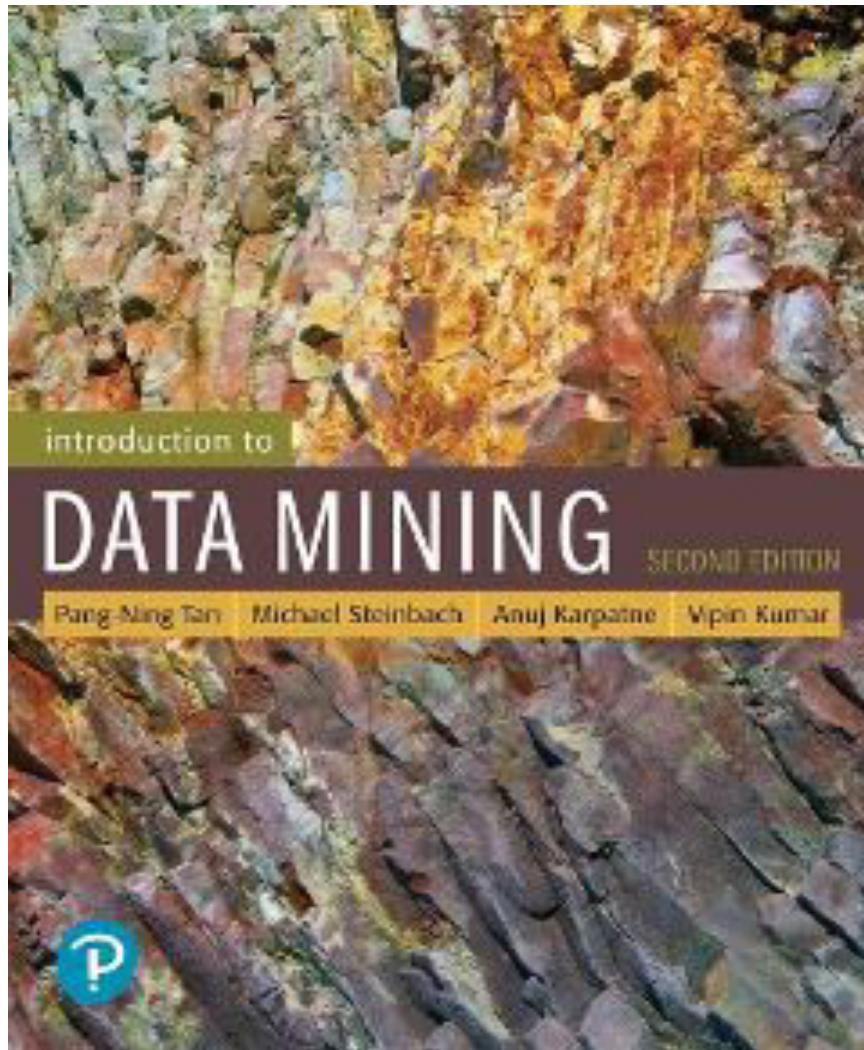


CSCE 5380/4380 – Data Mining



Chapter Two: **Data**

Outline

- **Attributes and Objects**
- **Types of Data**
- **Data Quality**
- **Similarity/ Dissimilarity Measures**
- **Data Preprocessing**

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, dimension, 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 for a particular object
- 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 can be different than the properties of the values used to represent the attribute

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 {tall, medium, short}
 - Interval
 - ◆ Examples: calendar dates, temperatures in Celsius or Fahrenheit.
 - Ratio
 - ◆ Examples: temperature in Kelvin, length, counts, elapsed time (e.g., time to run a race)

Properties of Attribute Values

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

Difference Between Ratio and Interval

- Is it physically meaningful to say that a temperature of 10° is twice that of 5° on
 - the Celsius scale?
 - the Fahrenheit scale?
 - the Kelvin scale?
- Consider measuring the height above average
 - If Bill's height is three inches above average and Bob's height is six inches above average, then would we say that Bob is twice as tall as Bill?
 - Is this situation analogous to that of temperature?

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

This categorization of attributes is due to S. S. Stevens

Attribute Type	Transformation	Comments
Categorical Qualitative	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., $new_value = f(old_value)$ 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}.
Numeric Quantitative	Interval	$new_value = a * old_value + b$ 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	$new_value = a * old_value$ Length can be measured in meters or feet.

This categorization of attributes is due to S. S. Stevens

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.

Asymmetric Attributes

- Only presence (a non-zero attribute value) is regarded as important
 - ◆ Words present in documents
 - ◆ Items present in customer transactions
- If we met a friend in the grocery store would we ever say the following?

“I see our purchases are very similar since we didn’t buy most of the same things.”

Critiques of the attribute categorization

- Incomplete
 - Asymmetric binary
 - Cyclical
 - Multivariate
 - Partially ordered
 - Partial membership
 - Relationships between the data
- Real data is approximate and noisy
 - This can complicate recognition of the proper attribute type
 - Treating one attribute type as another may be approximately correct

Key Messages for Attribute Types

- The types of operations you choose should be “meaningful” for the type of data you have
 - Distinctness, order, meaningful intervals, and meaningful ratios are only four (among many possible) properties of data
 - The data type you see – often numbers or strings – may not capture all the properties or may suggest properties that are not present
 - Analysis may depend on these other properties of the data
 - ◆ Many statistical analyses depend only on the distribution
 - In the end, what is meaningful can be specific to domain

Important Characteristics of Data

- Dimensionality (number of attributes)
 - ◆ High dimensional data brings a number of challenges
- Sparsity
 - ◆ Only presence counts
- Resolution
 - ◆ Patterns depend on the scale
- Size
 - ◆ Type of analysis may depend on size of data

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

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 a 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	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

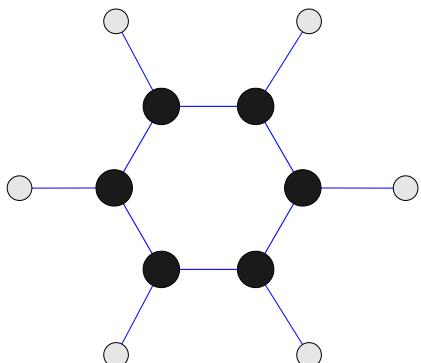
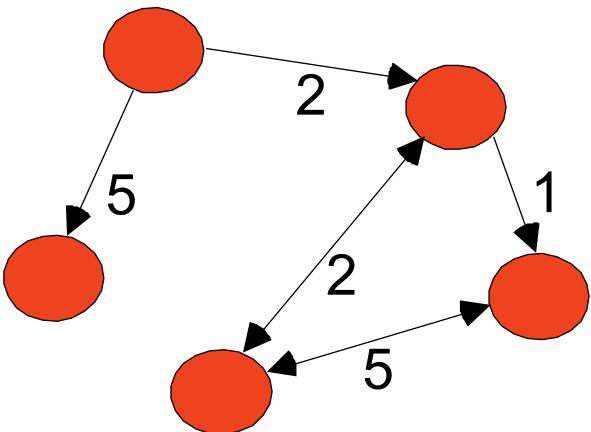
Transaction Data

- A special type of data, where
 - Each 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.
 - Can represent transaction data as record data

<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, a molecule, and webpages



Benzene Molecule: C_6H_6

Useful Links:

- [Bibliography](#)
- Other Useful Web sites
 - [ACM SIGKDD](#)
 - [KDnuggets](#)
 - [The Data Mine](#)

Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

- [Books](#)
- [General Data Mining](#)

Book References in Data Mining and Knowledge Discovery

Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Iyer, "Advances in Knowledge Discovery and Data Mining", AAAI Press/the MIT Press, 1996.

J. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993. Michael Berry and Gordon Linoff, "Data Mining Techniques (For Marketing, Sales, and Customer Support)", John Wiley & Sons, 1997.

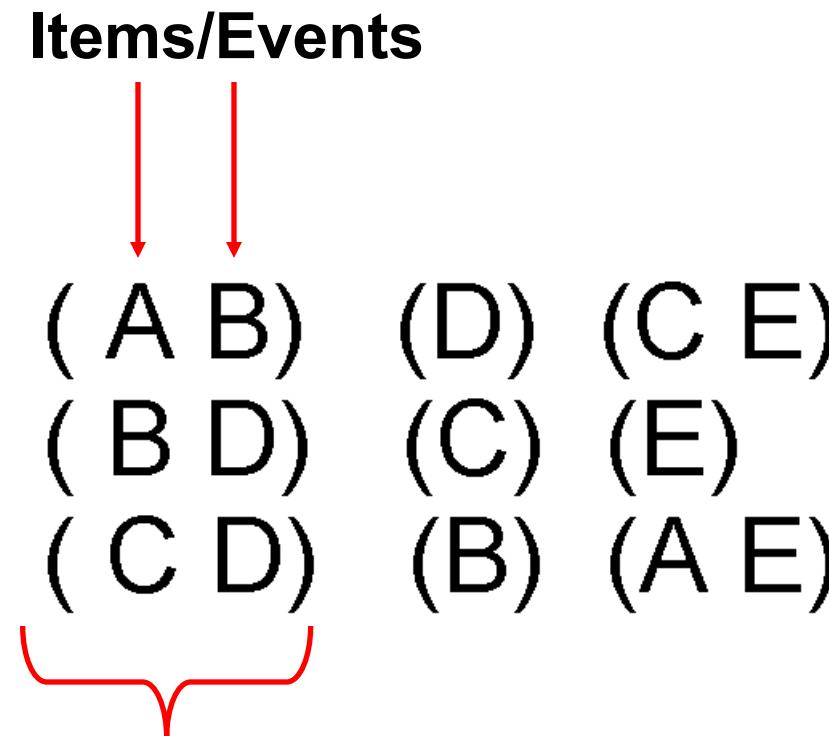
General Data Mining

Usama Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 21, no. 1, March 1998.

Christopher Matheus, Philip Chan, and Gregory Piatetsky-Shapiro, "Systems for knowledge Discovery in databases", IEEE Transactions on Knowledge and Data Engineering, 5(6):903-913, December 1993.

Ordered Data

- Sequences of transactions



Ordered Data

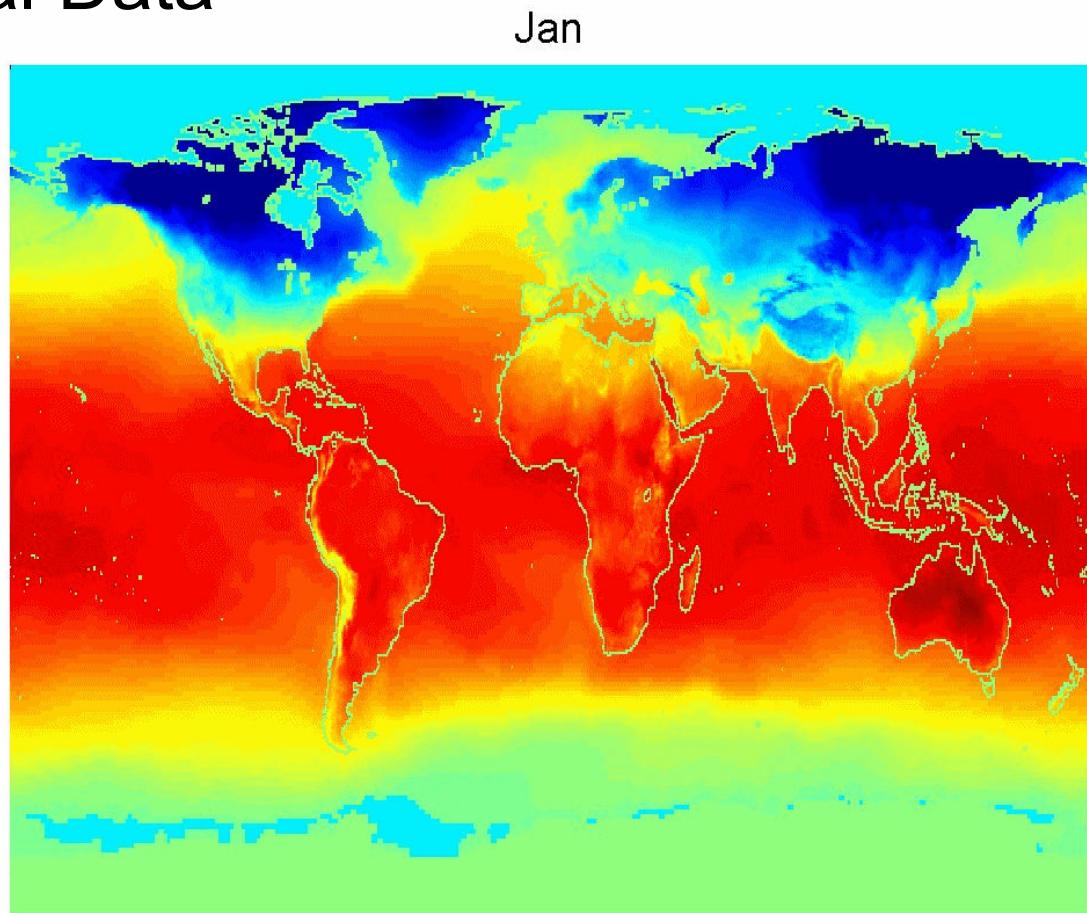
- Genomic sequence data

GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCAGGGGCCGCCCAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCAGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG

Ordered Data

- Spatio-Temporal Data

Average Monthly Temperature of land and ocean



Data Quality

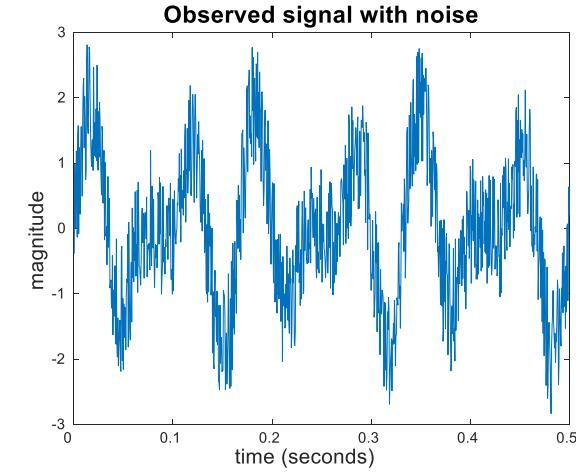
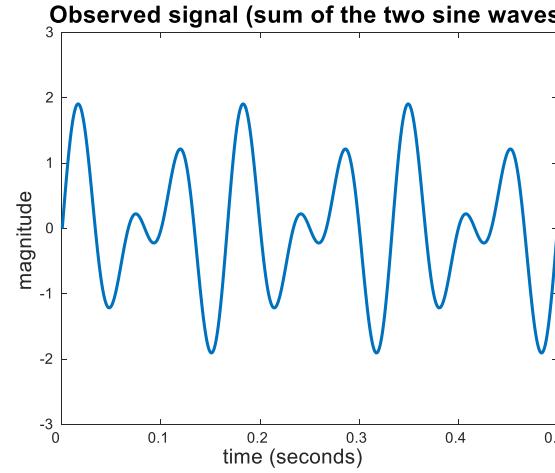
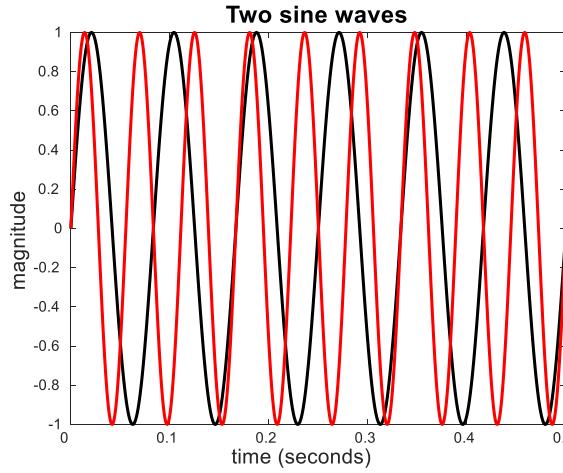
- Poor data quality negatively affects many data processing efforts
- Data mining example: a classification model for detecting people who are loan risks is built using poor data
 - Some credit-worthy candidates are denied loans
 - More loans are given to individuals that default

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
 - Wrong data
 - Fake data
 - Missing values
 - Duplicate data

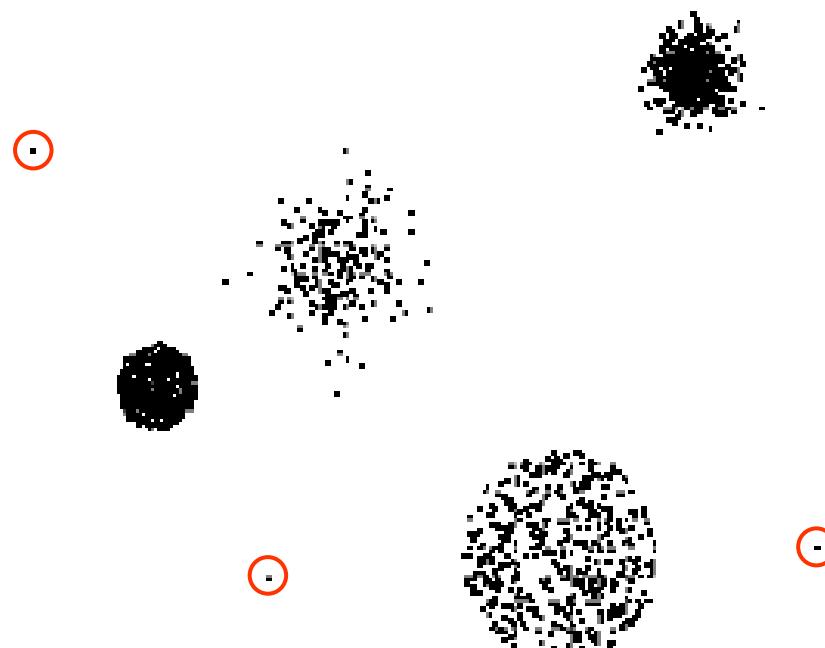
Noise

- For objects, noise is an extraneous object
- For attributes, noise refers to modification of original values
 - Examples: distortion of a person's voice when talking on a poor phone and "snow" on television screen
 - The figures below show two sine waves of the same magnitude and different frequencies, the waves combined, and the two sine waves with random noise
 - ◆ The magnitude and shape of the original signal is distorted



Outliers

- **Outliers** are data objects with characteristics that are considerably different than most of the other data objects in the data set
 - **Case 1:** Outliers are noise that interferes with data analysis
 - **Case 2:** Outliers are the goal of the analysis
 - ◆ Credit card fraud
 - ◆ Intrusion detection



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 or variables
 - Estimate missing values
 - ◆ Example: time series of temperature
 - ◆ Example: census results
 - Ignore the missing value during analysis

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
- When should duplicate data not be removed?

Similarity and Dissimilarity Measures

- Similarity measure
 - Numerical measure of how alike two data objects are.
 - Is higher when objects are more alike.
 - Often falls in the range [0, 1]
- Dissimilarity measure
 - Numerical measure of how different two data objects are
 - 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

The following table shows the similarity and dissimilarity between two objects, x and y , with respect to a single, simple attribute.

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

Euclidean Distance

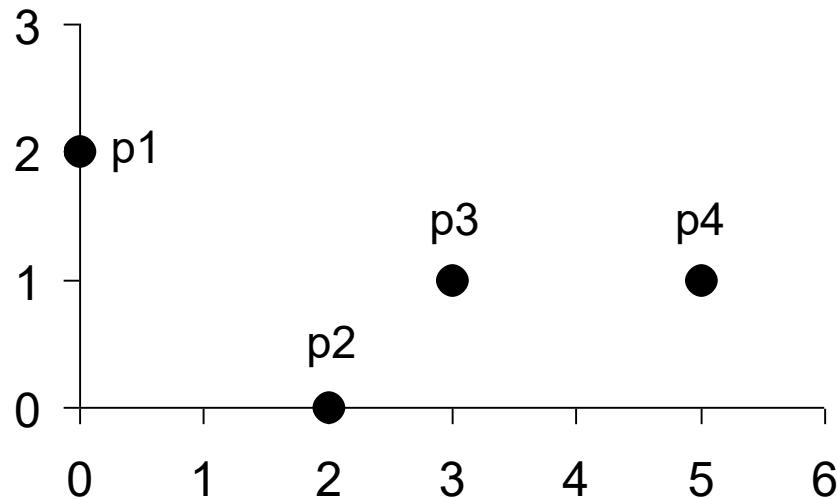
- Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects \mathbf{x} and \mathbf{y} .

- 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

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{1/r}$$

Where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects x and y .

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
 - A common example of this for binary vectors 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

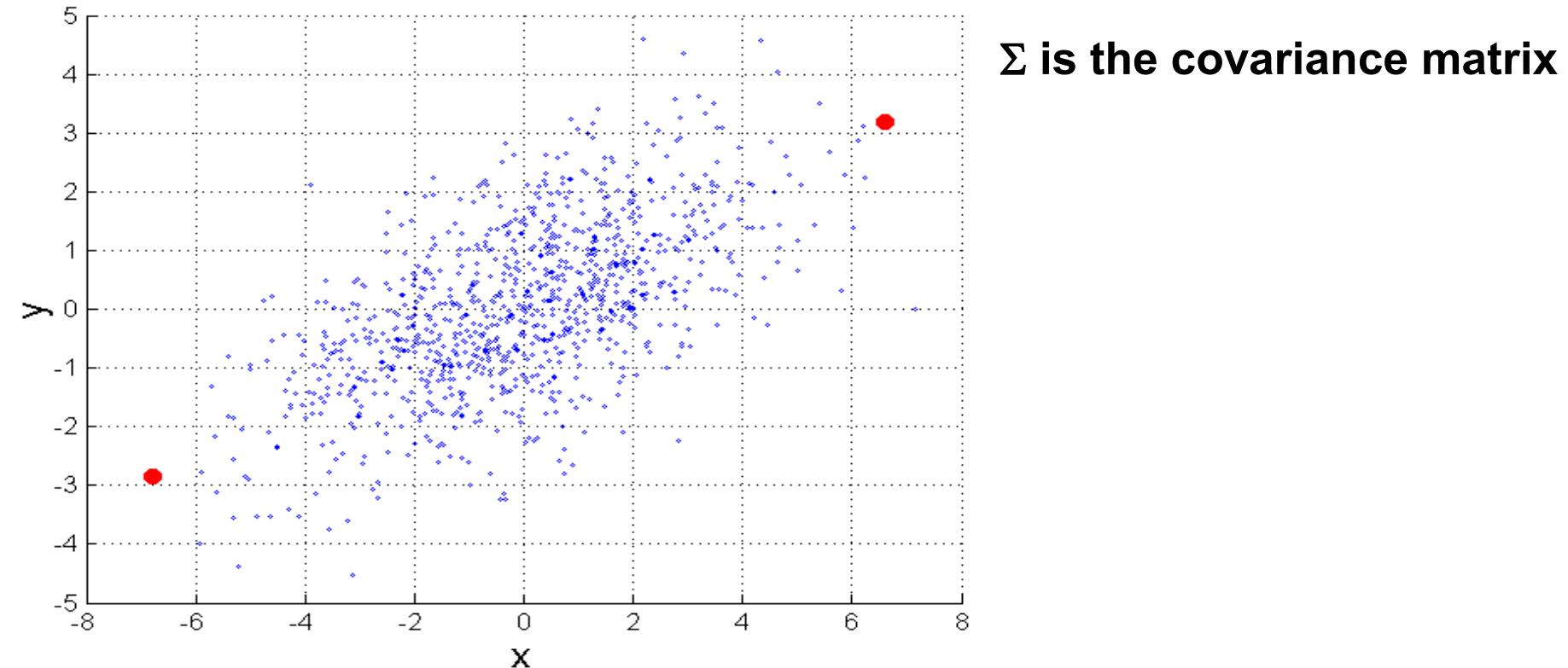
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

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

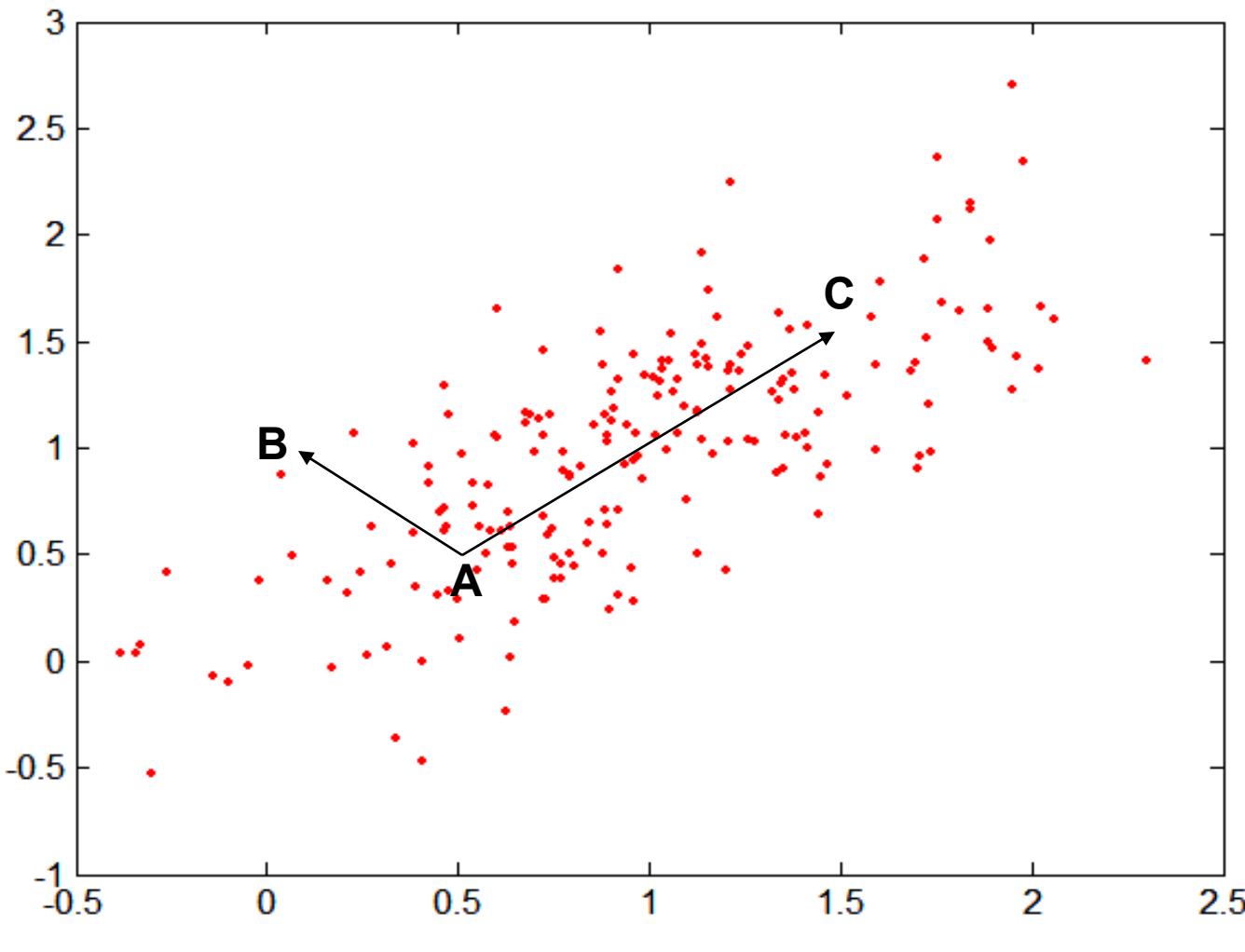
Mahalanobis Distance

$$\text{mahalanobis}(x, y) = ((x - y)^T \Sigma^{-1} (x - y))^{-0.5}$$



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. $d(\mathbf{x}, \mathbf{y}) \geq 0$ for all \mathbf{x} and \mathbf{y} and $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$.
 2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all \mathbf{x} and \mathbf{y} . (Symmetry)
 3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ for all points \mathbf{x} , \mathbf{y} , and \mathbf{z} . (Triangle Inequality)

where $d(\mathbf{x}, \mathbf{y})$ is the distance (dissimilarity) between points (data objects), \mathbf{x} and \mathbf{y} .

- A distance that satisfies these properties is a **metric**

Common Properties of a Similarity

- Similarities, also have some well known properties.
 1. $s(\mathbf{x}, \mathbf{y}) = 1$ (or maximum similarity) only if $\mathbf{x} = \mathbf{y}$.
(does not always hold, e.g., cosine)
 2. $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$ for all \mathbf{x} and \mathbf{y} . (Symmetry)

where $s(\mathbf{x}, \mathbf{y})$ is the similarity between points (data objects), \mathbf{x} and \mathbf{y} .

Similarity Between Binary Vectors

- Common situation is that objects, x and y , have only binary attributes

- Compute similarities using the following quantities

f_{01} = the number of attributes where x was 0 and y was 1

f_{10} = the number of attributes where x was 1 and y was 0

f_{00} = the number of attributes where x was 0 and y was 0

f_{11} = the number of attributes where x was 1 and y was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

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

J = number of 11 matches / number of non-zero attributes

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

SMC versus Jaccard: Example

$\mathbf{x} = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$\mathbf{y} = 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$

$f_{01} = 2$ (the number of attributes where \mathbf{x} was 0 and \mathbf{y} was 1)

$f_{10} = 1$ (the number of attributes where \mathbf{x} was 1 and \mathbf{y} was 0)

$f_{00} = 7$ (the number of attributes where \mathbf{x} was 0 and \mathbf{y} was 0)

$f_{11} = 0$ (the number of attributes where \mathbf{x} was 1 and \mathbf{y} was 1)

$$\begin{aligned}\text{SMC} &= (f_{11} + f_{00}) / (f_{01} + f_{10} + f_{11} + f_{00}) \\ &= (0+7) / (2+1+0+7) = 0.7\end{aligned}$$

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

Cosine Similarity

- If \mathbf{d}_1 and \mathbf{d}_2 are two document vectors, then

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = \langle \mathbf{d}_1, \mathbf{d}_2 \rangle / \|\mathbf{d}_1\| \|\mathbf{d}_2\|,$$

where $\langle \mathbf{d}_1, \mathbf{d}_2 \rangle$ indicates inner product or vector dot product of vectors, \mathbf{d}_1 and \mathbf{d}_2 , and $\|\mathbf{d}\|$ is the length of vector \mathbf{d} .

- Example:

$$\mathbf{d}_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$\mathbf{d}_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

$$\langle \mathbf{d}_1, \mathbf{d}_2 \rangle = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|\mathbf{d}_1\| = (3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$$

$$\|\mathbf{d}_2\| = (1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2)^{0.5} = (6)^{0.5} = 2.449$$

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = 0.3150$$

Correlation measures the linear relationship between objects

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard_deviation}(\mathbf{x}) * \text{standard_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x s_y}, \quad (2.11)$$

where we are using the following standard statistical notation and definitions

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y}) \quad (2.12)$$

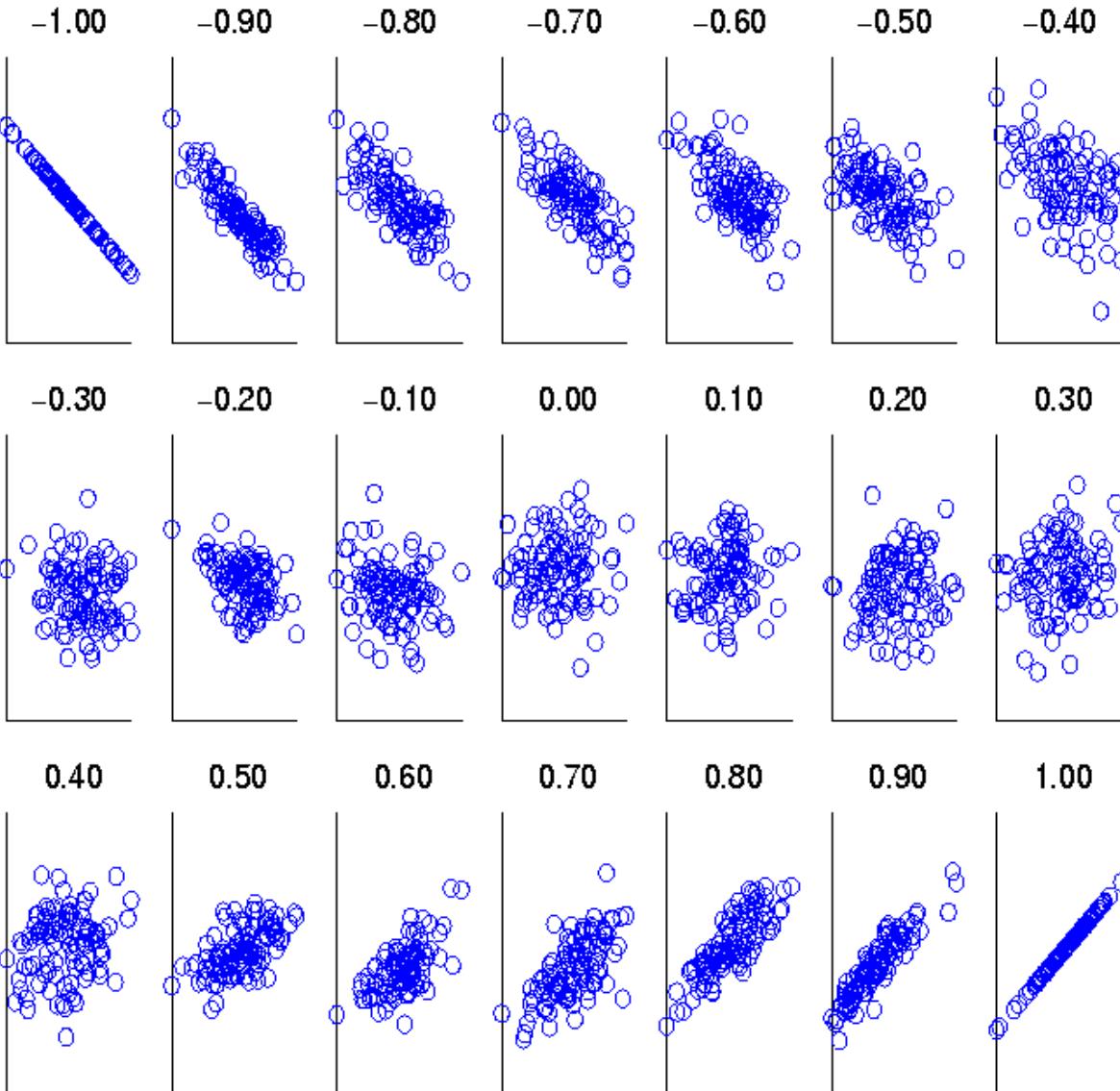
$$\text{standard_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$$

$$\text{standard_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \bar{y})^2}$$

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k \text{ is the mean of } \mathbf{x}$$

$$\bar{y} = \frac{1}{n} \sum_{k=1}^n y_k \text{ is the mean of } \mathbf{y}$$

Visually Evaluating Correlation



**Scatter plots
showing the
similarity from
-1 to 1.**

Correlation vs Cosine vs Euclidean Distance

- Compare the three proximity measures according to their behavior under variable transformation
 - scaling: multiplication by a value
 - translation: adding a constant

Property	Cosine	Correlation	Euclidean Distance
Invariant to scaling (multiplication)	Yes	Yes	No
Invariant to translation (addition)	No	Yes	No

- Consider the example
 - $\mathbf{x} = (1, 2, 4, 3, 0, 0, 0)$, $\mathbf{y} = (1, 2, 3, 4, 0, 0, 0)$
 - $\mathbf{y}_s = \mathbf{y} * 2$ (scaled version of \mathbf{y}), $\mathbf{y}_t = \mathbf{y} + 5$ (translated version)

Measure	(\mathbf{x}, \mathbf{y})	$(\mathbf{x}, \mathbf{y}_s)$	$(\mathbf{x}, \mathbf{y}_t)$
Cosine	0.9667	0.9667	0.7940
Correlation	0.9429	0.9429	0.9429
Euclidean Distance	1.4142	5.8310	14.2127

Correlation vs cosine vs Euclidean distance

- Choice of the right proximity measure depends on the domain
- What is the correct choice of proximity measure for the following situations?
 - Comparing documents using the frequencies of words
 - ◆ Documents are considered similar if the word frequencies are similar
 - Comparing the temperature in Celsius of two locations
 - ◆ Two locations are considered similar if the temperatures are similar in magnitude
 - Comparing two time series of temperature measured in Celsius
 - ◆ Two time series are considered similar if their “shape” is similar, i.e., they vary in the same way over time, achieving minimums and maximums at similar times, etc.

Comparison of Proximity Measures

- Domain of application
 - Similarity measures tend to be specific to the type of attribute and data
 - Record data, images, graphs, sequences, 3D-protein structure, etc. tend to have different measures
- However, one can talk about various properties that you would like a proximity measure to have
 - Symmetry is a common one
 - Tolerance to noise and outliers is another
 - Ability to find more types of patterns?
 - Many others possible
- The measure must be applicable to the data and produce results that agree with domain knowledge

Information Based Measures

- Information theory is a well-developed and fundamental discipline with broad applications
- Some similarity measures are based on information theory
 - Mutual information in various versions
 - Maximal Information Coefficient (MIC) and related measures
 - General and can handle non-linear relationships
 - Can be complicated and time intensive to compute

Information and Probability

- Information relates to possible outcomes of an event
 - transmission of a message, flip of a coin, or measurement of a piece of data



- The more certain an outcome, the less information that it contains and vice-versa
 - For example, if a coin has two heads, then an outcome of heads provides no information
 - More quantitatively, the information is related to the probability of an outcome
 - ◆ The smaller the probability of an outcome, the more information it provides and vice-versa
 - Entropy is the commonly used measure

Entropy

- For
 - a variable (event), X ,
 - with n possible values (outcomes), $x_1, x_2 \dots, x_n$
 - each outcome having probability, $p_1, p_2 \dots, p_n$
 - the entropy of X , $H(X)$, is given by
$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i$$
- Entropy is between 0 and $\log_2 n$ and is measured in bits
 - Thus, entropy is a measure of how many bits it takes to represent an observation of X on average

Entropy Examples

- For a coin with probability p of heads and probability $q = 1 - p$ of tails

$$H = -p \log_2 p - q \log_2 q$$

- For $p=0.5, q=0.5$ (fair coin) $H=1$
 - For $p = 1$ or $q = 1$, $H = 0$
-
- What is the entropy of a fair four-sided die?

Entropy for Sample Data: Example

Hair Color	Count	p	$-p \log_2 p$
Black	75	0.75	0.3113
Brown	15	0.15	0.4105
Blond	5	0.05	0.2161
Red	0	0.00	0
Other	5	0.05	0.2161
Total	100	1.0	1.1540

Maximum entropy is $\log_2 5 = 2.3219$

Entropy for Sample Data

- Suppose we have
 - a number of observations (m) of some attribute, X , e.g., the hair color of students in the class,
 - where there are n different possible values
 - And the number of observation in the i^{th} category is m_i
 - Then, for this sample

$$H(X) = - \sum_{i=1}^n \frac{m_i}{m} \log_2 \frac{m_i}{m}$$

- For continuous data, the calculation is harder

Mutual Information

- Information one variable provides about another

Formally, $I(X, Y) = H(X) + H(Y) - H(X, Y)$, where

$H(X, Y)$ is the joint entropy of X and Y ,

$$H(X, Y) = - \sum_i \sum_j p_{ij} \log_2 p_{ij}$$

Where p_{ij} is the probability that the i^{th} value of X and the j^{th} value of Y occur together

- For discrete variables, this is easy to compute
- Maximum mutual information for discrete variables is $\log_2(\min(n_X, n_Y))$, where n_X (n_Y) is the number of values of X (Y)

Mutual Information Example

Student Status	Count	p	$-p \log_2 p$
Undergrad	45	0.45	0.5184
Grad	55	0.55	0.4744
Total	100	1.00	0.9928

Grade	Count	p	$-p \log_2 p$
A	35	0.35	0.5301
B	50	0.50	0.5000
C	15	0.15	0.4105
Total	100	1.00	1.4406

Student Status	Grade	Count	p	$-p \log_2 p$
Undergrad	A	5	0.05	0.2161
Undergrad	B	30	0.30	0.5211
Undergrad	C	10	0.10	0.3322
Grad	A	30	0.30	0.5211
Grad	B	20	0.20	0.4644
Grad	C	5	0.05	0.2161
Total		100	1.00	2.2710

$$\text{Mutual information of Student Status and Grade} = 0.9928 + 1.4406 - 2.2710 = 0.1624$$

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, $s_k(\mathbf{x}, \mathbf{y})$, in the range [0, 1].

2: Define an indicator variable, δ_k , for the k^{th} attribute as follows:

$\delta_k = 0$ if the k^{th} attribute is an asymmetric attribute and both objects have a value of 0, or if one of the objects has a missing value for the k^{th} attribute

$\delta_k = 1$ otherwise

3. Compute $\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^n \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^n \delta_k}$

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
 - Use non-negative weights ω_k
 - $similarity(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^n \omega_k \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^n \omega_k \delta_k}$
- Can also define a weighted form of distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n w_k |x_k - y_k|^r \right)^{1/r}$$

Data Preprocessing

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

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.
 - Days aggregated into weeks, months, or years
 - More “stable” data - aggregated data tends to have less variability

Table 2.4. Data set containing information about customer purchases.

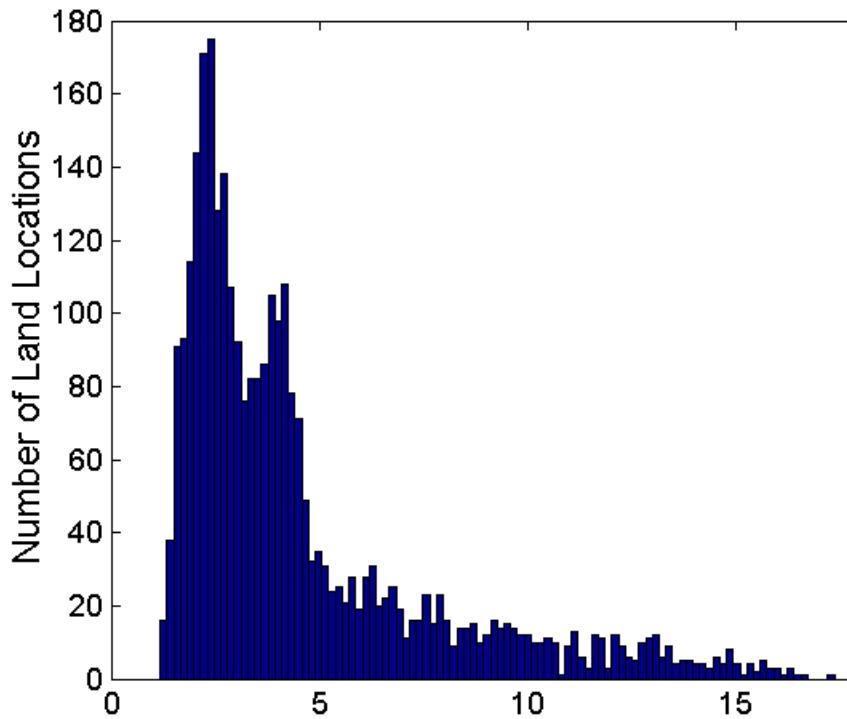
Transaction ID	Item	Store Location	Date	Price	...
:	:	:	:	:	
101123	Watch	Chicago	09/06/04	\$25.99	...
101123	Battery	Chicago	09/06/04	\$5.99	...
101124	Shoes	Minneapolis	09/06/04	\$75.00	...
:	:	:	:	:	

Example: Precipitation in Australia

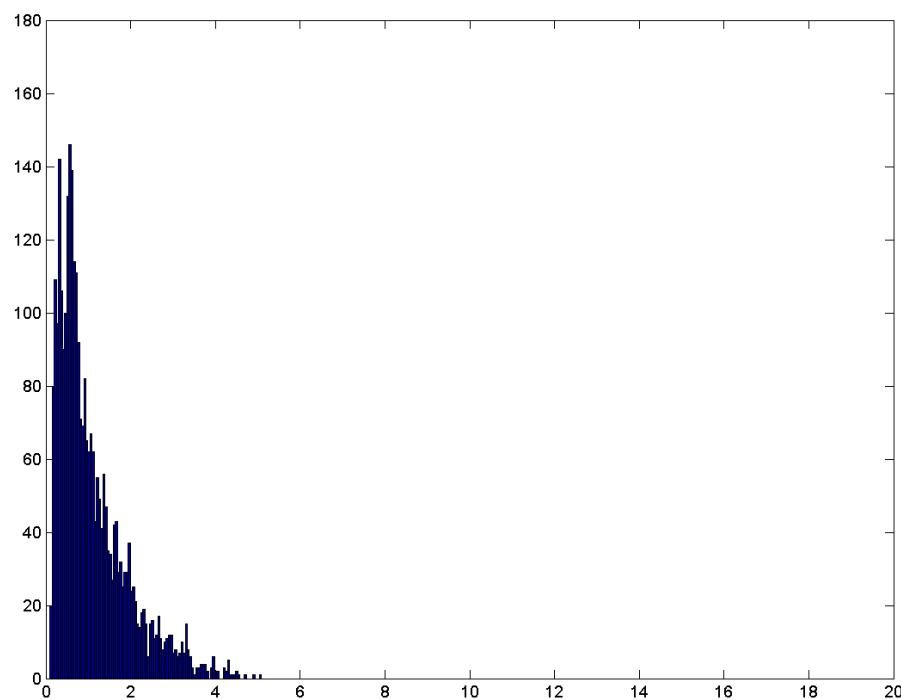
- This example is based on precipitation in Australia from the period 1982 to 1993.
 - The next slide shows
 - A histogram for the standard deviation of average monthly precipitation for 3,030 0.5° by 0.5° grid cells in Australia, and
 - A histogram for the standard deviation of the average yearly precipitation for the same locations.
- The average yearly precipitation has less variability than the average monthly precipitation.
- All precipitation measurements (and their standard deviations) are in centimeters.

Example: Precipitation in Australia ...

Variation of Precipitation in Australia



Standard Deviation of Average
Monthly Precipitation



Standard Deviation of
Average Yearly Precipitation

Sampling

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

Sampling ...

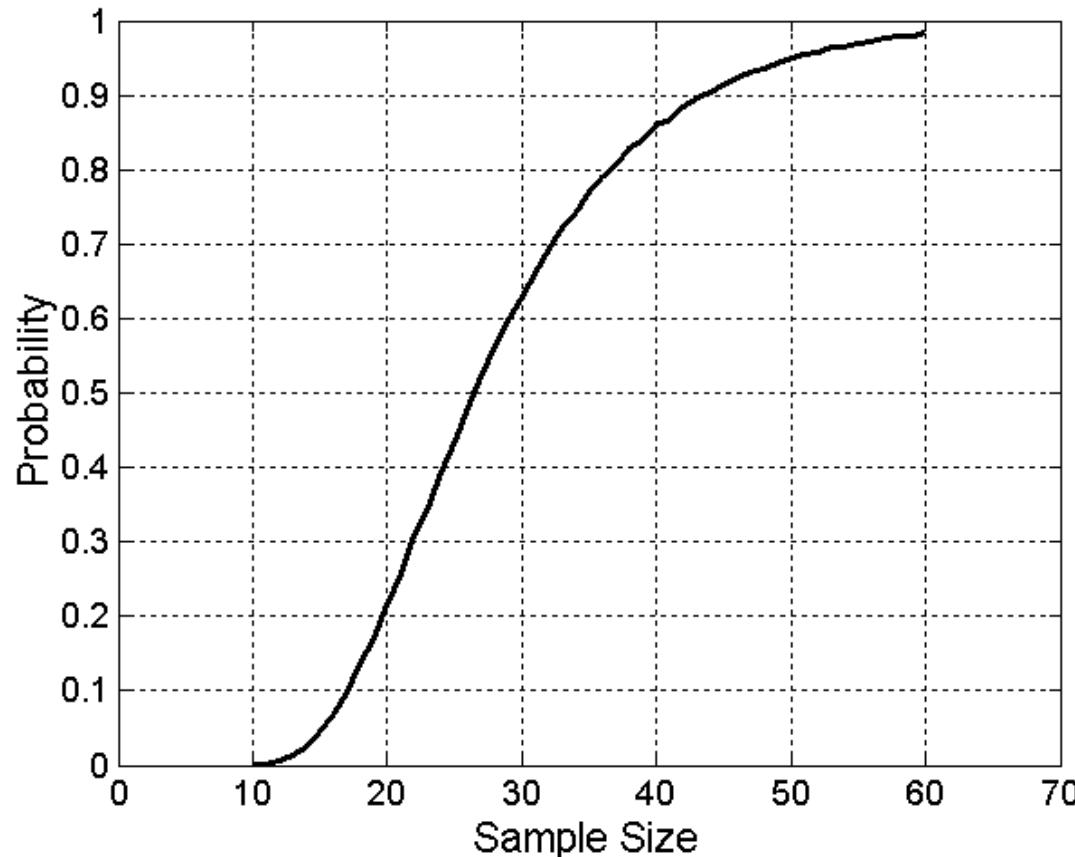
- The key principle for effective sampling is the following:
 - Using a sample will work almost as well as using the entire data set, if the sample is **representative**
 - A sample is **representative** if it has approximately the same properties (of interest) as the original set of data

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

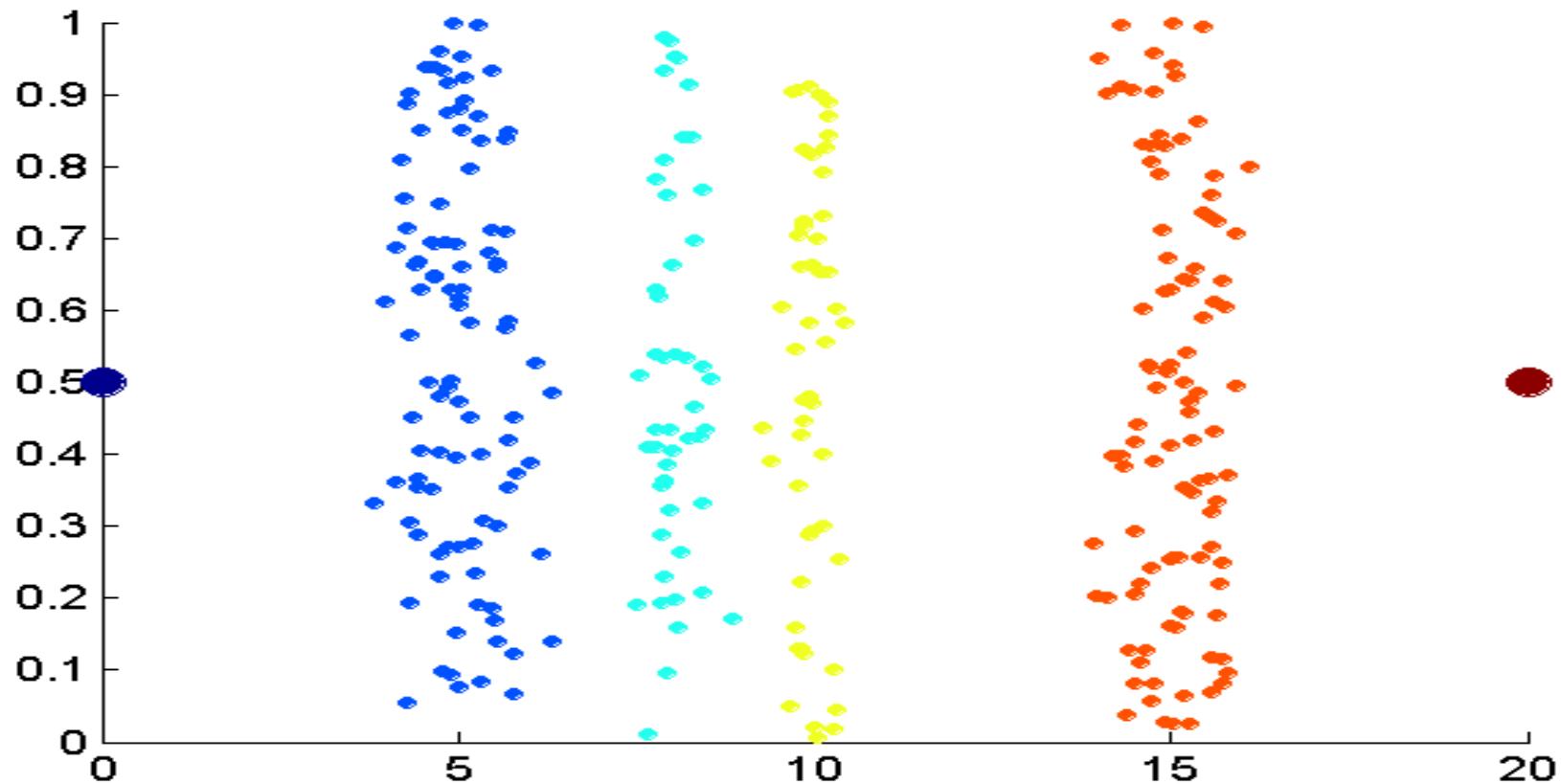
- What sample size is necessary to get at least one object from each of 10 equal-sized groups.



Discretization

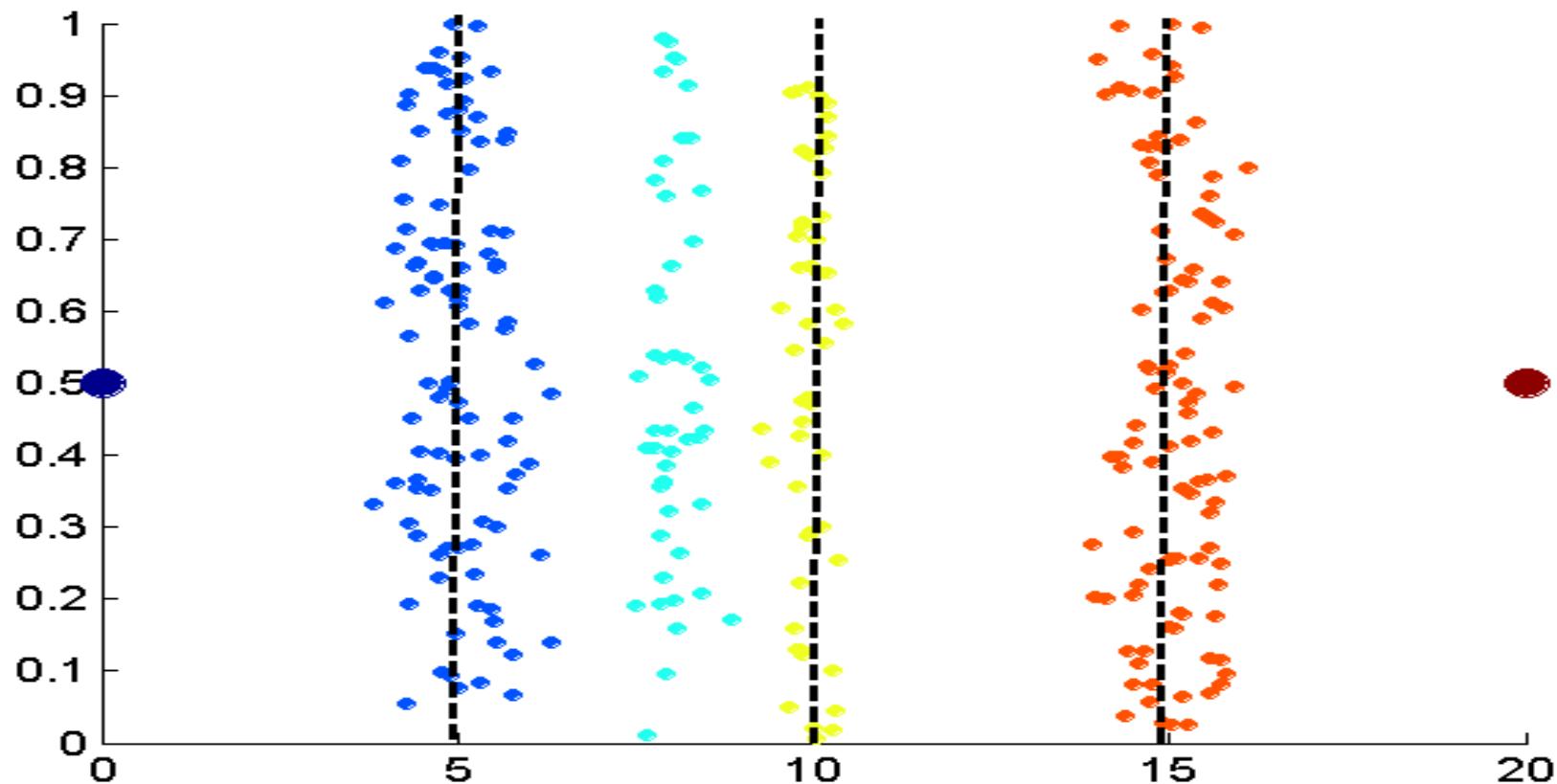
- **Discretization** is the process of converting a continuous attribute into an ordinal attribute
 - A potentially infinite number of values are mapped into a small number of categories
 - Discretization is used in both unsupervised and supervised settings

Unsupervised Discretization



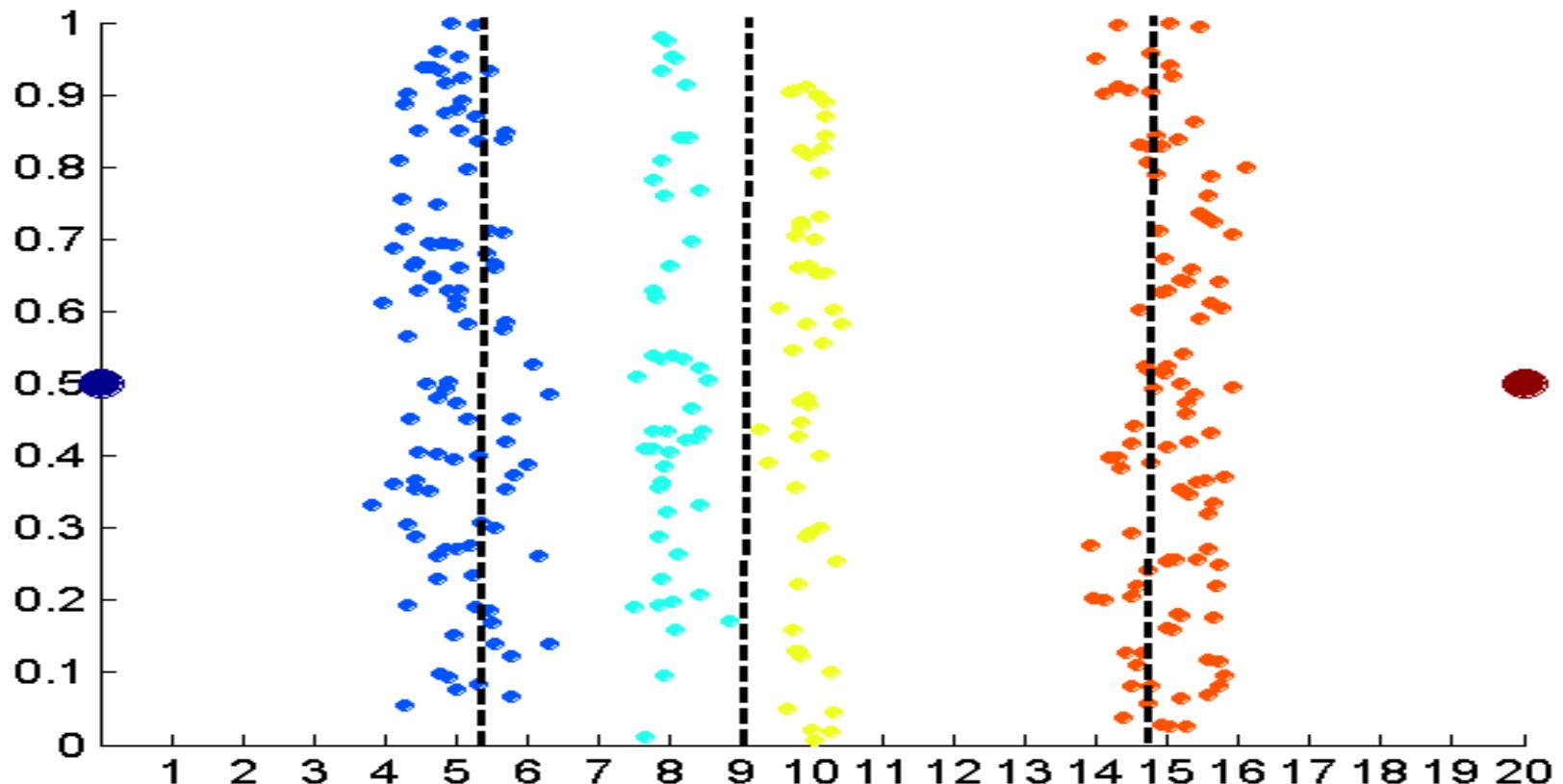
Data consists of four groups of points and two outliers. Data is one-dimensional, but a random y component is added to reduce overlap.

Unsupervised Discretization



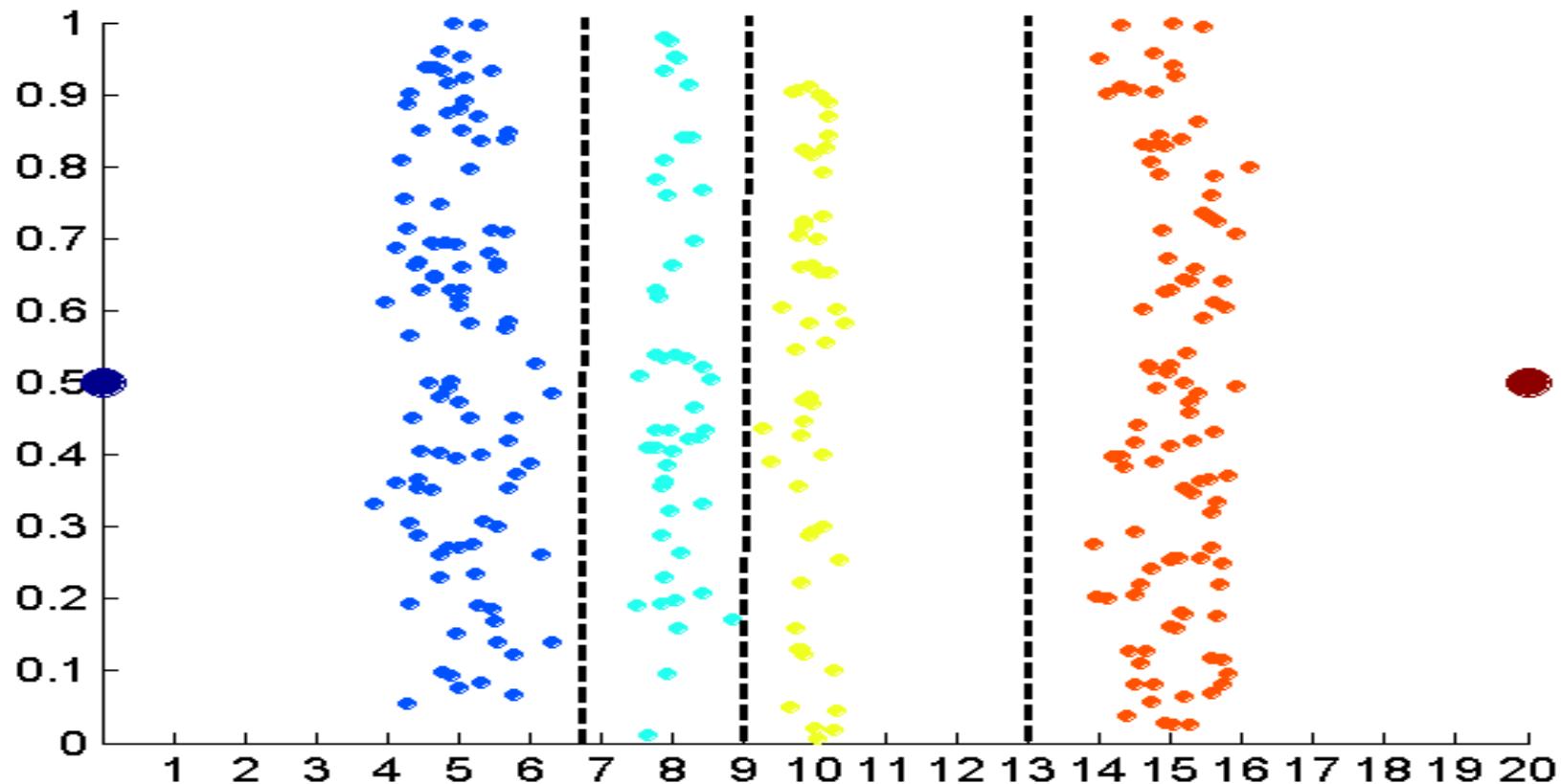
Equal interval width approach used to obtain 4 values.

Unsupervised Discretization



Equal frequency approach used to obtain 4 values.

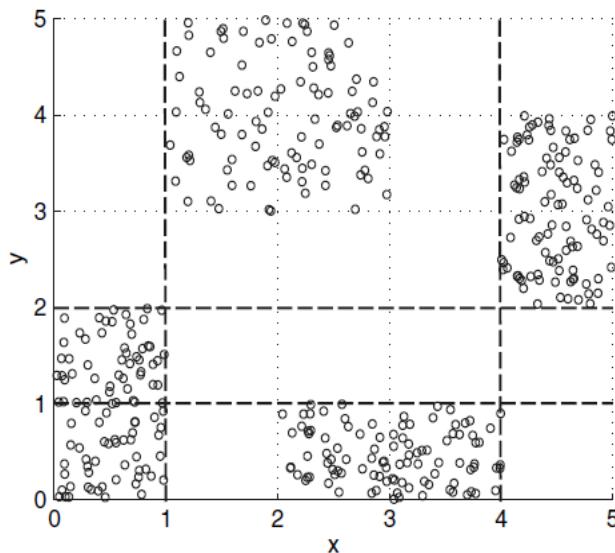
Unsupervised Discretization



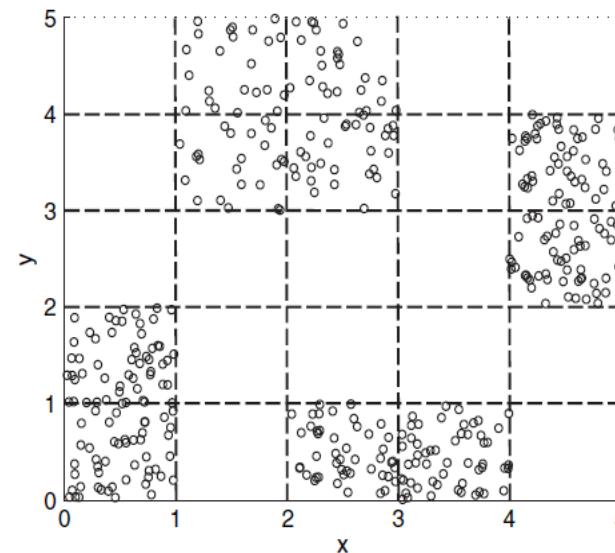
K-means approach to obtain 4 values.

Discretization in Supervised Settings

- Many classification algorithms work best if both the independent and dependent variables have only a few values
- We give an illustration of the usefulness of discretization using the following example.



(a) Three intervals



(b) Five intervals

Figure 2.14. Discretizing x and y attributes for four groups (classes) of points.

Binarization

- Binarization maps a continuous or categorical attribute into one or more binary variables

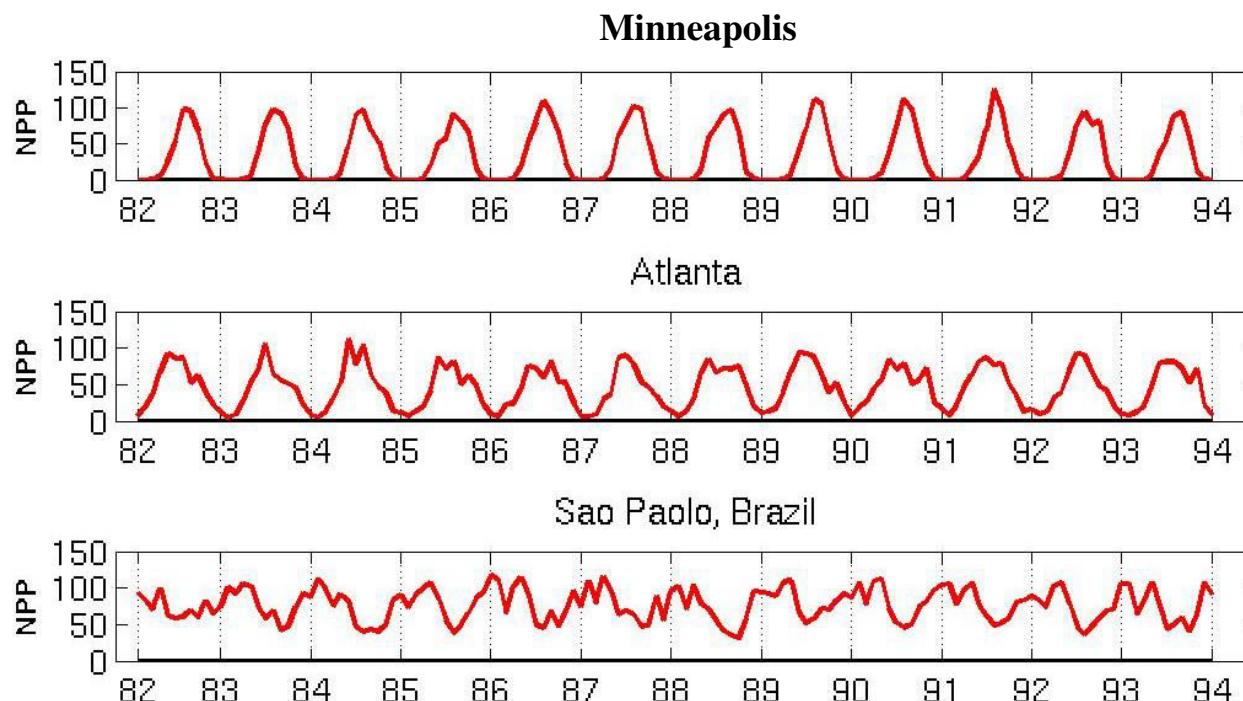
Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
<i>awful</i>	0	1	0	0	0	0
<i>poor</i>	1	0	1	0	0	0
<i>OK</i>	2	0	0	1	0	0
<i>good</i>	3	0	0	0	1	0
<i>great</i>	4	0	0	0	0	1

Attribute Transformation

- An **attribute transform** is 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 , $|x|$
 - **Normalization**
 - ◆ Refers to various techniques to adjust to differences among attributes in terms of frequency of occurrence, mean, variance, range
 - ◆ Take out unwanted, common signal, e.g., seasonality
 - In statistics, **standardization** refers to subtracting off the means and dividing by the standard deviation

Example: Sample Time Series of Plant Growth

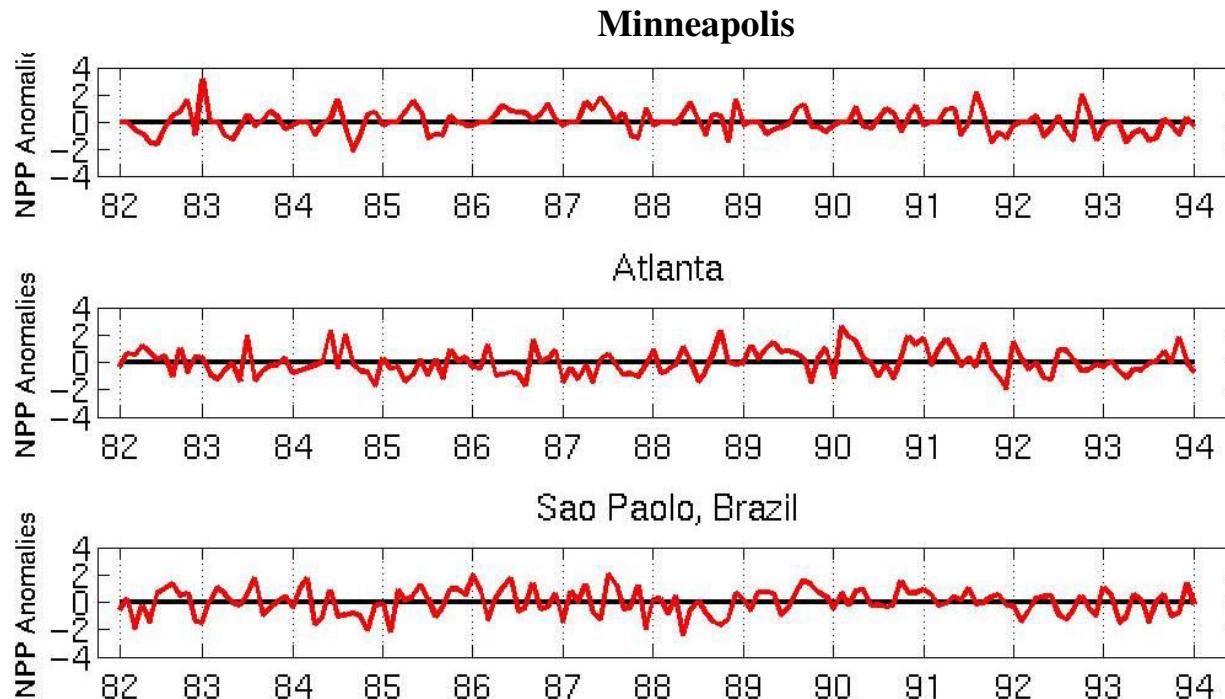


Net Primary Production (NPP) is a measure of plant growth used by ecosystem scientists.

Correlations between time series

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.7591	-0.7581
Atlanta	0.7591	1.0000	-0.5739
Sao Paolo	-0.7581	-0.5739	1.0000

Seasonality Accounts for Much Correlation



Normalized using monthly Z Score:

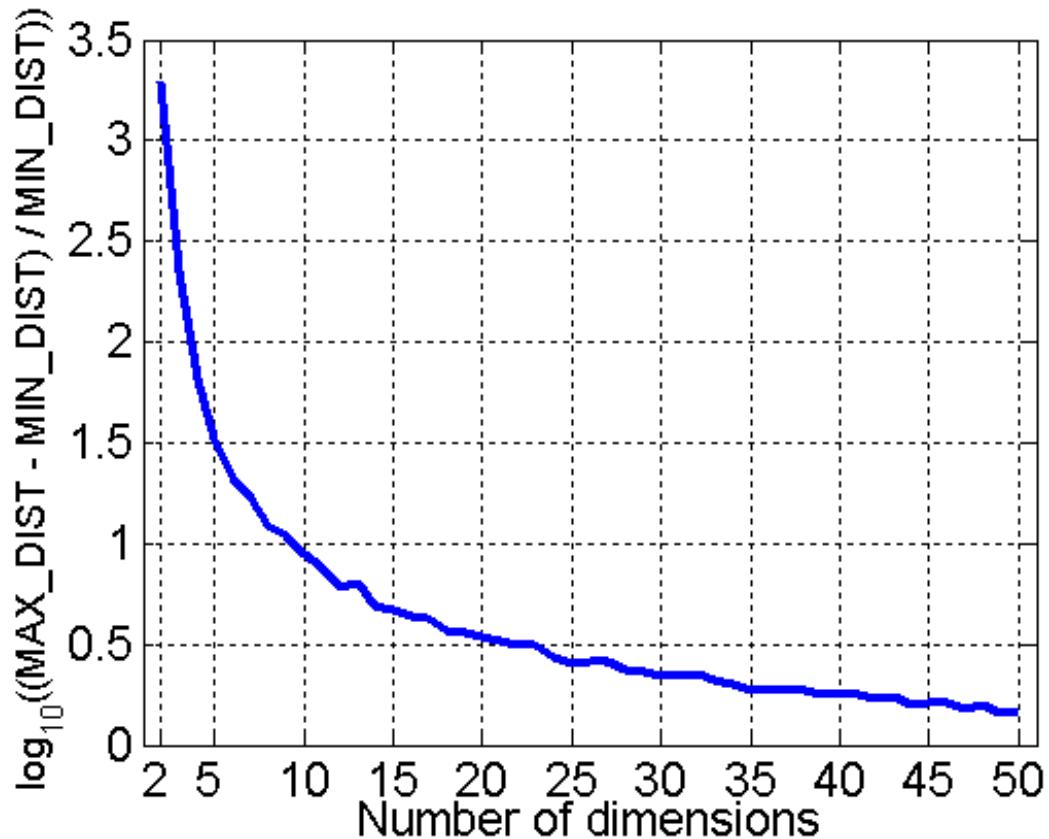
Subtract off monthly mean and divide by monthly standard deviation

Correlations between time series

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.0492	0.0906
Atlanta	0.0492	1.0000	-0.0154
Sao Paolo	0.0906	-0.0154	1.0000

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which are 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

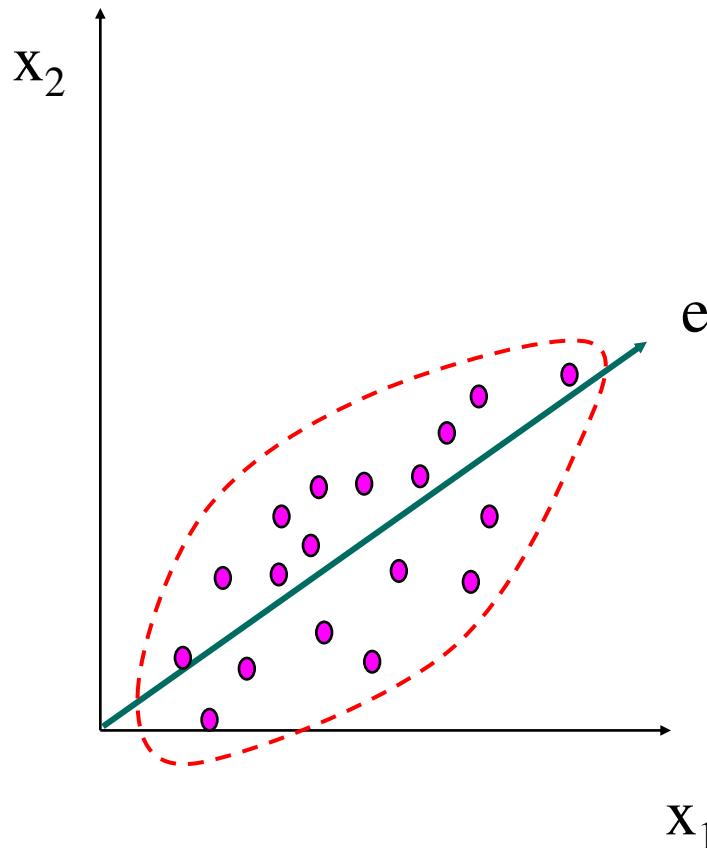
Dimensionality Reduction

- 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 Components Analysis (**PCA**)
 - 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



Feature Subset Selection

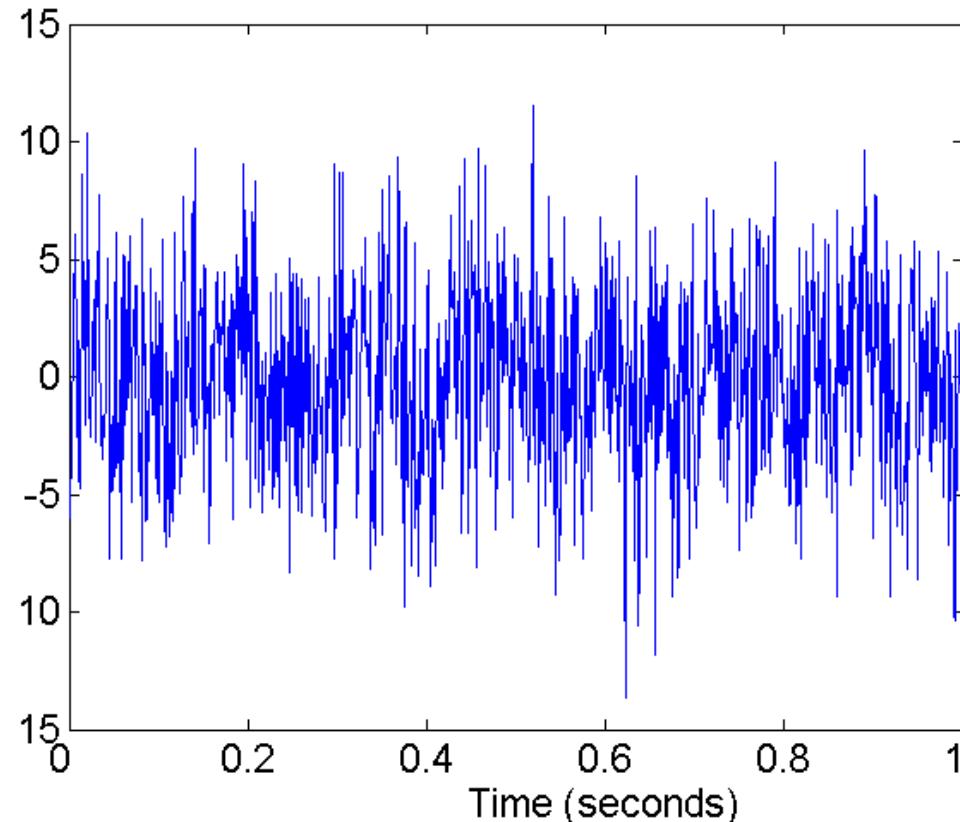
- Another way to reduce dimensionality of data
- Redundant features
 - Duplicate much or all of the information contained in one or more other attributes
 - Example: purchase price of a product and the amount of sales tax paid
- Irrelevant features
 - Contain no information that is useful for the data mining task at hand
 - Example: students' ID is often irrelevant to the task of predicting students' GPA
- Many techniques developed, especially for classification

Feature Creation

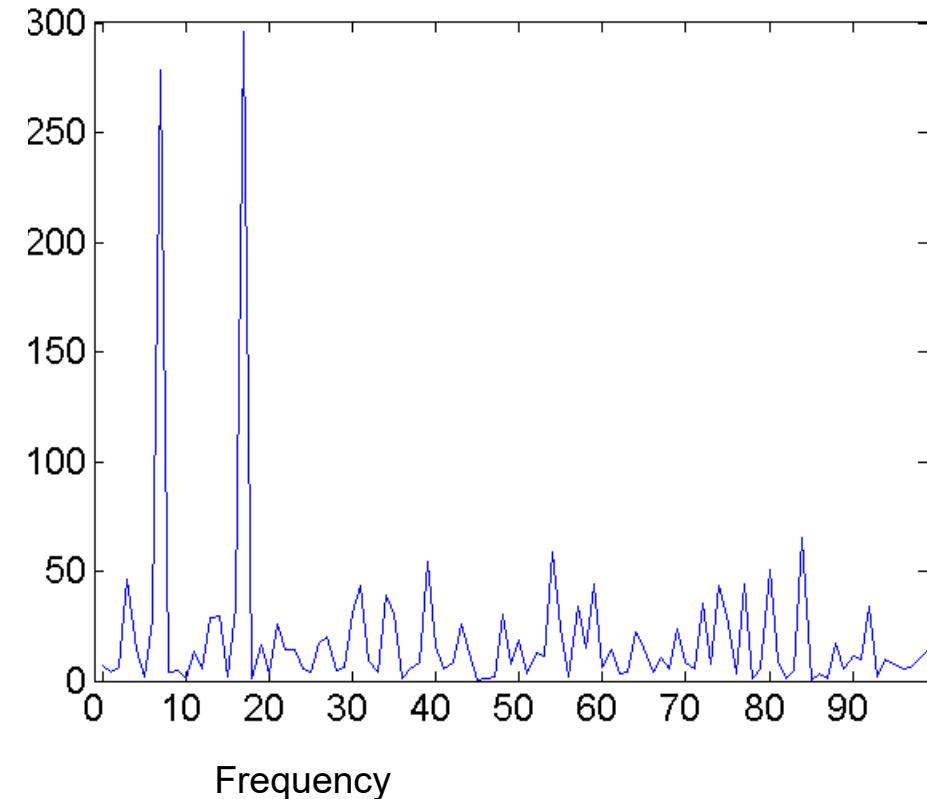
- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes
- Three general methodologies:
 - Feature extraction
 - ◆ Example: extracting edges from images
 - Feature construction
 - ◆ Example: dividing mass by volume to get density
 - Mapping data to new space
 - ◆ Example: Fourier and wavelet analysis

Mapping Data to a New Space

- Fourier and wavelet transform

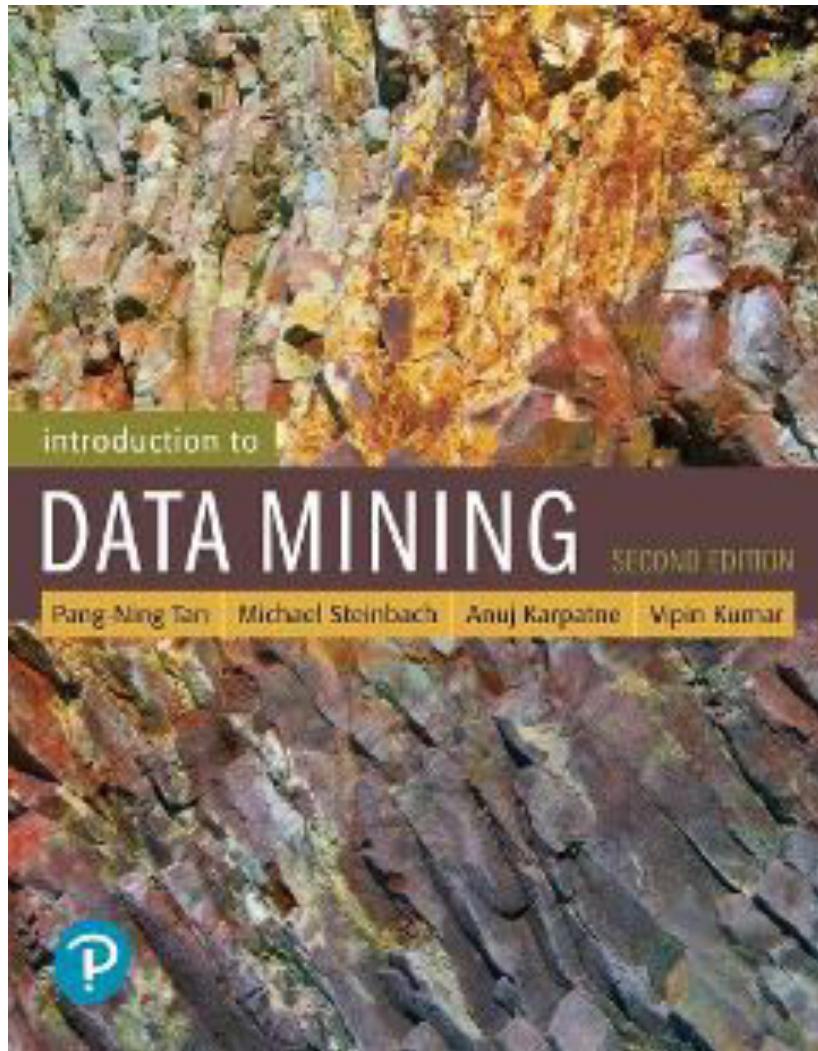


Two Sine Waves + Noise



Frequency

CSCE 5380/4380 – Data Mining



Chapter Three: **Classification: Basic Concepts and Techniques**

Outline

- **Classification: Definition**
- **Example of a Decision Tree**
- **Decision Tree Induction**
- **Decision Tree Based Classification**

Classification: Definition

- Given a collection of records (training set)
 - Each record is characterized by a tuple (x,y) , where x is the attribute set and y is the class label
 - ◆ x : attribute, predictor, independent variable, input
 - ◆ y : class, response, dependent variable, output
- Task:
 - Learn a model that maps each attribute set x into one of the predefined class labels y

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from x-rays or MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

General Approach for Building Classification Model

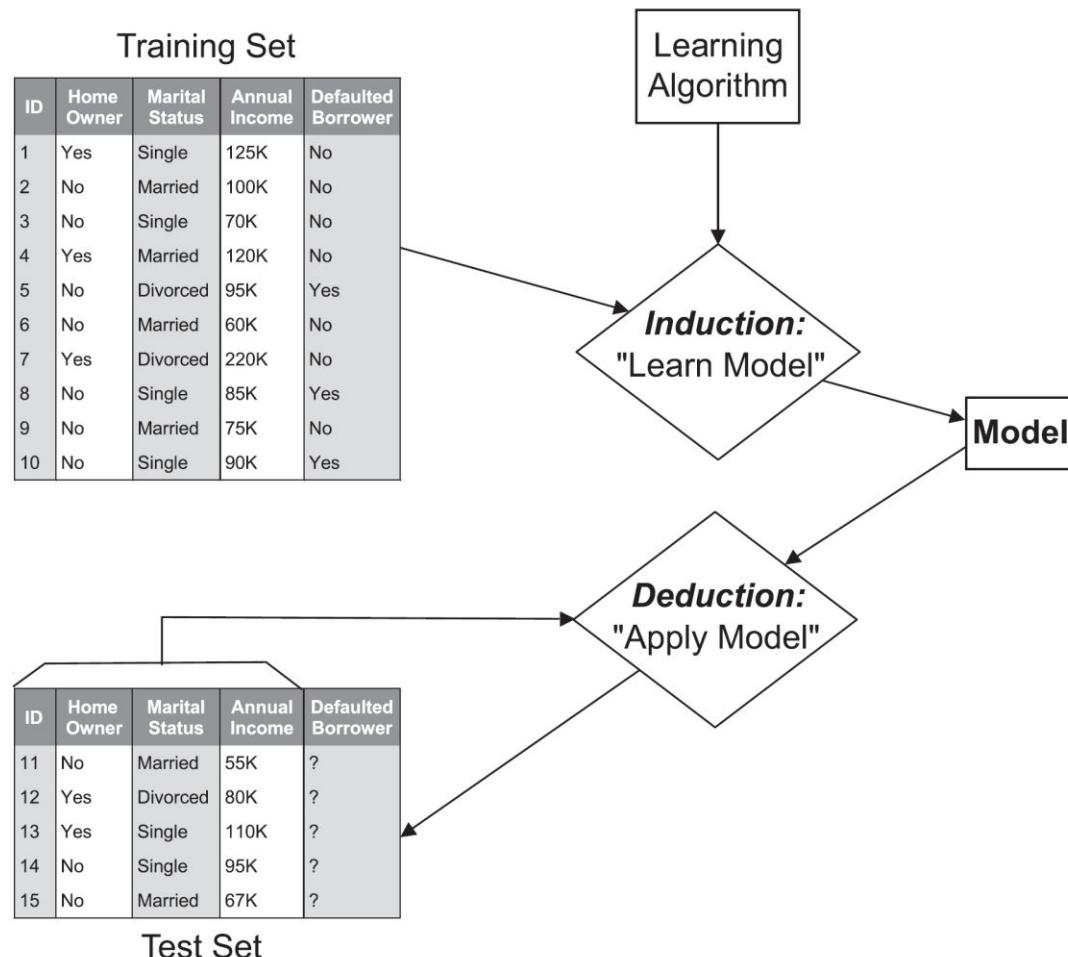


Figure 3.3. General framework for building a classification model.

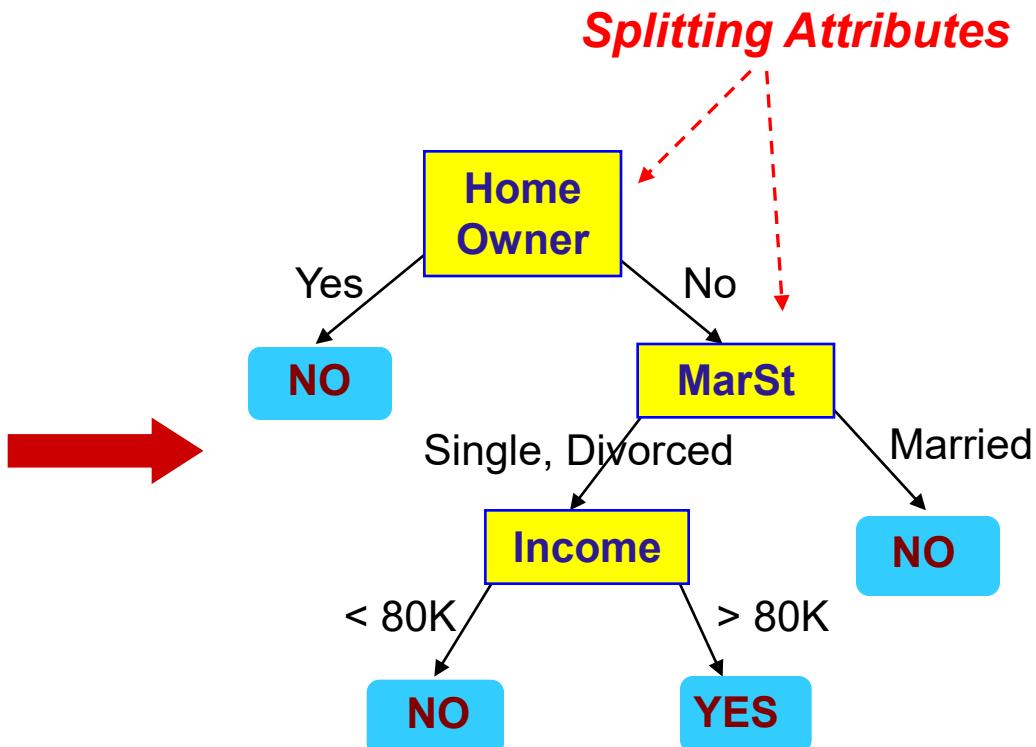
Classification Techniques

- Base Classifiers
 - Decision Tree based Methods
 - Rule-based Methods
 - Nearest-neighbor
 - Naïve Bayes and Bayesian Belief Networks
 - Support Vector Machines
 - Neural Networks, Deep Neural Nets

- Ensemble Classifiers
 - Boosting, Bagging, Random Forests

Example of a Decision Tree

ID	Training Data				class
	Home Owner	Marital Status	Annual Income	Defaulted Borrower	
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	

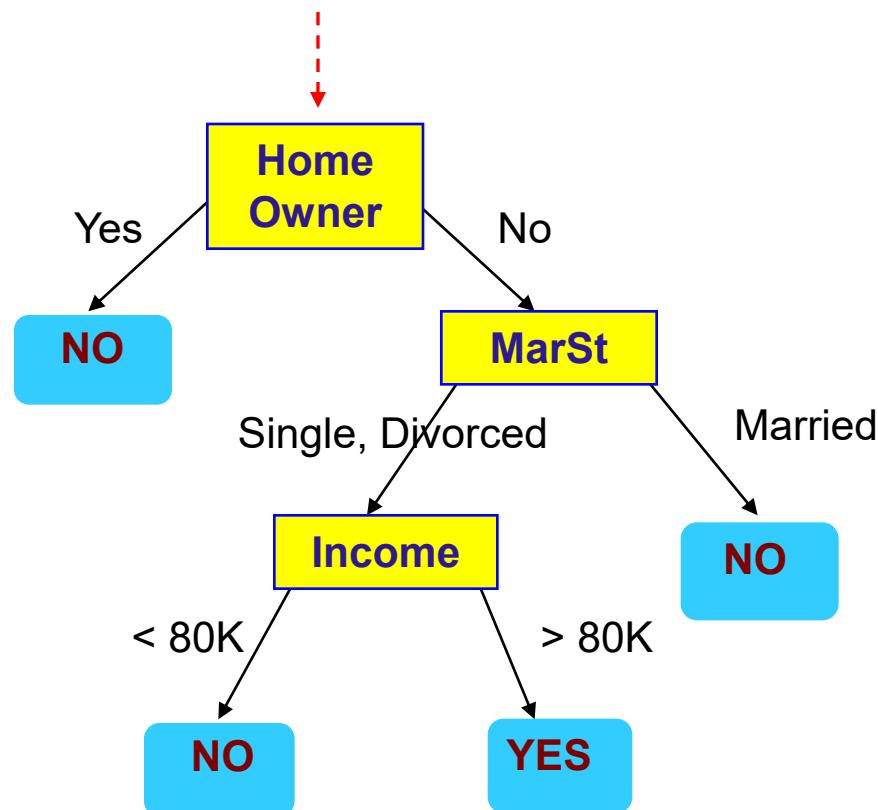


Training Data

Model: Decision Tree

Apply Model to Test Data

Start from the root of tree.



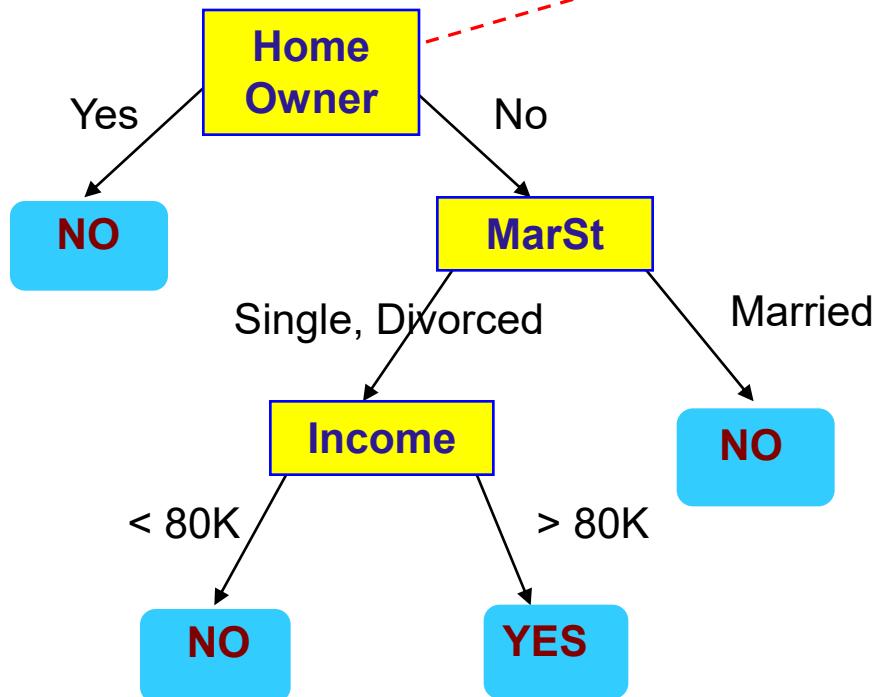
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data

Test Data

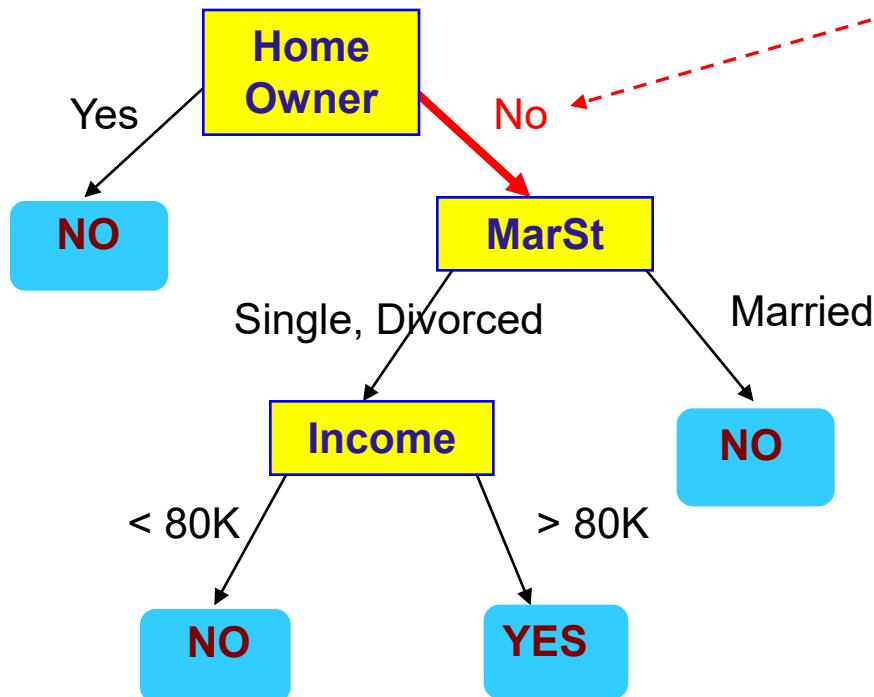
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

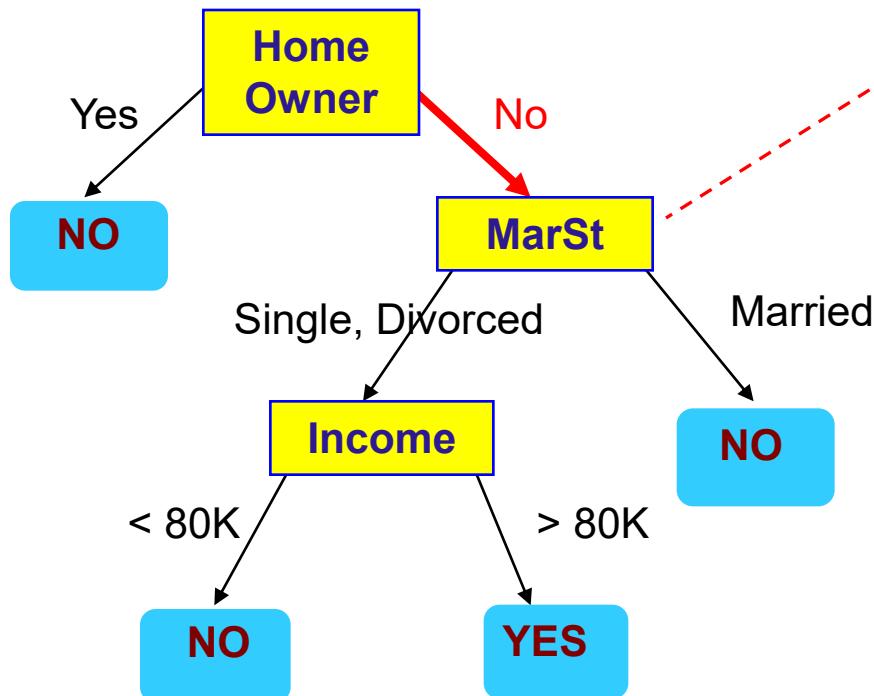
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

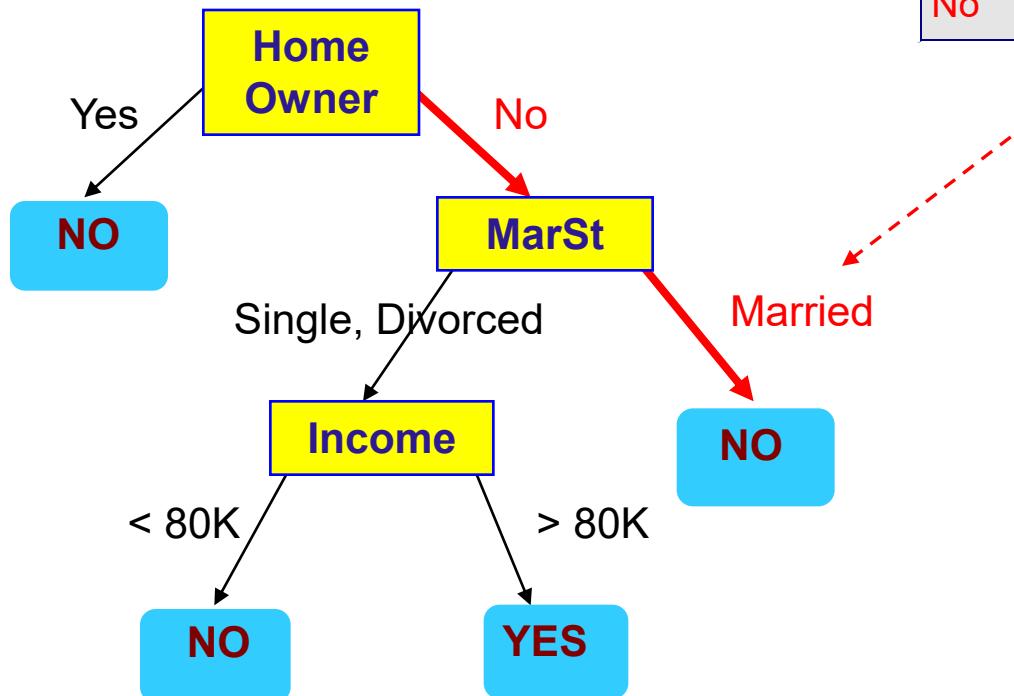
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

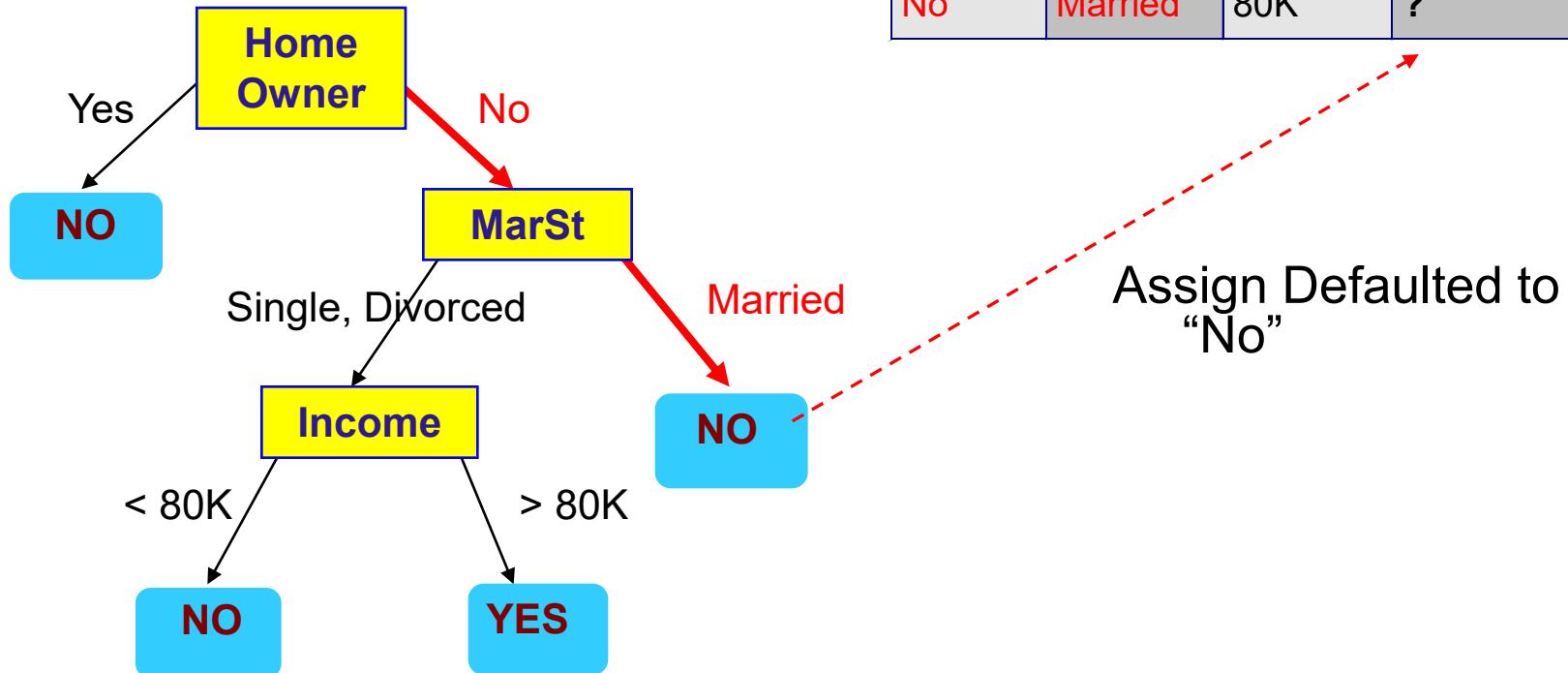
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

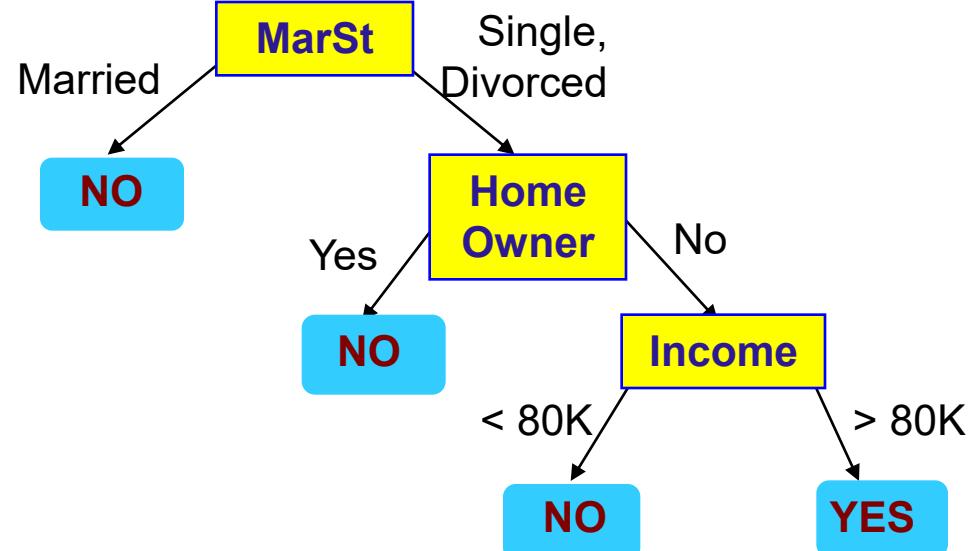
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Another Example of Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower	
				categorical	categorical
1	Yes	Single	125K	No	continuous
2	No	Married	100K	No	class
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!

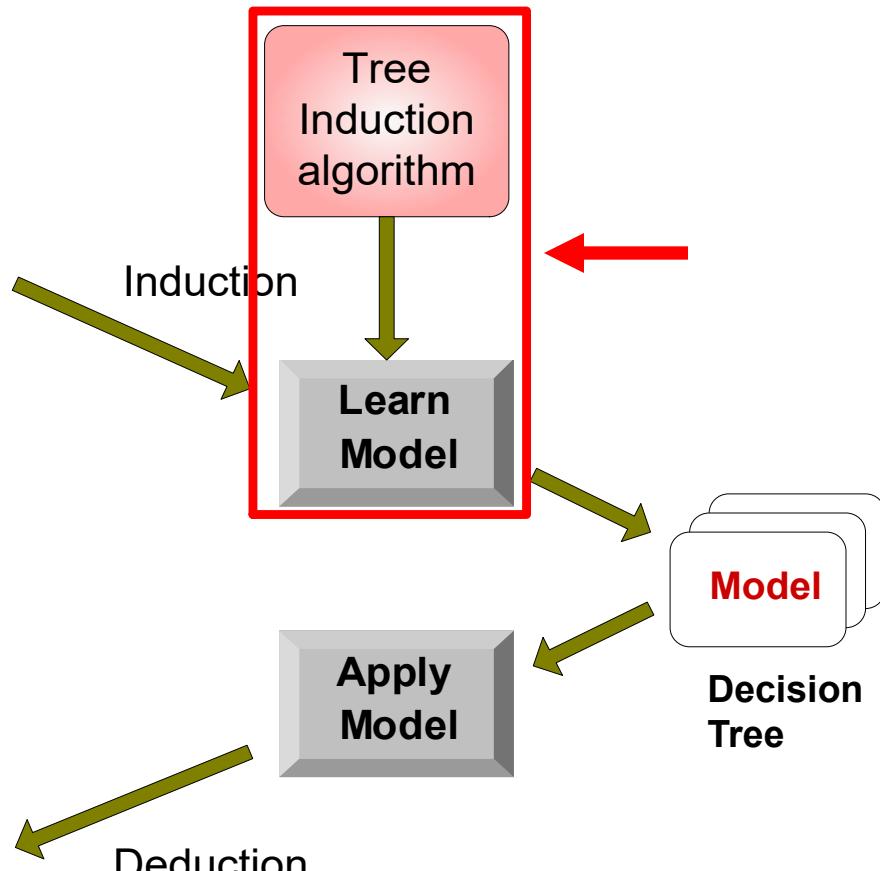
Decision Tree Classification Task

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



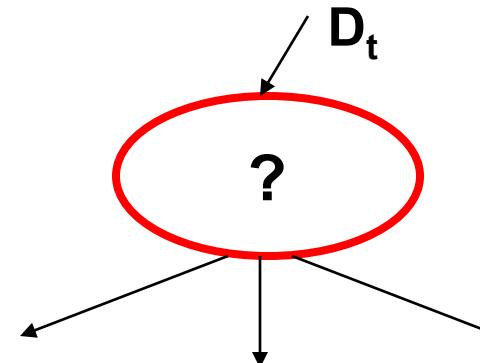
Decision Tree Induction

- Many Algorithms:
 - Hunt's Algorithm (one of the earliest)
 - CART
 - ID3, C4.5
 - SLIQ, SPRINT

General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- 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 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.

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Defaulted = No

(7,3)

(a)

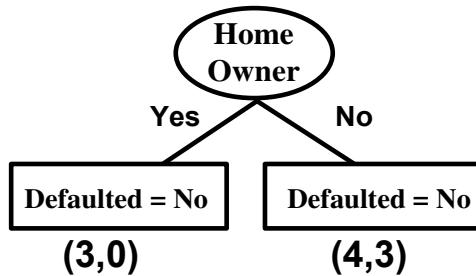
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm

Defaulted = No

(7,3)

(a)



(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Defaulted = No

(7,3)

(a)

Home
Owner

Yes

No

Defaulted = No

(3,0)

Defaulted = No

(4,3)

(b)

Home
Owner

Yes

No

Defaulted = No

(3,0) Single,
Divorced

Marital
Status

Married

Defaulted = Yes

(1,3)

Defaulted = No

(3,0)

(c)

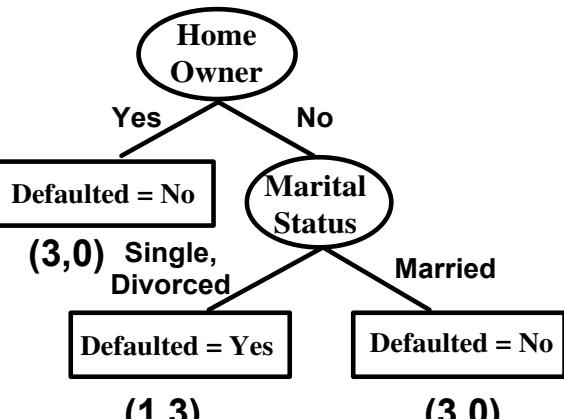
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm

Defaulted = No

(7,3)

(a)



Home Owner

Yes No

Defaulted = No

Defaulted = No

(3,0)

(4,3)

(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Defaulted = No
(3,0) Single, Divorced

Marital Status

Married

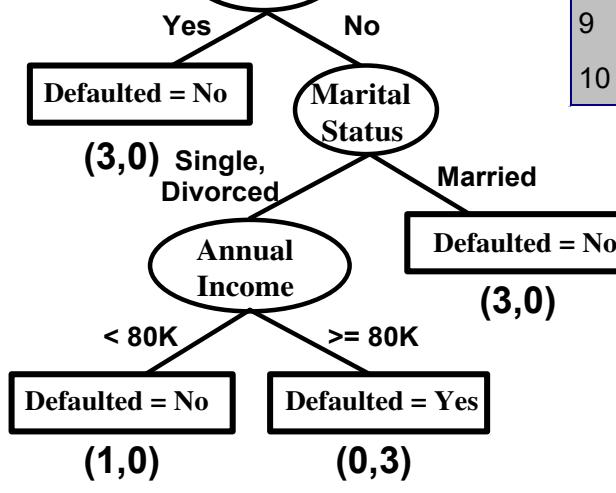
Defaulted = Yes

Defaulted = No

(1,3)

(3,0)

(c)



(d)

Design Issues of Decision Tree Induction

- How should training records be split?
 - Method for expressing test condition
 - ◆ depending on attribute types
 - Measure for evaluating the goodness of a test condition

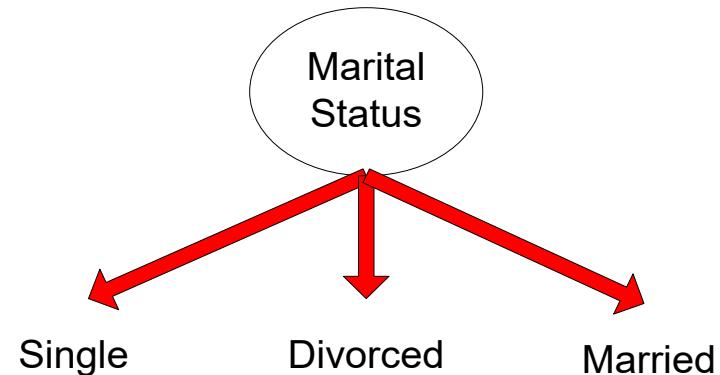
- How should the splitting procedure stop?
 - Stop splitting if all the records belong to the same class or have identical attribute values
 - Early termination

Methods for Expressing Test Conditions

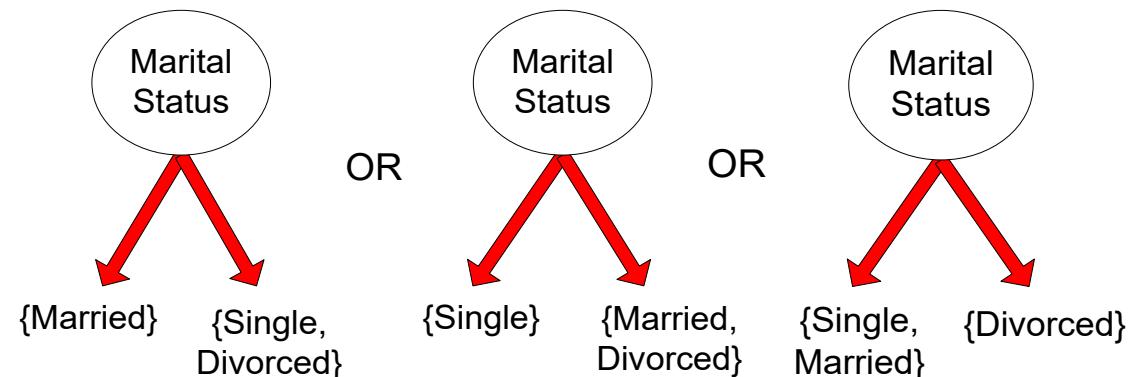
- Depends on attribute types
 - Binary
 - Nominal
 - Ordinal
 - Continuous

Test Condition for Nominal Attributes

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

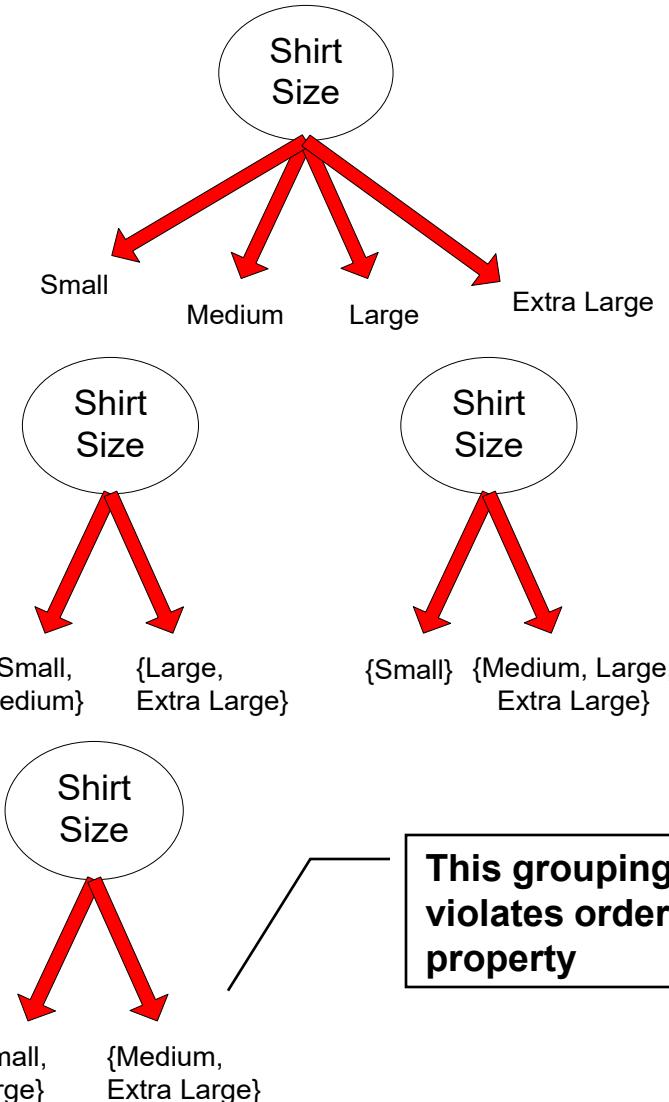


- Binary split:
 - Divides values into two subsets

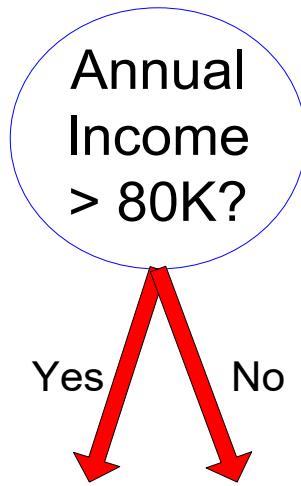


Test Condition for Ordinal Attributes

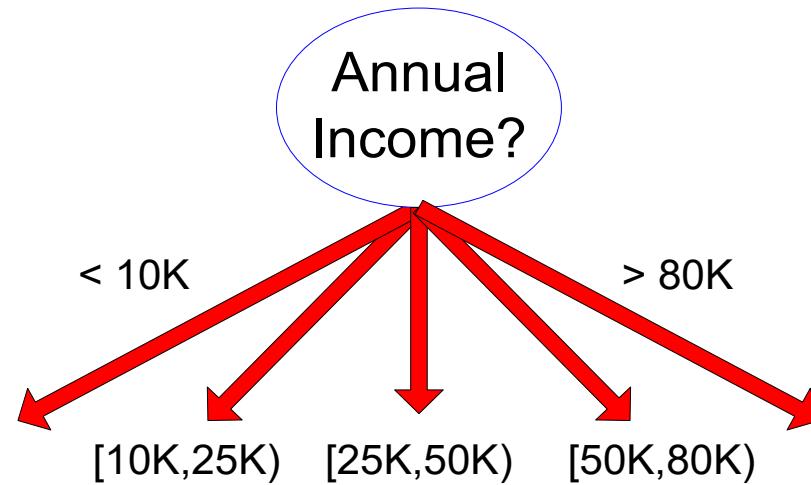
- Multi-way split:
 - Use as many partitions as distinct values
- Binary split:
 - Divides values into two subsets
 - Preserve order property among attribute values



Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

Splitting Based on Continuous Attributes

- Different ways of handling
 - **Discretization** to form an ordinal categorical attribute

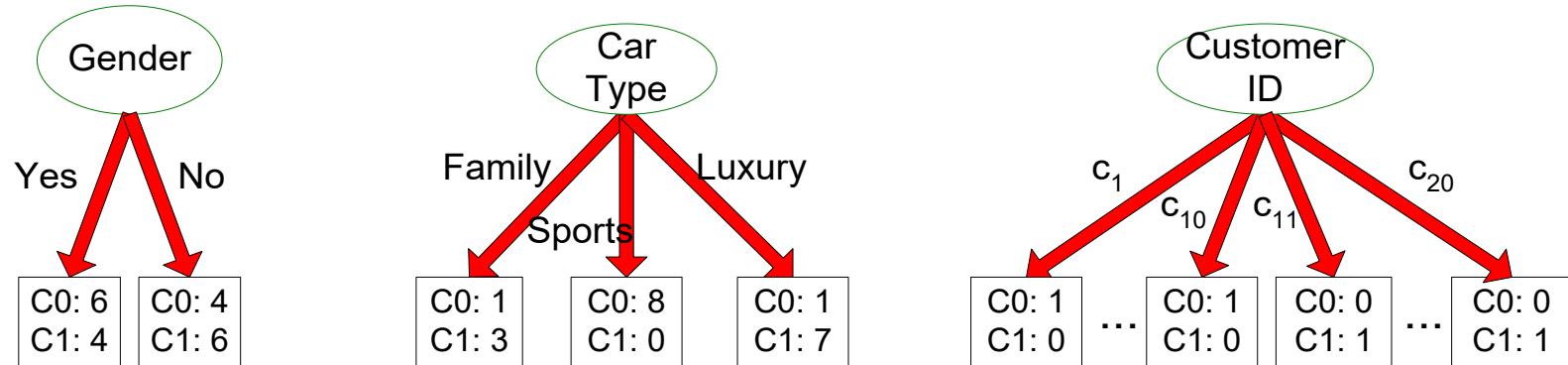
Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

 - ◆ Static – discretize once at the beginning
 - ◆ Dynamic – repeat at each node
 - **Binary Decision:** $(A < v)$ or $(A \geq v)$
 - ◆ consider all possible splits and finds the best cut
 - ◆ can be more compute intensive

How to determine the Best Split

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

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1



Which test condition is the best?

How to determine the Best Split

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

C0: 5
C1: 5

High degree of impurity

C0: 9
C1: 1

Low degree of impurity

Measures of Node Impurity

- Gini Index

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Entropy

$$Entropy = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

- Misclassification error

$$Classification\ error = 1 - \max[p_i(t)]$$

Finding the Best Split

1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting
 - Compute impurity measure of each child node
 - M is the weighted impurity of child nodes
3. Choose the attribute test condition that produces the highest gain

$$\text{Gain} = P - M$$

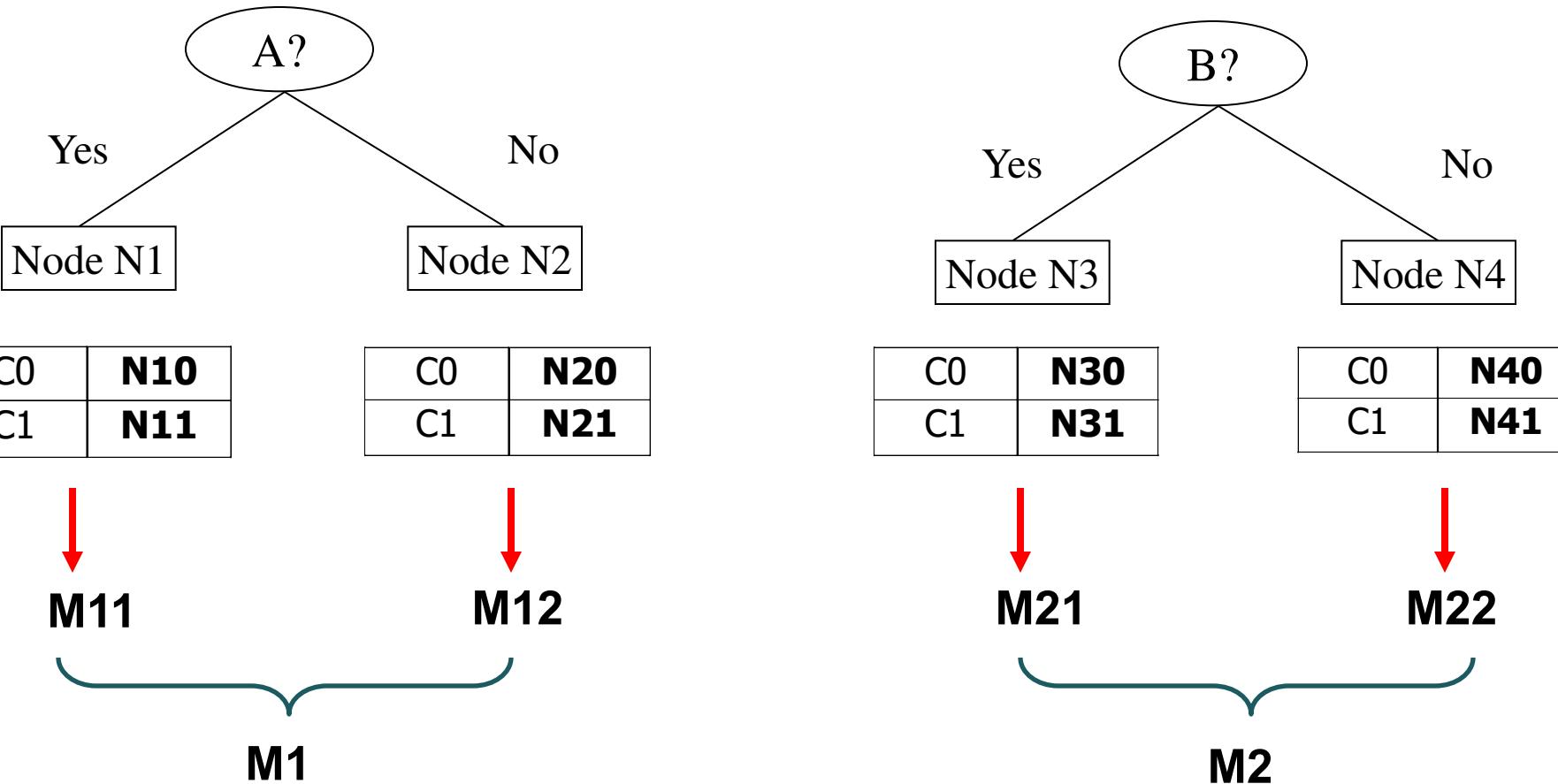
or equivalently, lowest impurity measure after splitting (M)

Finding the Best Split

Before Splitting:

C0	N00
C1	N01

→ P



$$\text{Gain} = P - M1 \quad \text{vs} \quad P - M2$$

Measure of Impurity: GINI

- Gini Index for a given node t

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
- Minimum of 0 when all records belong to one class, implying the most beneficial situation for classification
- Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

Measure of Impurity: GINI

- Gini Index for a given node t :

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

- For 2-class problem ($p, 1 - p$):
 - ◆ $GINI = 1 - p^2 - (1 - p)^2 = 2p(1-p)$

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Computing Gini Index of a Single Node

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1	0
C2	6

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

$$\text{Gini} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

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

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	2
C2	4

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

$$\text{Gini} = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

Computing Gini Index for a Collection of Nodes

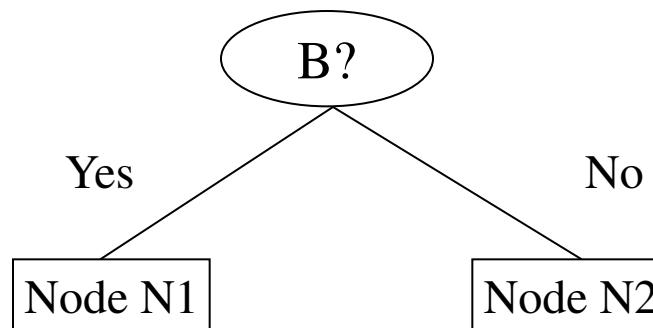
- When a node p is split into k partitions (children)

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i ,
 n = number of records at parent node p .

Binary Attributes: Computing GINI Index

- Splits into two partitions (child nodes)
- Effect of Weighing partitions:
 - Larger and purer partitions are sought



Gini(N1)

$$\begin{aligned} &= 1 - (5/6)^2 - (1/6)^2 \\ &= 0.278 \end{aligned}$$

Gini(N2)

$$\begin{aligned} &= 1 - (2/6)^2 - (4/6)^2 \\ &= 0.444 \end{aligned}$$

	N1	N2
C1	5	2
C2	1	4
Gini=0.361		

	Parent
C1	7
C2	5
Gini = 0.486	

Weighted Gini of N1 N2

$$\begin{aligned} &= 6/12 * 0.278 + \\ &\quad 6/12 * 0.444 \\ &= 0.361 \end{aligned}$$

$$\text{Gain} = 0.486 - 0.361 = 0.125$$

Categorical Attributes: Computing Gini Index

- 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	8	1
C2	3	0	7
Gini	0.163		

Two-way split

(find best partition of values)

CarType		
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

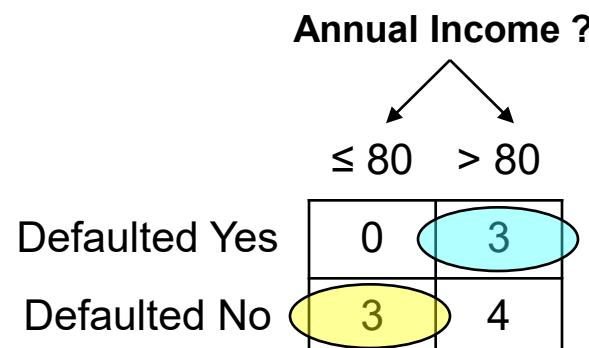
CarType		
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

Which of these is the best?

Continuous Attributes: Computing Gini Index

- Use Binary Decisions based on one value
- Several 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, $A \leq v$ and $A > v$
- Simple method to choose best v
 - For each v , scan the database to gather count matrix and compute its Gini index
 - Computationally Inefficient! Repetition of work.

ID	Home Owner	Marital Status	Annual Income	Defaulted
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: Computing Gini Index...

- For efficient computation: for each attribute,
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	
Annual Income											
Sorted Values	→	60	70	75	85	90	95	100	120	125	220

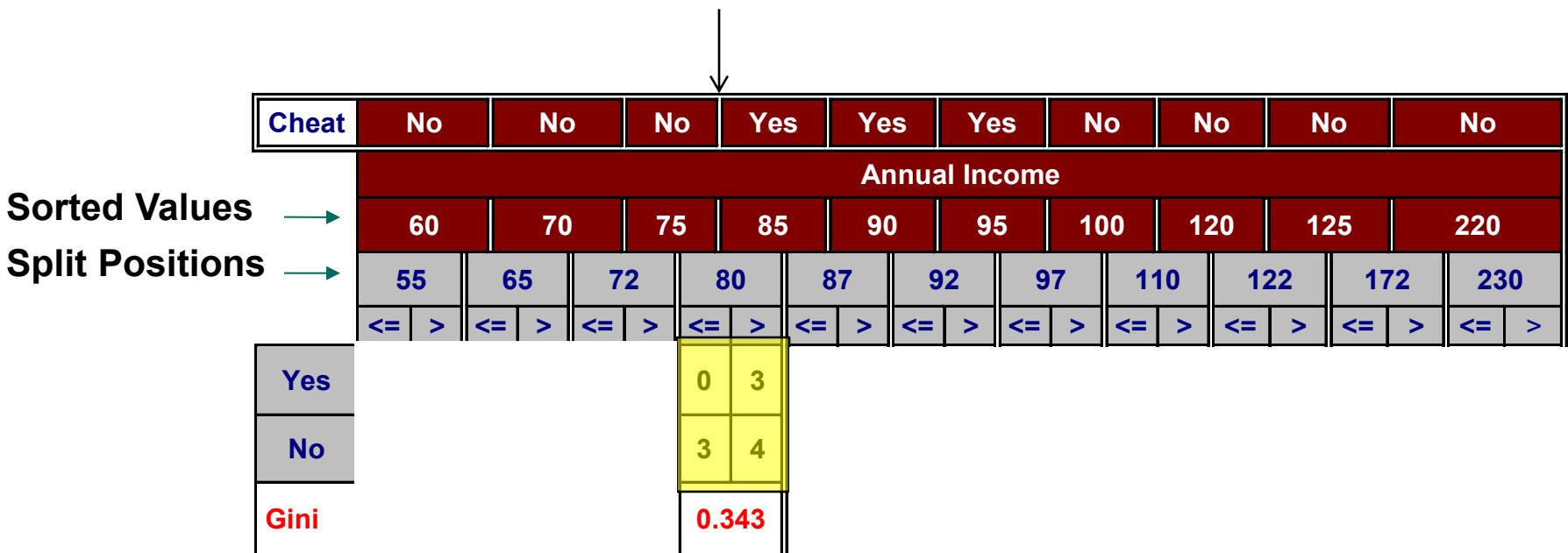
Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	
Annual Income											
Sorted Values	60	70	75	85	90	95	100	120	125	220	
Split Positions	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >

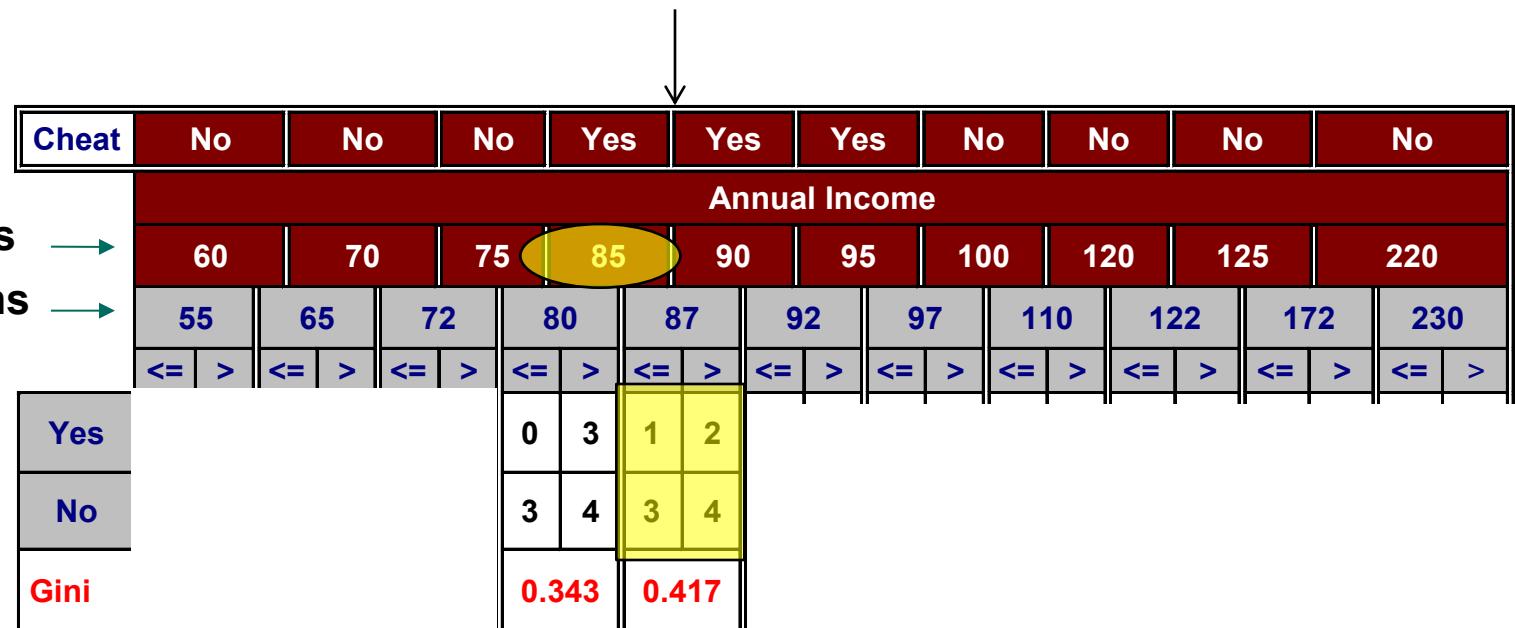
Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index



Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index



Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
 - Sort the attribute on values
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	
Annual Income											
Sorted Values →	60	70	75	85	90	95	100	120	125	220	
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	
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	3 4	4 3	5 2	6 1	
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	

Measure of Impurity: Entropy

- Entropy at a given node t

$$Entropy = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- ◆ Maximum of $\log_2 c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
 - ◆ Minimum of 0 when all records belong to one class, implying most beneficial situation for classification
-
- Entropy based computations are quite similar to the GINI index computations

Computing Entropy of a Single Node

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

C1	0
C2	6

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

$$\text{Entropy} = - 0 \log 0 - 1 \log 1 = - 0 - 0 = 0$$

C1	1
C2	5

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

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

C1	2
C2	4

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

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

Computing Information Gain After Splitting

- Information Gain:

$$Gain_{split} = Entropy(p) - \sum_{i=1}^k \frac{n_i}{n} Entropy(i)$$

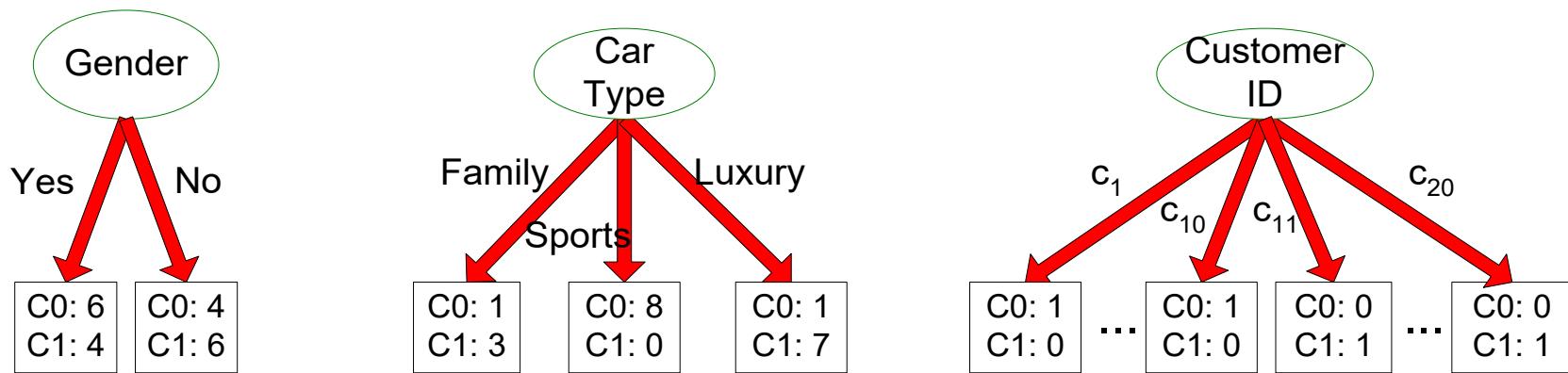
Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

- Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms
- Information gain is the mutual information between the class variable and the splitting variable

Problem with large number of partitions

- Node impurity measures tend to prefer splits that result in large number of partitions, each being small but pure



- Customer ID has highest information gain because entropy for all the children is zero

Gain Ratio

- Gain Ratio:

$$\text{Gain Ratio} = \frac{\text{Gain}_{\text{split}}}{\text{Split Info}}$$
$$\text{Split Info} = - \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

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

Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info}$$

$$Split\ Info = \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

$$\text{SplitINFO} = 1.52$$

CarType		
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

$$\text{SplitINFO} = 0.72$$

CarType		
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

$$\text{SplitINFO} = 0.97$$

Measure of Impurity: Classification Error

- Classification error at a node t

$$Error(t) = 1 - \max_i[p_i(t)]$$

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least interesting situation
- Minimum of 0 when all records belong to one class, implying the most interesting situation

Computing Error of a Single Node

$$\text{Error}(t) = 1 - \max_i[p_i(t)]$$

C1	0
C2	6

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

$$\text{Error} = 1 - \max(0, 1) = 1 - 1 = 0$$

C1	1
C2	5

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

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

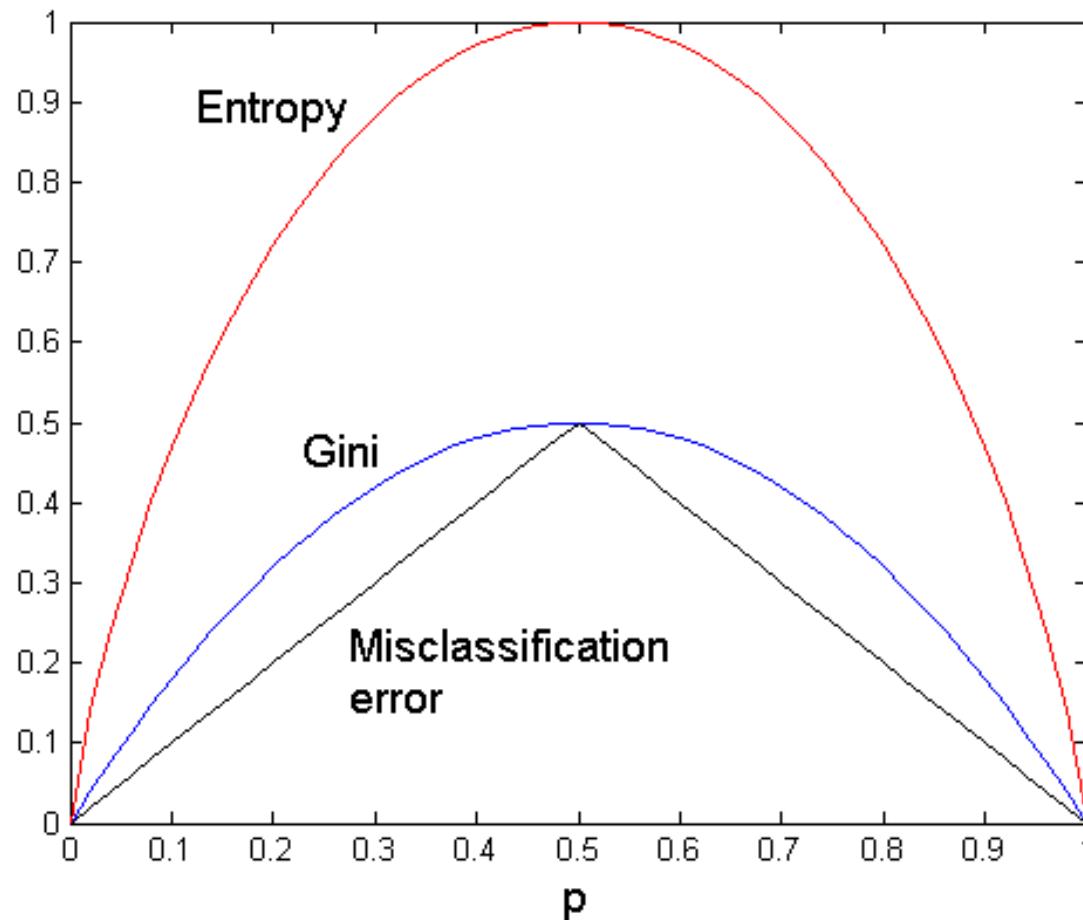
C1	2
C2	4

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

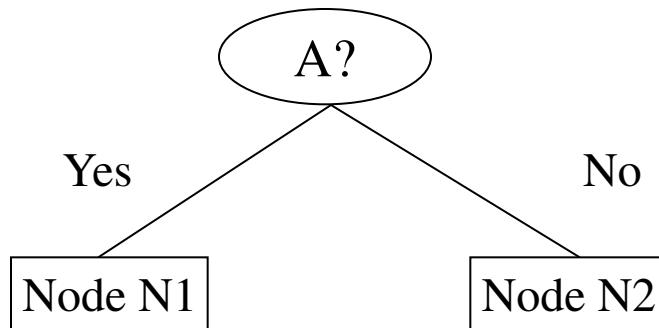
$$\text{Error} = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

Comparison among Impurity Measures

For a 2-class problem:



Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

Gini(N1)

$$\begin{aligned} &= 1 - (3/3)^2 - (0/3)^2 \\ &= 0 \end{aligned}$$

Gini(N2)

$$\begin{aligned} &= 1 - (4/7)^2 - (3/7)^2 \\ &= 0.489 \end{aligned}$$

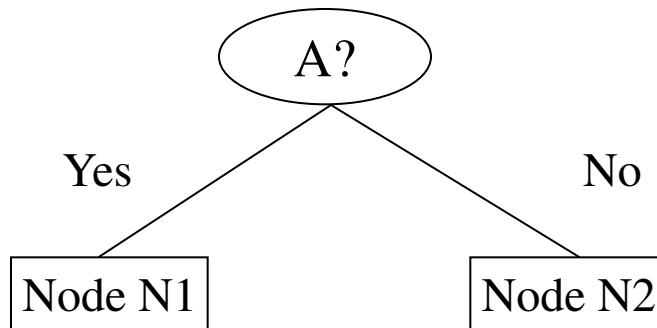
	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

Gini(Children)

$$\begin{aligned} &= 3/10 * 0 \\ &+ 7/10 * 0.489 \\ &= 0.342 \end{aligned}$$

**Gini improves but
error remains the
same!!**

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

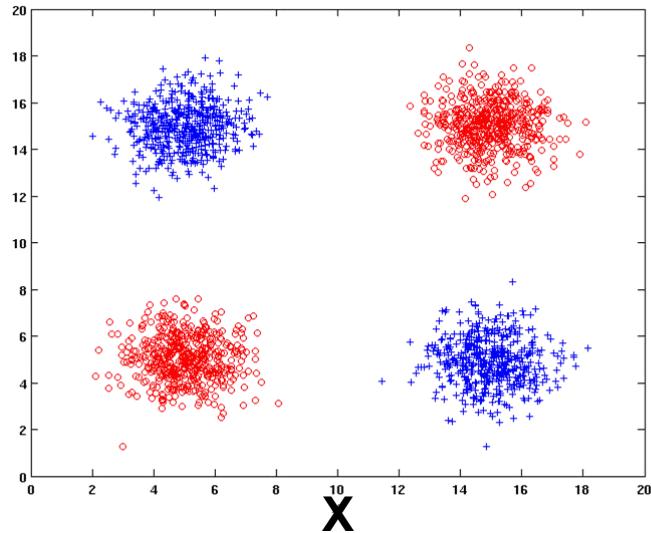
	N1	N2
C1	3	4
C2	1	2
Gini=0.416		

Misclassification error for all three cases = 0.3 !

Decision Tree Based Classification

- Advantages:
 - Relatively inexpensive to construct
 - Extremely fast at classifying unknown records
 - Easy to interpret for small-sized trees
 - Robust to noise (especially when methods to avoid overfitting are employed)
 - Can easily handle redundant attributes
 - Can easily handle irrelevant attributes (unless the attributes are interacting)
- Disadvantages: .
 - Due to the greedy nature of splitting criterion, interacting attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributed that are less discriminating.
 - Each decision boundary involves only a single attribute

Handling interactions

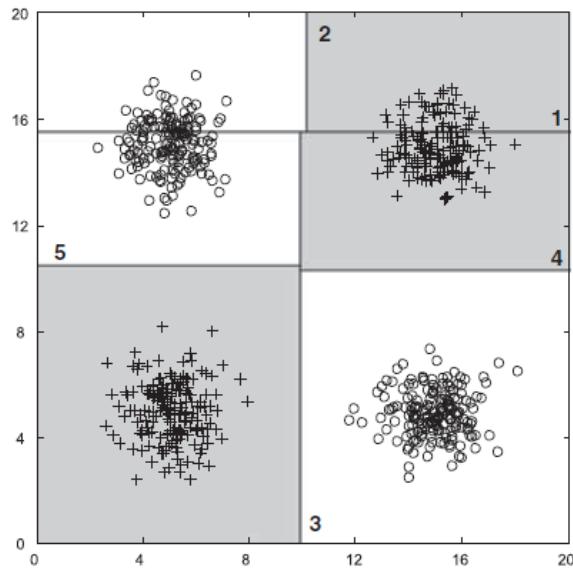


+ : 1000 instances

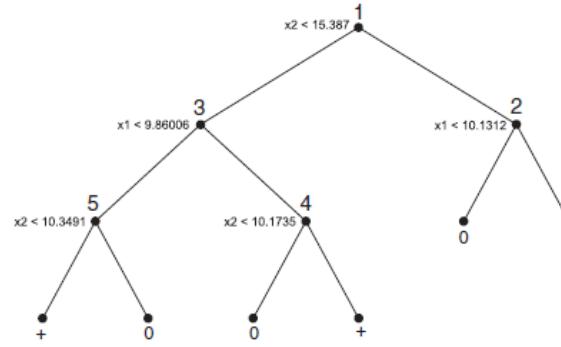
o : 1000 instances

Entropy (X) : 0.99
Entropy (Y) : 0.99

Handling interactions



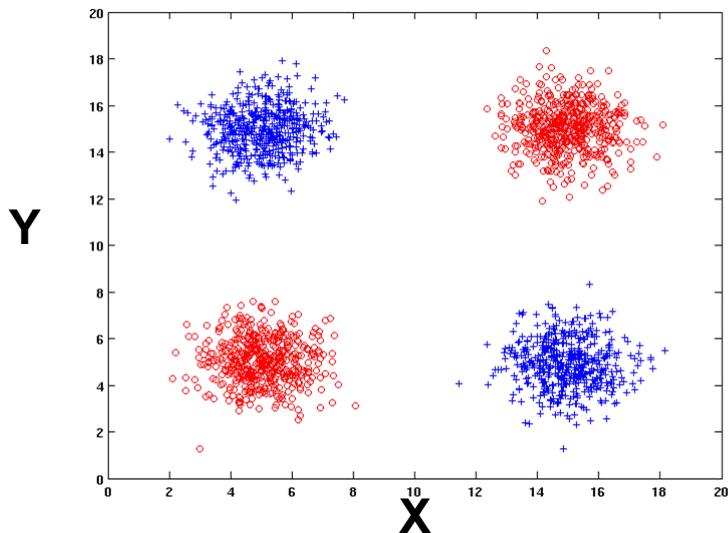
(a) Decision boundary for tree with 6 leaf nodes.



(b) Decision tree with 6 leaf nodes.

Figure 3.28. Decision tree with 6 leaf nodes using X and Y as attributes. Splits have been numbered from 1 to 5 in order of other occurrence in the tree.

Handling interactions given irrelevant attributes



+ : 1000 instances

o : 1000 instances

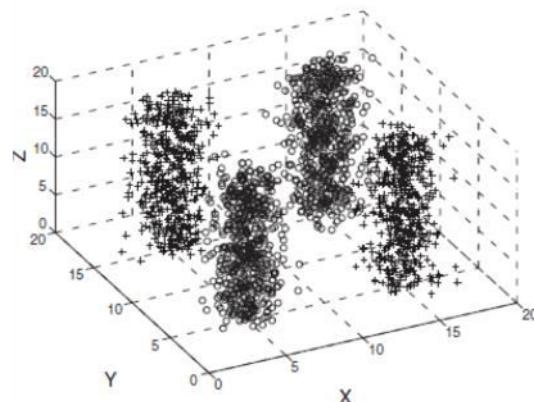
Adding Z as a noisy attribute generated from a uniform distribution

Entropy (X) : 0.99

Entropy (Y) : 0.99

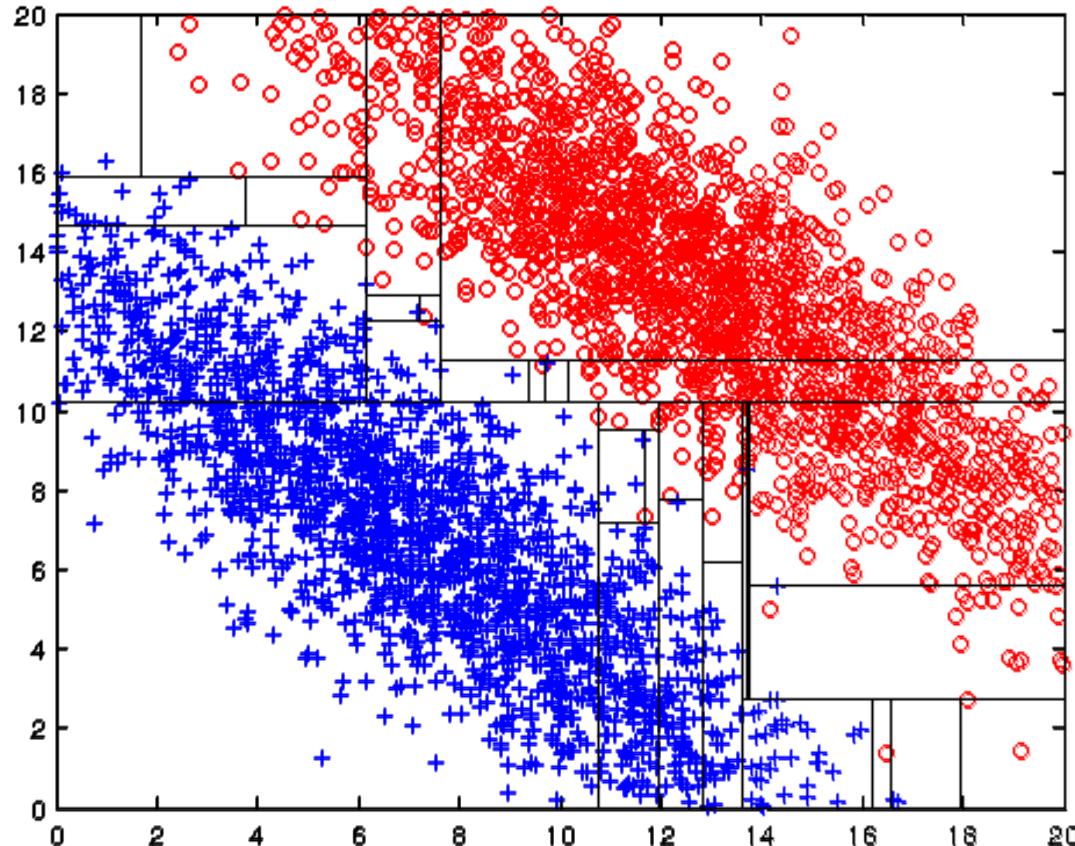
Entropy (Z) : 0.98

Attribute Z will be chosen for splitting!



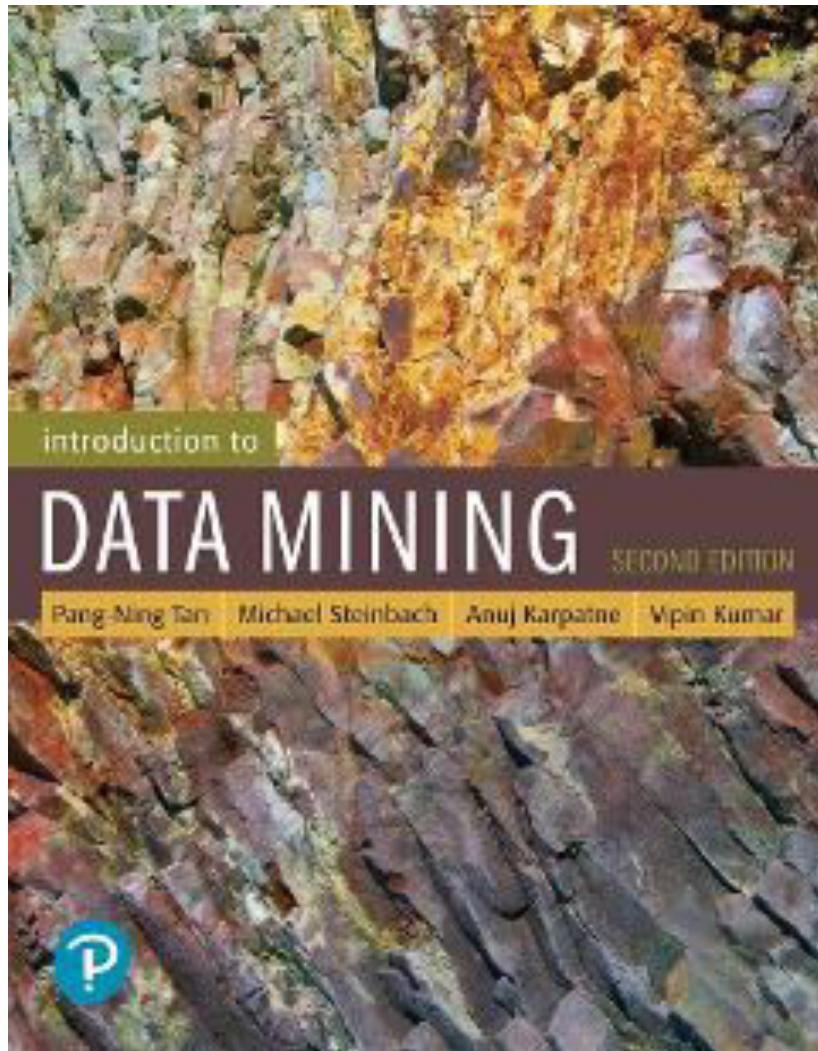
(a) Three-dimensional data with attributes X , Y , and Z .

Limitations of single attribute-based decision boundaries



Both **positive (+)** and **negative (o)** classes generated from skewed Gaussians with centers at (8,8) and (12,12) respectively.

CSCE 5380/4380 – Data Mining

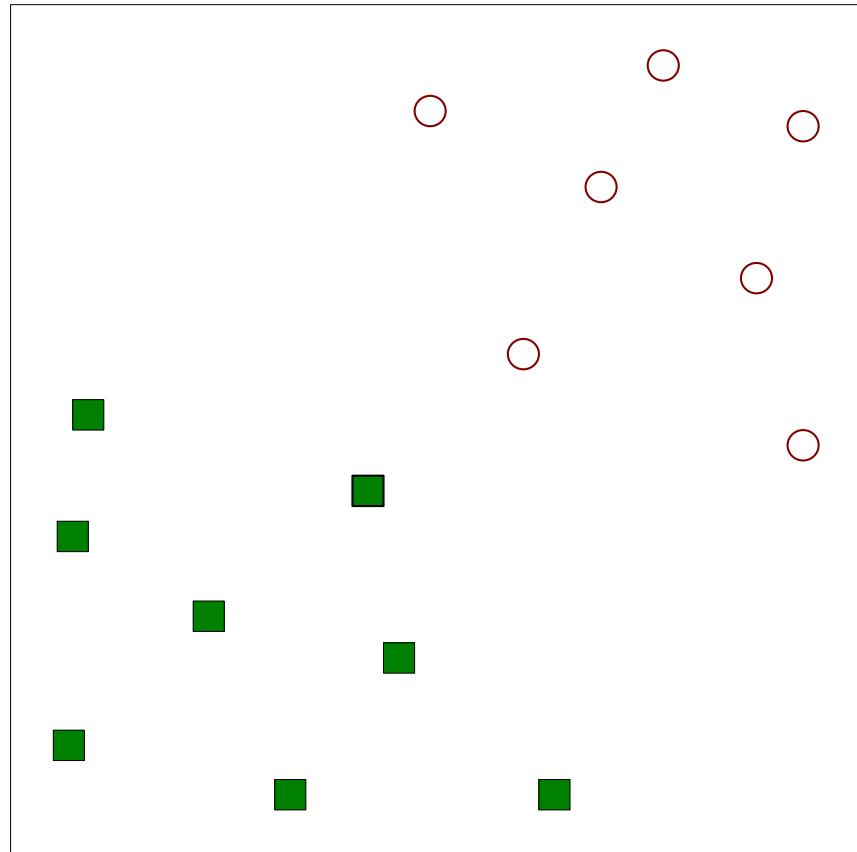


Chapter Four: **Support Vector Machines & Imbalanced Classes**

Outline

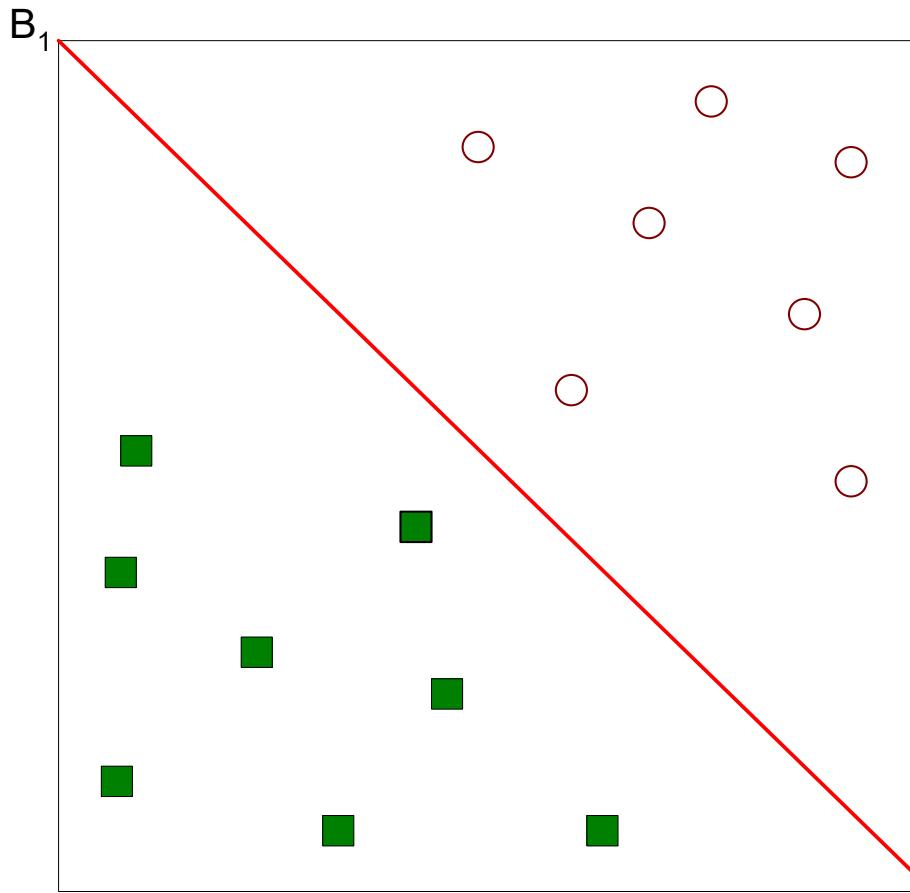
- **Support Vector Machines**
- **Nonlinear Support Vector Machines**
- **Characteristics of SVM**
- **Class Imbalance Problem**
- **Measures of Classification Performance & Imbalanced Classes**
- **ROC (Receiver Operating Characteristic)**
- **Building Classifiers with Imbalanced Training Set**

Support Vector Machines



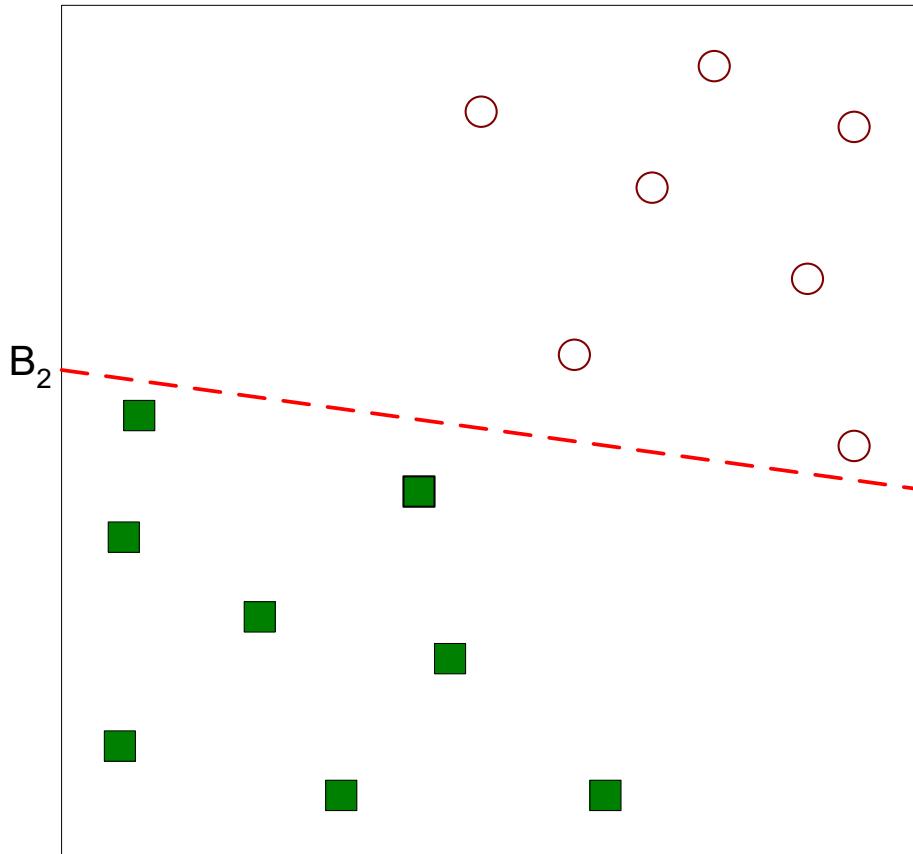
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



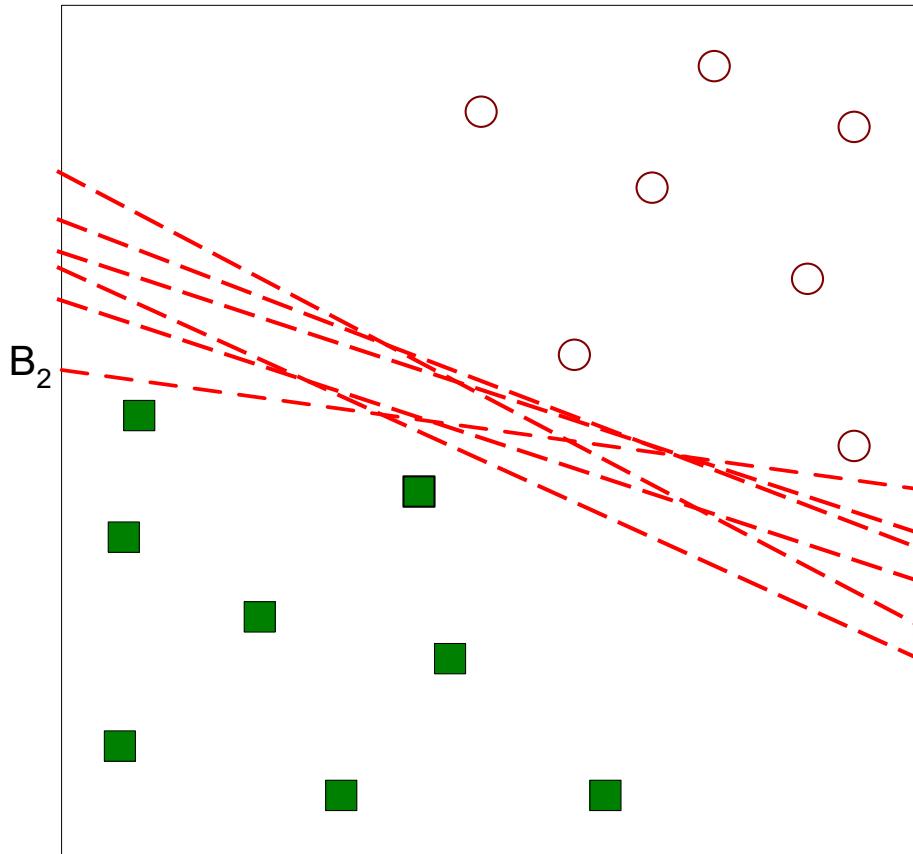
- One Possible Solution

Support Vector Machines



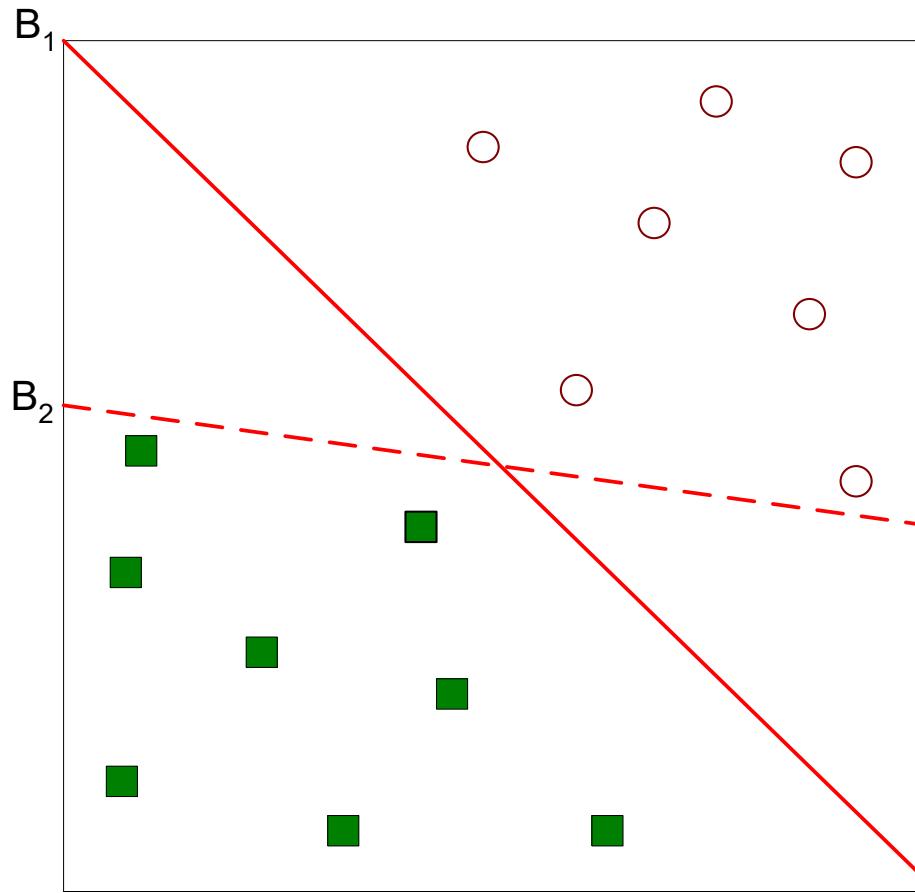
- Another possible solution

Support Vector Machines



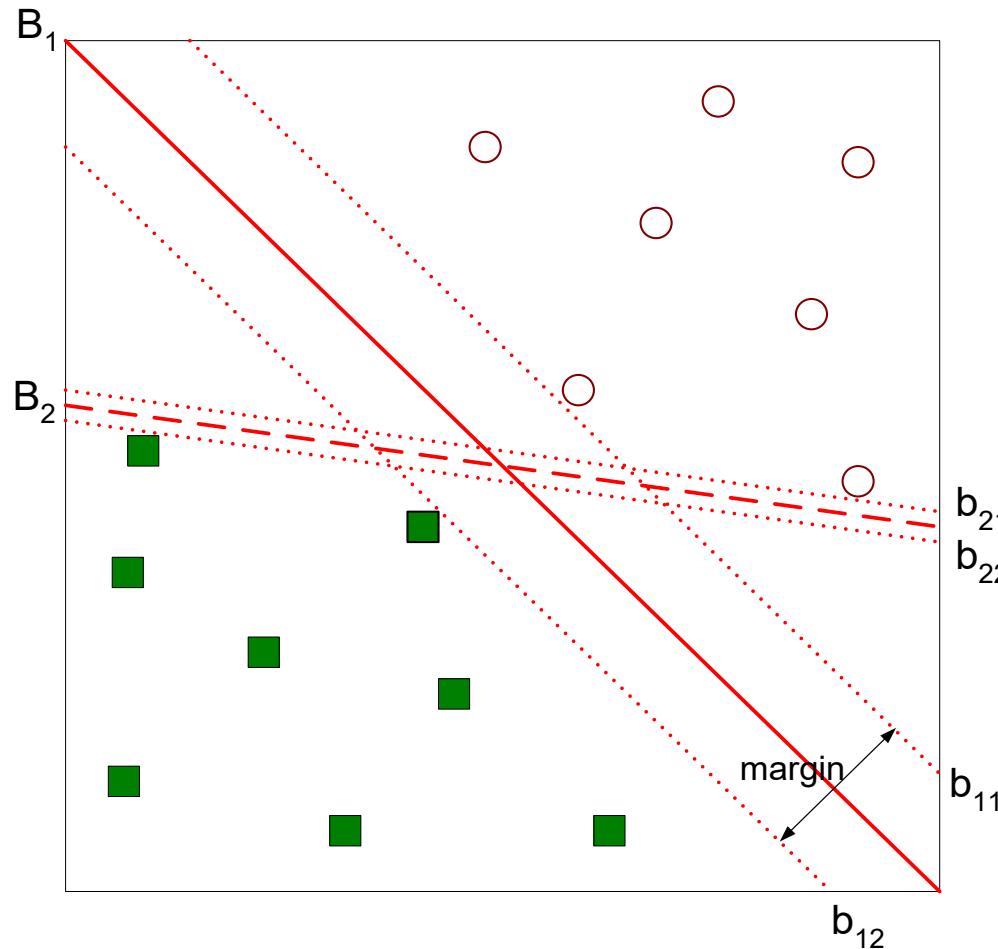
- Other possible solutions

Support Vector Machines



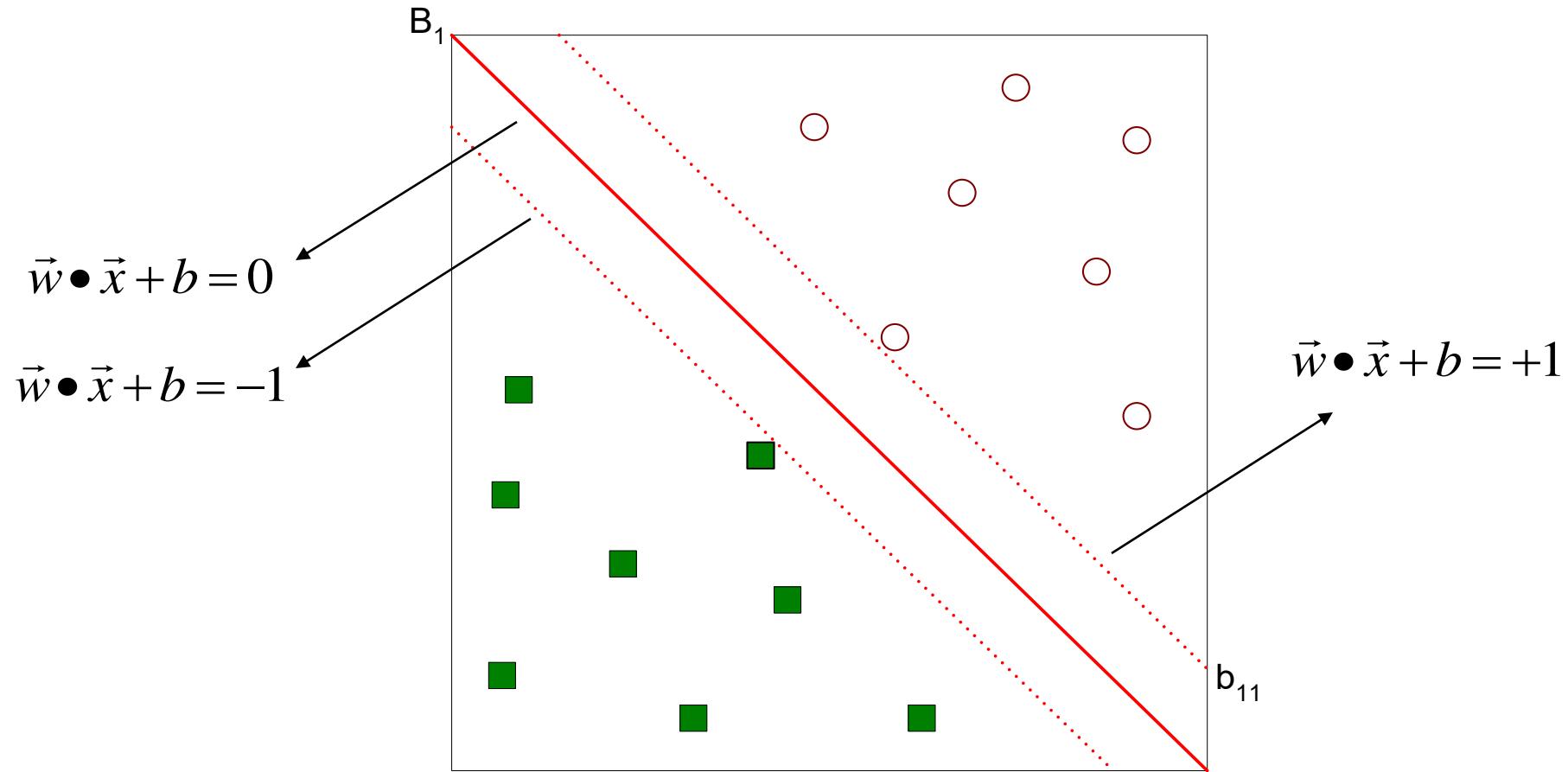
- Which one is better? B_1 or B_2 ?
- How do you define better?

Support Vector Machines



- Find hyperplane **maximizes** the margin => B1 is better than B2

Support Vector Machines



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|}$$

Linear SVM

- Linear model:

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

- Learning the model is equivalent to determining the values of \vec{w} and b
 - How to find \vec{w} and b from training data?

Learning Linear SVM

- Objective is to maximize: Margin = $\frac{2}{\|\vec{w}\|}$
 - Which is equivalent to minimizing: $L(\vec{w}) = \frac{\|\vec{w}\|^2}{2}$
 - Subject to the following constraints:

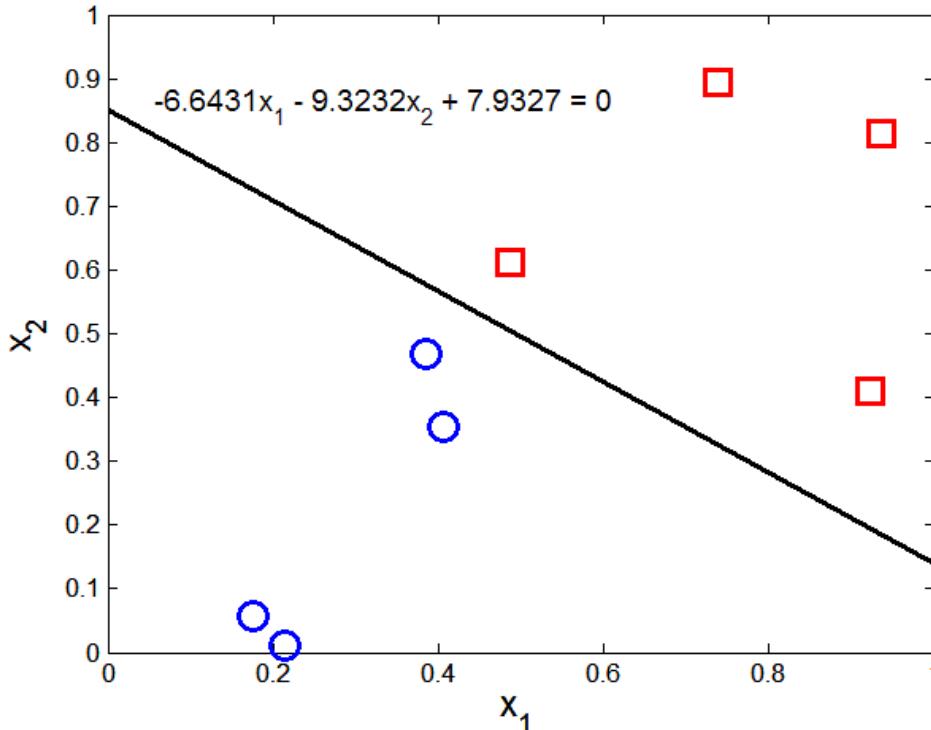
$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

or

$$y_i(w \bullet x_i + b) \geq 1, \quad i = 1, 2, \dots, N$$

- ◆ This is a constrained optimization problem
 - Solve it using Lagrange multiplier method

Example of Linear SVM



Support vectors

x1	x2	y	λ
0.3858	0.4687	1	65.5261
0.4871	0.611	-1	65.5261
0.9218	0.4103	-1	0
0.7382	0.8936	-1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	-1	0
0.2146	0.0099	1	0

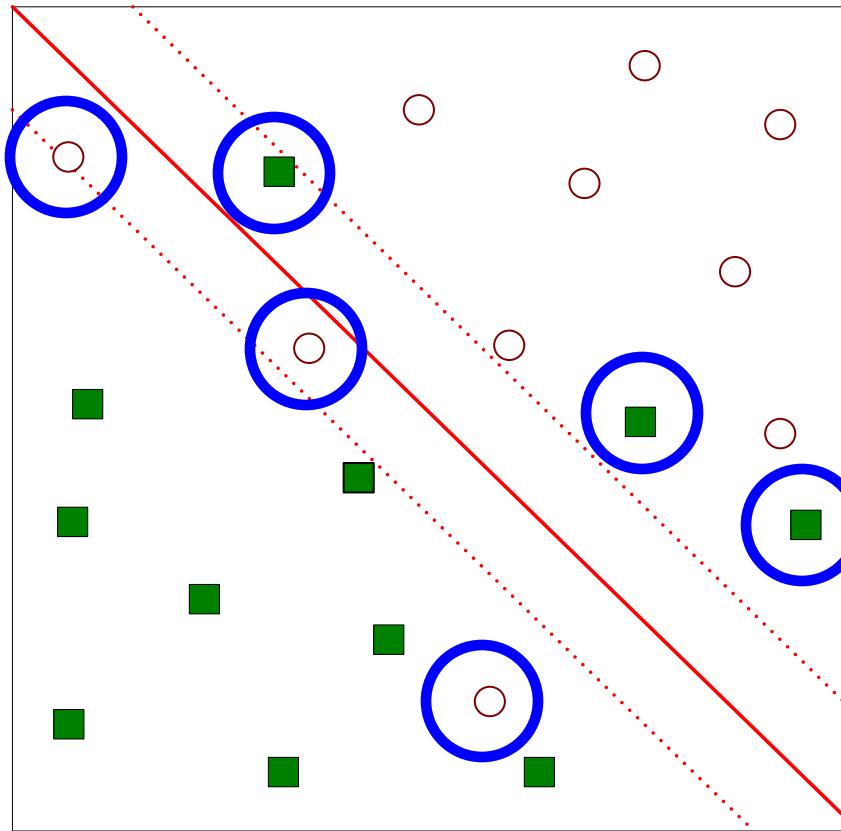
Learning Linear SVM

- Decision boundary depends only on support vectors
 - If you have data set with same support vectors, decision boundary will not change
 - How to classify using SVM once w and b are found? Given a test record, x_i

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?
 - Introduce slack variables

◆ Need to minimize:

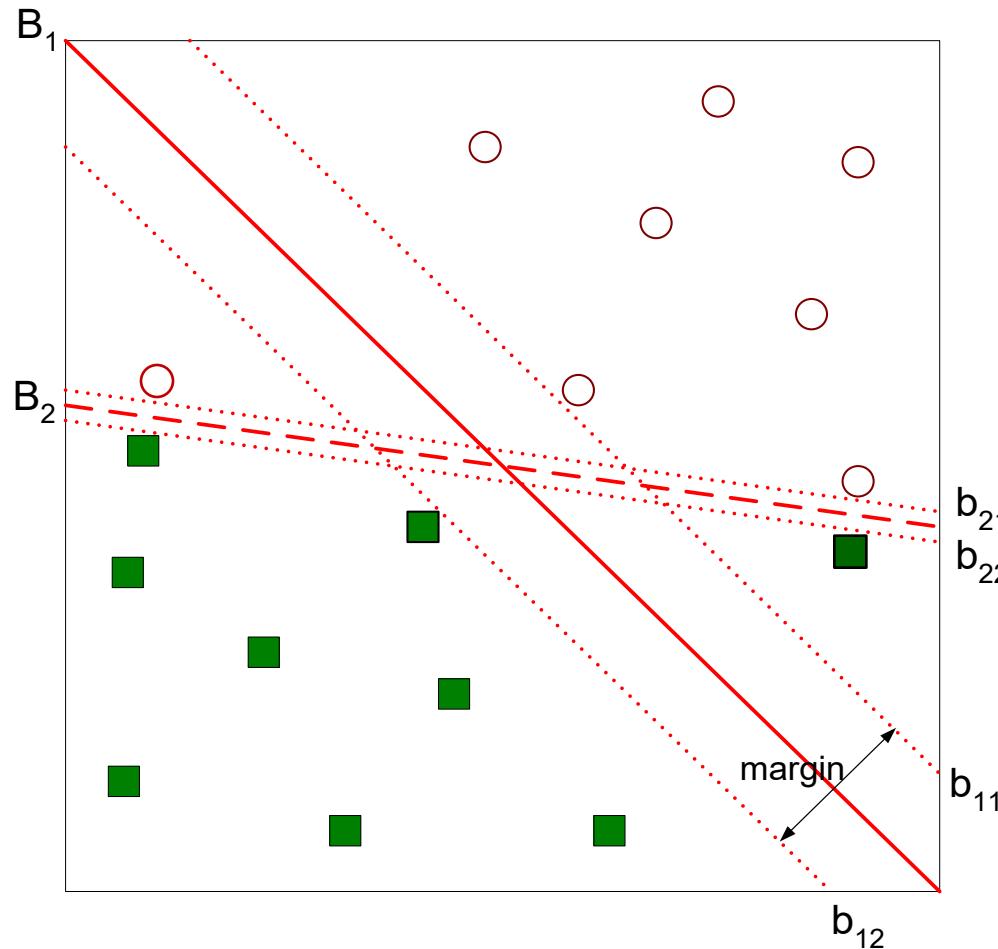
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C \left(\sum_{i=1}^N \xi_i^k \right)$$

◆ Subject to:

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

◆ If k is 1 or 2, this leads to similar objective function as linear SVM but with different constraints (see textbook)

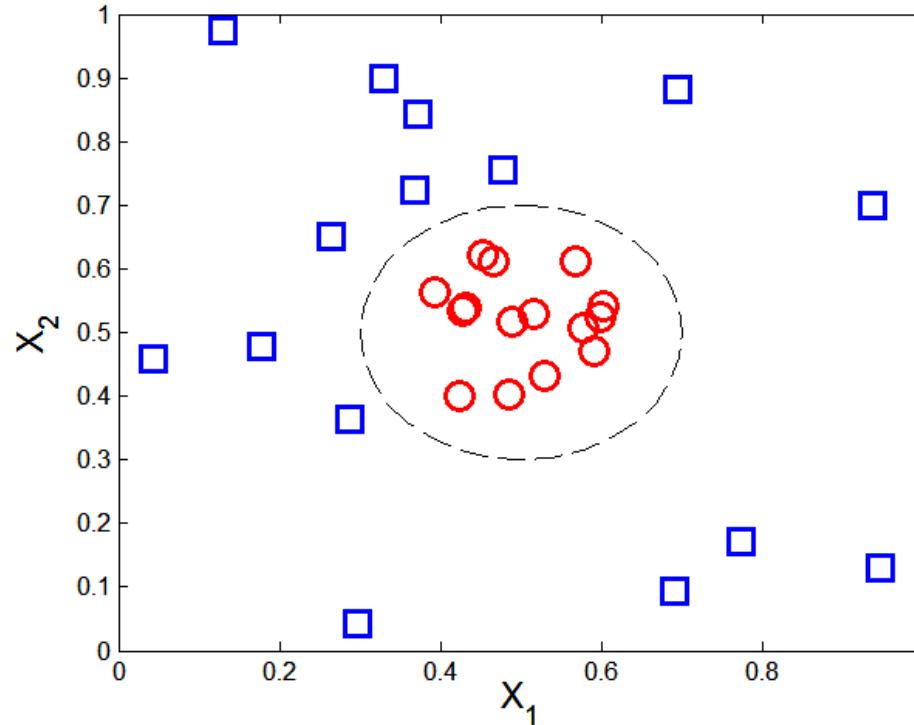
Support Vector Machines



- Find the hyperplane that optimizes both factors

Nonlinear Support Vector Machines

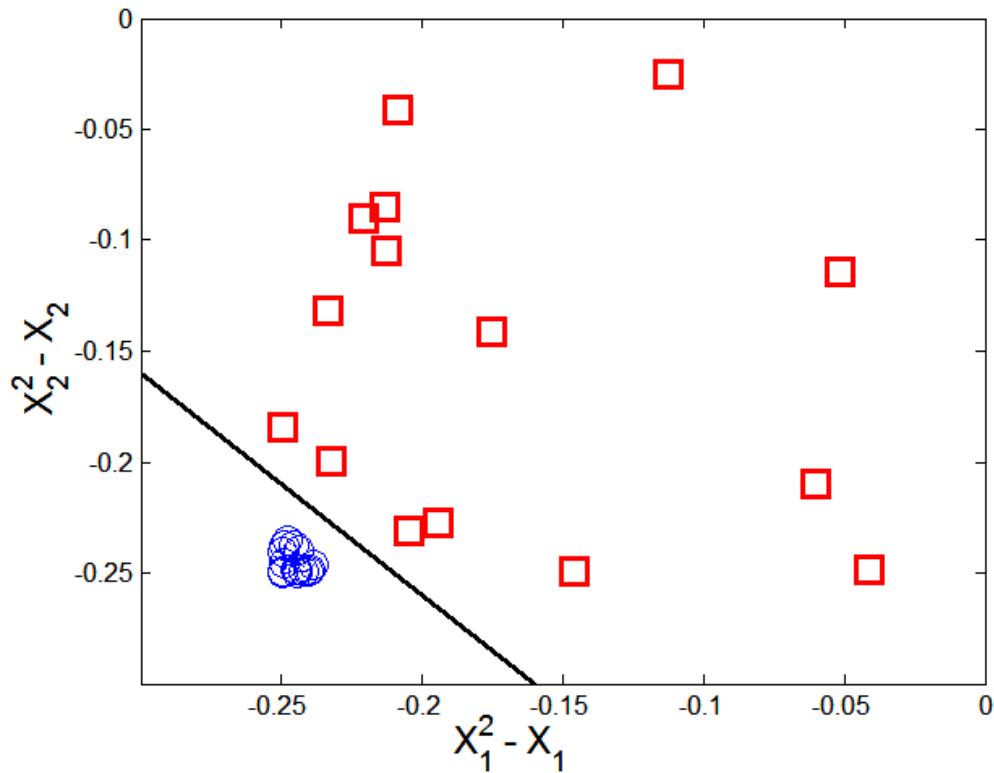
- What if decision boundary is not linear?



$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

Nonlinear Support Vector Machines

- Transform data into higher dimensional space



$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$w_4x_1^2 + w_3x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Decision boundary:

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

Learning Nonlinear SVM

- Optimization problem:

$$\min_w \frac{\|\mathbf{w}\|^2}{2}$$

subject to $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \forall \{(x_i, y_i)\}$

- Which leads to the same set of equations (but involve $\Phi(\mathbf{x})$ instead of \mathbf{x})

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \quad \mathbf{w} = \sum_i \lambda_i y_i \Phi(\mathbf{x}_i)$$
$$\lambda_i \{y_i (\sum_j \lambda_j y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) + b) - 1\} = 0,$$

$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \Phi(\mathbf{z}) + b) = sign(\sum_{i=1}^n \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b).$$

Learning NonLinear SVM

- Issues:
 - What type of mapping function Φ should be used?
 - How to do the computation in high dimensional space?
 - ◆ Most computations involve dot product $\Phi(x_i) \bullet \Phi(x_j)$
 - ◆ Curse of dimensionality?

Learning Nonlinear SVM

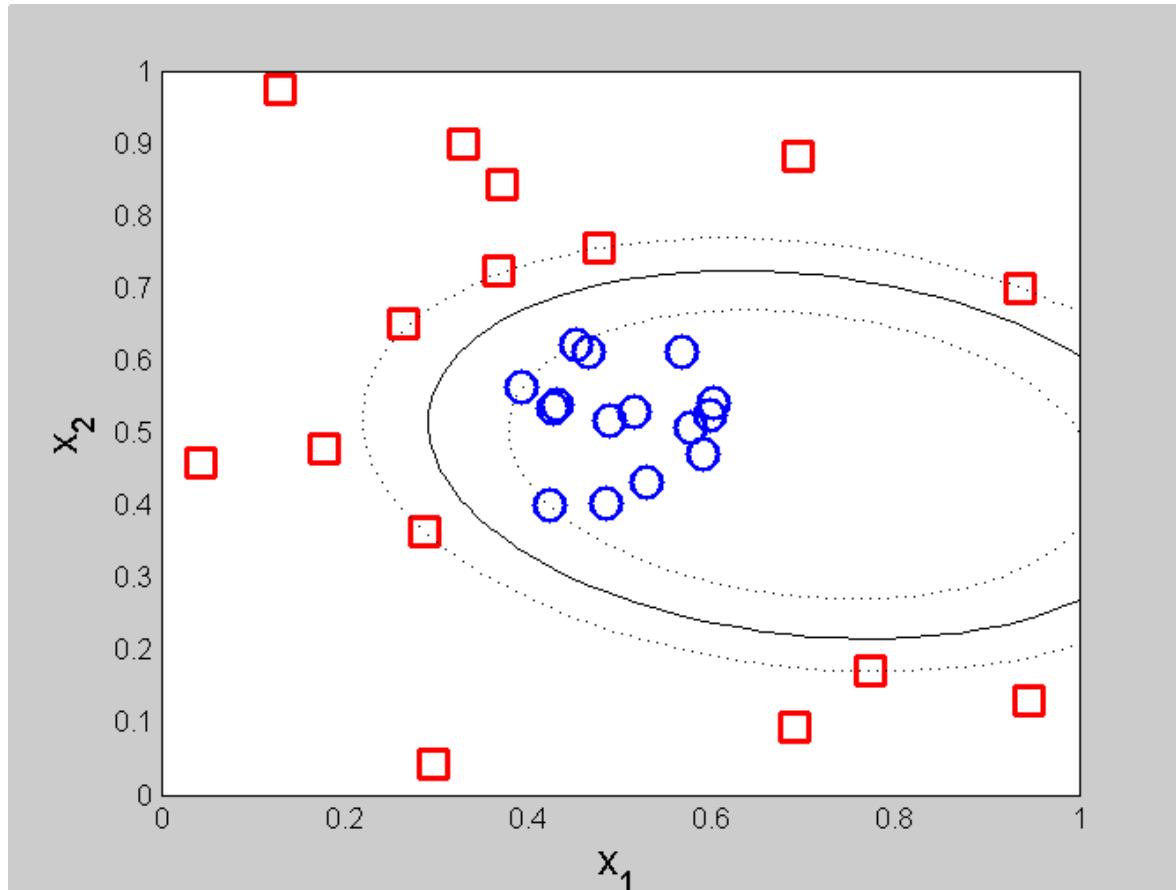
- Kernel Trick:
 - $\Phi(\mathbf{x}_i) \bullet \Phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$
 - $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function (expressed in terms of the coordinates in the original space)
 - ◆ Examples:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$

Example of Nonlinear SVM



SVM with polynomial
degree 2 kernel

Learning Nonlinear SVM

- Advantages of using kernel:
 - Don't have to know the mapping function Φ
 - Computing dot product $\Phi(x_i) \bullet \Phi(x_j)$ in the original space avoids curse of dimensionality
- Not all functions can be kernels
 - Must make sure there is a corresponding Φ in some high-dimensional space
 - Mercer's theorem (see textbook)

Characteristics of SVM

- The learning problem is formulated as a convex optimization problem
 - Efficient algorithms are available to find the global minima
 - Many of the other methods use greedy approaches and find locally optimal solutions
 - High computational complexity for building the model
- Robust to noise
- Overfitting is handled by maximizing the margin of the decision boundary,
- SVM can handle irrelevant and redundant attributes better than many other techniques
- The user needs to provide the type of kernel function and cost function
- Difficult to handle missing values
- What about categorical variables?

Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
 - Credit card fraud
 - Intrusion detection
 - Defective products in manufacturing assembly line
 - COVID-19 test results on a random sample
- **Key Challenge:**
 - Evaluation measures such as accuracy are not well-suited for imbalanced class

Measures of Classification Performance

	Predicted Class		
Actual Class		Yes	No
	Yes	TP	FN
	No	FP	TN

α is the probability that we reject the null hypothesis when it is true. This is a Type I error or a false positive (FP).

β is the probability that we accept the null hypothesis when it is false. This is a Type II error or a false negative (FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{ErrorRate} = 1 - \text{accuracy}$$

$$\text{Precision} = \text{Positive Predictive Value} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \text{Sensitivity} = \text{TP Rate} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \text{TN Rate} = \frac{TN}{TN + FP}$$

$$\text{FP Rate} = \alpha = \frac{FP}{TN + FP} = 1 - \text{specificity}$$

$$\text{FN Rate} = \beta = \frac{FN}{FN + TP} = 1 - \text{sensitivity}$$

$$\text{Power} = \text{sensitivity} = 1 - \beta$$

Problem with Accuracy

- Consider a 2-class problem
 - Number of Class NO examples = 990
 - Number of Class YES examples = 10
- If a model predicts everything to be class NO, accuracy is $990/1000 = 99\%$
 - This is misleading because this trivial model does not detect any class YES example
 - Detecting the rare class is usually more interesting (e.g., frauds, intrusions, defects, etc)

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	0	10
	Class>No	0	990

ction to Data Mining, 2nd Edition

Which model is better?

A

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	0	10
	Class>No	0	990

Accuracy: 99%

B

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	10	0
	Class>No	500	490

Accuracy: 50%

Alternative Measures

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

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

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

$$\text{F - measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	10	0
	Class>No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2 * 1 * 0.5}{1 + 0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	1	9
	Class>No	0	990

$$\text{Precision (p)} = \frac{1}{1+0} = 1$$

$$\text{Recall (r)} = \frac{1}{1+9} = 0.1$$

$$\text{F - measure (F)} = \frac{2 * 0.1 * 1}{1 + 0.1} = 0.18$$

$$\text{Accuracy} = \frac{991}{1000} = 0.991$$

Which of these classifiers is better?

A

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	10	40

Precision (p) = 0.8

Recall (r) = 0.8

F - measure (F) = 0.8

Accuracy = 0.8

B

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	1000	4000

Precision (p) = ~ 0.04

Recall (r) = 0.8

F - measure (F) = ~ 0.08

Accuracy = ~ 0.8

Alternative Measures

A	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	10	40

Precision (p) = 0.8
TPR = Recall (r) = 0.8
FPR = 0.2
F-measure (F) = 0.8
Accuracy = 0.8

$$\frac{\text{TPR}}{\text{FPR}} = 4$$

B	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	1000	4000

Precision (p) = 0.038
TPR = Recall (r) = 0.8
FPR = 0.2
F-measure (F) = 0.07
Accuracy = 0.8

$$\frac{\text{TPR}}{\text{FPR}} = 4$$

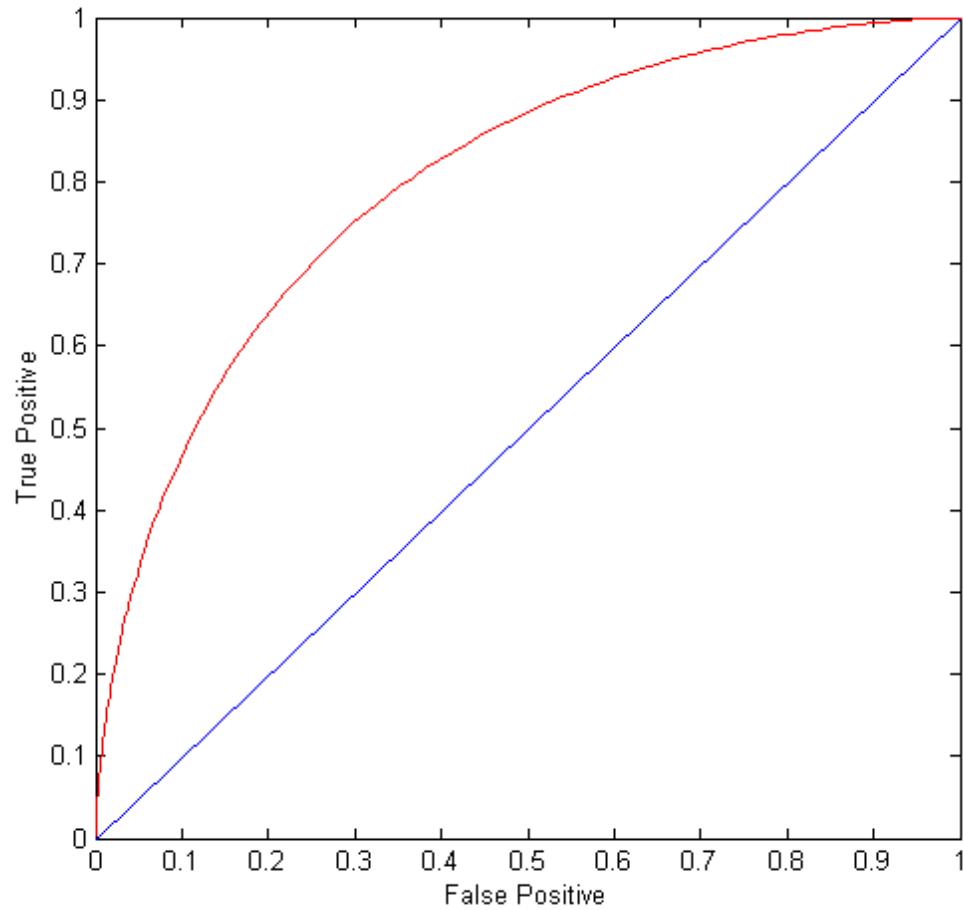
ROC (Receiver Operating Characteristic)

- A graphical approach for displaying trade-off between detection rate and false alarm rate
- Developed in 1950s for signal detection theory to analyze noisy signals
- ROC curve plots TPR against FPR
 - Performance of a model represented as a point in an ROC curve

ROC Curve

(TPR,FPR):

- (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

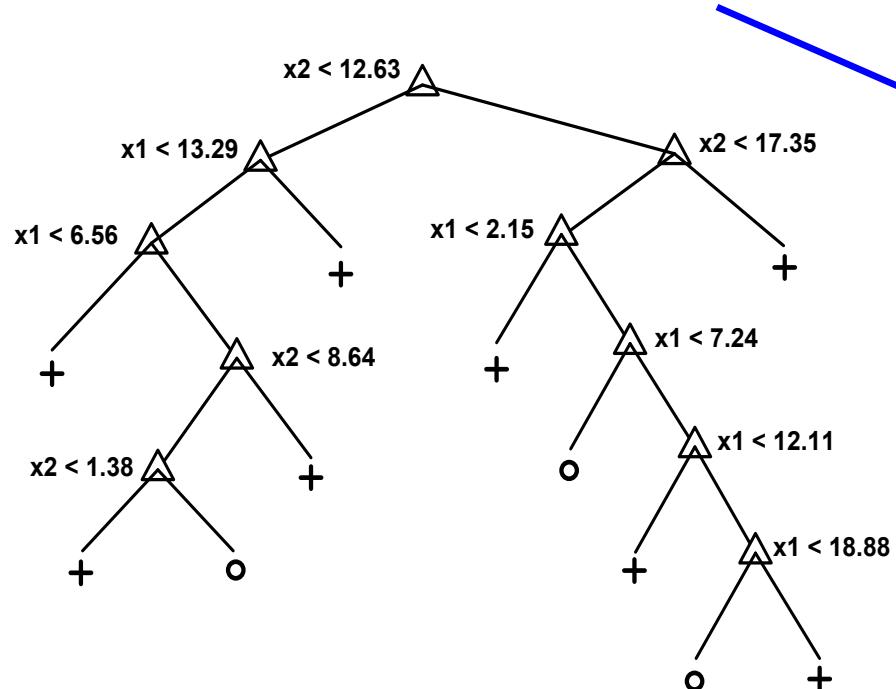


ROC (Receiver Operating Characteristic)

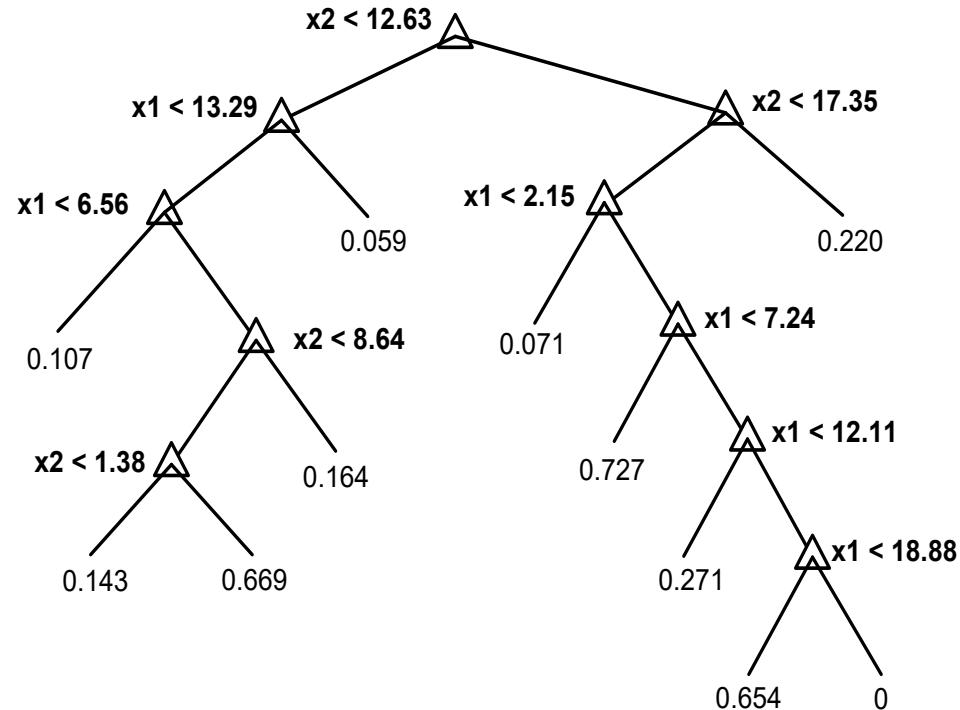
- To draw ROC curve, classifier must produce continuous-valued output
 - Outputs are used to rank test records, from the most likely positive class record to the least likely positive class record
 - By using different thresholds on this value, we can create different variations of the classifier with TPR/FPR tradeoffs
- Many classifiers produce only discrete outputs (i.e., predicted class)
 - How to get continuous-valued outputs?
 - ◆ Decision trees, rule-based classifiers, neural networks, Bayesian classifiers, k-nearest neighbors, SVM

Example: Decision Trees

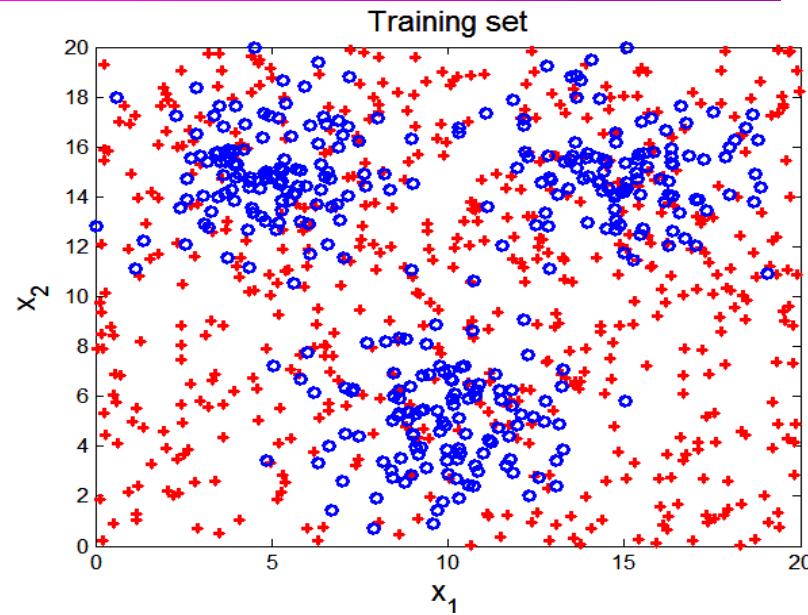
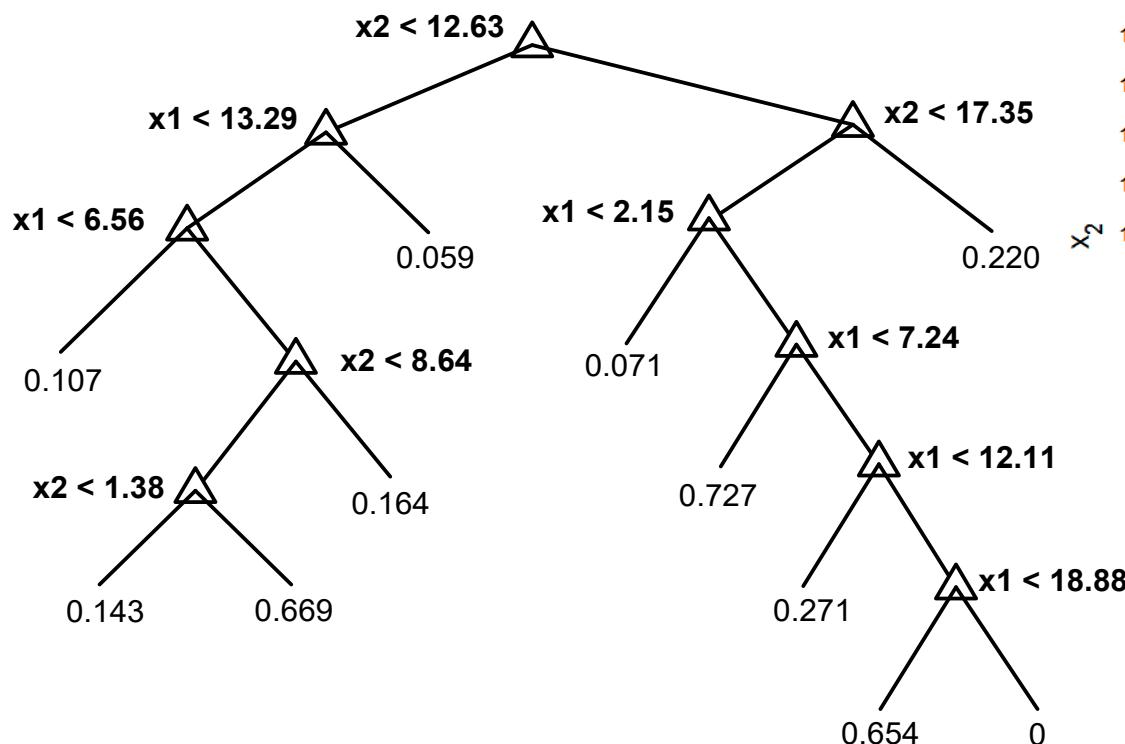
Decision Tree



Continuous-valued outputs



ROC Curve Example

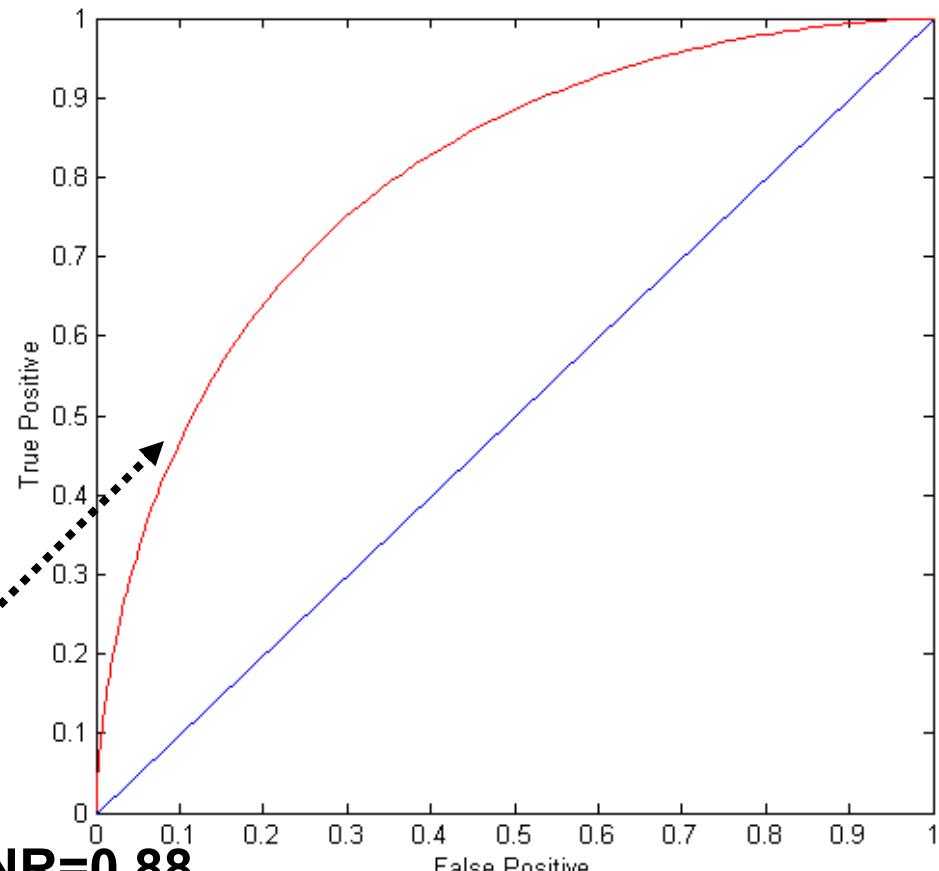
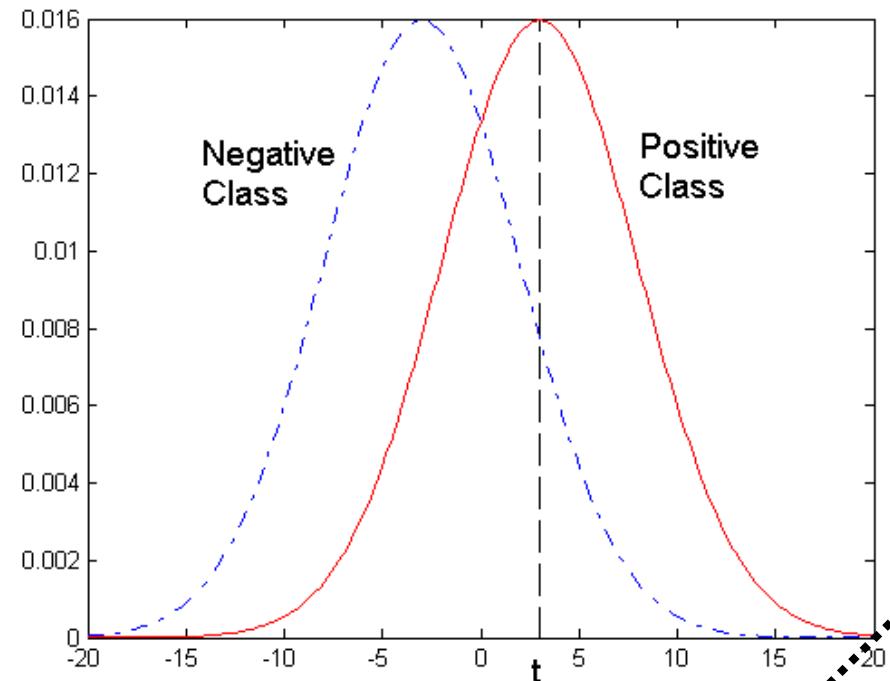


$\alpha = 0.3$		Predicted Class	
		Class 0	Class +
Actual Class	Class 0	645	209
	Class +	298	948

$\alpha = 0.7$		Predicted Class	
		Class 0	Class +
Actual Class	Class 0	181	673
	Class +	78	1168

ROC Curve Example

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



At threshold t :

$\text{TPR}=0.5$, $\text{FNR}=0.5$, $\text{FPR}=0.12$, $\text{TNR}=0.88$

How to Construct an ROC curve

Instance	Score	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

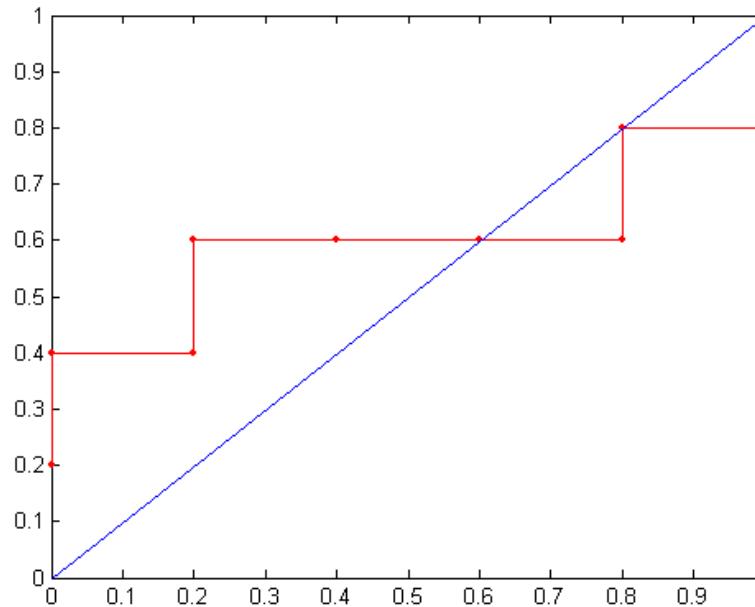
- Use a classifier that produces a continuous-valued score for each instance
 - The more likely it is for the instance to be in the + class, the higher the score
- Sort the instances in decreasing order according to the score
- Apply a threshold at each unique value of the score
- Count the number of TP, FP, TN, FN at each threshold
 - $TPR = TP/(TP+FN)$
 - $FPR = FP/(FP + TN)$

How to construct an ROC curve

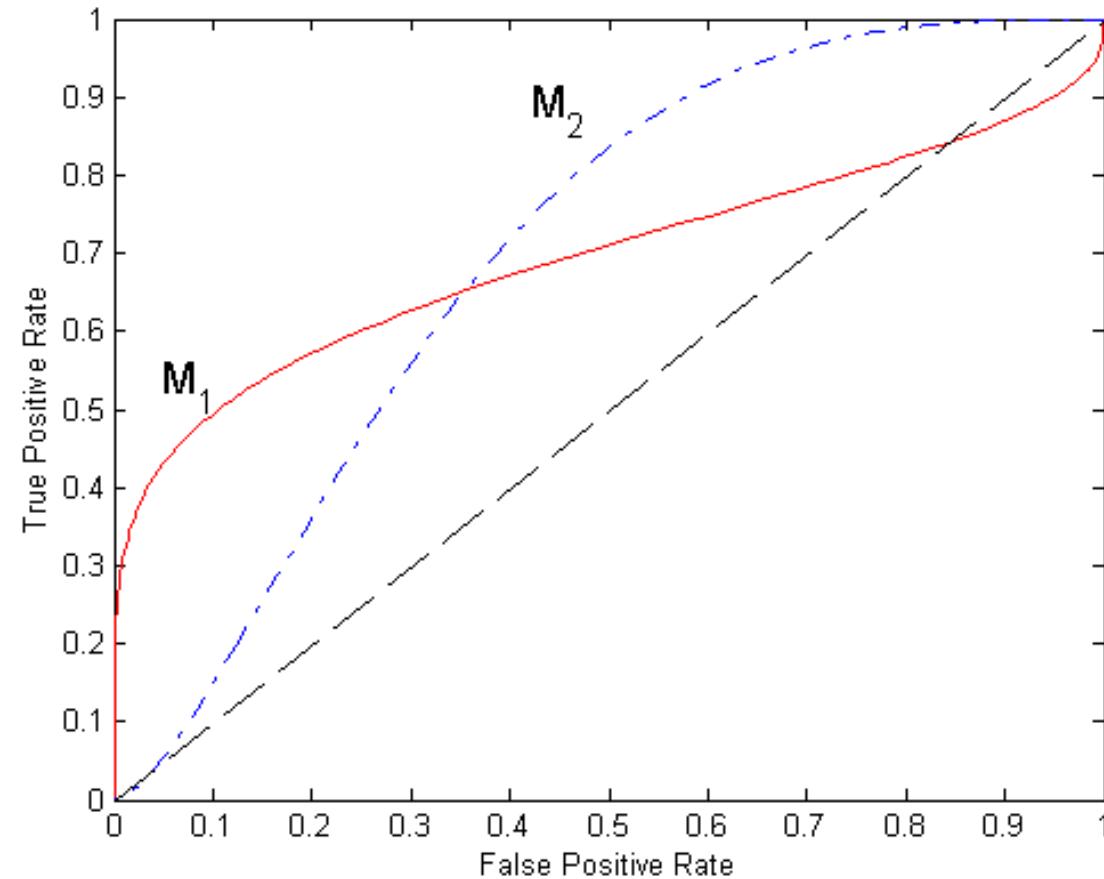
Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

→ →

ROC Curve:



Using ROC for Model Comparison



- No model consistently outperforms 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

Dealing with Imbalanced Classes - Summary

- Many measures exists, but none of them may be ideal in all situations
 - Random classifiers can have high value for many of these measures
 - TPR/FPR provides important information but may not be sufficient by itself in many practical scenarios
 - Given two classifiers, sometimes you can tell that one of them is strictly better than the other
 - ◆ C_1 is strictly better than C_2 if C_1 has strictly better TPR and FPR relative to C_2 (or same TPR and better FPR, and vice versa)
 - Even if C_1 is strictly better than C_2 , C_1 's F-value can be worse than C_2 's if they are evaluated on data sets with different imbalances
 - Classifier C_1 can be better or worse than C_2 depending on the scenario at hand (class imbalance, importance of TP vs FP, cost/time tradeoffs)

Which Classifier is better? Low Skew case

T1	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	50	50
	Class>No	1	99

Precision (p) = 0.98

TPR = Recall (r) = 0.5

FPR = 0.01

TPR/FPR = 50

F – measure = 0.66

T2	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	10	90

Precision (p) = 0.9

TPR = Recall (r) = 0.99

FPR = 0.1

TPR/FPR = 9.9

F – measure = 0.94

T3	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	1	99

Precision (p) = 0.99

TPR = Recall (r) = 0.99

FPR = 0.01

TPR/FPR = 99

F – measure = 0.99

Which Classifier is better? Medium Skew case

T1	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	50	50
	Class>No	10	990

Precision (p) = 0.83

TPR = Recall (r) = 0.5

FPR = 0.01

TPR/FPR = 50

F – measure = 0.62

T2	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	100	900

Precision (p) = 0.5

TPR = Recall (r) = 0.99

FPR = 0.1

TPR/FPR = 9.9

F – measure = 0.66

T3	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	10	990

Precision (p) = 0.9

TPR = Recall (r) = 0.99

FPR = 0.01

TPR/FPR = 99

F – measure = 0.94

Which Classifier is better? High Skew case

T1	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	50	50
	Class>No	100	9900

Precision (p) = 0.3

TPR = Recall (r) = 0.5

FPR = 0.01

TPR/FPR = 50

F – measure = 0.375

T2	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	1000	9000

Precision (p) = 0.09

TPR = Recall (r) = 0.99

FPR = 0.1

TPR/FPR = 9.9

F – measure = 0.165

T3	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	99	1
	Class>No	100	9900

Precision (p) = 0.5

TPR = Recall (r) = 0.99

FPR = 0.01

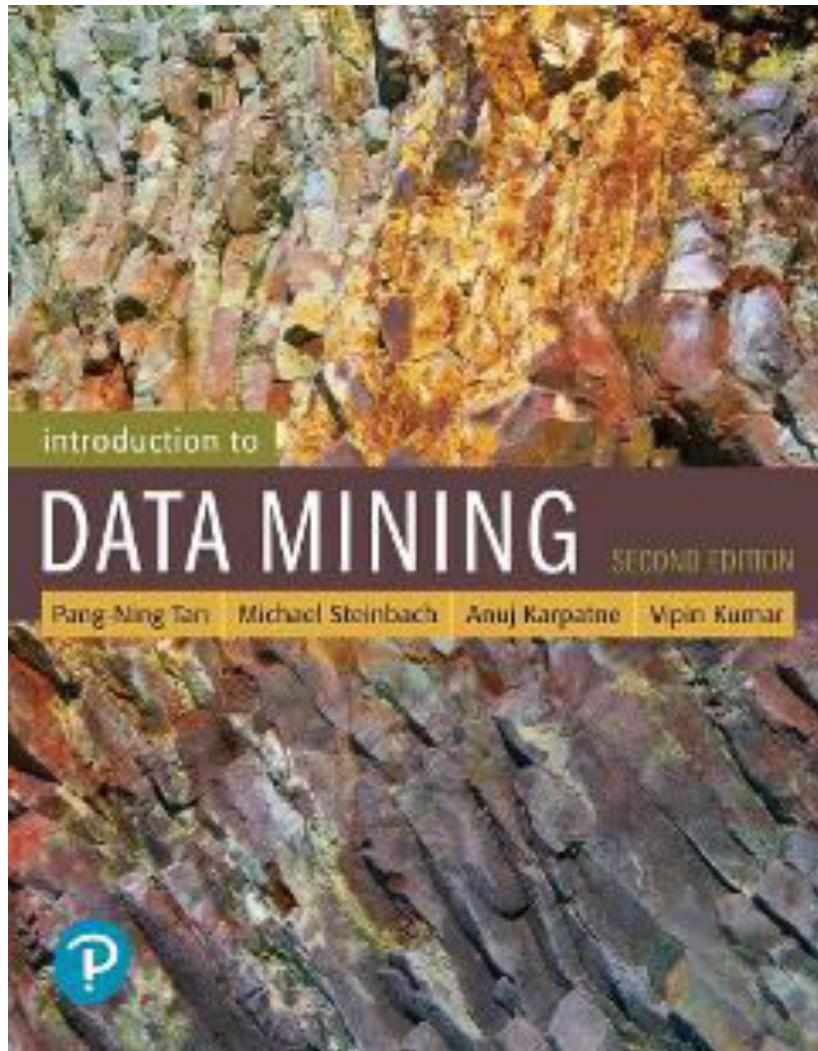
TPR/FPR = 99

F – measure = 0.66

Building Classifiers with Imbalanced Training Set

- Modify the distribution of training data so that rare class is well-represented in training set
 - Undersample the majority class
 - Oversample the rare class

CSCE 5380/4380 – Data Mining



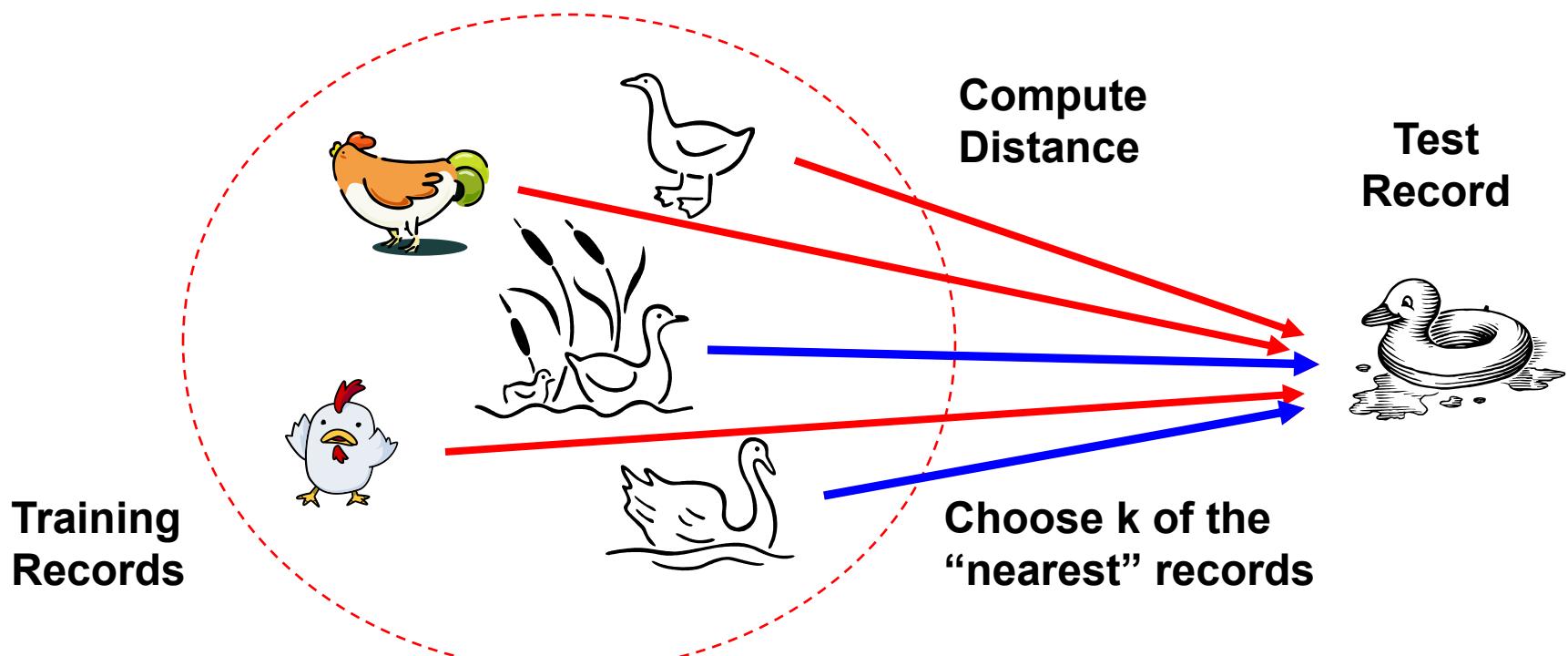
Chapter Four: **K-Nearest Neighbor & Logistic Regression**

Outline

- **Nearest Neighbor Classifiers**
- **K-NN: Algorithms**
- **Characteristics of K-NN**
- **Logistic Regression**
- **Learning a Logistic Regression Model**
- **Characteristics of Logistic Regression**

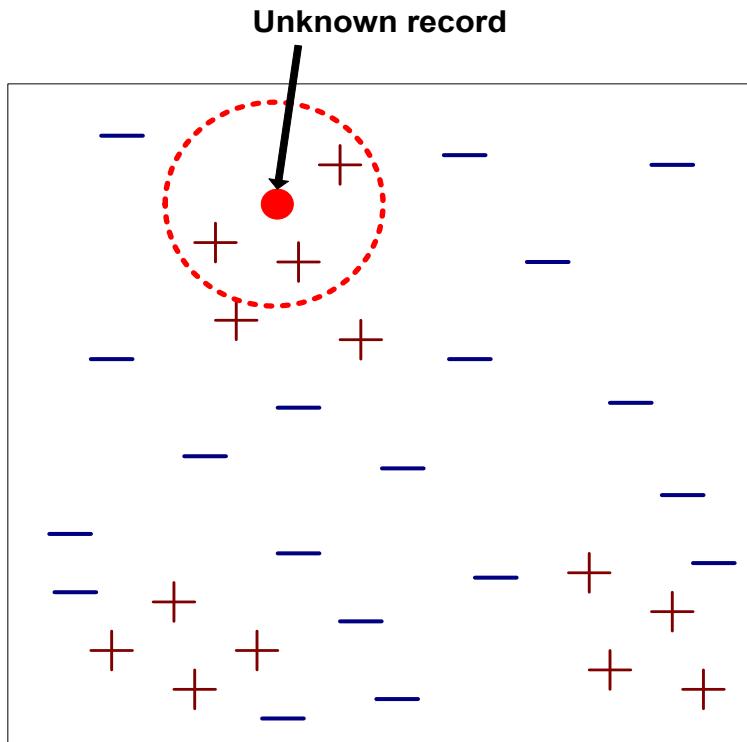
Nearest Neighbor Classifiers

- Basic idea:
 - If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest Neighbor Classifiers

The k -nearest neighbors of a given test instance z refer to the k training examples that are closest to z

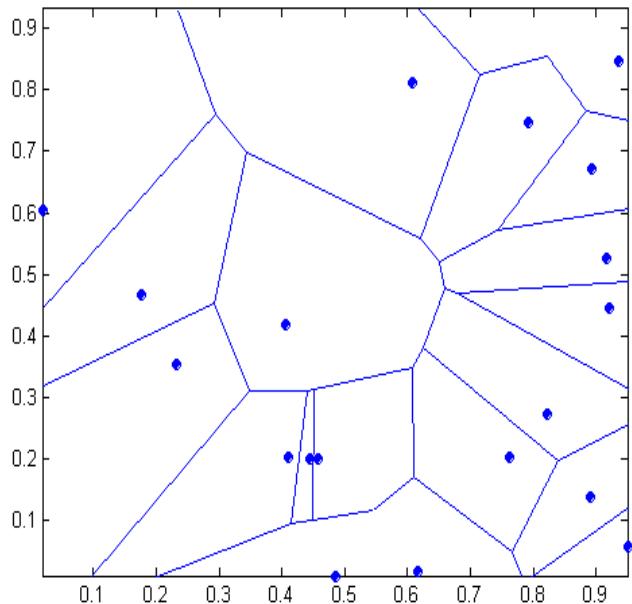


- Requires the following:
 - A set of labeled records
 - Proximity metric to compute distance/similarity between a pair of records
 - e.g., Euclidean distance
 - The value of k , the number of nearest neighbors to retrieve
 - A method for using class labels of K nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Nearest Neighbor classifiers

- Nearest neighbor classifiers are known as ***Lazy Learners*** (Strategy of delaying the process of modeling the training data until it is needed to classify the test instances)
- Nearest neighbor classifiers are local classifiers
- They can produce decision boundaries of arbitrary shapes.

1-nn decision boundary is a Voronoi Diagram



K-NN: Algorithms

- 1: Let k be the number of nearest neighbors and D be the set of training examples.
- 2: **for** each test instance $z = (\mathbf{x}', y')$ **do**
- 3: Compute $z = (\mathbf{x}', \mathbf{x})$, the distance between z and every example, $(\mathbf{x}, y) \in D$.
- 4: Select $D_z \subseteq D$, the set of k closest training examples to z .
- 5: $y' = \arg \max_v \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
- 6: **end for**

How to Determine the class label of a Test Sample?

- Take the majority vote of class labels among the k-nearest neighbors

$$\text{Majority Voting : } y' = \arg \max_v \sum_{(x_i, y_i) \in D_z} I(v = y_i),$$

- To reduce the impact of K neighbors, Weight the vote according to distance
 - weight factor, $w = 1/d^2$

Choice of proximity measure matters

- For documents, cosine is better than correlation or Euclidean

1 1 1 1 1 1 1 1 1 1 0

0 1 1 1 1 1 1 1 1 1 1

vs

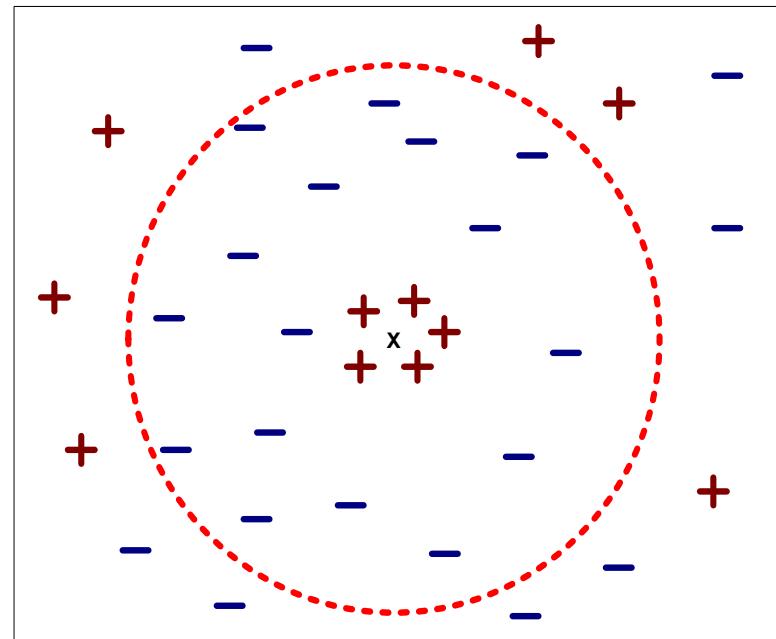
0 0 0 0 0 0 0 0 0 0 1

1 0 0 0 0 0 0 0 0 0 0

Euclidean distance = 1.4142 for both pairs, but
the cosine similarity measure has different
values for these pairs.

Nearest Neighbor Classification...

- Choosing the value of k :
 - If k is too small, sensitive to noise points (overfitting)
 - If k is too large, neighborhood may include points from other classes (misclassification)



Nearest Neighbor Classification...

- Data preprocessing is often required
 - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
 - ◆ Example:
 - height of a person may vary from 1.5m to 1.8m
 - weight of a person may vary from 90lb to 300lb
 - income of a person may vary from \$10K to \$1M
 - Time series are often standardized to have 0 means a standard deviation of 1

Nearest Neighbor Classification...

- How to handle missing values in training and test sets?
 - Proximity computations normally require the presence of all attributes
 - Some approaches use the subset of attributes present in two instances
 - ◆ This may not produce good results since it effectively uses different proximity measures for each pair of instances
 - ◆ Thus, proximities are not comparable

Characteristics of K-NN

- K-NN is part of instance-based learning, which uses the training examples to make predictions for a test instance.
- Nearest neighbor classifiers make their predictions based on local information and can produce decision boundaries of arbitrary shape.
- Nearest neighbor classifiers have difficulty handling missing values in both the training and test sets since proximity computations normally require the presence of all attributes.
- Nearest neighbor classifiers can handle the presence of interacting attributes, but the presence of irrelevant and redundant attributes can adversely affect the performance of nearest neighbor classifiers.
- K-NN can produce wrong predictions unless the appropriate proximity measure and data preprocessing steps are taken.

Logistic Regression

- Logistic regression is a classification algorithm used to assign observations to a discrete set of classes (e.g. to classify instances, in some classification problems, to Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign).
- Logistic regression transforms its output using the *logistic (sigmoid) function* to return a probability value.
- logistic regression is a *probabilistic discriminative model*, which directly estimates the odds of a data instance x using its attribute values.

Different ways of expressing probability

- Consider a two-outcome probability space, where:
 - $p(O_1) = p$
 - $p(O_2) = 1 - p = q$
- Can express probability of O_1 as:

	notation	range equivalents		
standard probability	p	0	0.5	1
odds	p / q	0	1	$+\infty$
log odds (logit)	$\log(p / q)$	$-\infty$	0	$+\infty$

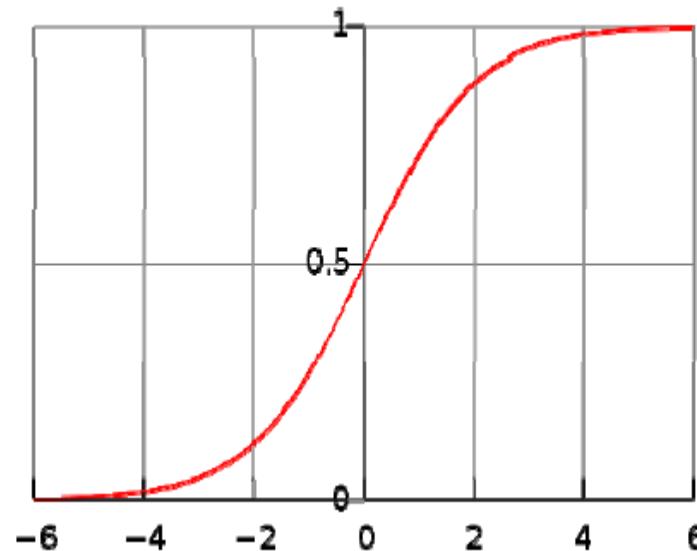
Log odds (logit function) & logistic function

- Numeric treatment of outcomes O_1 and O_2 is equivalent
 - If neither outcome is favored over the other, then log odds = 0.
 - If one outcome is favored with log odds = x , then other outcome is disfavored with log odds = $-x$.

$$z = \log\left(\frac{p}{1-p}\right) \quad \text{logit function}$$

$$\frac{p}{1-p} = e^z$$

$$p = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \quad \text{logistic function}$$



Logistic Regression: The Scenario

- A multidimensional feature space (features can be categorical or continuous).
- Outcome is discrete, not continuous.
 - We'll focus on case of two classes.
- It seems plausible that a linear decision boundary (hyperplane) will give good predictive accuracy.

Logistic Regression: The Idea

- Model consists of a vector θ in n -dimensional feature space (Model Parameters)
- For a point x in feature space, project it onto θ to convert it into a real number z in the range $-\infty$ to $+\infty$

$$z = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

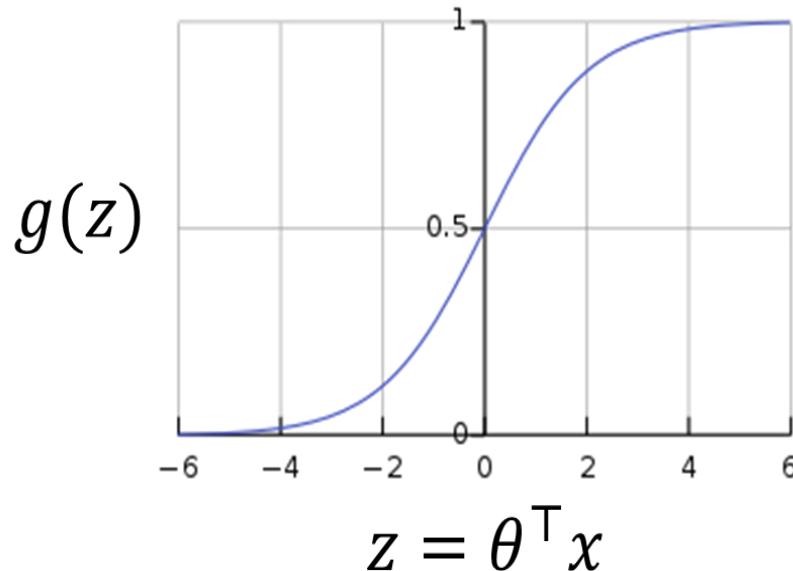
- Map z to the range 0 to 1 using the logistic function (Hypothesis Representation)

$$h_{\theta}(x) = g(\theta^T x), \text{ where } g(z) = \frac{1}{1+e^{-z}}$$

- Overall, logistic regression maps a point x in n -dimensional feature space to a value in the range 0 to 1

Logistic Regression: The Model

$$h_{\theta}(x) = g(\theta^T x)$$
$$g(z) = \frac{1}{1 + e^{-z}}$$



Suppose predict “y = 1” if $h_{\theta}(x) \geq 0.5$

$$z = \theta^T x \geq 0$$

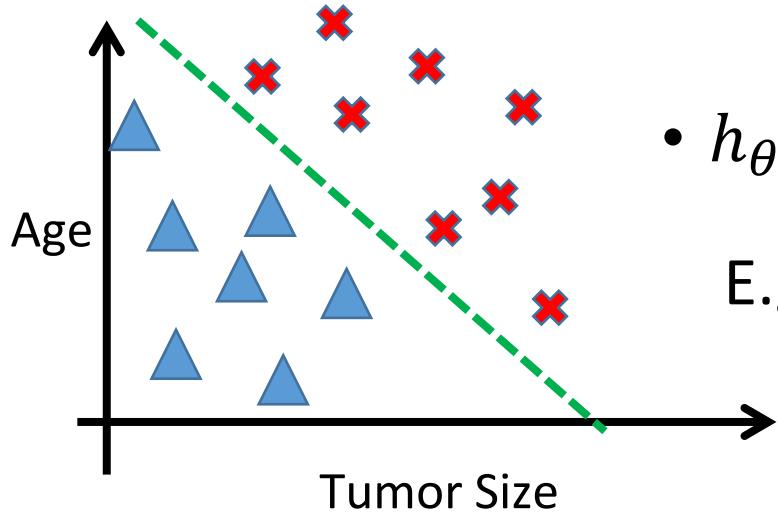
predict “y = 0” if $h_{\theta}(x) < 0.5$

$$z = \theta^T x < 0$$

Logistic Regression: Decision Boundary

Example:

Classification of Tumor Malignant ($y=1$) or Benign ($y=0$), based on two features (Age and Tumor Size)



- $$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

E.g., $\theta_0 = -3, \theta_1 = 1, \theta_2 = 1$

- Predict " $y = 1$ " if $-3 + x_1 + x_2 \geq 0$

Learning a Logistic Regression Model

- Need to optimize θ so the model gives the best possible reproduction of training set labels
 - Done by numerical approximation of maximum likelihood
- or
- By using stochastic gradient descent

Logistic Regression: Cost Function

How to learn a *logistic regression model* $\mathbf{h}_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x})$,
where $\boldsymbol{\theta} = [\theta_0, \dots, \theta_n]$ and $\mathbf{x} = [x_0, \dots, x_n]$?

- By minimizing the following cost function:

$$\text{Cost}(\mathbf{h}_{\theta}(\mathbf{x}), y) = -y \log\left(\frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}\right) - (1 - y) \log\left(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}\right)$$

- That is:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \frac{1}{n} \sum_{i=1}^m \text{Cost}(\mathbf{h}_{\theta}(\mathbf{x}^{(i)}), y^{(i)})$$

Cost function
 $J(\boldsymbol{\theta})$

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \frac{1}{n} \sum_{i=1}^m \boxed{-y^{(i)} \log\left(\frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}^{(i)}}}\right) - (1 - y) \log\left(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}^{(i)}}}\right)}$$

Gradient Descent For Logistic Regression

Outline:

- Have cost function $J(\theta)$, where $\theta = [\theta_0, \dots, \theta_n]$
- Start off with some guesses for $\theta_0, \dots, \theta_n$
 - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
- Repeat until convergence {

}

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

Partial derivative

Note: Update all θ_j simultaneously

Learning rate, which controls how big a step we take when we update θ_j

Gradient Descent For Logistic Regression

Outline:

- Have cost function $J(\theta)$, where $\theta = [\theta_0, \dots, \theta_n]$
- Start off with some guesses for $\theta_0, \dots, \theta_n$
 - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
- Repeat until convergence {

$$\theta_j = \theta_j - \alpha \sum_{i=1}^m \left(\frac{1}{1 + e^{-\theta^T x^{(i)}}} - y^{(i)} \right) x_j^{(i)}$$

*The final formula
after applying
partial derivatives*

}

Logistic Regression: Inference after learning

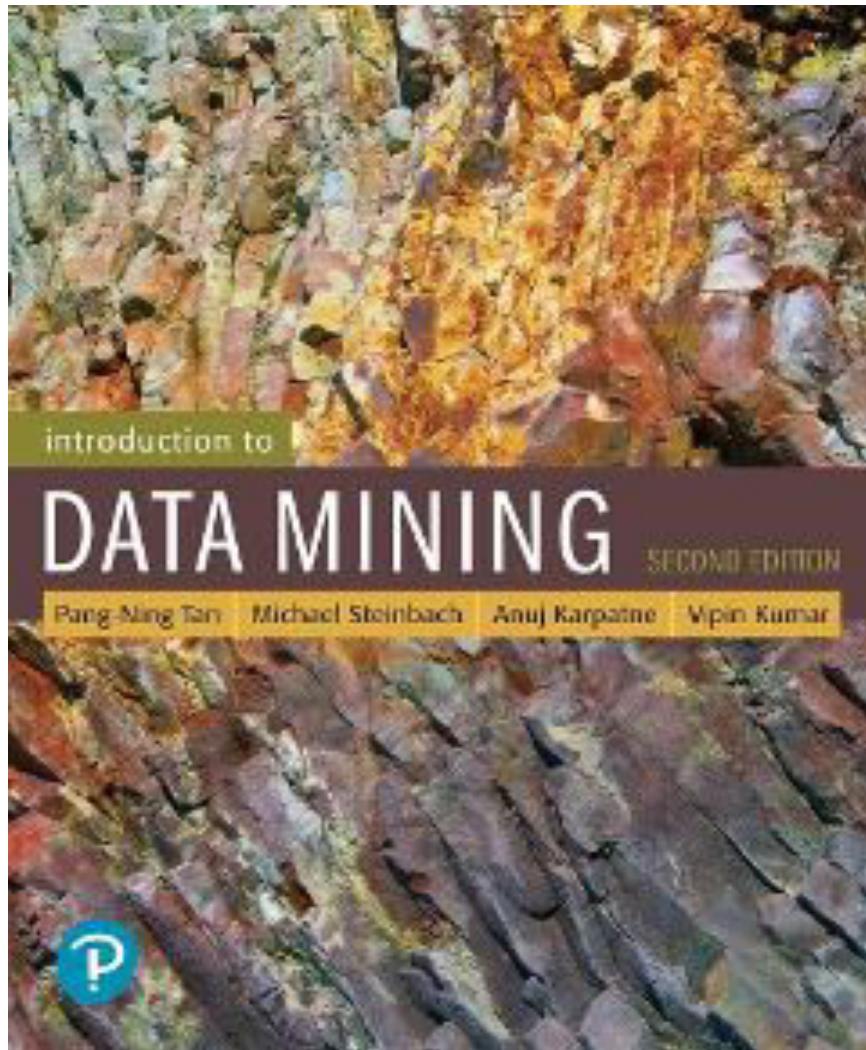
- After learning the parameters $\theta = [\theta_0, \dots, \theta_n]$, we can predict the output of any new unseen $x = [x_0, \dots, x_n]$ as follows:

$$\begin{cases} \text{if } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} < 0.5 \text{ predict 0} \\ \text{Else if } h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \geq 0.5 \text{ predict 1} \end{cases}$$

Characteristics of Logistic Regression

- The learned parameters of logistic regression can be analyzed to understand the relationships between attributes and class labels.
- Because logistic regression does not involve computing densities and distances in the attribute space, it can work more robustly even in high-dimensional settings.
- Logistic regression can handle irrelevant attributes by learning weight parameters close to 0 for attributes that do not provide any gain in performance during training. It can also handle interacting attributes since the learning of model parameters is achieved in a joint fashion by considering the effects of all attributes together.
- Logistic regression cannot handle data instances with missing values, since the posterior probabilities are only computed by taking a weighted sum of all the attributes.

CSCE 5380 – Data Mining



Chapter One: **Introduction**

Large-scale Data is Everywhere!

- There has been enormous data growth in both commercial and scientific databases due to advances in data generation and collection technologies
- New mantra
 - Gather whatever data you can whenever and wherever possible.
- Expectations
 - Gathered data will have value either for the purpose collected or for a purpose not envisioned.



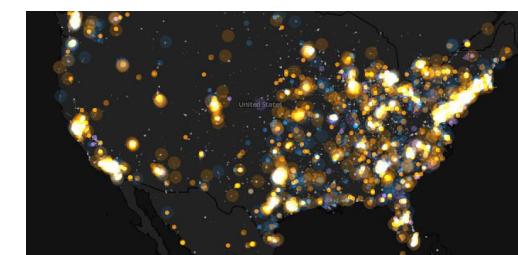
Cyber Security



E-Commerce



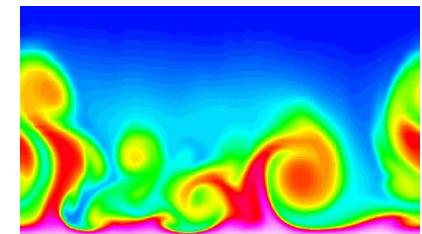
Traffic Patterns



Social Networking: Twitter



Sensor Networks



Computational Simulations

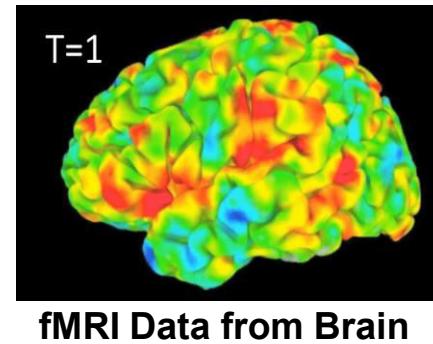
Why Data Mining? Commercial Viewpoint

- Lots of data is being collected and warehoused
 - Web data
 - ◆ Google has Peta Bytes of web data
 - ◆ Facebook has billions of active users
 - purchases at department/grocery stores, e-commerce
 - ◆ Amazon handles millions of visits/day
 - Bank/Credit Card transactions
- Computers have become cheaper and more powerful
- Competitive Pressure is Strong
 - Provide better, customized services for an edge (e.g. in Customer Relationship Management)

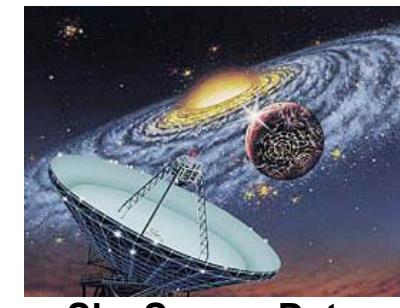


Why Data Mining? Scientific Viewpoint

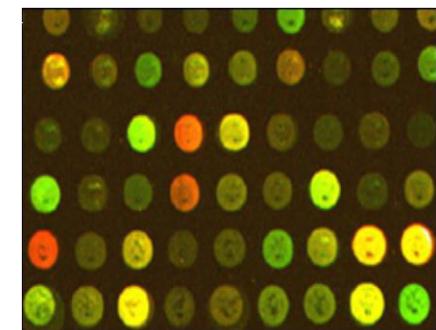
- Data collected and stored at enormous speeds
 - remote sensors on a satellite
 - ◆ NASA EOSDIS archives over petabytes of earth science data / year
 - telescopes scanning the skies
 - ◆ Sky survey data
 - High-throughput biological data
 - scientific simulations
 - ◆ terabytes of data generated in a few hours
- Data mining helps scientists
 - in automated analysis of massive datasets
 - In hypothesis formation



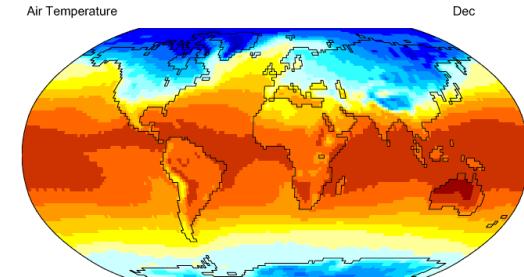
fMRI Data from Brain



Sky Survey Data



Gene Expression Data



Surface Temperature of Earth

Great opportunities to improve productivity in all walks of life

McKinsey Global Institute

Big data: The next frontier for innovation, competition, and productivity

Big data—a growing torrent

\$600 to buy a disk drive that can store all of the world's music

5 billion mobile phones in use in 2010

30 billion pieces of content shared on Facebook every month

40% projected growth in global data generated per year vs. 5% growth in global IT spending

235 terabytes data collected by the US Library of Congress in April 2011

15 out of 17 sectors in the United States have more data stored per company than the US Library of Congress

Big data—capturing its value

\$300 billion potential annual value to US health care—more than double the total annual health care spending in Spain

€250 billion potential annual value to Europe's public sector administration—more than GDP of Greece

\$600 billion potential annual consumer surplus from using personal location data globally

60% potential increase in retailers' operating margins possible with big data

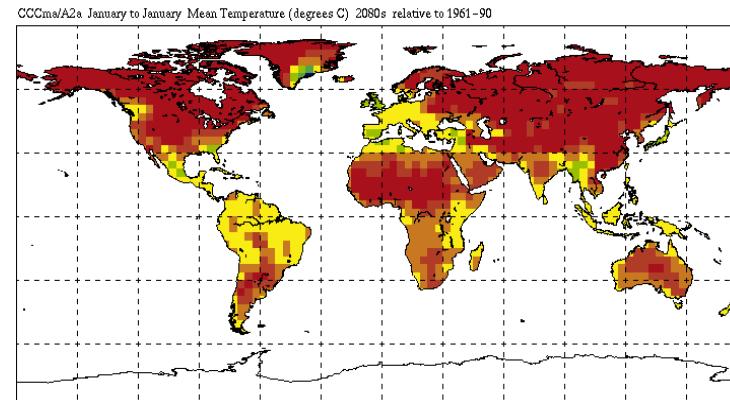
140,000–190,000 more deep analytical talent positions, and

1.5 million more data-savvy managers needed to take full advantage of big data in the United States

Great Opportunities to Solve Society's Major Problems



Improving health care and reducing costs



Predicting the impact of climate change



Finding alternative/ green energy sources

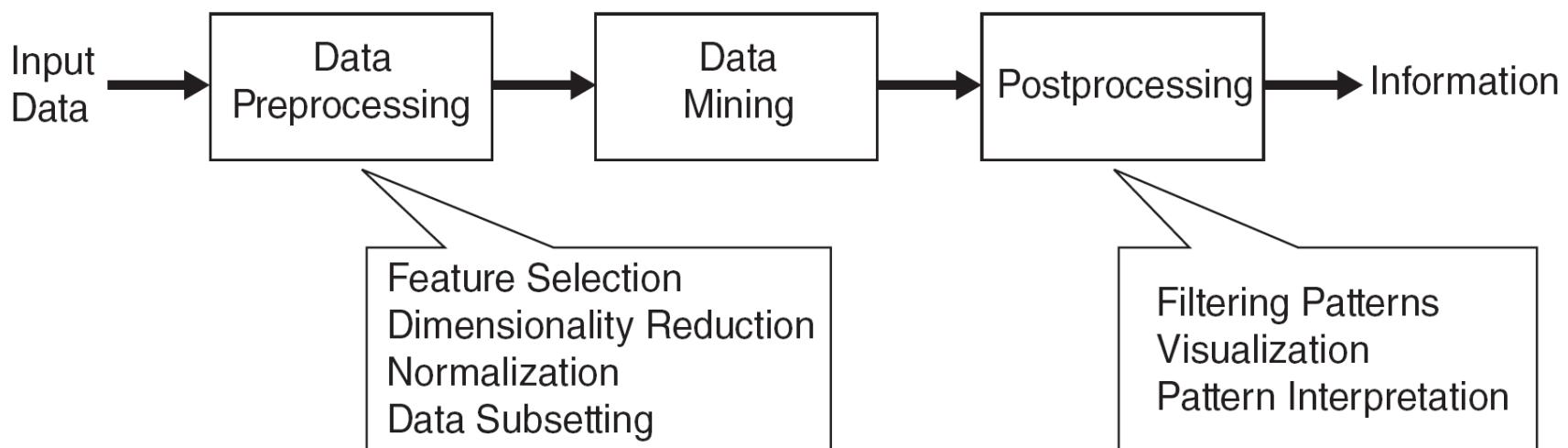


Reducing hunger and poverty by increasing agriculture production

What is Data Mining?

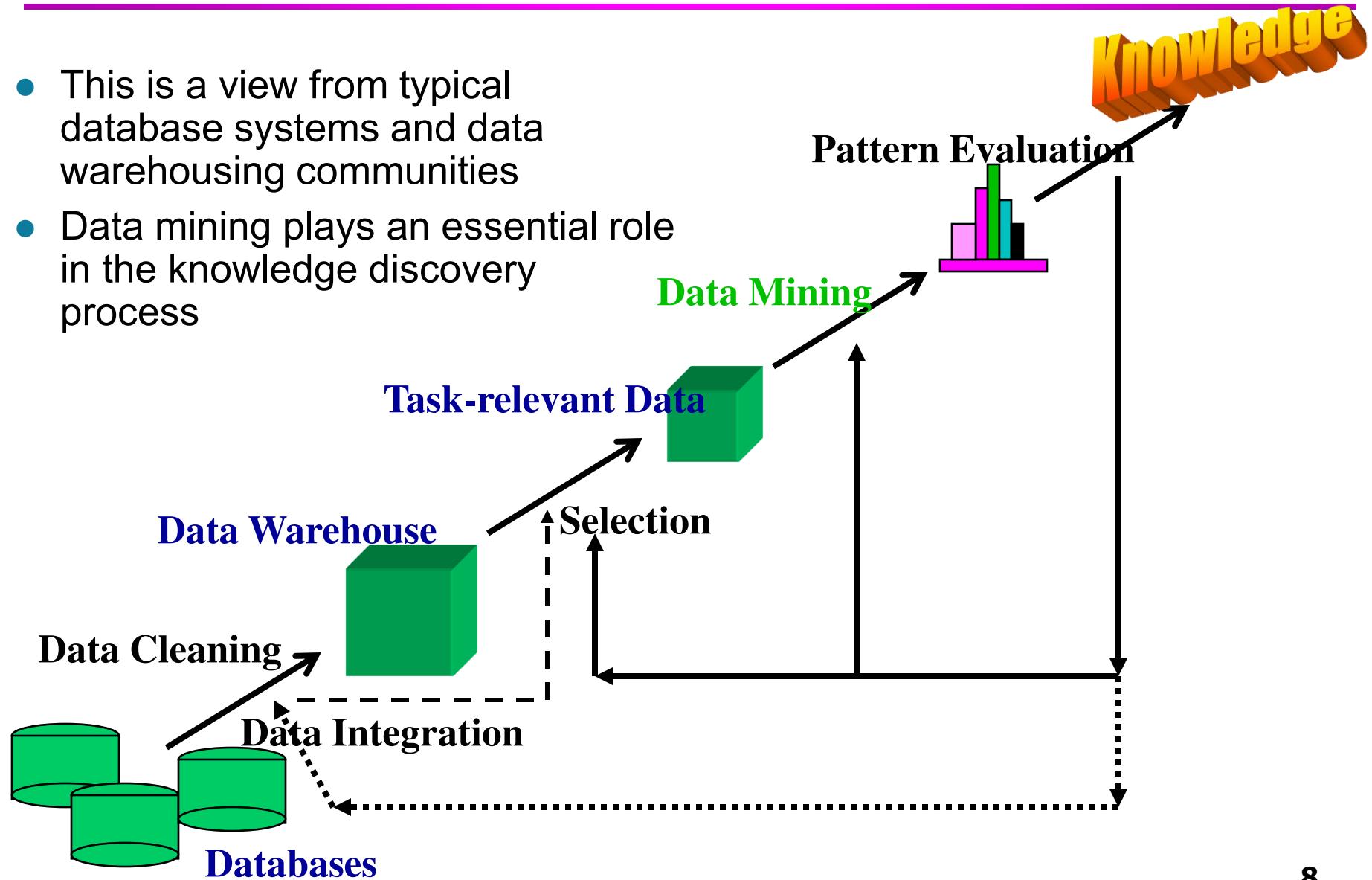
• Many Definitions

- Non-trivial extraction of implicit, previously unknown and potentially useful information from data
- Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns



Knowledge Discovery (KDD) Process

- This is a view from typical database systems and data warehousing communities
- Data mining plays an essential role in the knowledge discovery process

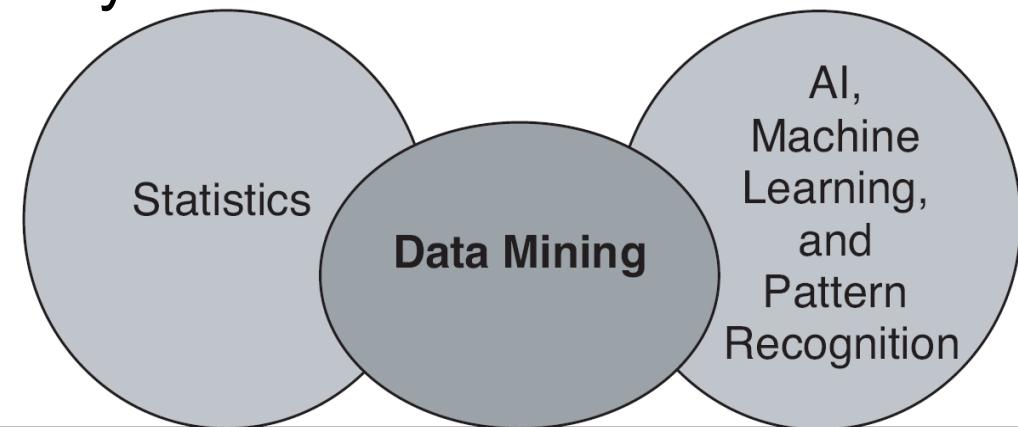


Data Mining: On What Kinds of Data?

- **Database-oriented data sets and applications**
 - Relational database, data warehouse, transactional database
- **Advanced data sets and advanced applications**
 - Data streams and sensor data
 - Time-series data, temporal data, sequence data (incl. bio-sequences)
 - Structure data, graphs, social networks and multi-linked data
 - Object-relational databases
 - Heterogeneous databases and legacy databases
 - Spatial data and spatiotemporal data
 - Multimedia database
 - Text databases
 - The World-Wide Web

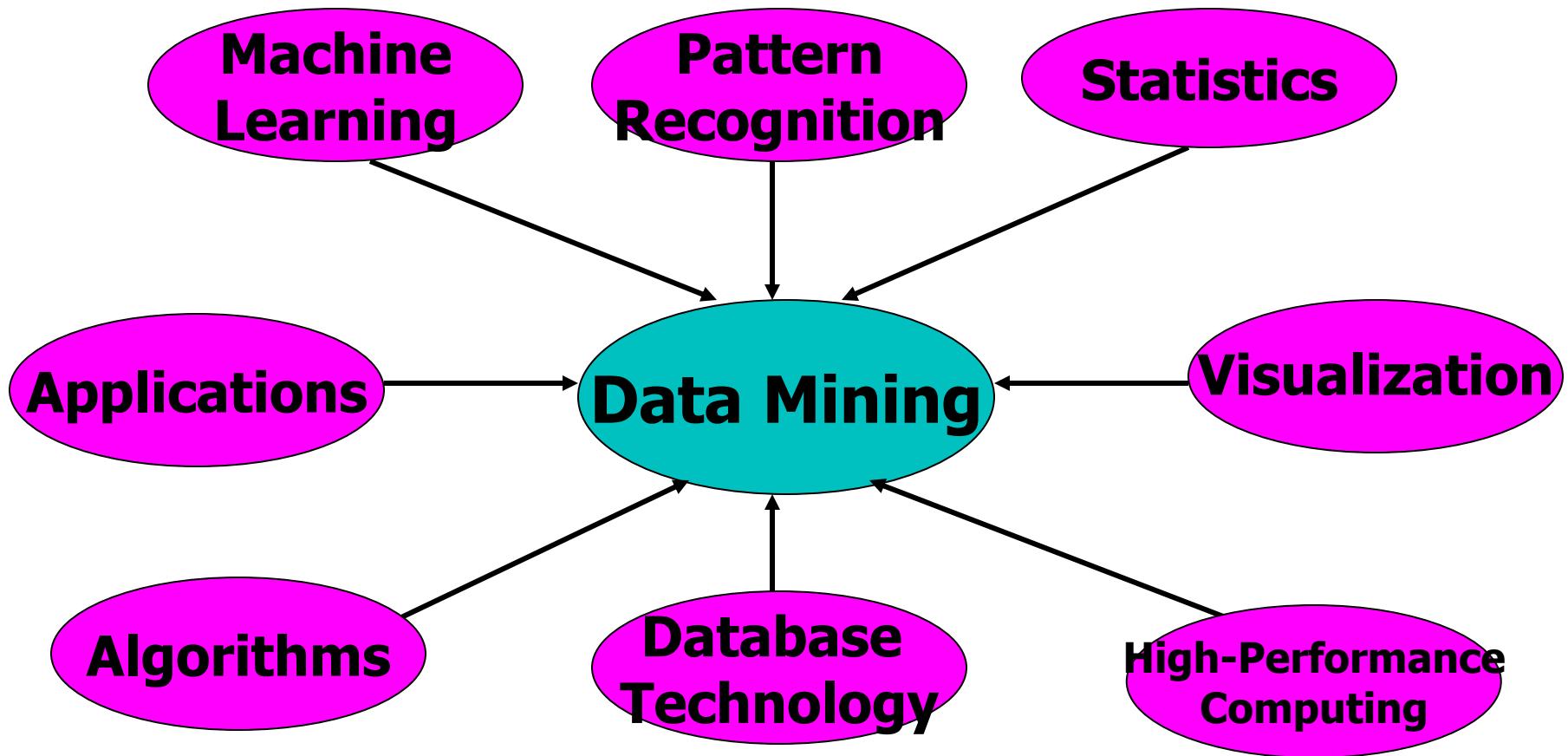
Origins of Data Mining

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional techniques may be unsuitable due to data that is
 - Large-scale
 - High dimensional
 - Heterogeneous
 - Complex
 - Distributed
- A key component of the emerging field of data science and data-driven discovery



Database Technology, Parallel Computing, Distributed Computing

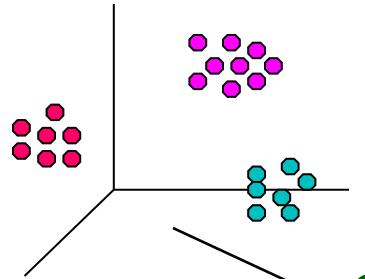
Data Mining: Confluence of Multiple Disciplines



Data Mining Tasks

- Prediction Methods
 - Use some variables to predict unknown or future values of other variables.
- Description Methods
 - Find human-interpretable patterns that describe the data.

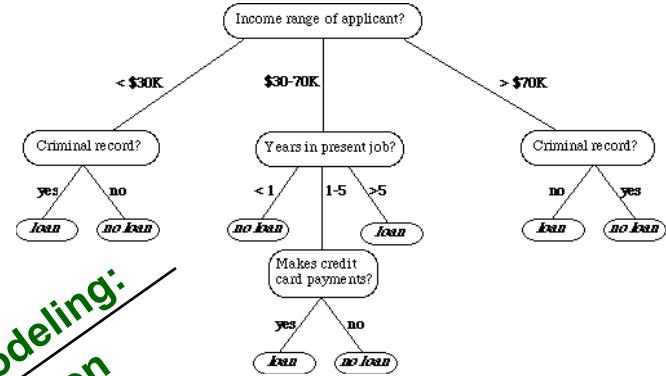
Data Mining Tasks ...



Clustering

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
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

*Predictive Modeling:
Classification*



*Anomaly
Detection*



Data Mining Tasks: (1) Classification

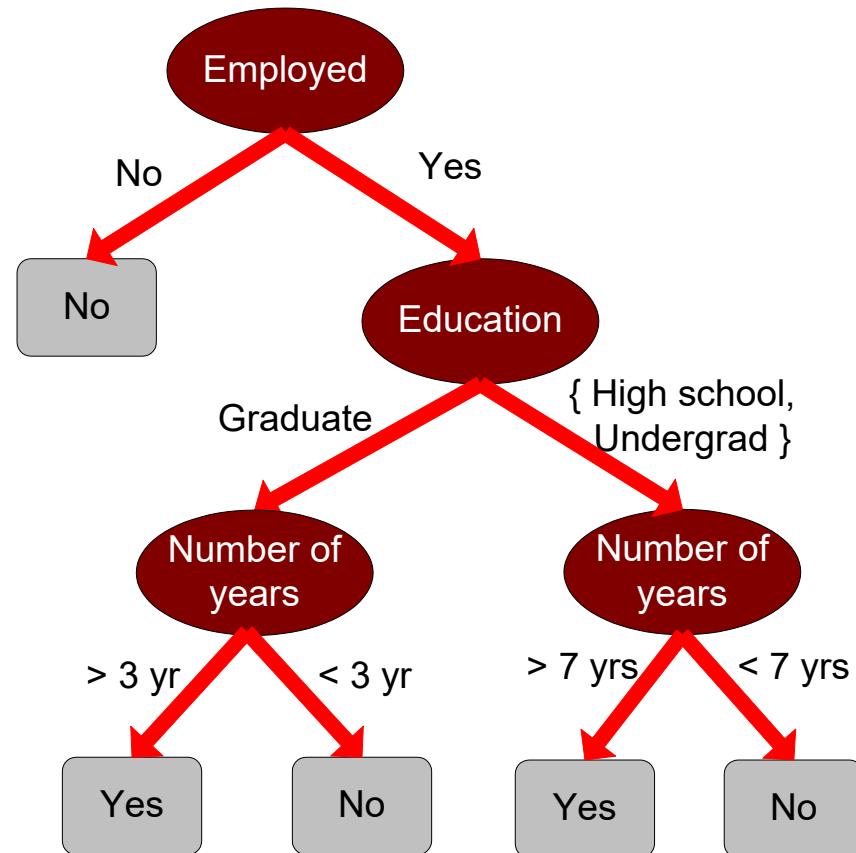
- **Classification and label prediction**
 - Construct models based on some training examples
 - Describe and distinguish classes or concepts for future prediction
 - ◆ E.g., classify countries based on (climate), or classify cars based on (gas mileage)
 - Predict some unknown class labels
- **Typical methods**
 - Decision trees, naïve Bayesian classification, support vector machines, neural networks, rule-based classification, pattern-based classification, logistic regression, ...
- **Typical applications:**
 - Credit card fraud detection, direct marketing, classifying stars, diseases, web-pages, ...

Predictive Modeling: Classification

- Find a model for class attribute as a function of the values of other attributes

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
...

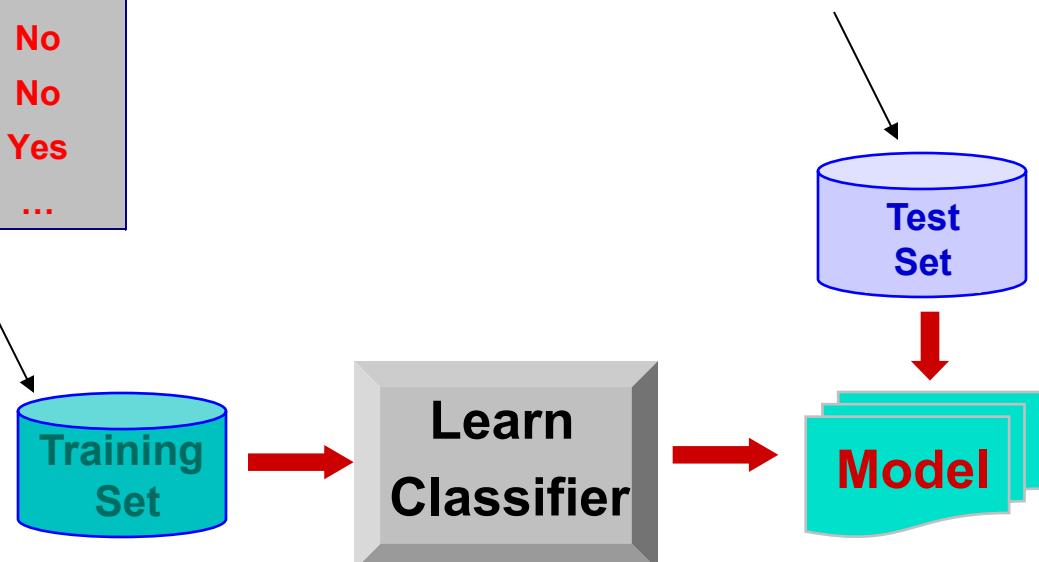
Model for predicting credit worthiness



Classification Example

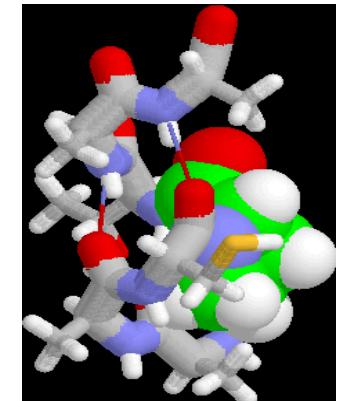
categorical categorical quantitative class				
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
...

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
...



Examples of Classification Task

- Classifying credit card transactions as legitimate or fraudulent
- Classifying land covers (water bodies, urban areas, forests, etc.) using satellite data
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Identifying intruders in the cyberspace
- Predicting tumor cells as benign or malignant
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil



Classification: Application 1

- 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.

Classification: Application 2

- Churn prediction for telephone customers
 - **Goal:** To predict whether a customer is likely to be lost to a competitor.
 - **Approach:**
 - ◆ Use detailed record of transactions with each of the past and present customers, to find attributes.
 - How often the customer calls, where he calls, what time-of-the day he calls most, his financial status, marital status, etc.
 - ◆ Label the customers as loyal or disloyal.
 - ◆ Find a model for loyalty.

From [Berry & Linoff] Data Mining Techniques, 1997

Classification: Application 3

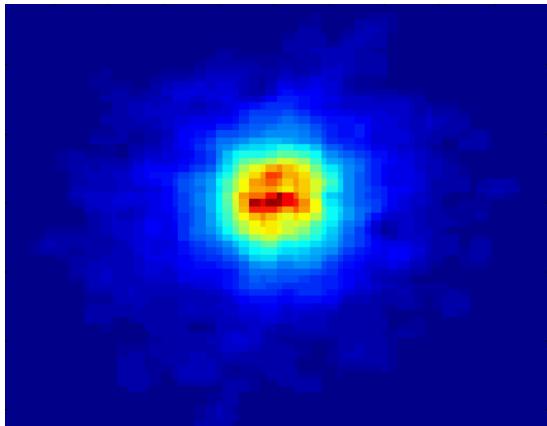
- Sky Survey Cataloging
 - **Goal:** To predict class (star or galaxy) of sky objects, especially visually faint ones, based on the telescopic survey images (from Palomar Observatory).
 - 3000 images with 23,040 x 23,040 pixels per image.
 - **Approach:**
 - ◆ Segment the image.
 - ◆ Measure image attributes (features) - 40 of them per object.
 - ◆ Model the class based on these features.
 - ◆ Success Story: Could find 16 new high red-shift quasars, some of the farthest objects that are difficult to find!

From [Fayyad, et.al.] Advances in Knowledge Discovery and Data Mining, 1996

Classifying Galaxies

Courtesy: <http://aps.umn.edu>

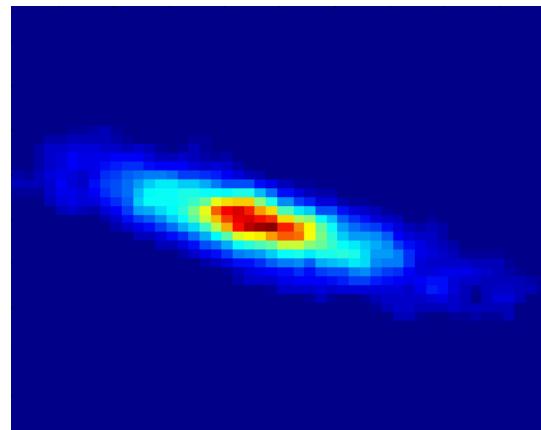
Early



Class:

- Stages of Formation

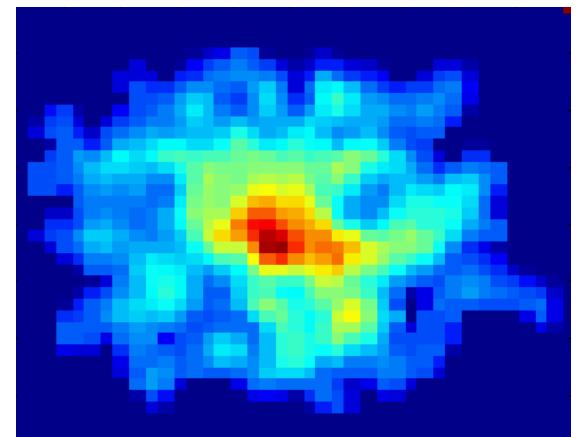
Intermediate



Attributes:

- Image features,
- Characteristics of light waves received, etc.

Late



Data Size:

- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

Regression

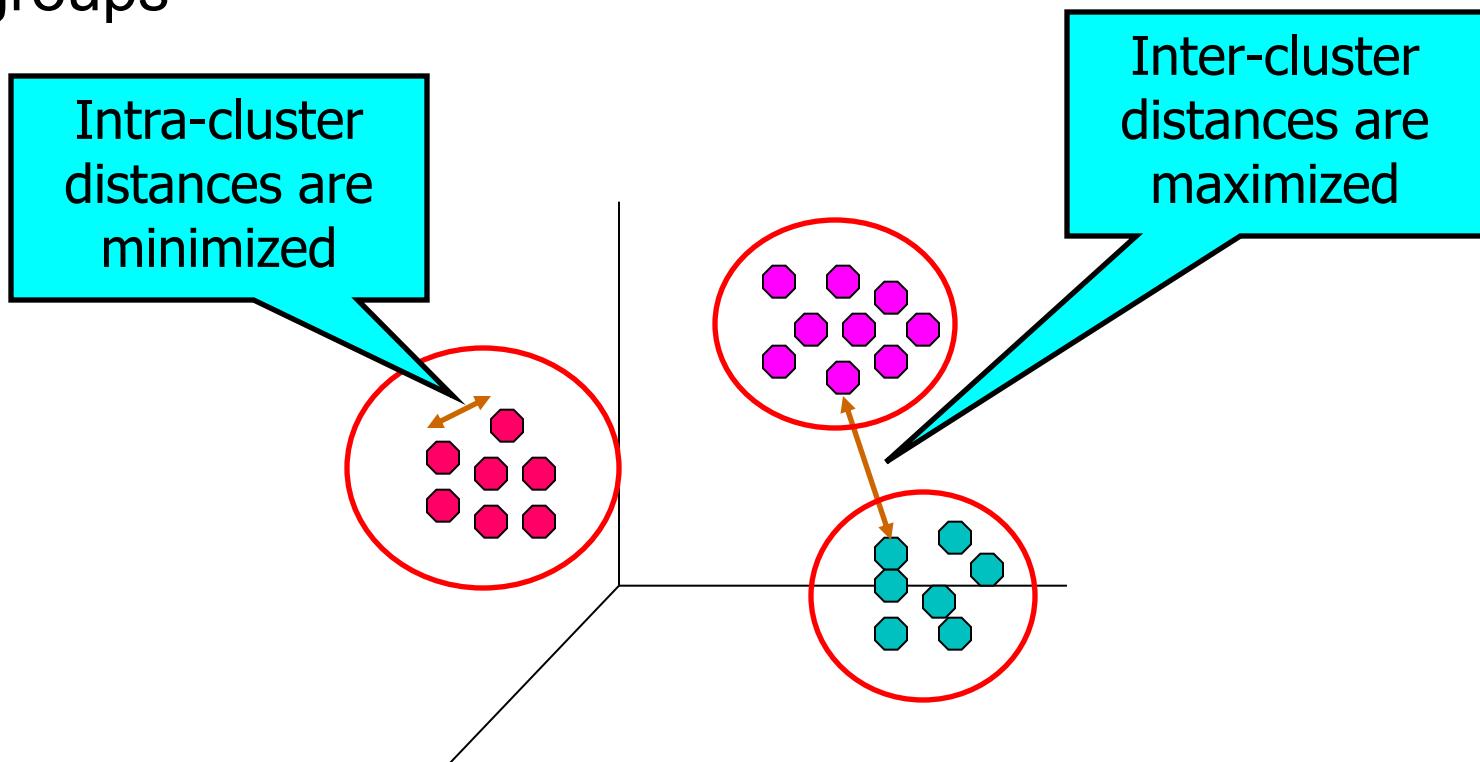
- Predict a value of a given continuous valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.
- Extensively studied in statistics, neural network fields.
- Examples:
 - Predicting sales amounts of new product based on advertising expenditure.
 - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
 - Time series prediction of stock market indices.

Data Mining Tasks: (2) Cluster Analysis

- Unsupervised learning (i.e., Class label is unknown)
- Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
- Principle: **Minimizing intra-class similarity maximizing interclass similarity**
- Many methods and applications

Clustering

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



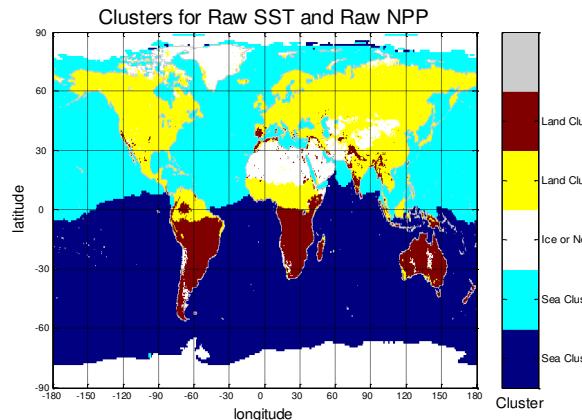
Applications of Cluster Analysis

● Understanding

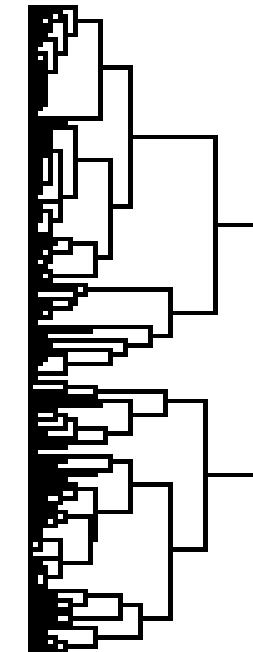
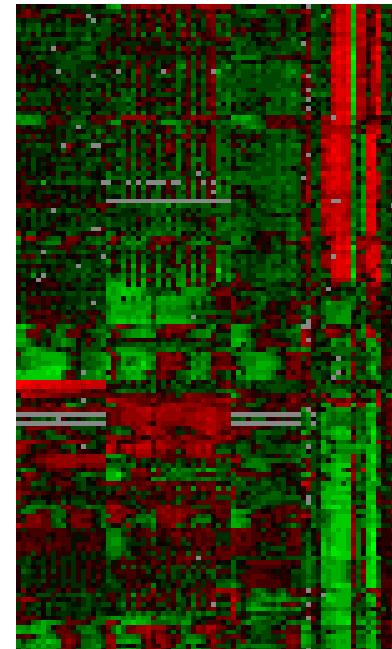
- Custom profiling for targeted marketing
- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations

● Summarization

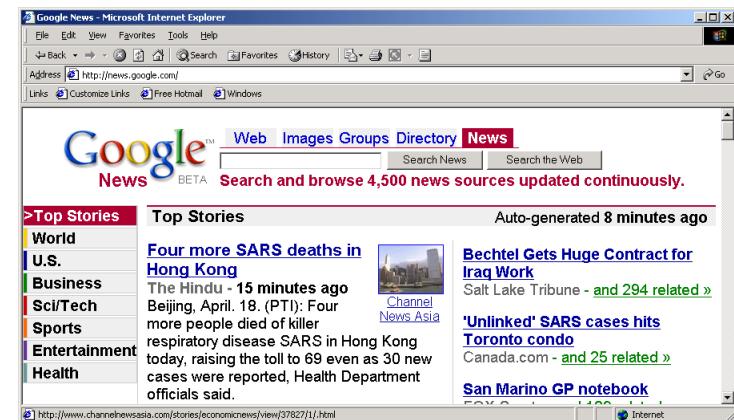
- Reduce the size of large data sets



Use of K-means to partition Sea Surface Temperature (SST) and Net Primary Production (NPP) into clusters that reflect the Northern and Southern Hemispheres.



Courtesy: Michael Eisen



Clustering: Application 1

- Market Segmentation:

- **Goal:** subdivide a market into distinct subsets of customers where any subset may conceivably be selected as a market target to be reached with a distinct marketing mix.
- **Approach:**
 - ◆ Collect different attributes of customers based on their geographical and lifestyle related information.
 - ◆ Find clusters of similar customers.
 - ◆ Measure the clustering quality by observing buying patterns of customers in same cluster vs. those from different clusters.

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.

Enron email dataset



Data Mining Tasks: (3) Association Rules

- Frequent patterns (or frequent itemsets)
 - What items are frequently purchased together in your Walmart?
- Association, correlation vs. causality
 - A typical association rule
 - ◆ Milk → Coke [0.5%, 75%] (support, confidence)
 - Are strongly associated items also strongly correlated?
- How to mine such patterns and rules efficiently in large datasets?
- How to use such patterns for classification, clustering, and other applications?

Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection
 - 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

Rules Discovered:

$\{\text{Milk}\} \rightarrow \{\text{Coke}\}$

$\{\text{Diaper}, \text{Milk}\} \rightarrow \{\text{Beer}\}$

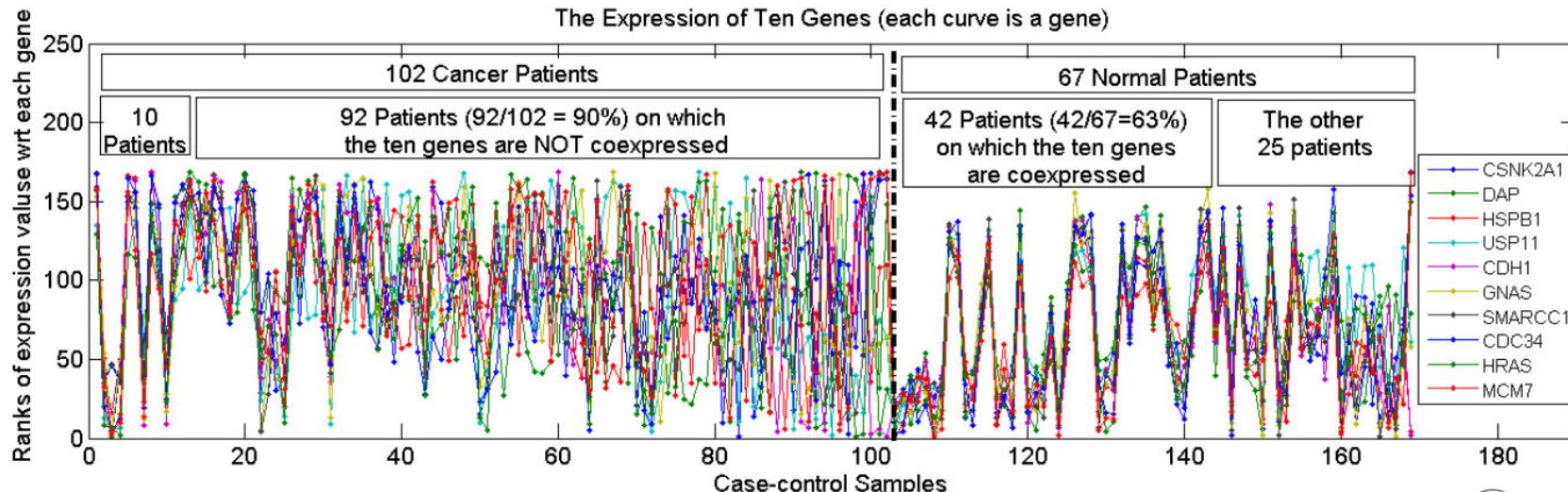
Association Analysis: Applications

- Market-basket analysis
 - Rules are used for sales promotion, shelf management, and inventory management
- Telecommunication alarm diagnosis
 - Rules are used to find combination of alarms that occur together frequently in the same time period
- Medical Informatics
 - Rules are used to find combination of patient symptoms and test results associated with certain diseases

Association Analysis: Applications

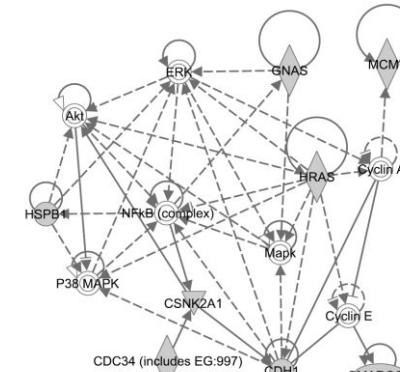
- An Example Subspace Differential Coexpression Pattern from lung cancer dataset

Three lung cancer datasets [Bhattacharjee et al. 2001], [Stearman et al. 2005], [Su et al. 2007]



Enriched with the TNF/NFB signaling pathway
which is well-known to be related to lung cancer
P-value: 1.4×10^{-5} (6/10 overlap with the pathway)

[Fang et al PSB 2010]

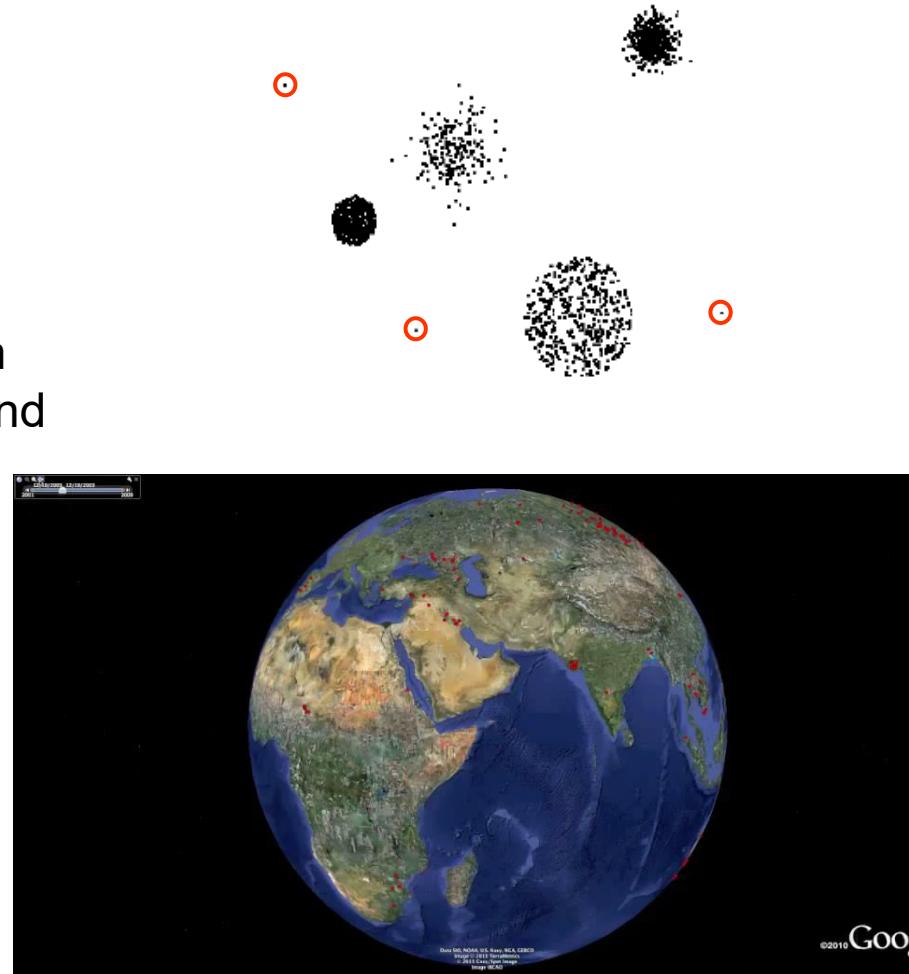
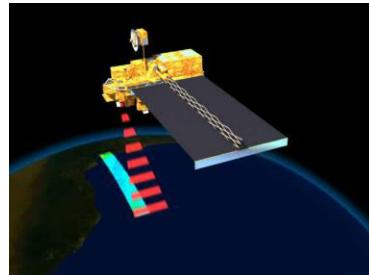


Data Mining Tasks: (4) Anomaly Detection

- Anomaly Detection (or Outlier analysis)
 - Outlier: A data object that does not comply with the general behavior of the data
 - Noise or exception? — One person's garbage could be another person's treasure
 - Methods: by product of clustering or regression analysis, ...
 - Useful in fraud detection, rare events analysis

Deviation/Anomaly/Change Detection

- Detect significant deviations from normal behavior
- Applications:
 - Credit Card Fraud Detection
 - Network Intrusion Detection
 - Identify anomalous behavior from sensor networks for monitoring and surveillance.
 - Detecting changes in the global forest cover.



DM Tasks: Structure and Network Analysis

- **Graph mining**
 - Finding frequent subgraphs (e.g., chemical compounds), trees (XML), substructures (web fragments, communities, etc)
- **Information network analysis**
 - Social networks: actors (objects, nodes) and relationships (edges)
 - ◆ e.g., author networks in CS, terrorist networks
 - Multiple heterogeneous networks
 - ◆ A person could be multiple information networks: friends, family, classmates, ...
 - Links carry a lot of semantic information: Link mining
- **Web mining**
 - Web is a big information network: from PageRank to Google
 - Analysis of Web information networks
 - ◆ Web community discovery, opinion mining, usage mining, ...

Motivating Challenges and Issues

- Scalability
 - Efficiency and scalability of data mining algorithms
- High Dimensionality
 - Mining knowledge in multi-dimensional space
- Heterogeneous and Complex Data
 - Mining various and new kinds of knowledge
- Data Ownership and Distribution
 - Social impacts of data mining
 - Privacy-preserving data mining