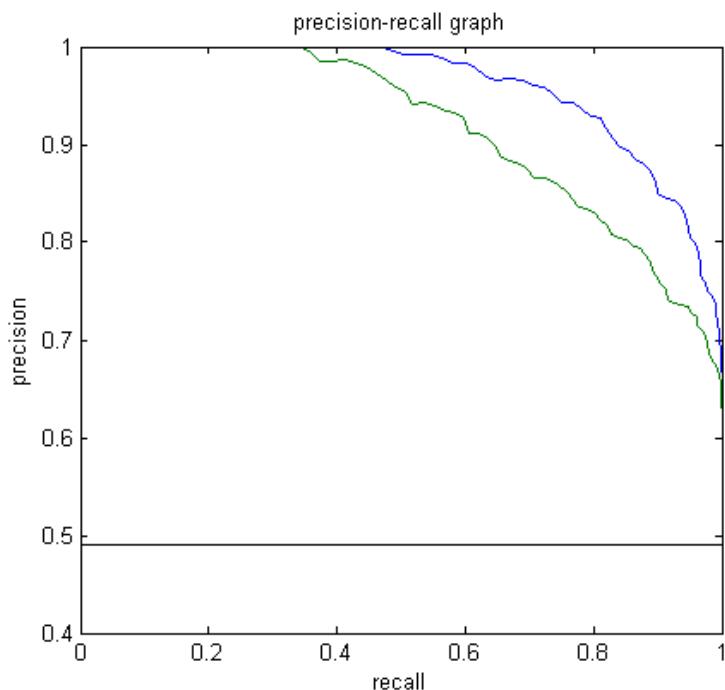
rate of true results
out of the total positive
records detected by the model

Count	PREDICTED CLASS	
	Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a
	Class>No	c

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

Precision-Recall plot

- Usually for parameterized models, it controls the precision/recall tradeoff



Model Evaluation

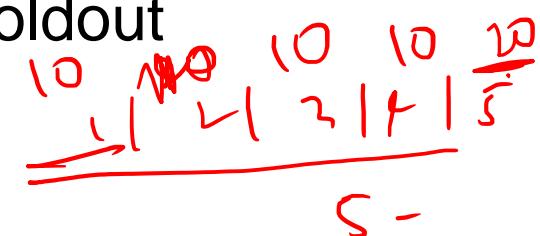
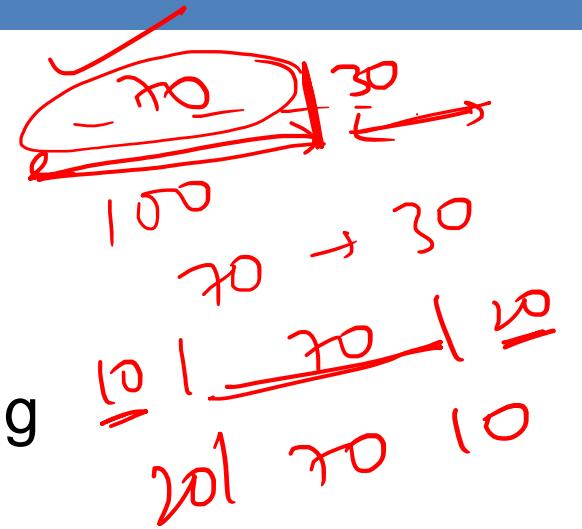
- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?
- Methods for Model Comparison
 - How to compare the relative performance among competing models?

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?
- Performance of a model may depend on other factors besides the learning algorithm:
 - Class distribution ✓
 - Cost of misclassification ✓
 - Size of training and test sets ✓

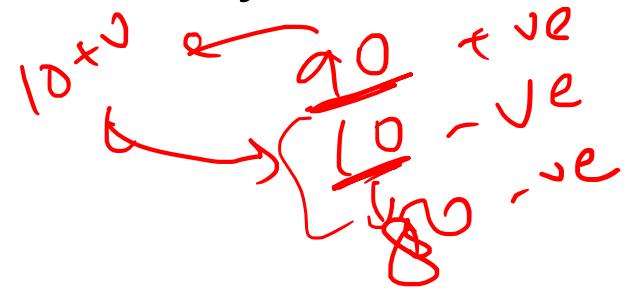
Methods of Estimation

- Holdout
 - Reserve **2/3** for training and **1/3** for testing
- Random subsampling
 - One sample may be biased -- Repeated holdout
- Cross validation
 - Partition data into **k** disjoint subsets
 - **k**-fold: train on **k-1** partitions, test on the remaining one
 - Leave-one-out: **k=n**
 - Guarantees that each record is used the same number of times for training and testing
- Bootstrap
 - Sampling with replacement
 - ~63% of records used for training, ~27% for testing

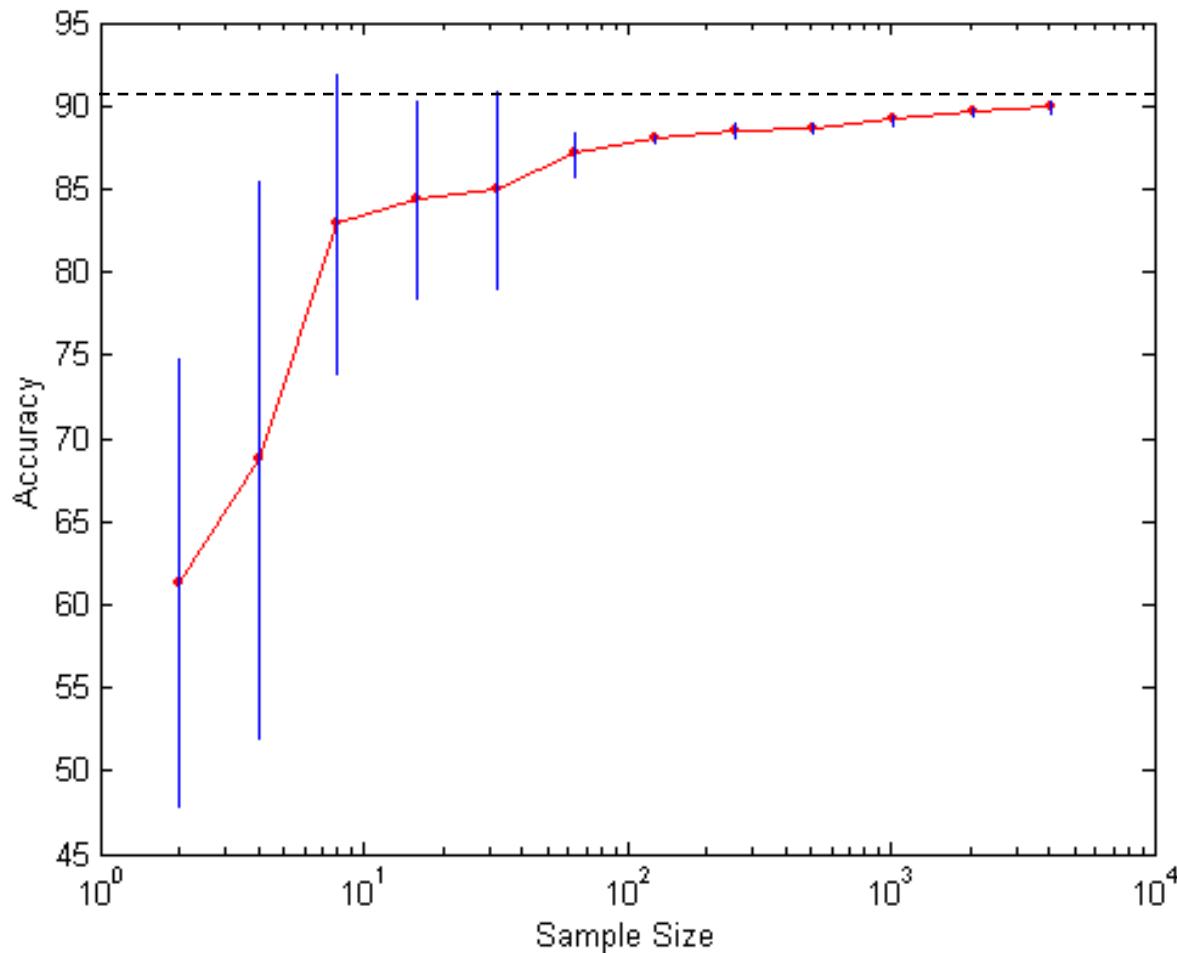


Dealing with class Imbalance

- If the class we are interested in is very rare, then the classifier will ignore it.
 - The class imbalance problem
- Solution
 - We can modify the optimization criterion by using a cost sensitive metric
 - We can **balance** the class distribution
 - Sample from the larger class so that the size of the two classes is the same
 - Replicate the data of the class of interest so that the classes are balanced
 - Over-fitting issues



Learning Curve



- Learning curve shows how accuracy changes with varying sample size
 - Requires a sampling schedule for creating learning curve
- Effect of small sample size:
- Bias in the estimate
 - Variance of estimate

Model Evaluation

- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?
- Methods for Model Comparison
 - How to compare the relative performance among competing models?

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
 - Characterize the trade-off between positive hits and false alarms
- **ROC** curve plots **TPR** (on the **y**-axis) against **FPR** (on the **x**-axis)

$$\underline{\text{TPR}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \underline{\text{Recall}}$$

Fraction of **positive instances** predicted **correctly**

$$\underline{\text{FPR}} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Fraction of **negative instances** predicted **incorrectly**

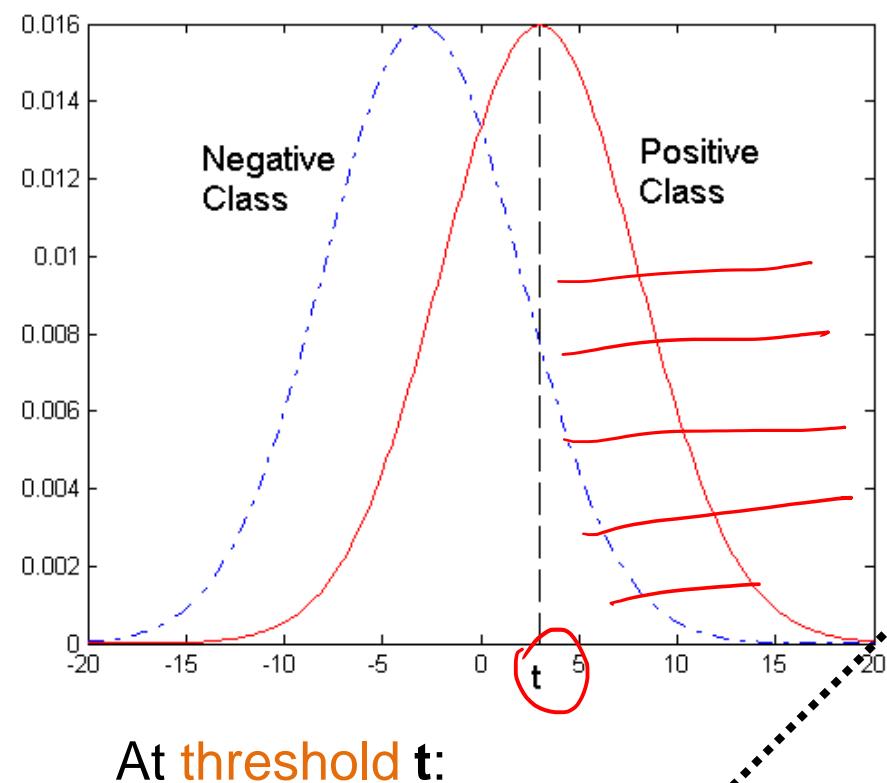
		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

ROC (Receiver Operating Characteristic)

- Performance of a classifier represented as a **point** on the **ROC** curve
- Changing some parameter of the algorithm, sample distribution or cost matrix changes the location of the point

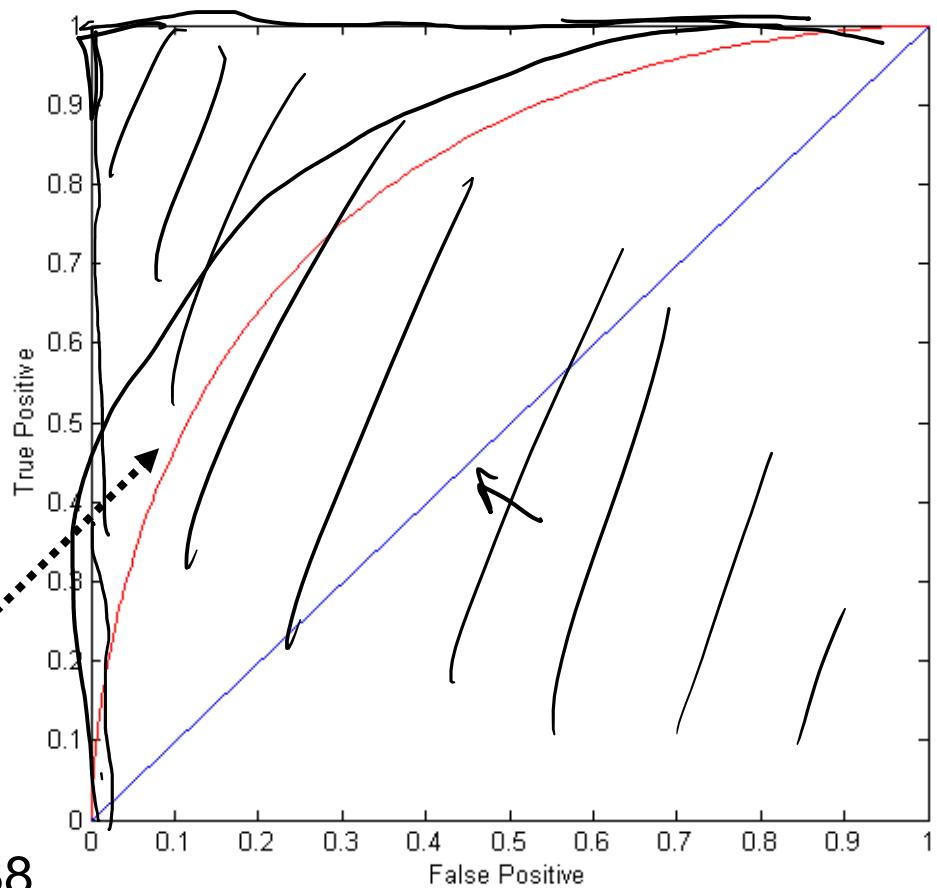
ROC Curve

- 1-dimensional data set containing 2 classes (**positive** and **negative**)
- any points located at $x > t$ is classified as **positive**



At threshold t :

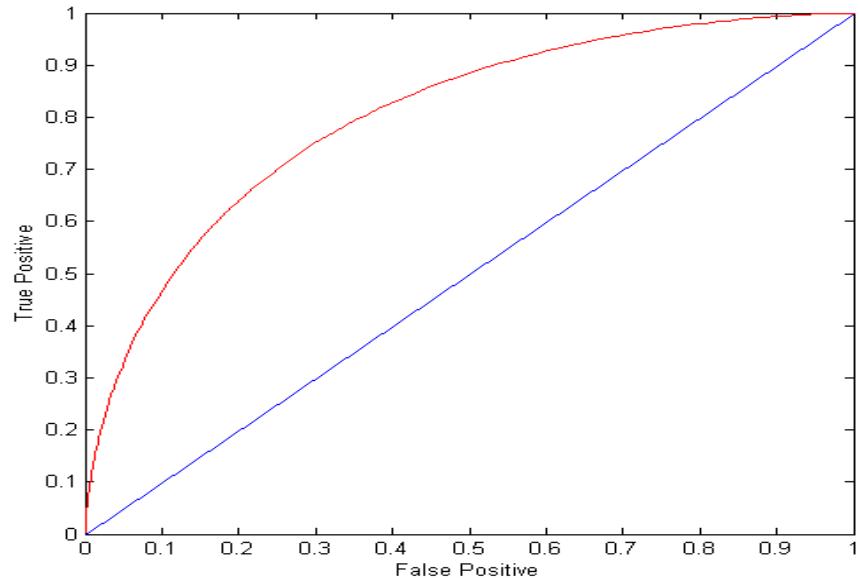
TP=0.5, FN=0.5, FP=0.12, FN=0.88



ROC Curve

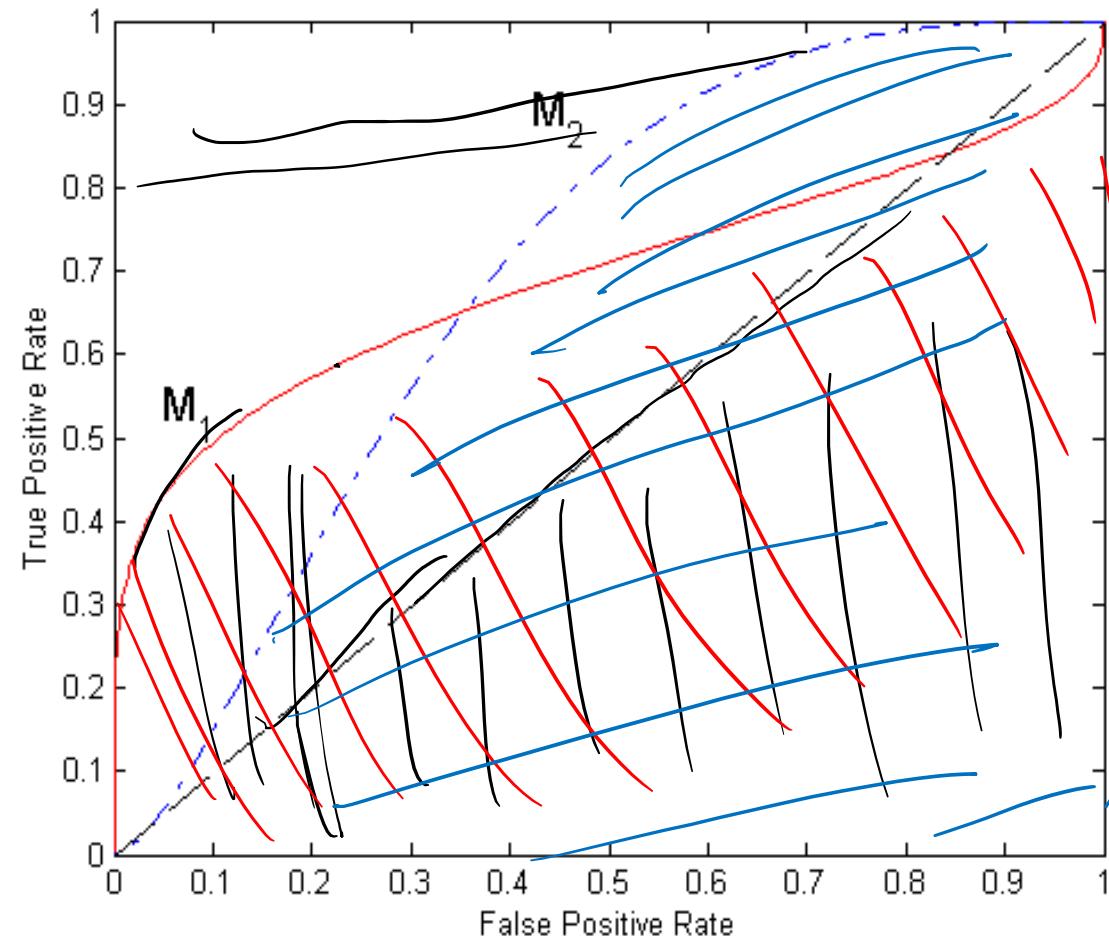
(TP,FP):

- (0,0): declare everything to be negative class
 - (1,1): declare everything to be positive class
 - (1,0): ideal
-
- Diagonal line:
 - Random guessing
 - Below diagonal line:
 - prediction is opposite of the true class



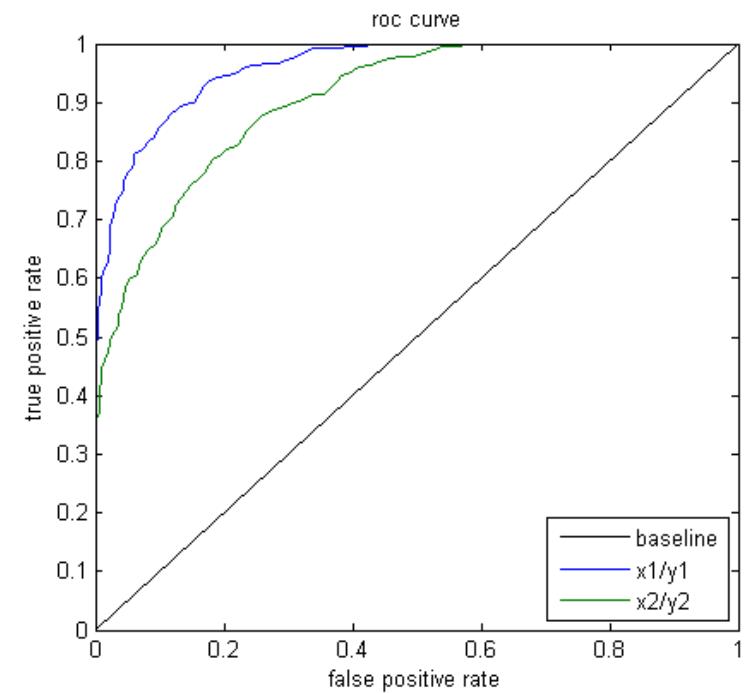
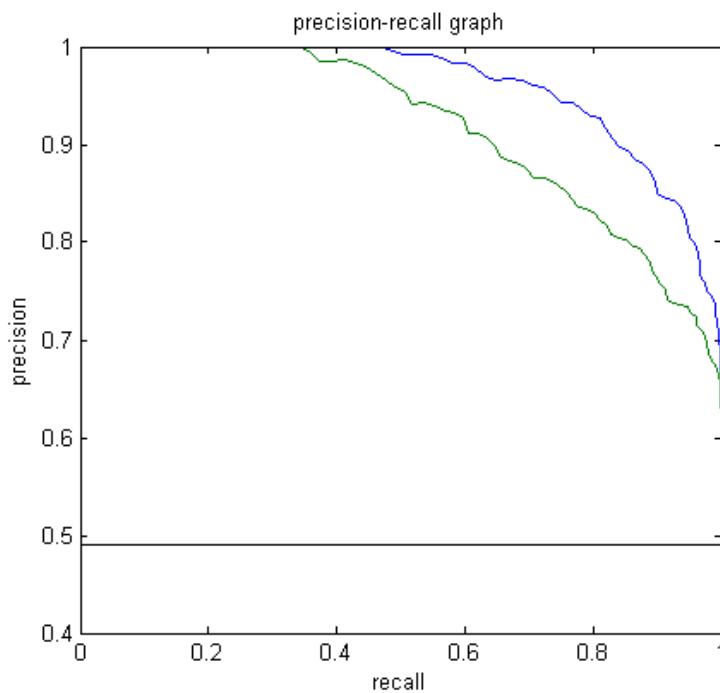
		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Using ROC for Model Comparison



- No model consistently outperform the other
 - M_1 is better for small FPR
 - M_2 is better for large FPR
- Area Under the ROC curve (**AUC**)
 - Ideal: Area = 1
 - Random guess:
 - Area = 0.5

ROC curve vs Precision-Recall curve



Area Under the Curve (AUC) as a single number for evaluation

Rule Based Classification

Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule: $(Condition) \rightarrow y$
 - where
 - *Condition* is a conjunctions of attributes
 - y is the class label
 - LHS: rule antecedent or pre condition
 - RHS: rule consequent
- Rules : $R = (R_1 \vee R_2 \vee \dots \vee R_k)$
 - Examples of classification rules:
 - (Blood Type=Warm) \wedge (Lay Eggs=Yes) \rightarrow Birds
 - (Taxable Income < 50K) \wedge (Refund=Yes) \rightarrow Evade=No

$$\begin{array}{lcl} \vee = \cup = OR \\ \wedge = \cap = AND \end{array}$$

Rule-based Classifier (Example) for Vertebrate Classification Problem

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Application of Rule-Based Classifier

- A rule r **covers** a record x if the attributes of the record x satisfy the condition of the rule. Rule r is also said to be **triggered or fired** whenever it covers a given record.

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- Coverage of a rule:
 - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
 - Fraction of records that satisfy both the antecedent and consequent of a rule

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	<u>Single</u>	125K	No
2	No	Married	100K	No
3	No	<u>Single</u>	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	<u>Single</u>	85K	Yes X
9	No	Married	75K	No
10	No	<u>Single</u>	90K	Yes X

$R \rightarrow (\text{Status} = \underline{\text{Single}}) \rightarrow \underline{\text{No}}$

Coverage = 40%, Accuracy = 50%

How does Rule-based Classifier Work?

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm ✓	yes ✓	no	no	?
turtle	cold	no ✓	no ✓	sometimes	? <i>Reptile/Amp</i>
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

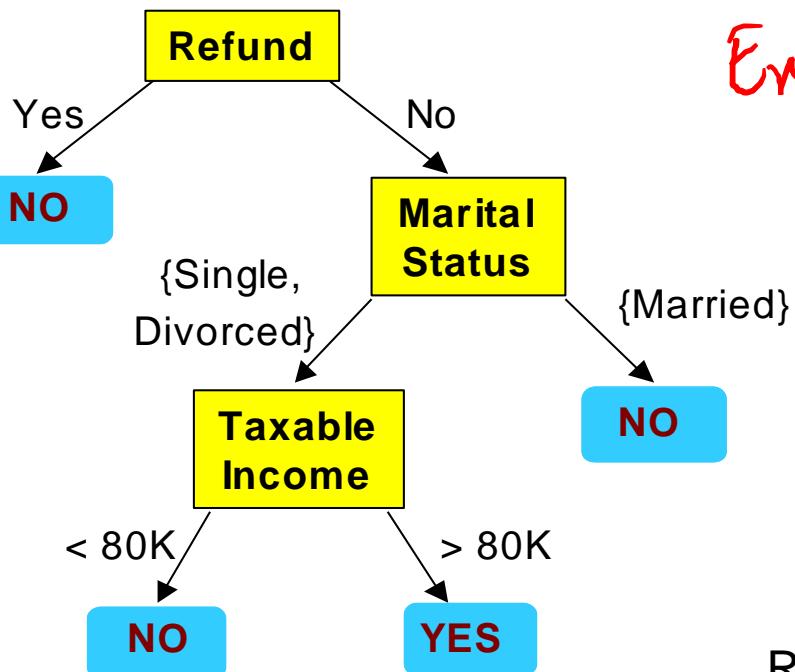
A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Characteristics of Rule-Based Classifier

- Mutually exclusive rules
 - Classifier contains mutually exclusive rules if no two rules are triggered by the same record.
 - Every record is covered by at most one rule
- Exhaustive rules
 - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
 - Each record is covered by at least one rule

From Decision Trees To Rules



Exhaustive

Exclusive

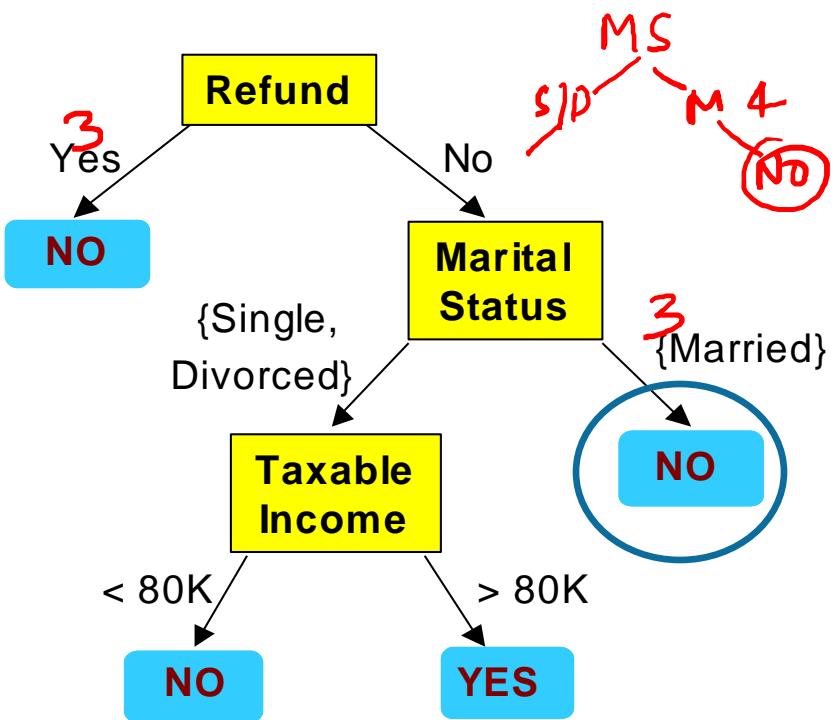
Classification Rules	
(Refund=Yes) ==> No	<i>3 (1,4,7)</i>
R ₂ (Refund>No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No	<i>-1 (3)</i>
R ₃ (Refund>No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes	<i>-3 (5,8)</i>
R₄ (Refund-No, Marital Status={Married}) ==> No	<i>3 (2,6,9)</i>

*R₅ {MS = married} => NO
4 (12,19,6,9)*

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Initial Rule: $\neg(\text{Refund}=\text{No}) \wedge (\text{Status}=\text{Married}) \rightarrow \text{No}$

Simplified Rule: $(\text{Status}=\text{Married}) \rightarrow \text{No}$

Annotations include a red circle around 'Married' in the 'Status' column, a bracket under 'Married' with '2 4 nos', and a bracket under 'Single' with '2 4 ok'.

Effect of Rule Simplification

- Rules are no longer mutually exclusive
 - A record may trigger more than one rule
 - Solution?
 - Ordered rule set
 - Unordered rule set – use voting schemes
- Rules are no longer exhaustive
 - A record may not trigger any rules
 - Solution?
 - Use a default class

Ordered Rule Set

- Rules are rank ordered according to their priority
 - An ordered rule set is known as a decision list
- When a test record is presented to the classifier
 - It is assigned to the class label of the highest ranked rule it has triggered
 - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

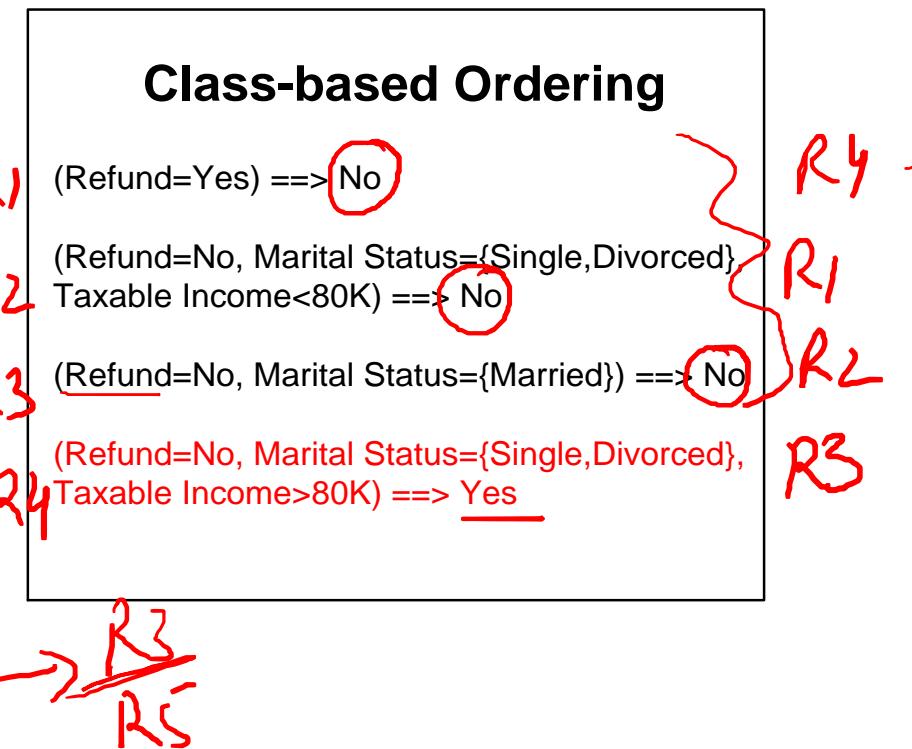
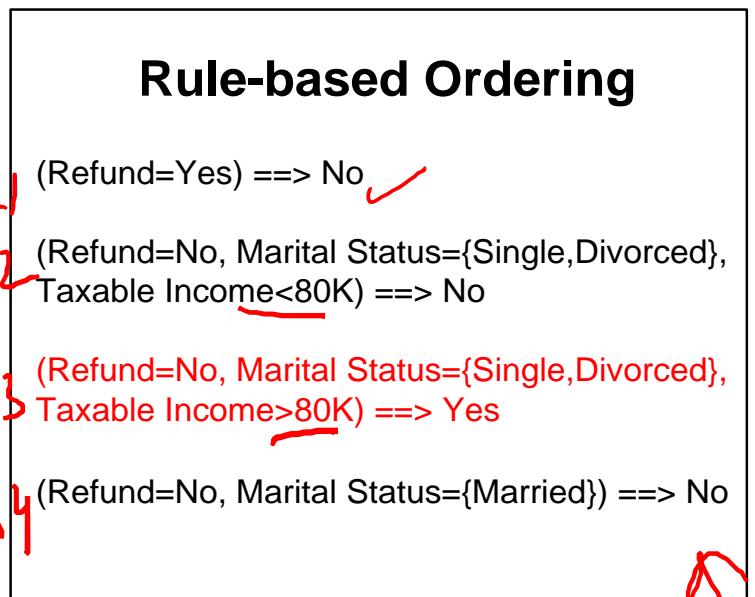
R5: (Live in Water = sometimes) \rightarrow Amphibians



Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Rule Ordering Schemes

- Rule-based ordering
 - Individual rules are ranked based on their quality
- Class-based ordering
 - Rules that belong to the same class appear together



Building Classification Rules

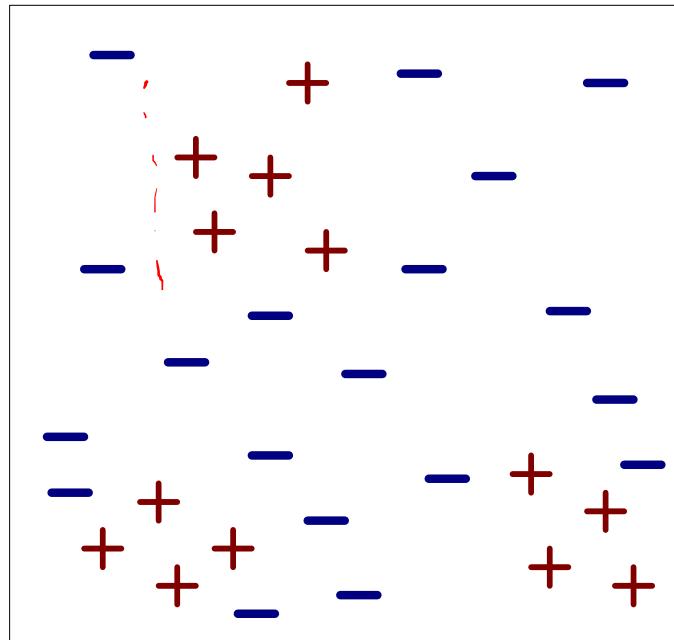
- Direct Method:
 - Extract rules directly from data
 - e.g.: RIPPER, CN2, Holte's 1R
- Indirect Method:
 - Extract rules from other classification models (e.g. decision trees, neural networks, etc.).
 - e.g: C4.5rules

Direct Method: Sequential Covering

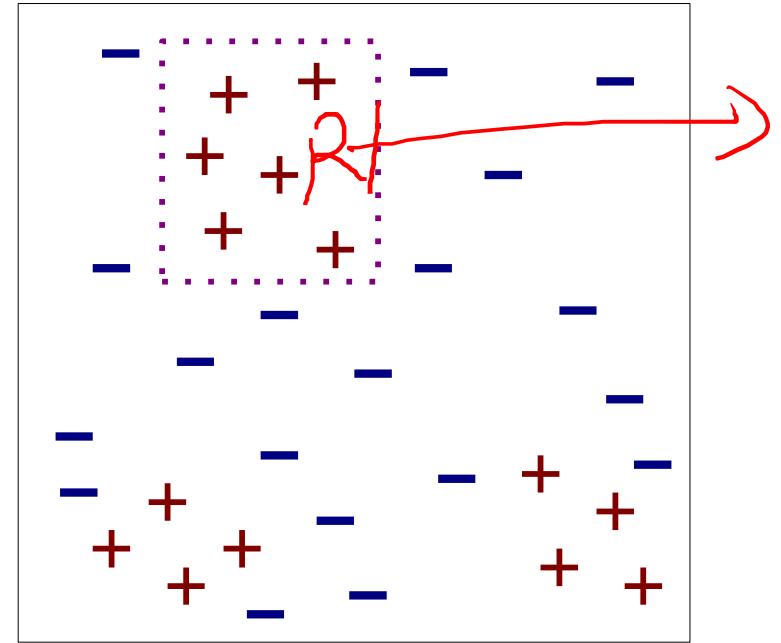
1. Start from an empty rule $\{ \rightarrow y_d$,
default class
2. Grow a rule using the Learn-One-Rule function
3. Remove training records covered by the rule
4. Repeat Step (2) and (3) until stopping criterion is met

① Class prevalence
② Cost of misclassifying records from a given class.

Example of Sequential Covering



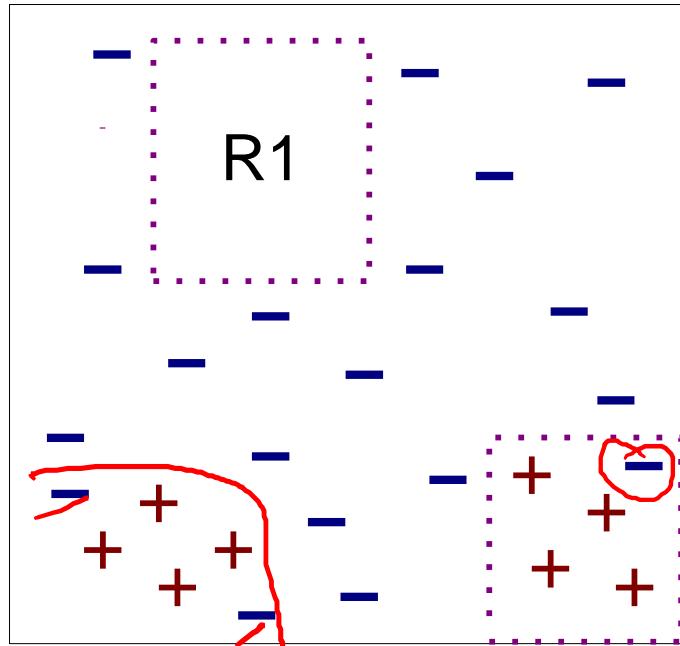
(i) Original Data



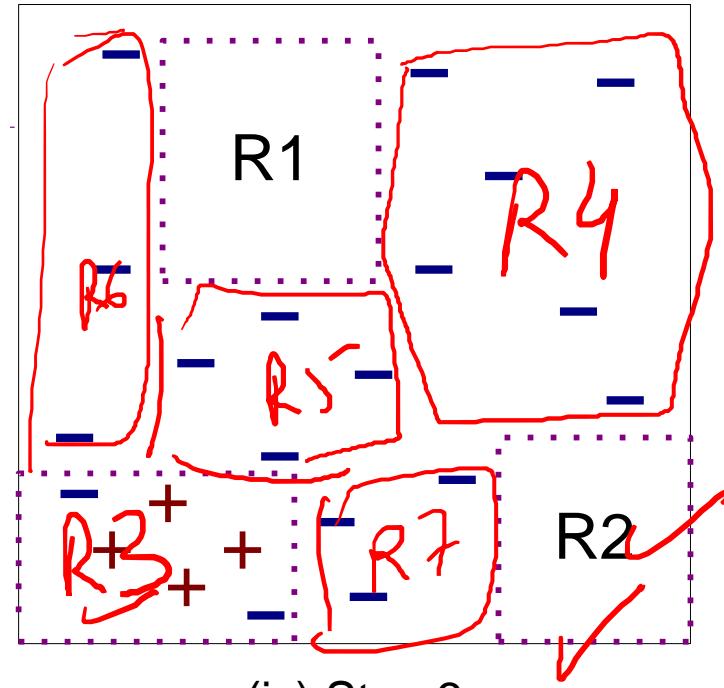
(ii) Step 1

class i + detection extraction

Example of Sequential Covering...



(iii) Step 2



(iv) Step 3

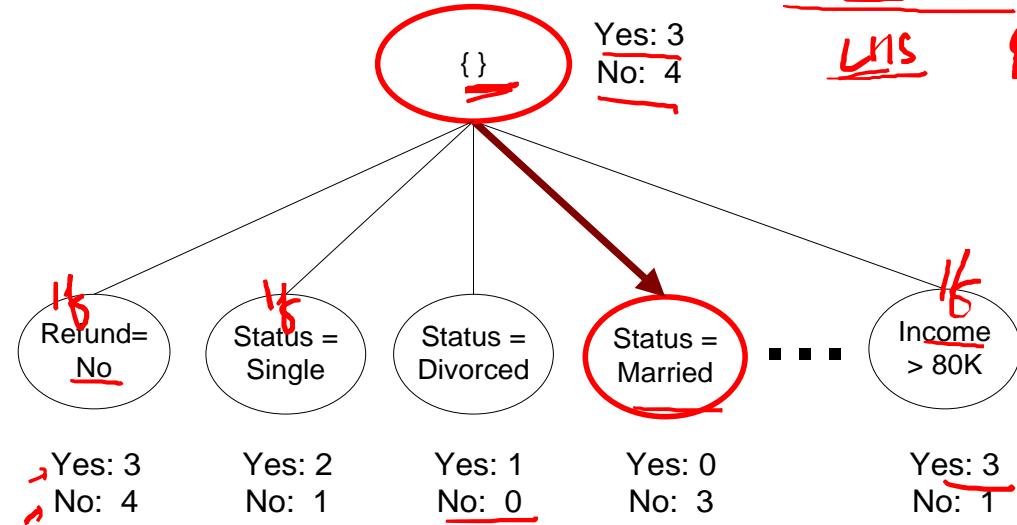
Learn one rule \rightarrow greedy search
 \rightarrow refining until stopping \rightarrow Rule pruning

Aspects of Sequential Covering

- Rule Growing ✓
training raw sets
- Instance Elimination ✓
- Rule Evaluation ✓
- Stopping Criterion
- Rule Pruning ✓

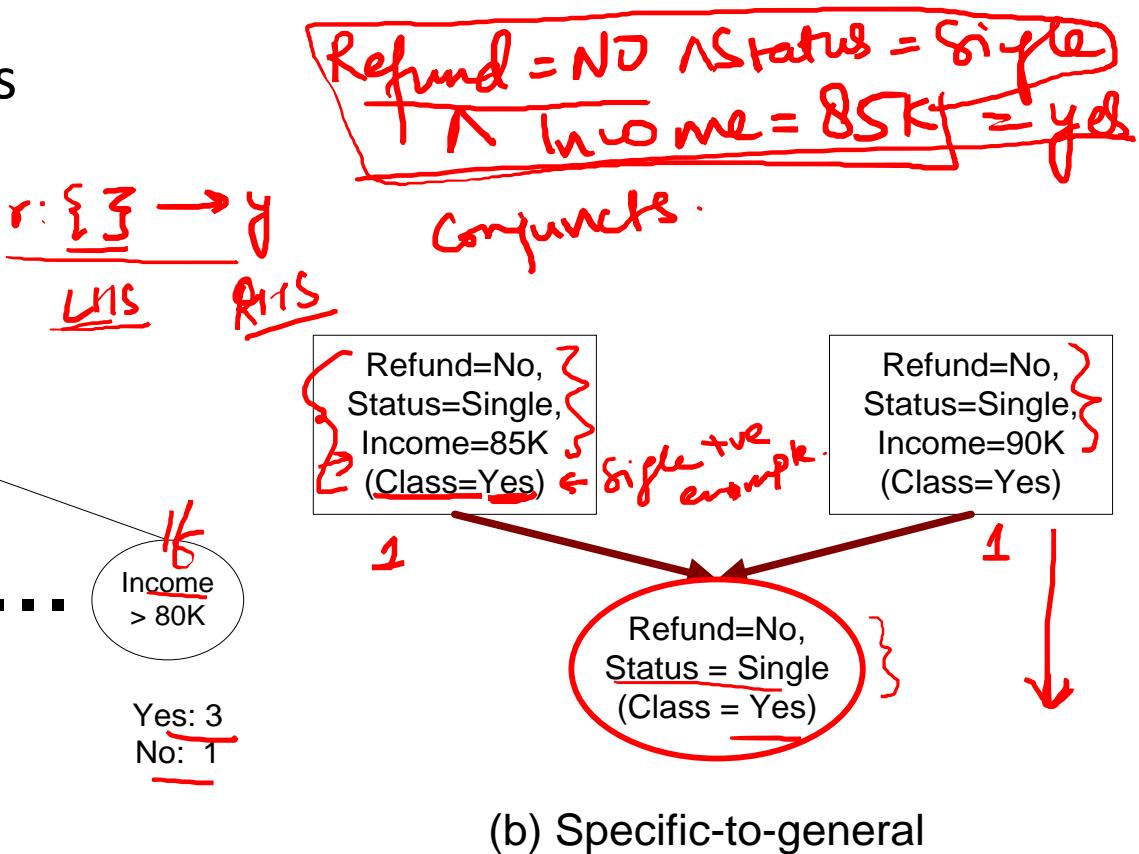
Rule Growing

- Two common strategies



(a) General-to-specific

$1 \rightarrow 1$ class
 $K \rightarrow K$ classes



(b) Specific-to-general

K rules
—
 3 attributes

Rule Growing (Examples)

- CN2 Algorithm:

- Start from an empty conjunct: {}
- Add conjuncts that minimizes the entropy measure: {A}, {A,B}, ...
- Determine the rule consequent by taking majority class of instances covered by the rule

- RIPPER Algorithm:

- Start from an empty rule: {} => class
- Add conjuncts that maximizes FOIL's information gain measure:

- R0: {} => class (initial rule)
- R1: {A} => class (rule after adding conjunct)
- $\text{Gain}(R0, R1) = t [\log(p1/(p1+n1)) - \log(p0/(p0 + n0))]$
- where t: number of positive instances covered by both R0 and R1

p0: number of positive instances covered by R0

n0: number of negative instances covered by R0

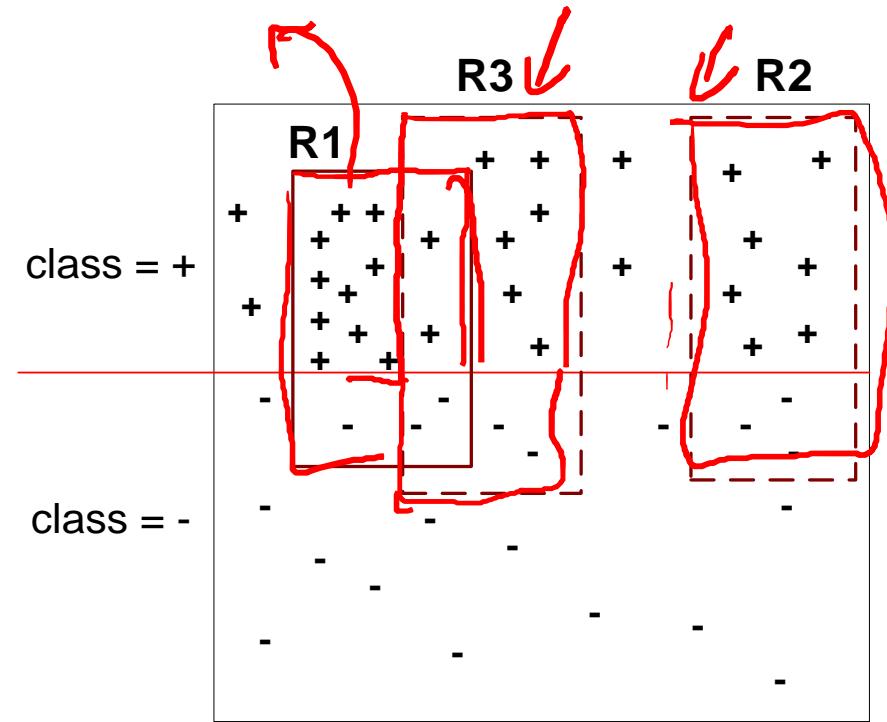
p1: number of positive instances covered by R1

n1: number of negative instances covered by R1

Conjunction examples
evaluator metric

Instance Elimination

- Why do we need to eliminate instances?
 - Otherwise, the next rule is identical to previous rule
- Why do we remove positive instances?
 - Ensure that the next rule is different
- Why do we remove negative instances?
 - Prevent underestimating accuracy of rule
 - Compare rules R2 and R3 in the diagram



$$\frac{50}{55} = 90.91\% \quad \text{Rule } r_1: 2 \text{ - (50\%)}.$$

- Metrics: $\frac{2}{n} = \frac{n_c}{n}$
- Accuracy

$$= \frac{n_c + 1}{n + k}$$

- Laplace

$$r_1 = \frac{50+1}{55+2}$$

- = 89.47%
- M-estimate

$$r_2 = \frac{2+1}{2+2} = \frac{n_c + kp}{n + k} = 1$$

60 true examples { training set
100 ave examples { test set

Rule r_1 : 50 true, 5 ave ex.

Rule r_2 : 2 true, no ave ex.

$$r_1 = 90.91\%$$

$$r_2 = 100\%$$

n : Number of instances covered by rule

n_c : Number of instances covered by rule

k : Number of classes

p : Prior probability

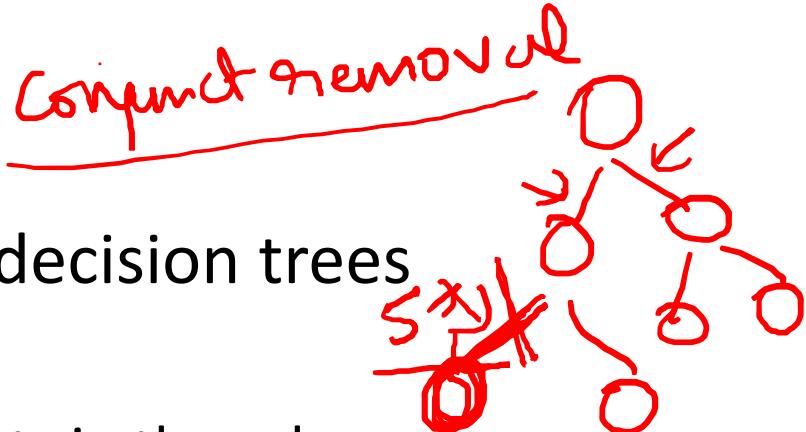
$$p = 1/k \quad \frac{2}{1/2} = 2^{0.5} \quad P = 1/k$$

Stopping Criterion and Rule Pruning

- Stopping criterion
 - Compute the gain
 - If gain is not significant, discard the new rule

- Rule Pruning

- Similar to post-pruning of decision trees
- Reduced Error Pruning:
 - Remove one of the conjuncts in the rule
 - Compare error rate on validation set before and after pruning
 - If error improves, prune the conjunct



Summary of Direct Method

- Grow a single rule
- Remove Instances from rule
- Prune the rule (if necessary)
- Add rule to Current Rule Set
- Repeat

Repeated Incremental Pruning to Produce Better Direct Method: RIPPER Reduction.

- ① to for datasets with high class imbalance
- ② it works well for noisy datasets. → vs to reduce overfitting
- For 2-class problem, choose one of the classes as positive class, and the other as negative class
 - Learn rules for positive class
 - Negative class will be default class
- For multi-class problem
 - Order the classes according to increasing class prevalence
(fraction of instances that belong to a particular class)
 - Learn the rule set for smallest class first, treat the rest as negative class
 - Repeat with next smallest class as positive class

$C_1 \rightarrow$ least frequent class

$C_n \rightarrow$ most frequent class

$\rightarrow C_1 \rightarrow \dots \rightarrow C_n$

$\left\{ \begin{array}{l} \text{! -ve} \\ \text{con-} \end{array} \right.$

Direct Method: RIPPER

general to specific
Strategy

- Growing a rule:
 - Start from empty rule
 - Add conjuncts as long as they improve FOIL's information gain
 - Stop when rule no longer covers negative examples
 - Prune the rule immediately using incremental reduced error pruning
 - Measure for pruning: $v = (p-n)/(p+n)$
 - p: number of positive examples covered by the rule in the validation set
 - n: number of negative examples covered by the rule in the validation set
 - Pruning method: delete any final sequence of conditions that maximizes v

$$\begin{array}{l} \xrightarrow{\text{A B C D}} \gamma \\ \xrightarrow{\text{A B C}} \gamma \\ \xrightarrow{\text{A C D}, \text{B C D}} \gamma \end{array}$$

Direct Method: RIPPER

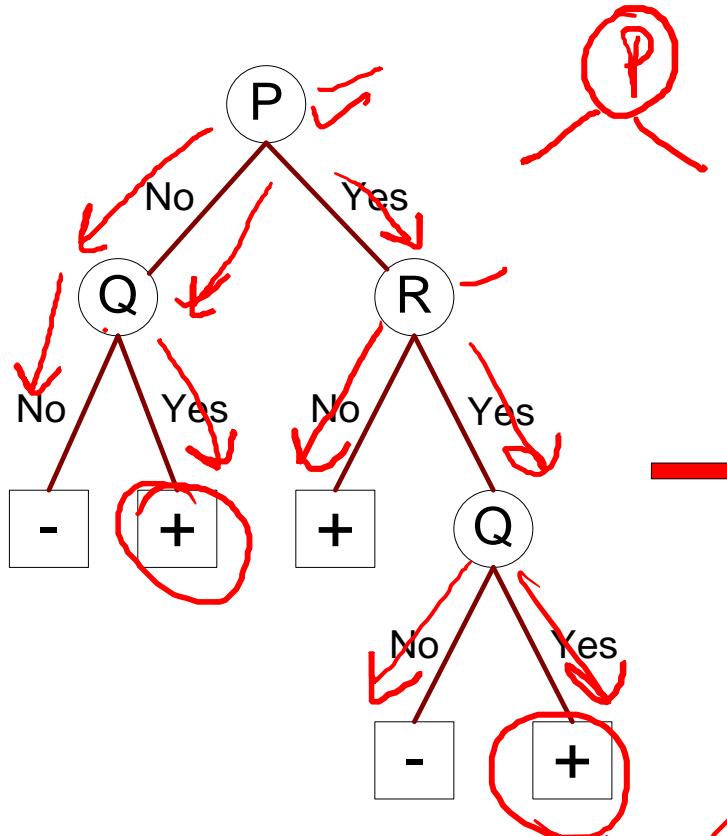
- Building a Rule Set:
 - Use sequential covering algorithm
 - Finds the best rule that covers the current set of positive examples
 - Eliminate both positive and negative examples covered by the rule
 - Each time a rule is added to the rule set, compute the new description length
 - stop adding new rules when the new description length is d bits longer than the smallest description length obtained so far

Direct Method: RIPPER

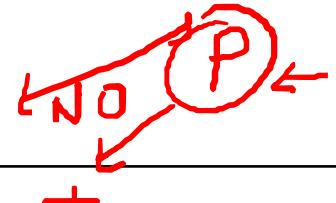
- Optimize the rule set:
 - For each rule r in the rule set R
 - Consider 2 alternative rules:
 - Replacement rule (r^*): grow new rule from scratch
 - Revised rule(r'): add conjuncts to extend the rule r
 - Compare the rule set for r against the rule set for r^* and r'
 - Choose rule set that minimizes MDL principle
 - Repeat rule generation and rule optimization for the remaining positive examples

$$R = \{ r_1, r_2, r_3, \dots \}$$
$$r_1 = \overbrace{ABCD}^{\text{r}} \rightarrow y$$
$$r^* = \overbrace{EF}^{\text{r}^*} \rightarrow y$$
$$r' = \overbrace{ABCDE}^{\text{r}'} \rightarrow y$$

Indirect Methods



$$\begin{aligned} P = \text{No} \wedge Q = \text{No} &\rightarrow - \\ P = \underline{S=R} \end{aligned}$$



Rule Set

- r1: $(P=\text{No}, Q=\text{No}) \implies -$
- r2: $(P=\text{No}, Q=\text{Yes}) \implies +$
- r3: $(P=\text{Yes}, R=\text{No}) \implies +$
- r4: $(P=\text{Yes}, R=\text{Yes}, Q=\text{No}) \implies -$
- r5: $(P=\text{Yes}, R=\text{Yes}, Q=\text{Yes}) \implies +$

$\theta \rightarrow \text{Yes} \rightarrow +$
 $Q': Q = \text{Yes} \rightarrow +$

$\tau_2: (P = \text{No}) \wedge (\theta = \text{Yes}) \rightarrow +$
 $\tau_3: (P = \text{Yes}) \wedge (R = \text{No}) \rightarrow +$
 $\tau_5: (P = \text{Yes}) \wedge (R = \text{Yes}) \wedge (\theta = \text{Y}) \rightarrow +$

Q3: same

Indirect Method: C4.5rules

Rule Generation

- Extract rules from an unpruned decision tree
- For each rule, $r: A \rightarrow y$,
 - consider an alternative rule $r': A' \rightarrow y$ where A' is obtained by removing one of the conjuncts in A
 - Compare the pessimistic error rate for r against all r' s $e(A') \geq e(A)$
 - Prune if one of the r' s has lower pessimistic error rate
 - Repeat until we can no longer improve generalization error

$$e(r_4) > e(r_4') > e(r_4'')$$

$\overbrace{r_4: P=4 \wedge R=4 \wedge Q=N}^{\text{generalization error}} \rightarrow -$

$\overbrace{r_4': P=4 \wedge R=4}^{\text{generalization error}} \rightarrow -$

$\overbrace{r_4'': P=4}^{\text{generalization error}} \rightarrow -$

$A' : A \cap B$

$A : A \cap B \cap C \rightarrow y$

Indirect Method: C4.5rules

Rule Ordering:

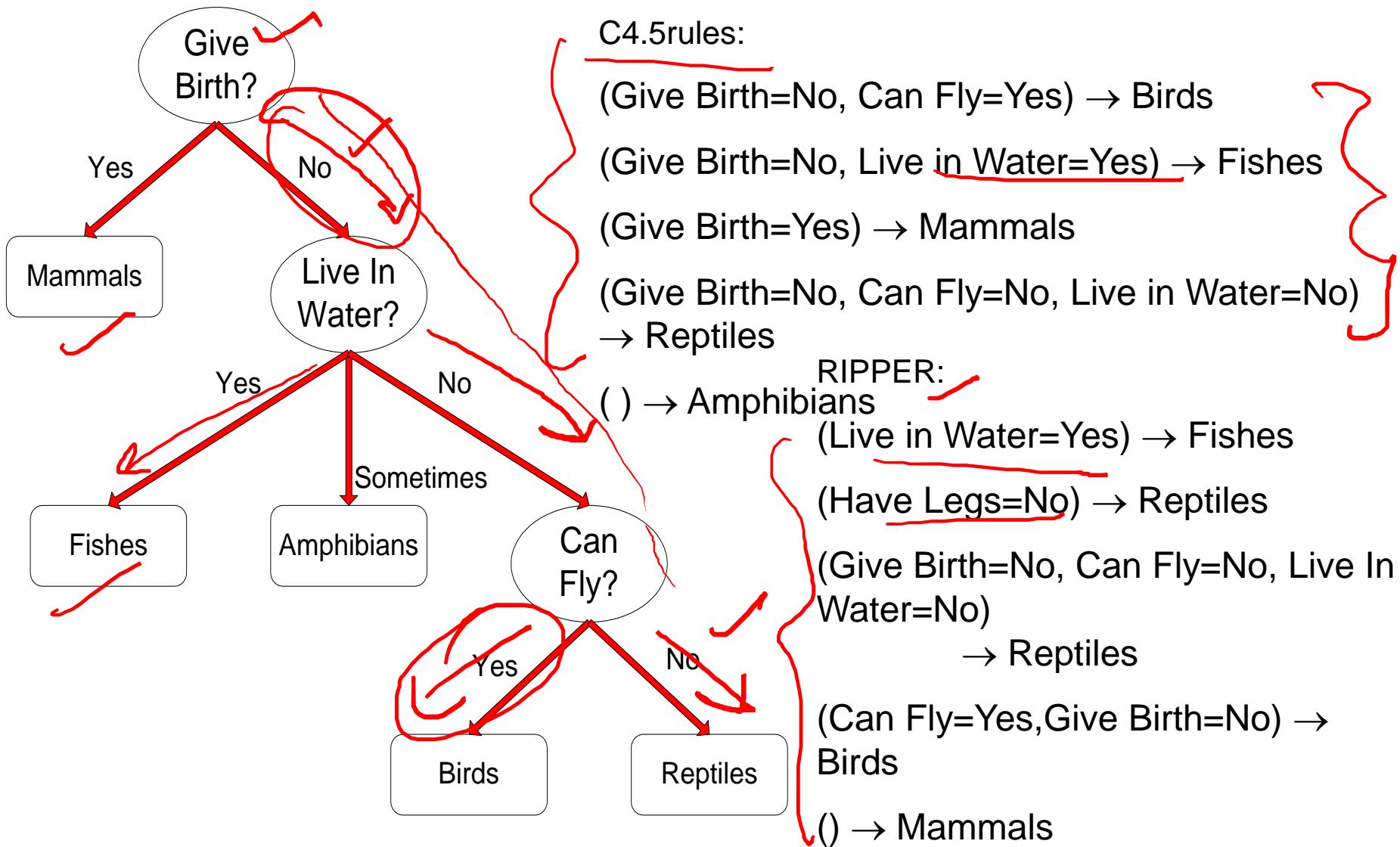
- Instead of ordering the rules, order subsets of rules (class ordering)
 - Each subset is a collection of rules with the same rule consequent (class)
 - Compute description length of each subset
 - Description length = $L(\text{error}) + g L(\text{model})$
 - g is a parameter that takes into account the presence of redundant attributes in a rule set
(default value = 0.5)

+ve $DL <$ +ve
-ve $DL - ve$

Example

Name	Give Birth	Lay Eggs	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	no	yes	mammals
python	no	yes	no	no	no	reptiles
salmon	no	yes	no	yes	no	fishes
whale	yes	no	no	yes	no	mammals
frog	no	yes	no	sometimes	yes	amphibians
komodo	no	yes	no	no	yes	reptiles
bat	yes	no	yes	no	yes	mammals
pigeon	no	yes	yes	no	yes	birds
cat	yes	no	no	no	yes	mammals
leopard shark	yes	no	no	yes	no	fishes
turtle	no	yes	no	sometimes	yes	reptiles
penguin	no	yes	no	sometimes	yes	birds
porcupine	yes	no	no	no	yes	mammals
eel	no	yes	no	yes	no	fishes
salamander	no	yes	no	sometimes	yes	amphibians
gila monster	no	yes	no	no	yes	reptiles
platypus	no	yes	no	no	yes	mammals
owl	no	yes	yes	no	yes	birds
dolphin	yes	no	no	yes	no	mammals
eagle	no	yes	yes	no	yes	birds

C4.5rules versus RIPPER



C4.5rules versus RIPPER

C4.5 and C4.5rules:

TP

P, R, f

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

RIPPER:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

Advantages of Rule-Based Classifiers

- As highly expressive as decision trees
 - Easy to interpret
 - Easy to generate
 - Can classify new instances rapidly
 - Performance comparable to decision trees
- B** The class-based additive approach adopted by many rule-based classifiers is well suited for handling data sets with imbalanced class distribution

Data Mining

Unit 3

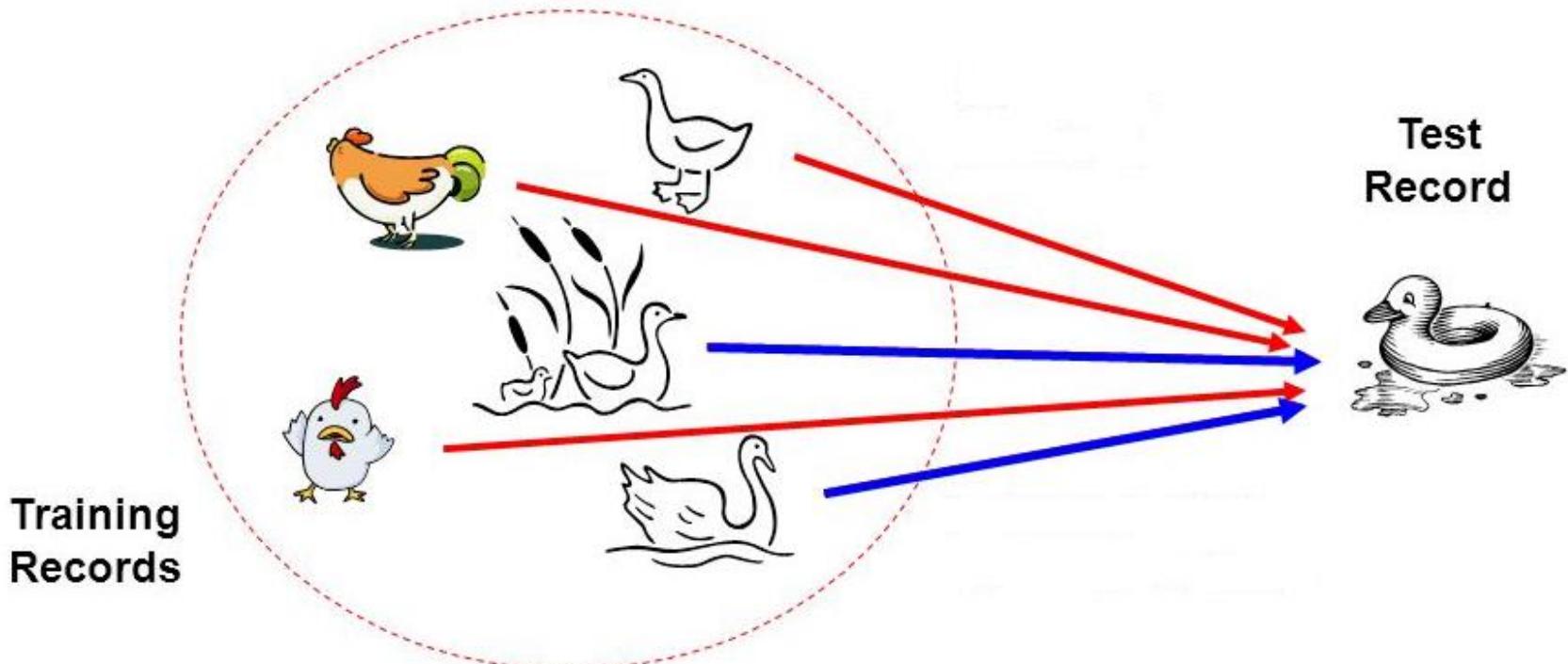
Statistical Classifier: Naïve Bayes' Classifier

Slides credit: Dr. Debasis Samanta (IITkgp)

Bayesian Classifier

Bayesian Classifier

- Principle
 - If it walks like a duck, quacks like a duck, then it is **probably** a duck

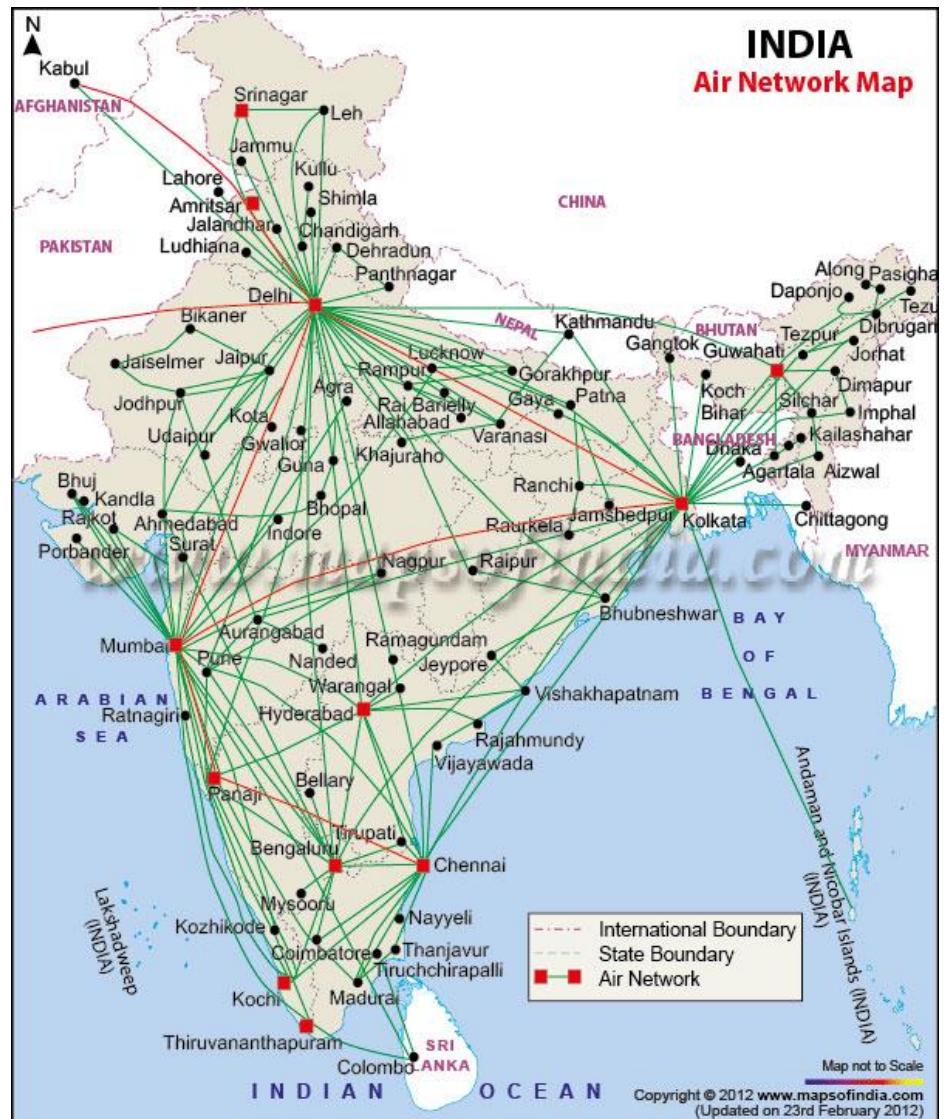


Bayesian Classifier

- A statistical classifier
 - Performs *probabilistic prediction*, i.e., predicts class membership probabilities
- Foundation
 - Based on Bayes' Theorem.
- Assumptions
 1. The classes are mutually exclusive and exhaustive.
 2. The attributes are independent given the class.
- Called “Naïve” classifier because of these assumptions.
 - Empirically proven to be useful.
 - Scales very well.

Example: Bayesian Classification

- **Example 8.2:** Air Traffic Data
 - Let us consider a set observation recorded in a database
 - Regarding the arrival of airplanes in the routes from any airport to New Delhi under certain conditions.



Air-Traffic Data

Days	Season	Fog	Rain	Class
Weekday	Spring	None	None	On Time
Weekday	Winter	None	Slight	On Time
Weekday	Winter	None	None	On Time
Holiday	Winter	High	Slight	Late
Saturday	Summer	Normal	None	On Time
Weekday	Autumn	Normal	None	Very Late
Holiday	Summer	High	Slight	On Time
Sunday	Summer	Normal	None	On Time
Weekday	Winter	High	Heavy	Very Late
Weekday	Summer	None	Slight	On Time

Cond. to next slide...

Air-Traffic Data

Cond. from previous slide...

Days	Season	Fog	Rain	Class
Saturday	Spring	High	Heavy	Cancelled
Weekday	Summer	High	Slight	On Time
Weekday	Winter	Normal	None	Late
Weekday	Summer	High	None	On Time
Weekday	Winter	Normal	Heavy	Very Late
Saturday	Autumn	High	Slight	On Time
Weekday	Autumn	None	Heavy	On Time
Holiday	Spring	Normal	Slight	On Time
Weekday	Spring	Normal	None	On Time
Weekday	Spring	Normal	Heavy	On Time

Air-Traffic Data

- In this database, there are four attributes

$$A = [\text{Day}, \text{Season}, \text{Fog}, \text{Rain}]$$

with 20 tuples.

- The categories of classes are:

$$C = [\text{On Time}, \text{Late}, \text{Very Late}, \text{Cancelled}]$$

- Given this is the knowledge of data and classes, we are to find most likely classification for any other **unseen instance**, for example:

Week Day	Winter	High	None	???
----------	--------	------	------	-----

- Classification technique eventually to map this tuple into an accurate class.

Bayesian Classifier

- In many applications, the relationship between the attributes set and the class variable is **non-deterministic**.
 - In other words, a test cannot be classified to a class label with certainty.
 - In such a situation, the classification can be achieved **probabilistically**.
- The Bayesian classifier is an approach for **modelling probabilistic relationships** between the attribute set and the class variable.
- More precisely, Bayesian classifier use **Bayes' Theorem of Probability** for classification.
- Before going to discuss the Bayesian classifier, we should have a quick look at the **Theory of Probability** and then **Bayes' Theorem**.

Bayes' Theorem of Probability

Simple Probability

Definition 8.2: Simple Probability

If there are n elementary events associated with a random experiment and m of n of them are favorable to an event A , then the probability of happening or occurrence of A is

$$P(A) = \frac{m}{n}$$

Simple Probability

- Suppose, A and B are any two events and $P(A)$, $P(B)$ denote the probabilities that the events A and B will occur, respectively.
- **Mutually Exclusive Events:**
 - Two events are mutually exclusive, if the occurrence of one precludes the occurrence of the other.

Example: Tossing a coin (two events)

Tossing a ludo cube (Six events)

💡 Can you give an example, so that two events are not mutually exclusive?

Hint: Tossing two identical coins, Weather (sunny, foggy, warm)

Simple Probability

- **Independent events:** Two events are independent if occurrences of one does not alter the occurrence of other.

Example: Tossing both coin and ludo cube together.
(How many events are here?)

💡 Can you give an example, where an event is dependent on one or more other events(s)?

Hint: Receiving a message (A) through a communication channel (B) over a computer (C), rain and dating.

Joint Probability

Definition 8.3: Joint Probability

If $P(A)$ and $P(B)$ are the probability of two events, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

If A and B are mutually exclusive, then $P(A \cap B) = 0$

If A and B are independent events, then $P(A \cap B) = P(A).P(B)$

Thus, for mutually exclusive events

$$P(A \cup B) = P(A) + P(B)$$

Conditional Probability

Definition 8.2: Conditional Probability

If events are dependent, then their probability is expressed by conditional probability. The probability that A occurs given that B is denoted by $P(A|B)$.

Suppose, A and B are two events associated with a random experiment. The probability of A under the condition that B has already occurred and $P(B) \neq 0$ is given by

$$\begin{aligned} P(A|B) &= \frac{\text{Number of events in } B \text{ which are favourable to } A}{\text{Number of events in } B} \\ &= \frac{\text{Number of events favourable to } A \cap B}{\text{Number of events favourable to } B} \\ &= \frac{P(A \cap B)}{P(B)} \end{aligned}$$

Conditional Probability

Corollary 8.1: Conditional Probability

$$P(A \cap B) = P(A) \cdot P(B|A), \quad \text{if } P(A) \neq 0$$

or $P(A \cap B) = P(B) \cdot P(A|B), \quad \text{if } P(B) \neq 0$

For three events A , B and C

$$P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C|A \cap B)$$

For n events A_1, A_2, \dots, A_n and if all events are mutually independent to each other

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1) \cdot P(A_2) \cdots \cdots \cdots P(A_n)$$

Note:

$$P(A|B) = 0 \quad \text{if events are mutually exclusive}$$

$$P(A|B) = P(A) \quad \text{if } A \text{ and } B \text{ are independent}$$

$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A) \text{ otherwise,}$$

$$P(A \cap B) = P(B \cap A)$$

Conditional Probability

- Generalization of Conditional Probability:

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)} \\ &= \frac{P(B|A) \cdot P(A)}{P(B)} \quad \because P(A \cap B) = P(B|A) \cdot P(A) = P(A|B) \cdot P(B) \end{aligned}$$

By the law of total probability : $P(B) = P[(B \cap A) \cup (B \cap \bar{A})]$, where \bar{A} denotes the compliment of event A. Thus,

$$\begin{aligned} P(A|B) &= \frac{P(B|A) \cdot P(A)}{P[(B \cap A) \cup (B \cap \bar{A})]} \\ &= \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(B|\bar{A}) \cdot P(\bar{A})} \end{aligned}$$

Conditional Probability

In general,

$$P(A|D) = \frac{P(A) \cdot P(D|A)}{P(A) \cdot P(D|A) + P(B) \cdot P(D|B) + P(C) \cdot P(D | C)}$$

Total Probability

Definition 8.3: Total Probability

Let E_1, E_2, \dots, E_n be n mutually exclusive and exhaustive events associated with a random experiment. If A is any event which occurs with E_1 or E_2 or \dots, E_n , then

$$P(A) = P(E_1).P(A|E_1) + P(E_2).P(A|E_2) + \dots + P(E_n).P(A|E_n)$$

Total Probability: An Example

Example 8.3

A bag contains 4 red and 3 black balls. A second bag contains 2 red and 4 black balls. One bag is selected at random. From the selected bag, one ball is drawn. What is the probability that the ball drawn is red?

This problem can be answered using the concept of Total Probability

E_1 =Selecting bag I

E_2 =Selecting bag II

A = Drawing the red ball

Thus, $P(A) = P(E_1).P(A|E_1) + P(E_2).P(A|E_2)$

where, $P(A|E_1)$ = Probability of drawing red ball when first bag has been chosen

and $P(A|E_2)$ = Probability of drawing red ball when second bag has been chosen

Reverse Probability

Example 8.3:

A bag (Bag I) contains 4 red and 3 black balls. A second bag (Bag II) contains 2 red and 4 black balls. You have chosen one ball at random. It is found as red ball. What is the probability that the ball is chosen from Bag I?

Here,

E_1 = Selecting bag I

E_2 = Selecting bag II

A = Drawing the red ball

We are to determine $P(E_1|A)$. Such a problem can be solved using Bayes' theorem of probability.

Bayes' Theorem

Theorem 8.4: Bayes' Theorem

Let E_1, E_2, \dots, E_n be n mutually exclusive and exhaustive events associated with a random experiment. If A is any event which occurs with E_1 or E_2 or ... or E_n , then

$$P(E_i|A) = \frac{P(E_i) \cdot P(A|E_i)}{\sum_{i=1}^n P(E_i) \cdot P(A|E_i)}$$

Prior and Posterior Probabilities

- $P(A)$ and $P(B)$ are called prior probabilities
- $P(A|B)$, $P(B|A)$ are called posterior probabilities

Example 8.6: Prior versus Posterior Probabilities

- This table shows that the event Y has two outcomes namely A and B , which is dependent on another event X with various outcomes like x_1, x_2 and x_3 .
- **Case1:** Suppose, we don't have any information of the event A . Then, from the given sample space, we can calculate $P(Y = A) = \frac{5}{10} = 0.5$.
- **Case2:** Now, suppose, we want to calculate $P(X = x_2 | Y = A) = \frac{2}{5} = 0.4$.

The later is the conditional or posterior probability, whereas the former is the prior probability.

X	Y
x_1	A
x_2	A
x_3	B
x_3	A
x_2	B
x_1	A
x_1	B
x_3	B
x_2	B
x_2	A

Naïve Bayesian Classifier

- Suppose, Y is a class variable and $X = \{X_1, X_2, \dots, X_n\}$ is a set of attributes, with instance of Y .

INPUT (X)	CLASS(Y)
...	...
...	...
x_1, x_2, \dots, x_n	y_i
...	...

- The classification problem, then can be expressed as the class-conditional probability

$$P(Y = y_i | (X_1 = x_1) \text{ AND } (X_2 = x_2) \text{ AND } \dots \text{ AND } (X_n = x_n))$$

Naïve Bayesian Classifier

- Naïve Bayesian classifier calculate this posterior probability using Bayes' theorem, which is as follows.
- From Bayes' theorem on conditional probability, we have

$$\begin{aligned} P(Y|X) &= \frac{P(X|Y) \cdot P(Y)}{P(X)} \\ &= \frac{P(X|Y) \cdot P(Y)}{P(X|Y = y_1) \cdot P(Y = y_1) + \cdots + P(X|Y = y_k) \cdot P(Y = y_k)} \end{aligned}$$

where,

$$P(X) = \sum_{i=1}^k P(X|Y = y_i) \cdot P(Y = y_i)$$

Note:

- $P(X)$ is called the evidence (also the total probability) and it is a constant.
- The probability $P(Y|X)$ (also called class conditional probability) is therefore proportional to $P(X|Y) \cdot P(Y)$.
- Thus, $P(Y|X)$ can be taken as a measure of Y given that X .

$$P(Y|X) \approx P(X|Y) \cdot P(Y)$$

Naïve Bayesian Classifier

- Suppose, for a given instance of X (say $x = (X_1 = x_1)$ and $(X_n = x_n)$).
- There are any two class conditional probabilities namely $P(Y = y_i | X=x)$ and $P(Y = y_j | X=x)$.
- If $P(Y = y_i | X=x) > P(Y = y_j | X=x)$, then we say that y_i is more stronger than y_j for the instance $X = x$.
- The strongest y_i is the classification for the instance $X = x$.

Naïve Bayesian Classifier

- **Example:** With reference to the Air Traffic Dataset mentioned earlier, let us tabulate all the posterior and prior probabilities as shown below.

		Class			
Attribute		On Time	Late	Very Late	Cancelled
Day	Weekday	$9/14 = 0.64$	$\frac{1}{2} = 0.5$	$3/3 = 1$	$0/1 = 0$
	Saturday	$2/14 = 0.14$	$\frac{1}{2} = 0.5$	$0/3 = 0$	$1/1 = 1$
	Sunday	$1/14 = 0.07$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Holiday	$2/14 = 0.14$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
Season	Spring	$4/14 = 0.29$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Summer	$6/14 = 0.43$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Autumn	$2/14 = 0.14$	$0/2 = 0$	$1/3 = 0.33$	$0/1 = 0$
	Winter	$2/14 = 0.14$	$2/2 = 1$	$2/3 = 0.67$	$0/1 = 0$

Naïve Bayesian Classifier

		Class			
Attribute		On Time	Late	Very Late	Cancelled
Fog	None	$5/14 = 0.36$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	High	$4/14 = 0.29$	$1/2 = 0.5$	$1/3 = 0.33$	$1/1 = 1$
	Normal	$5/14 = 0.36$	$1/2 = 0.5$	$2/3 = 0.67$	$0/1 = 0$
Rain	None	$5/14 = 0.36$	$1/2 = 0.5$	$1/3 = 0.33$	$0/1 = 0$
	Slight	$8/14 = 0.57$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Heavy	$1/14 = 0.07$	$1/2 = 0.5$	$2/3 = 0.67$	$1/1 = 1$
Prior Probability		$14/20 = 0.70$	$2/20 = 0.10$	$3/20 = 0.15$	$1/20 = 0.05$

Naïve Bayesian Classifier

Instance:

Week Day	Winter	High	Heavy	???
----------	--------	------	-------	-----

Case1: Class = On Time : $0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$

Case2: Class = Late : $0.10 \times 0.50 \times 1.0 \times 0.50 \times 0.50 = 0.0125$

Case3: Class = Very Late : $0.15 \times 1.0 \times 0.67 \times 0.33 \times 0.67 = 0.0222$

Case4: Class = Cancelled : $0.05 \times 0.0 \times 0.0 \times 1.0 \times 1.0 = 0.0000$

Case3 is the strongest; Hence correct classification is **Very Late**

Naïve Bayesian Classifier

Algorithm: Naïve Bayesian Classification

Input: Given a set of k mutually exclusive and exhaustive classes $C = \{c_1, c_2, \dots, c_k\}$, which have prior probabilities $P(C_1), P(C_2), \dots, P(C_k)$.

There are n -attribute set $A = \{A_1, A_2, \dots, A_n\}$, which for a given instance have values $A_1 = a_1, A_2 = a_2, \dots, A_n = a_n$

Step: For each $c_i \in C$, calculate the class condition probabilities, $i = 1, 2, \dots, k$

$$p_i = P(C_i) \times \prod_{j=1}^n P(A_j = a_j | C_i)$$

$$p_x = \max\{p_1, p_2, \dots, p_k\}$$

Output: C_x is the classification

Note: $\sum p_i \neq 1$, because they are not probabilities rather proportion values (to posterior probabilities)

Naïve Bayesian Classifier

Pros and Cons

- The Naïve Bayes' approach is a very popular one, which often works well.
- However, it has a number of potential problems
 - It relies on all attributes being **categorical**.
 - If the data is **less**, then it **estimates poorly**.

Naïve Bayesian Classifier

Approach to overcome the limitations in Naïve Bayesian Classification

- Estimating the posterior probabilities for continuous attributes
 - In real life situation, all attributes are not necessarily be categorical, In fact, there is a mix of both categorical and continuous attributes.
 - In the following, we discuss the schemes to deal with continuous attributes in Bayesian classifier.
 1. We can discretize each continuous attributes and then replace the continuous values with its corresponding discrete intervals.
 2. We can assume a certain form of probability distribution for the continuous variable and estimate the parameters of the distribution using the training data. A Gaussian distribution is usually chosen to represent the posterior probabilities for continuous attributes. A general form of Gaussian distribution will look like

$$P(x: \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

where, μ and σ^2 denote mean and variance, respectively.

Naïve Bayesian Classifier

For each class C_i , the posterior probabilities for attribute A_j (it is the numeric attribute) can be calculated following Gaussian normal distribution as follows.

$$P(A_j = a_j | C_i) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(a_j - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

Here, the parameter μ_{ij} can be calculated based on the sample mean of attribute value of A_j for the training records that belong to the class C_i .

Similarly, σ_{ij}^2 can be estimated from the calculation of variance of such training records.

Naïve Bayesian Classifier

M-estimate of Conditional Probability

- The M-estimation is to deal with the potential problem of Naïve Bayesian Classifier when training data size is too poor.
 - If the posterior probability for one of the attribute is zero, then the overall class-conditional probability for the class vanishes.
 - In other words, if training data do not cover many of the attribute values, then we may not be able to classify some of the test records.
- This problem can be addressed by using the **M-estimate approach**.

M-estimate Approach

- M-estimate approach can be stated as follows

$$P(A_j = a_j | C_i) = \frac{n_{c_i} + mp}{n + m}$$

where, n = total number of instances from class C_i

n_{c_i} = number of training examples from class C_i that take the value $A_j = a_j$

m = it is a parameter known as the equivalent sample size, and

p = is a user specified parameter.

Note:

If $n = 0$, that is, if there is no training set available, then $P(a_j | C_i) = p$,
so, this is a different value, in absence of sample value.

A Practice Example

Example 8.4

Class:

C1:`buys_computer = 'yes'`

C2:`buys_computer = 'no'`

Data instance

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = fair)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

A Practice Example

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$

- Compute $P(X|C_i)$ for each class

$$P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$$

$$P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

$$P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$$

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

$$P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$$

$$P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$$

Therefore, X belongs to class ("buys_computer = yes")

Estimating conditional probability for continuous values: example

- Example: Continuous-valued Features
 - Temperature is naturally of continuous value.
Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8
No: 27.3, 30.1, 17.4, 29.5, 15.1
 - Estimate mean and variance for each class

$$\mu = \frac{1}{N} \sum_{n=1}^N x_n, \quad \sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2$$

– ~~Learning~~ Gaussian models for $P(\text{temp} | C)$

$$\begin{aligned}\mu_{\text{Yes}} &= 21.64, \quad \sigma_{\text{Yes}} = 2.35 \\ \mu_{\text{No}} &= 23.88, \quad \sigma_{\text{No}} = 7.09\end{aligned}$$

$$\hat{P}(x | \text{Yes}) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2 \times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x | \text{No}) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2 \times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

M-estimate: example

p: prior estimate of the probability
m: equivalent sample size (constant) In the absence of other information, assume a uniform prior: p = 1/k where k is the number of values that the attribute a_i can take.

Example: $P(\text{outlook}=\text{overcast}|\text{no})=0$ in the play-tennis dataset

- Adding m "virtual" examples (m : up to 1% of #training example)
 - In this dataset, # of training examples for the "no" class is 5.
 - We can only add $m=1$ "virtual" example in our m-estimate remedy.
- The "outlook" feature can takes only 3 values. So $p=1/3$.
- Re-estimate $P(\text{outlook}|\text{no})$ with the m-estimate

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \quad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$

E-mail classification

Training data: a corpus of email messages, each message annotated as spam or no spam.

Task: classify new email messages as spam/no spam.

To use a naive Bayes classifier for this task, we have to first find an **attribute representation** of the data.

Treat each text position as an attribute, with as its value the word at this position. Example: email starts: *get rich*.

The naive Bayes classifier is then:

$$\begin{aligned}v_{\text{NB}} &= \arg \max_{v_j \in \{\text{spam}, \text{nospam}\}} P(v_j) \prod_i P(a_i | v_j) \\&= \arg \max_{v_j \in \{\text{spam}, \text{nospam}\}} P(v_j) P(a_1 = \text{get} | v_j) P(a_2 = \text{rich} | v_j)\end{aligned}$$

Using naive Bayes means we assume that **words are independent of each** other. Clearly incorrect, but doesn't hurt a lot for our task.

The classifier uses $P(a_i = w_k | v_j)$, i.e., the probability that the i -th word in the email is the k -word in our vocabulary, given the email has been classified as v_j .

Simplify by assuming that **position is irrelevant**: estimate $P(w_k | v_j)$, i.e., the probability that word w_k occurs in the email, given class v_j .

Create a **vocabulary**: make a list of all words in the training corpus, discard words with very high or very low frequency.

Training: estimate priors:

$$P(v_j) = \frac{n}{N}$$

Estimate likelihoods using the ***m*-estimate**:

$$P(w_k|v_j) = \frac{n_k+1}{n+|\text{Vocabulary}|}$$

N : total number of words in all emails

n : number of words in emails with class v_j

n_k : number of times word w_k occurs in emails with class v_j

$|\text{Vocabulary}|$: size of the vocabulary

Testing: to classify a new email, assign it the class with the highest posterior probability. Ignore unknown words.

Bayes Error rate

- Suppose we know the true probability distribution that governs $P(X|Y)$.
- The Bayesian classification method allows us to determine the ideal decision boundary for the classification task
- Example:
 - Consider the task of identifying alligators and crocodiles based on their respective lengths. The average length of an adult crocodile is about 15 feet, while the average length of an adult alligator is about 12 feet

- Assuming that their length x follows a Gaussian distribution with a standard deviation equal to 2 feet, we can express their class-conditional probabilities as follows:

$$P(X|\text{Crocodile}) = \frac{1}{\sqrt{2\pi} \cdot 2} \exp \left[-\frac{1}{2} \left(\frac{X - 15}{2} \right)^2 \right]$$

$$P(X|\text{Alligator}) = \frac{1}{\sqrt{2\pi} \cdot 2} \exp \left[-\frac{1}{2} \left(\frac{X - 12}{2} \right)^2 \right]$$

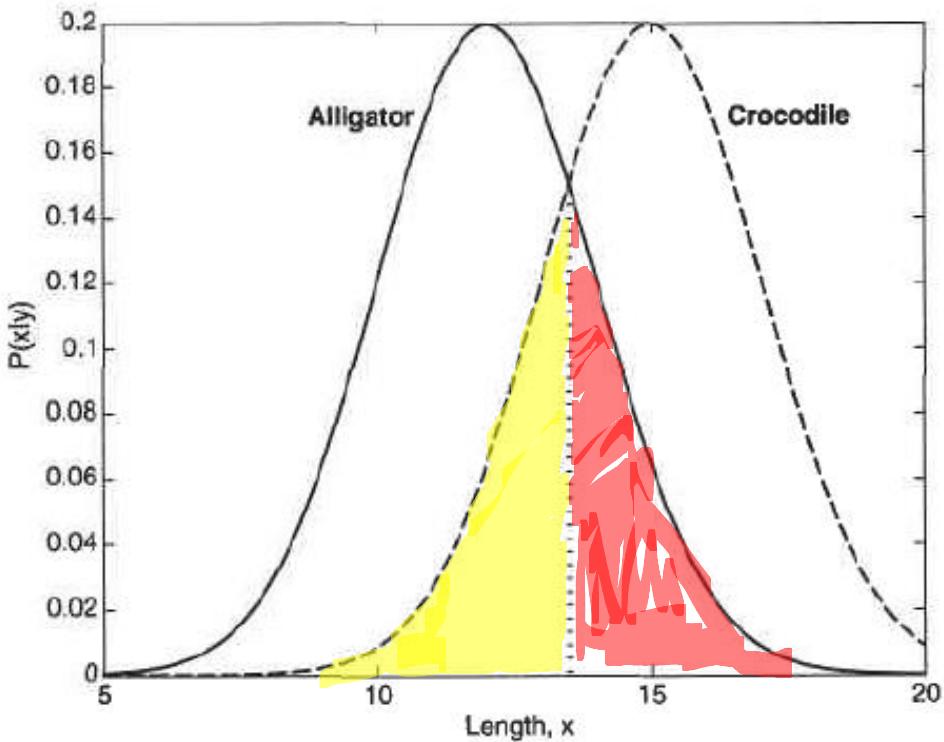
- Assuming that their prior probabilities are the same, the ideal decision boundary is located at some length \hat{x} such that

$$P(X = \hat{x}|\text{Crocodile}) = P(X = \hat{x}|\text{Alligator})$$

Using above equations:

$$\left(\frac{\hat{x} - 15}{2}\right)^2 = \left(\frac{\hat{x} - 12}{2}\right)^2,$$

which can be solved to yield $\hat{x} = 13.5$.
The decision boundary for this example
is located halfway between the two means.



What happens if the prior probabilities are different?

- The ideal decision boundary in the preceding example classifies all creatures whose lengths are less than \hat{x} as alligators and those whose lengths are greater than \hat{x} as crocodiles.
- The error rate of the classifier is given by the sum of the area under the posterior probability curve for crocodiles (from length 0 to \hat{x}) and the area under the posterior probability curve for alligators (from \hat{x} to ∞):

$$\text{Error} = \int_0^{\hat{x}} P(\text{Crocodile}|X)dX + \int_{\hat{x}}^{\infty} P(\text{Alligator}|X)dX.$$

- The total error rate is known as the **Bayes error rate**.

Multiclass Classification

- One vs. rest approach
- One against one approach

K classes

For each class $y_i \in Y$
a binary classifier. B_i

$(1 - r)$ approach =

