

Data Mining Classification: Basic Concepts, Decision Trees, and Model Evaluation

Lecture Notes for Chapter 4

Introduction to Data Mining

by

Tan, Steinbach, Kumar



Classification: Definition

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



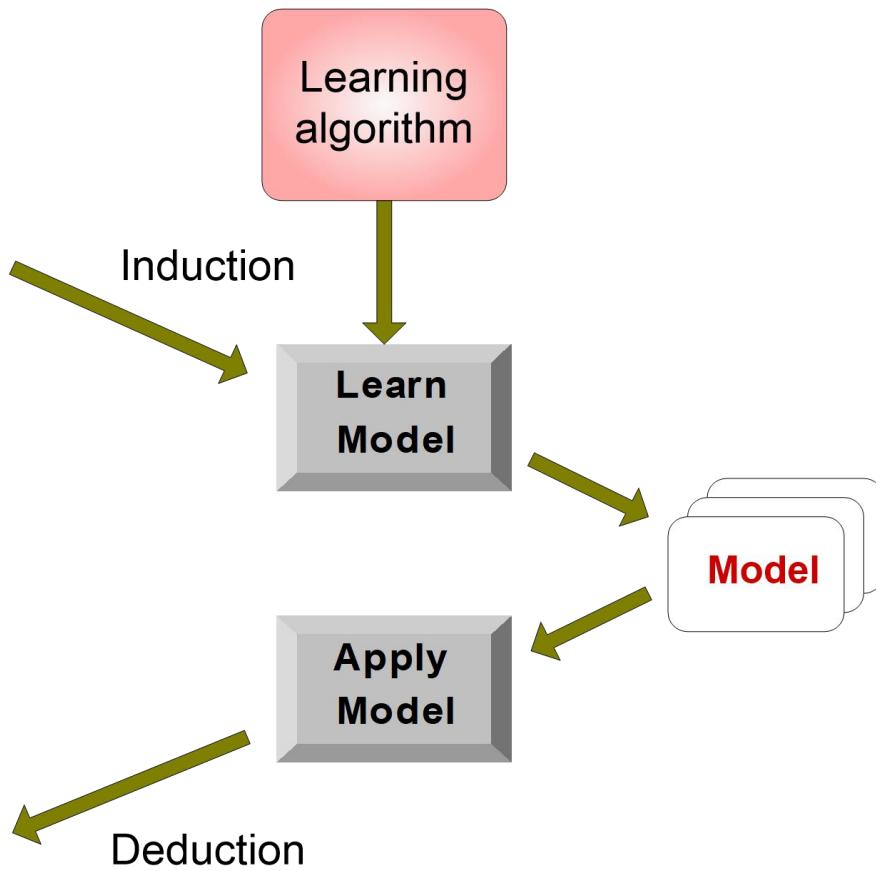
Illustrating Classification Task

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

Training Set

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

Test Set



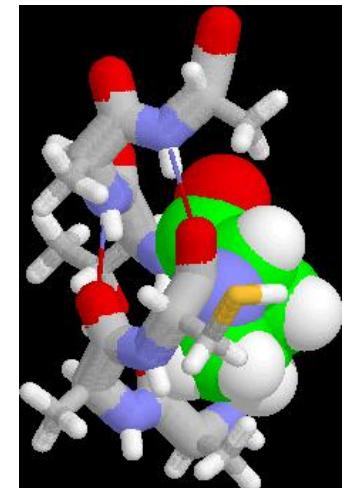
PRESIDENCY
UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013



Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc



Classification Techniques

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



**PRESIDENCY
UNIVERSITY**

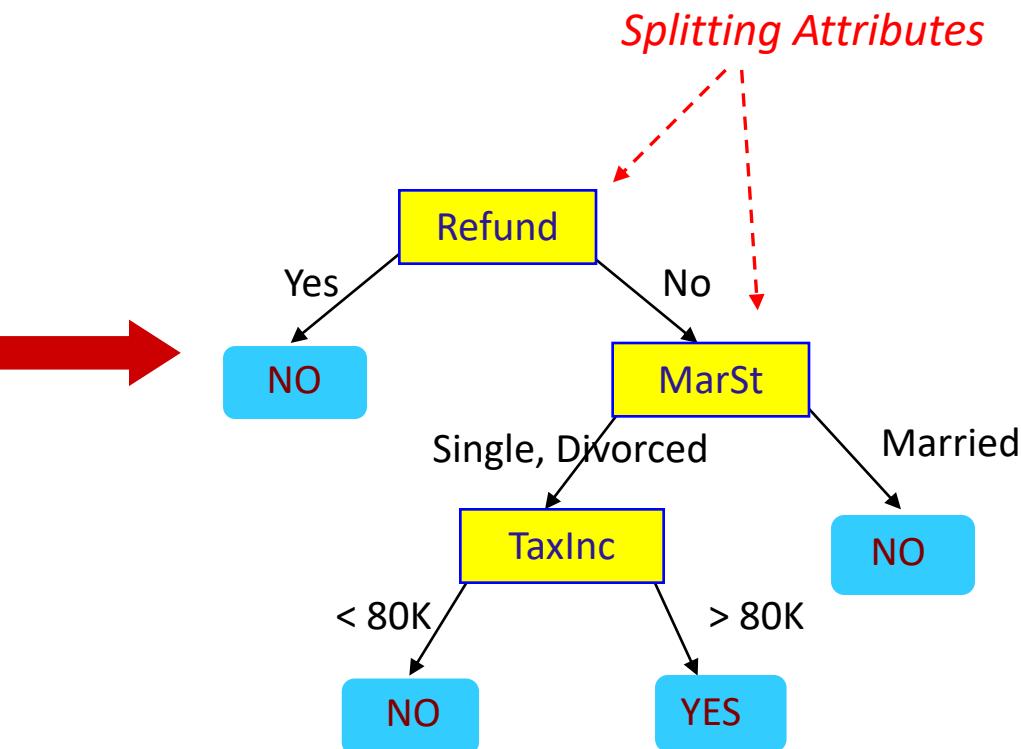
Private University Estd. in Karnataka State by Act No. 41 of 2013



Example of a Decision Tree

categorical
categorical
continuous
class

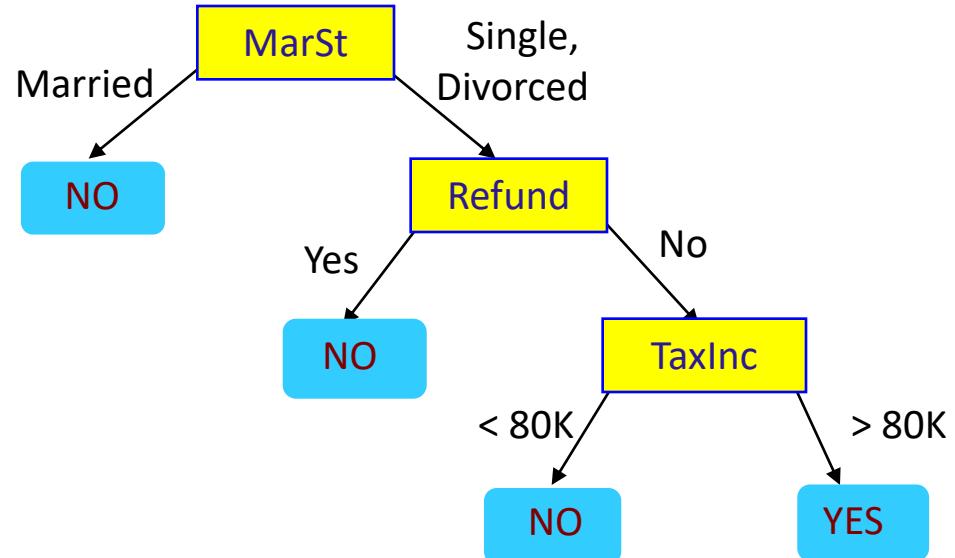
| Tid | Refund | Marital Status | Taxable Income | Cheat |
|-----|--------|----------------|----------------|-------|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married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



Model: Decision Tree

Another Example of Decision Tree

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



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

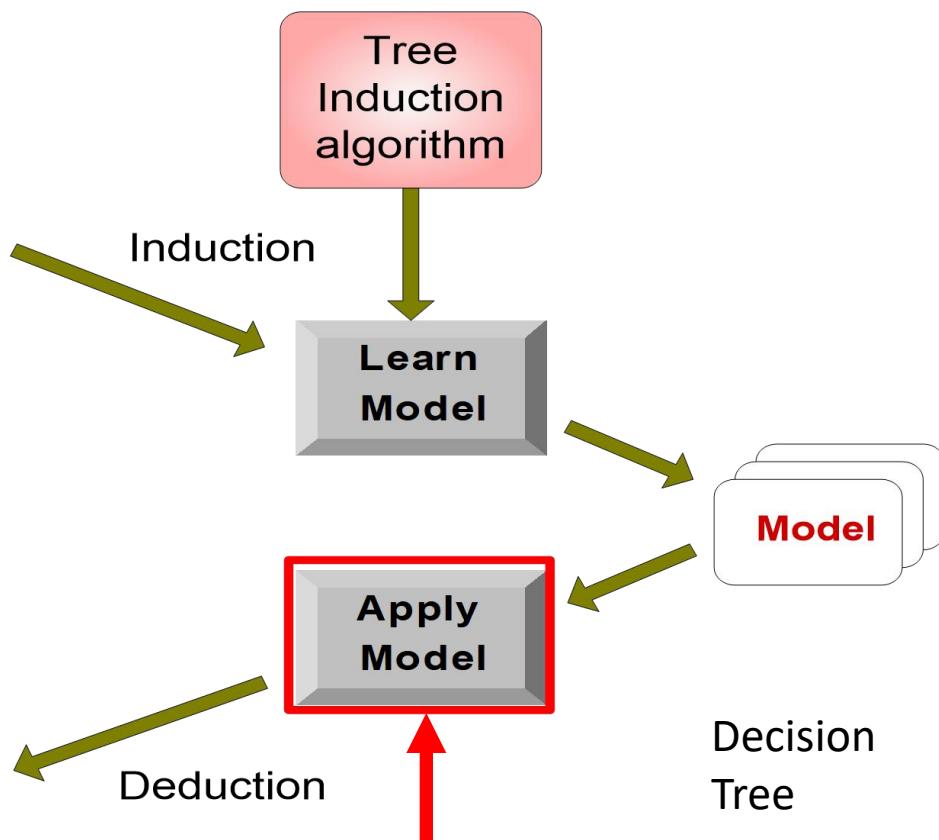
Decision Tree Classification Task

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

Training Set

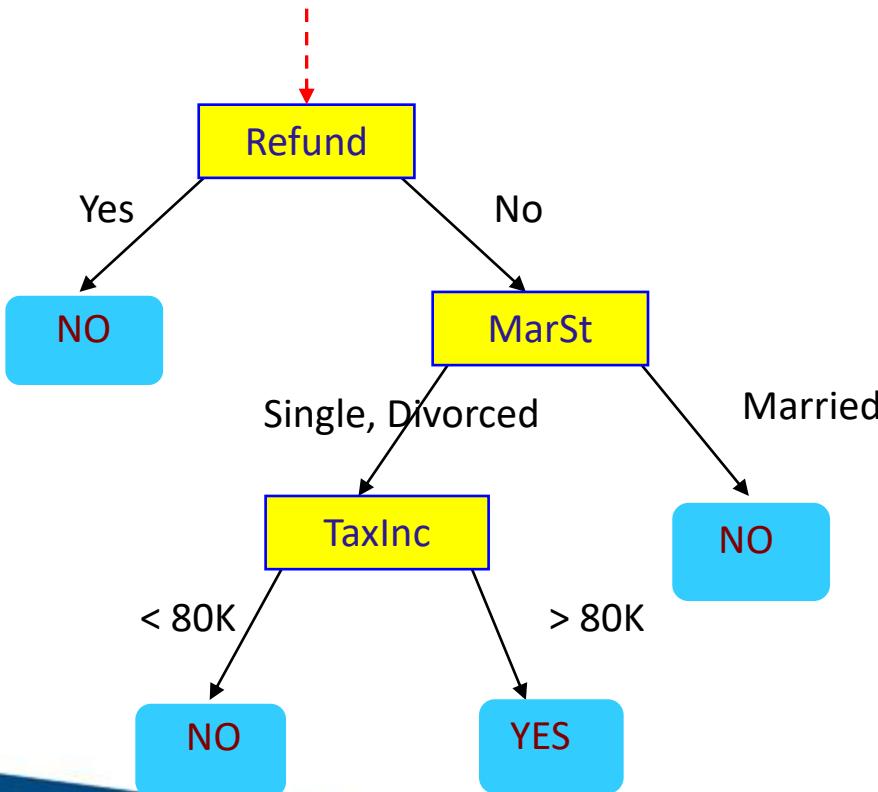
| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|-----|---------|---------|---------|-------|
| 11 | No | Small | 55K | ? |
| 12 | Yes | Medium | 80K | ? |
| 13 | Yes | Large | 110K | ? |
| 14 | No | Small | 95K | ? |
| 15 | No | Large | 67K | ? |

Test Set



Apply Model to Test Data

Start from the root of tree.



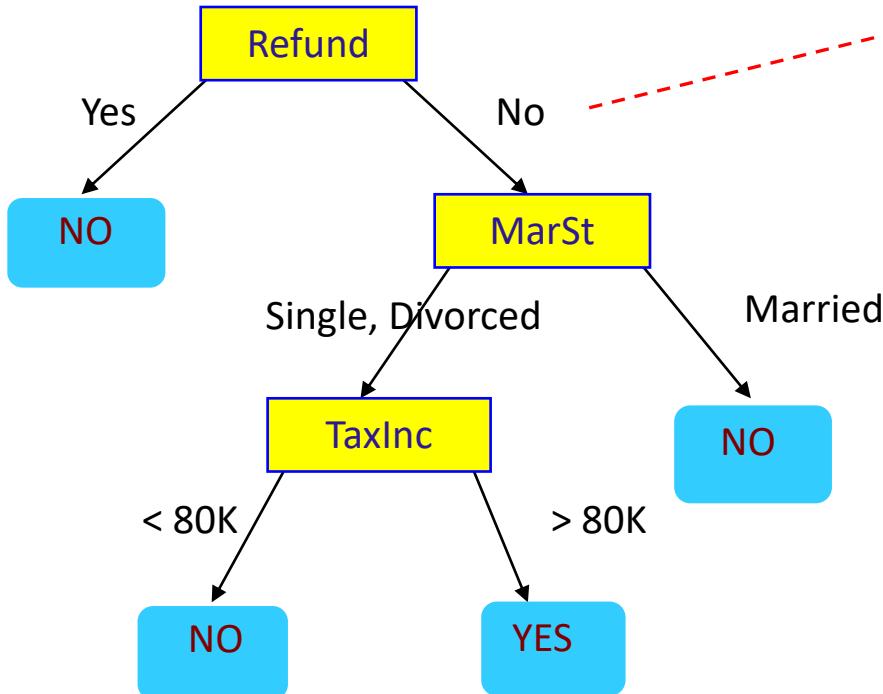
Test Data

| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |

Apply Model to Test Data

Test Data

| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



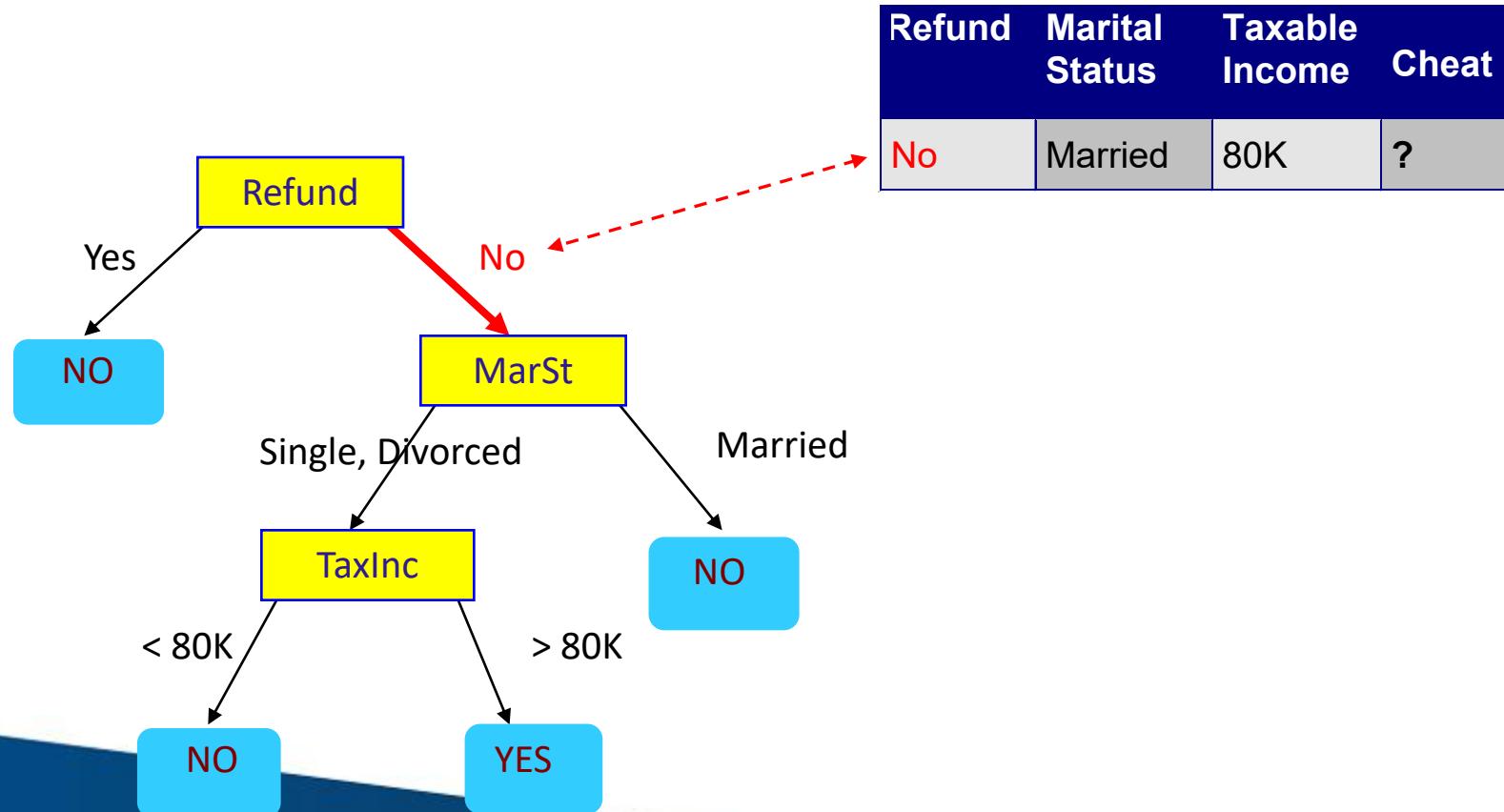
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Apply Model to Test Data

Test Data



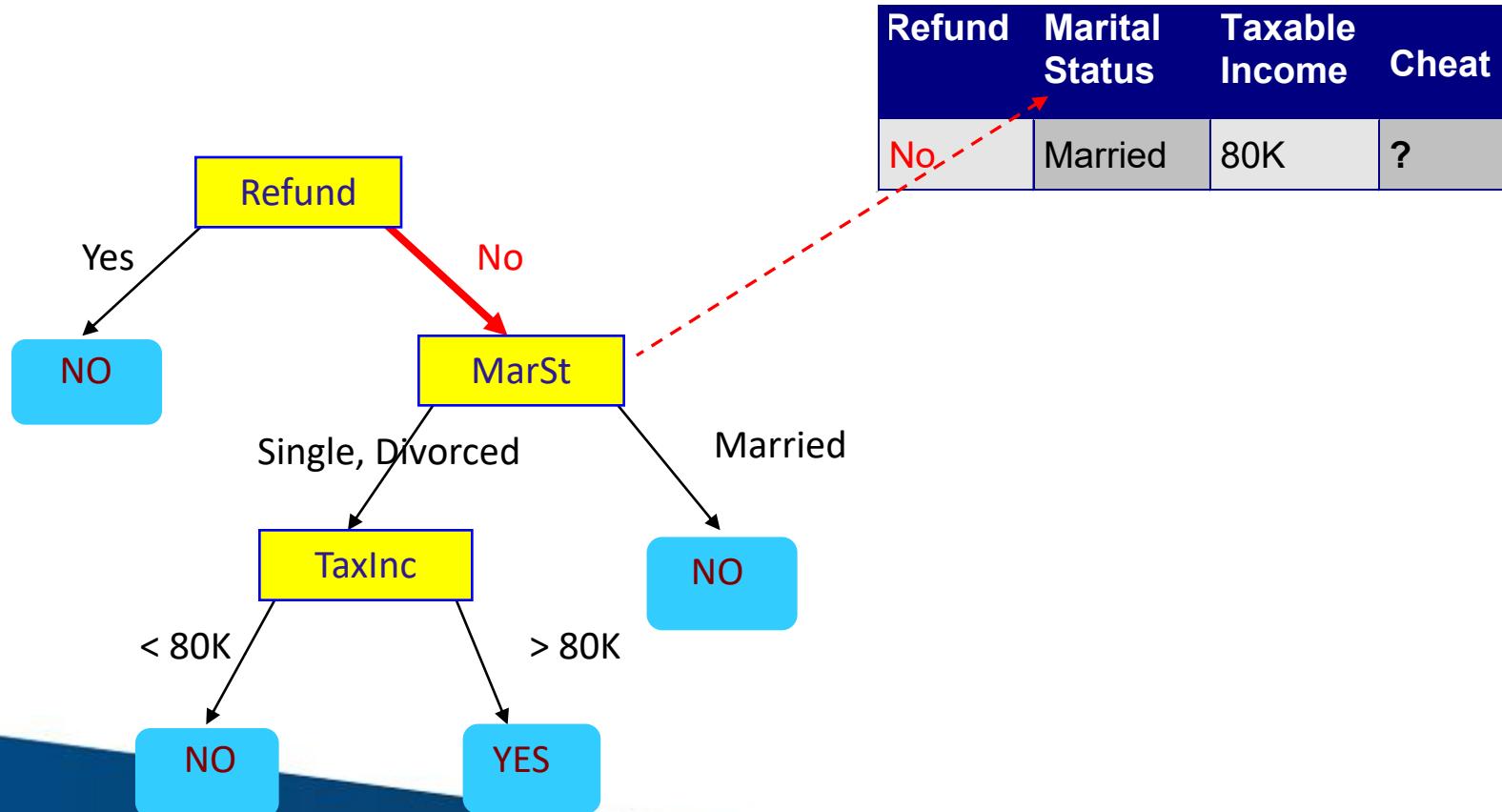
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Apply Model to Test Data

Test Data



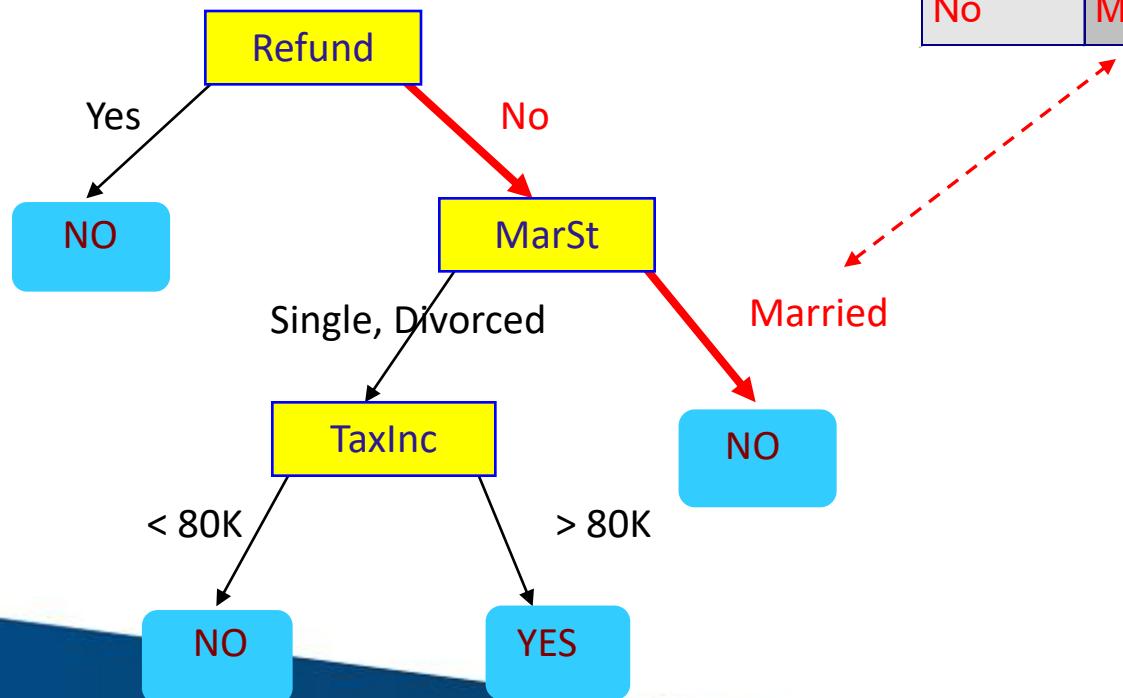
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Apply Model to Test Data

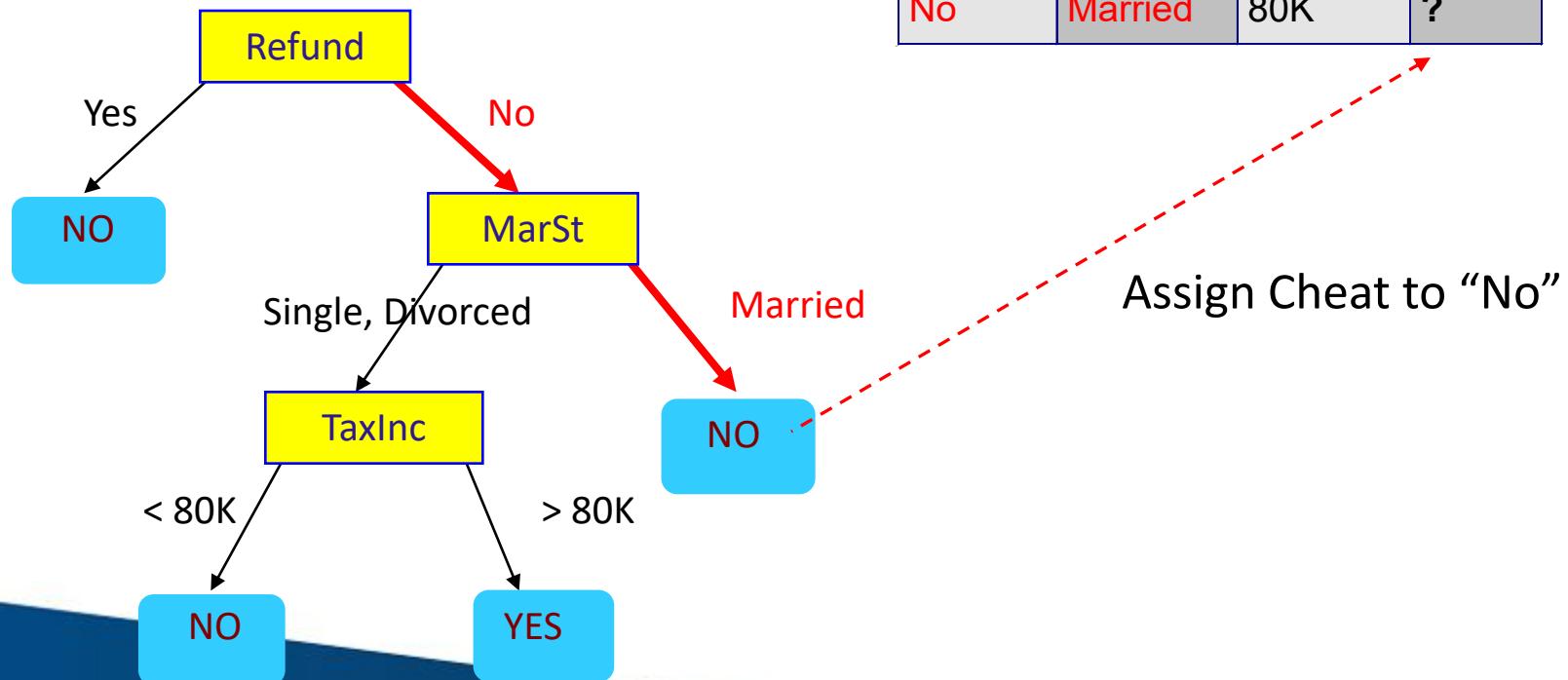
Test Data



| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |

Apply Model to Test Data

Test Data



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



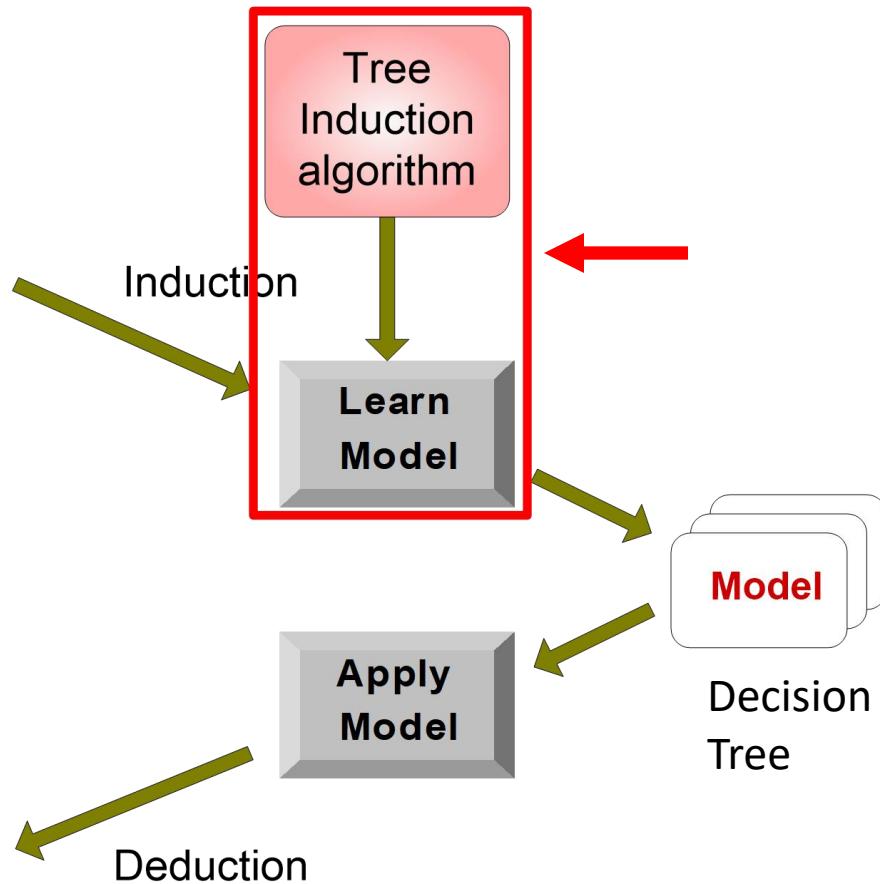
Decision Tree Classification Task

| 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



**PRESIDENCY
UNIVERSITY**

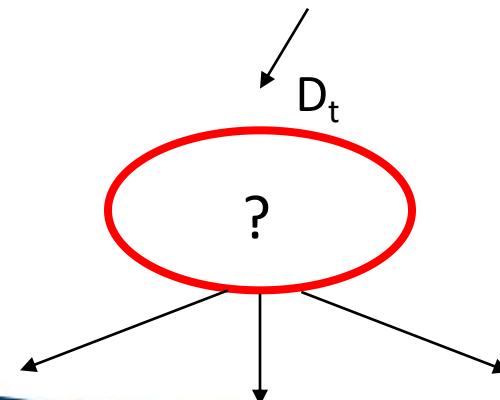
Private University Estd. in Karnataka State by Act No. 41 of 2013



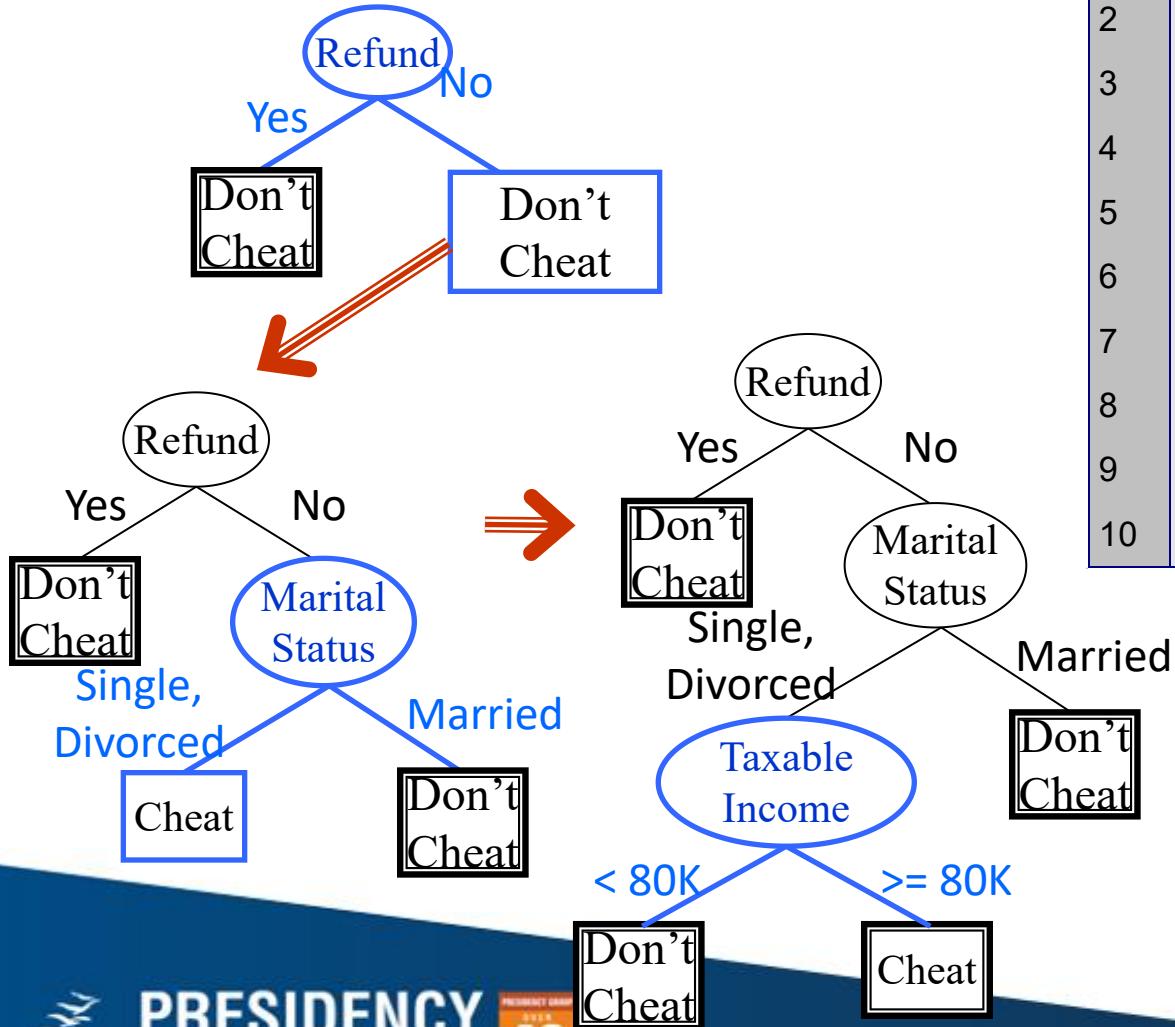
General Structure of Hunt's Algorithm

- 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.

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



Hunt's Algorithm



| Tid | Refund | Marital Status | Taxable Income | Cheat |
|-----|--------|----------------|----------------|-------|
| 1 | Yes | Single | 125K | No |
| 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 |



Tree Induction

- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - How to specify the attribute test condition?
 - How to determine the best split?
 - Determine when to stop splitting



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Tree Induction

- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - How to specify the attribute test condition?
 - How to determine the best split?
 - Determine when to stop splitting



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

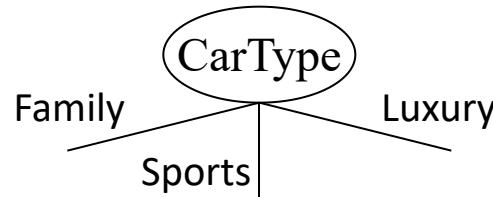


How to Specify Test Condition?

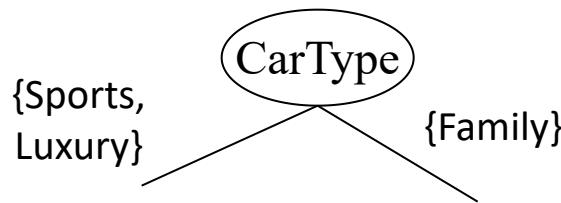
- Depends on attribute types
 - Nominal
 - Ordinal
 - Continuous
- Depends on number of ways to split
 - 2-way split
 - Multi-way split

Splitting Based on Nominal Attributes

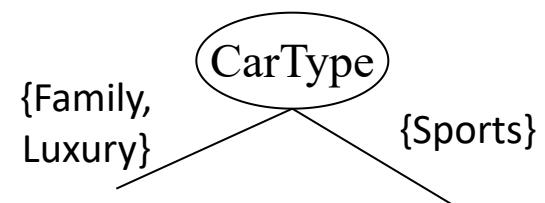
- **Multi-way split:** Use as many partitions as distinct values.



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

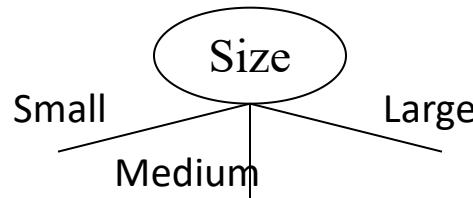


OR

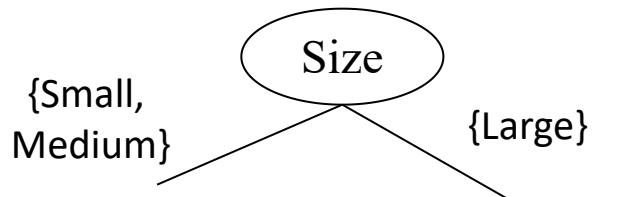


Splitting Based on Ordinal Attributes

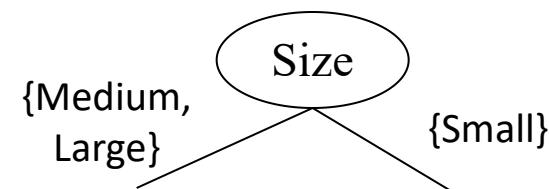
- **Multi-way split:** Use as many partitions as distinct values.



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



OR



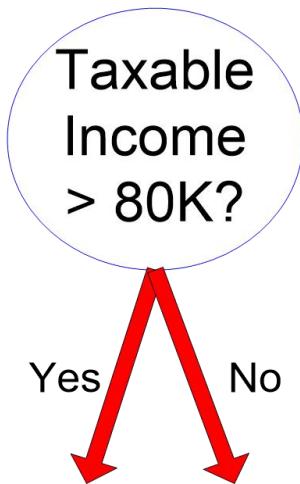
- What about this split?



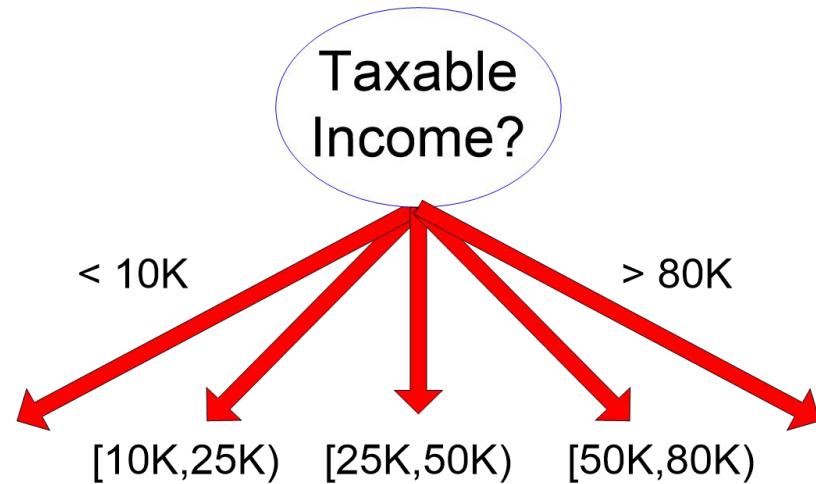
Splitting Based on Continuous Attributes

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

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

Tree Induction

- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - How to specify the attribute test condition?
 - **How to determine the best split?**
 - Determine when to stop splitting



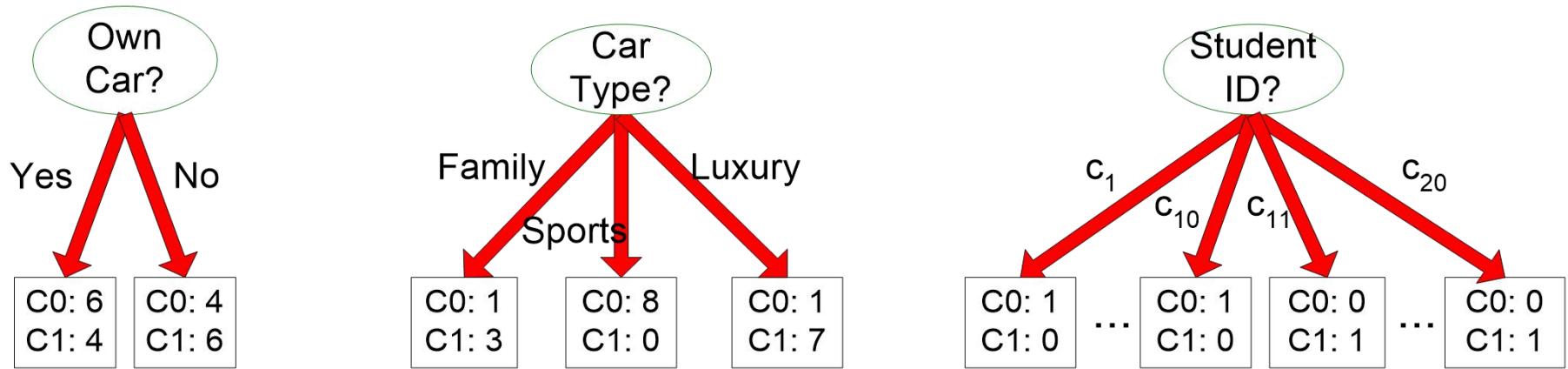
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to determine the Best Split

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



Which test condition is the best?

How to determine the Best Split

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

C0: 5
C1: 5

Non-homogeneous,
High degree of impurity

C0: 9
C1: 1

Homogeneous,
Low degree of impurity



Measures of Node Impurity

- Gini Index

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

- Entropy

$$Entropy(t) = -\sum_j p(j | t) \log_2 p(j | t)$$

- Misclassification error

$$Error(t) = 1 - \max_j P(j | t)$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

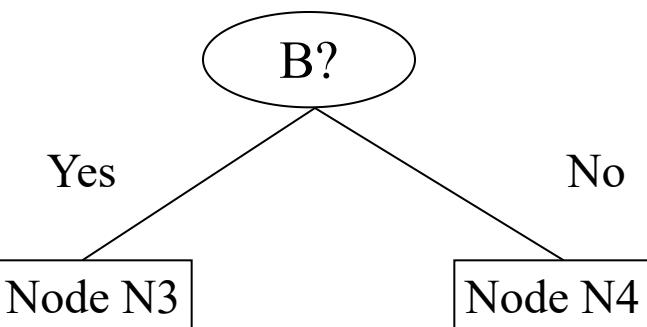
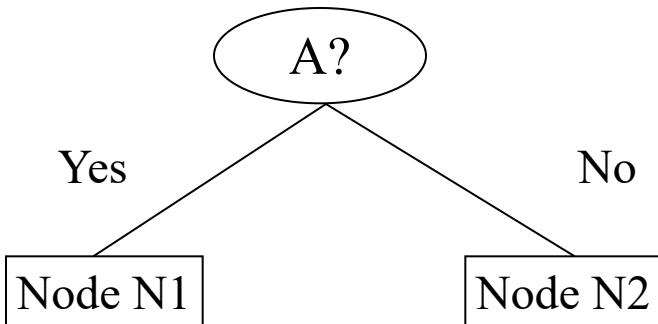


How to Find the Best Split

Before Splitting:

| | |
|----|------------|
| C0 | N00 |
| C1 | N01 |

→ M0

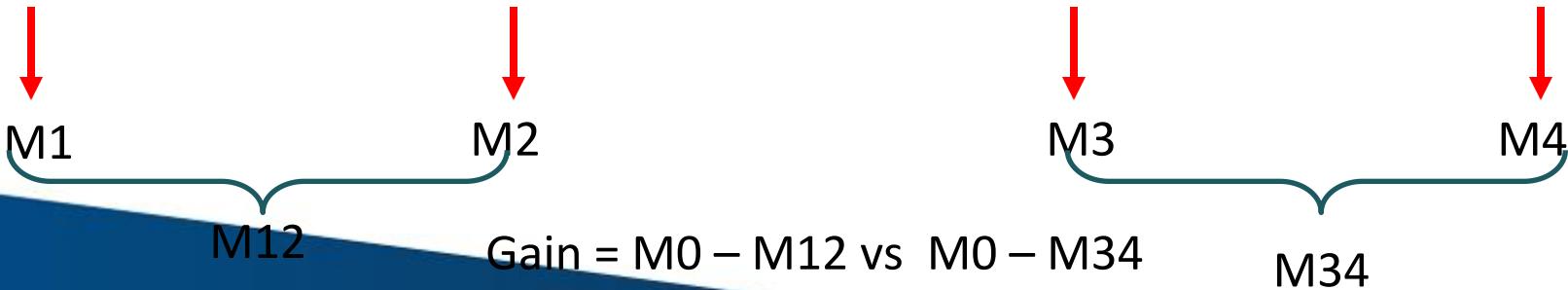


| | |
|----|------------|
| C0 | N10 |
| C1 | N11 |

| | |
|----|------------|
| C0 | N20 |
| C1 | N21 |

| | |
|----|------------|
| C0 | N30 |
| C1 | N31 |

| | |
|----|------------|
| C0 | N40 |
| C1 | N41 |



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

| | |
|-------------------|----------|
| 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 | |



Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

| | |
|----|----------|
| 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 |

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

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

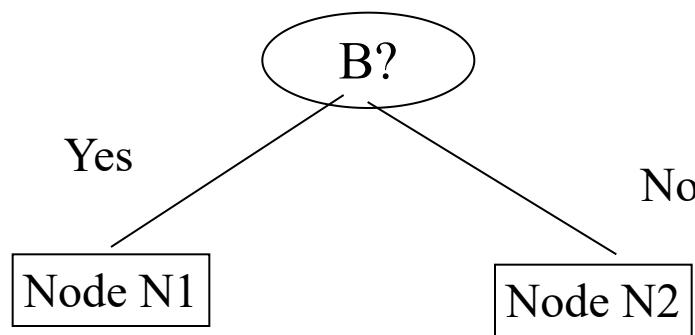
$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

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

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
 - Larger and Purer Partitions are sought for.

$$\begin{aligned} \text{Gini}(N_1) &= 1 - (5/7)^2 - (2/7)^2 \\ &= 0.408 \\ \text{Gini}(N_2) &= 1 - (1/5)^2 - (4/5)^2 \\ &= 0.32 \end{aligned}$$



| | N1 | N2 |
|-------------------|----|----|
| C1 | 5 | 1 |
| C2 | 2 | 4 |
| Gini=0.396 | | |

| | Parent |
|---------------------|--------|
| C1 | 6 |
| C2 | 6 |
| Gini = 0.500 | |

$$\begin{aligned} \text{Gini(Children)} &= 7/12 * 0.408 + \\ &\quad 5/12 * 0.32 \\ &= 0.396 \end{aligned}$$

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 | 2 | 1 |
| C2 | 4 | 1 | 1 |
| Gini | 0.393 | | |

Two-way split
(find best partition of values)

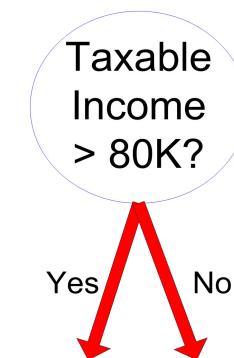
| | CarType | |
|------|------------------|----------|
| | {Sports, Luxury} | {Family} |
| C1 | 3 | 1 |
| C2 | 2 | 4 |
| Gini | 0.400 | |

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

Continuous Attributes: 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, $A < v$ and $A \geq v$
- 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.

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



Continuous Attributes: 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 | No |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Taxable Income | | | | | | | | | | | |
| Sorted Values | 60 | 70 | 75 | 85 | 90 | 95 | 100 | 120 | 125 | 172 | 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 | 3 0 |
| No | 0 7 | 1 6 | 2 5 | 3 4 | 3 4 | 3 4 | 3 4 | 4 3 | 5 2 | 6 1 | 7 0 |
| Gini | 0.420 | 0.400 | 0.375 | 0.343 | 0.417 | 0.400 | 0.300 | 0.343 | 0.375 | 0.400 | 0.420 |



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Ex 1: Decision Tree Induction to Predict the Risk Class of a bank loan Applicant using Gini Index

| Owes Home? | Married? | Gender? | Employed? | Credit Rating? | Risk Class |
|------------|----------|---------|-----------|----------------|------------|
| Yes | Yes | Male | Yes | A | B |
| No | No | Female | Yes | A | A |
| Yes | Yes | Female | Yes | B | C |
| Yes | No | Male | No | B | B |
| No | Yes | Female | Yes | B | C |
| No | No | Female | Yes | B | A |
| No | No | Male | No | B | B |
| Yes | No | Female | Yes | A | A |
| No | Yes | Female | Yes | A | C |
| Yes | Yes | Female | Yes | A | C |

Ex 1: Decision Tree Induction to Predict the Risk Class of a bank loan Applicant using Gini Index ...contn..

- Steps to be done on board
- Refer 5.3 – Example using Gini Index



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
 - Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
 - Minimum (0.0) when all records belong to one class, implying most information



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j | t) \log_2 p(j | t)$$

| | |
|----|----------|
| 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 |

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

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

| | |
|----|----------|
| C1 | 2 |
| C2 | 4 |

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

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Splitting Based on INFO...

- Information Gain:

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

Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.



ID3 – Iterative Dichotomizer

- Invented by J. Ross Quinlan
- Employs a top-down greedy search through the space of possible decision trees.

Greedy because there is no backtracking. It picks highest values first.

- Select attribute that is most useful for classifying examples (attribute that has the highest Information Gain).

Entropy

- Entropy measures the impurity of an arbitrary collection of examples.
- For a collection S, entropy is given as:

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

- For a collection S having positive and negative examples

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

where p_+ is the proportion of positive examples

and p_- is the proportion of negative examples

In general, $Entropy(S) = 0$ if all members of S belong to the same class.

$Entropy(S) = 1$ (maximum) when all members are split equally.

Information Gain

- Measures the expected reduction in entropy. The higher the IG, more is the expected reduction in entropy.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Values(A)$ is the set of all possible values for attribute A,

S_v is the subset of S for which attribute A has value v.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Example : Determine whether an animal lays eggs using entropy

| Independent/Condition attributes | | | | | Dependent/ Decision attributes |
|----------------------------------|--------------|----------|-----|-------|--------------------------------------|
| Animal | Warm-blooded | Feathers | Fur | Swims | Lays Eggs |
| Ostrich | Yes | Yes | No | No | Yes |
| Crocodile | No | No | No | Yes | Yes |
| Raven | Yes | Yes | No | No | Yes |
| Albatross | Yes | Yes | No | No | Yes |
| Dolphin | Yes | No | No | Yes | No |
| Koala | Yes | No | Yes | No | No |

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

$$\begin{aligned} \text{Entropy(4Y,2N)} &: -(4/6)\log_2(4/6) - (2/6)\log_2(2/6) \\ &= 0.91829 \end{aligned}$$

Now, we have to find the IG for all four attributes
Warm-blooded, Feathers, Fur, Swims

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

For attribute ‘Warm-blooded’:

Values(Warm-blooded) : [Yes, No]

S = [5Y, 1N]

S_{Yes} = [3Y, 2N] E(S_{Yes}) = 0.97095

S_{No} = [1Y, 0N] E(S_{No}) = 0 (all members belong to same class)

$$\begin{aligned} Gain(S, \text{Warm-blooded}) &= 0.91829 - [(5/6)*0.97095 + (1/6)*0] \\ &= 0.10916 \end{aligned}$$

For attribute ‘Feathers’:

Values(Feathers) : [Yes, No]

S = [4Y, 2N]

S_{Yes} = [3Y, 0N] E(S_{Yes}) = 0

S_{No} = [1Y, 2N] E(S_{No}) = 0.91829

$$Gain(S, \text{Feathers}) = 0.91829 - [(3/6)*0 + (3/6)*0.91829]$$

$$= 0.45914$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

For attribute ‘Fur’:

Values(Fur) : [Yes, No]

S = [5Y, 1N]

S_{Yes} = [0Y, 1N] E(S_{Yes}) = 0

S_{No} = [4Y, 1N] E(S_{No}) = 0.7219

$$\begin{aligned} Gain(S, \text{Fur}) &= 0.91829 - [(1/6)*0 + (5/6)*0.7219] \\ &= 0.3167 \end{aligned}$$

For attribute ‘Swims’:

Values(Swims) : [Yes, No]

S = [4Y, 2N]

S_{Yes} = [1Y, 1N] E(S_{Yes}) = 1 (equal members in both classes)

S_{No} = [3Y, 1N] E(S_{No}) = 0.81127

$$\begin{aligned} Gain(S, \text{Swims}) &= 0.91829 - [(2/6)*1 + (4/6)*0.81127] \\ &= 0.04411 \end{aligned}$$



$\text{Gain}(S, \text{Warm-blooded}) = 0.10916$

$\text{Gain}(S, \text{Feathers}) = 0.45914$

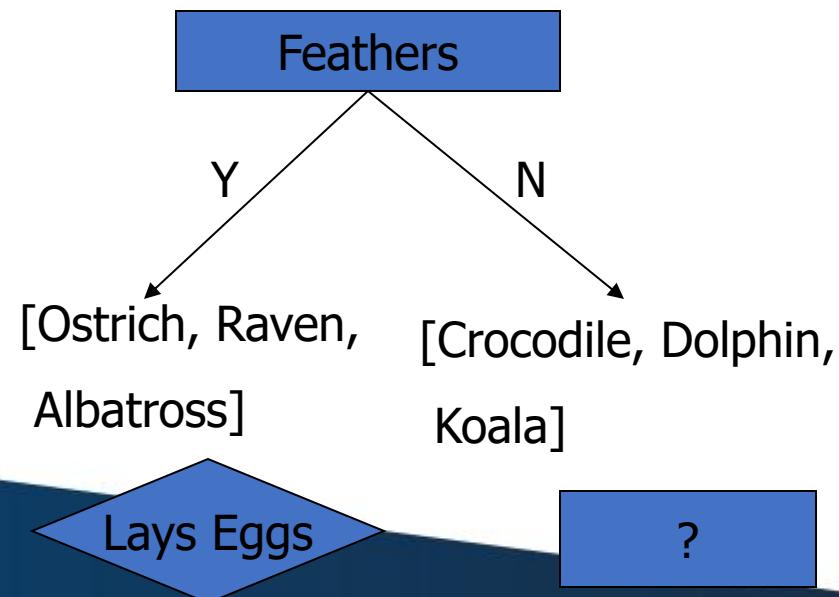
$\text{Gain}(S, \text{Fur}) = 0.31670$

$\text{Gain}(S, \text{Swims}) = 0.04411$

$\text{Gain}(S, \text{Feathers})$ is maximum, so it is considered as the root node

| Anim al | War m-blood ed | Feath ers | Fur | Swim s | Lays Eggs |
|-----------|----------------|-----------|-----|--------|-----------|
| Ostrich | Yes | Yes | No | No | Yes |
| Crocodile | No | No | No | Yes | Yes |
| Raven | Yes | Yes | No | No | Yes |
| Albatross | Yes | Yes | No | No | Yes |
| Dolphin | Yes | No | No | Yes | No |
| Koala | Yes | No | Yes | No | No |

The 'Y' descendant has only positive examples and becomes the leaf node with classification 'Lays Eggs'



| Animal | Warm-blooded | Feathers | Fur | Swims | Lays Eggs |
|---------------|---------------------|-----------------|------------|--------------|------------------|
| Crocodile | No | No | No | Yes | Yes |
| Dolphin | Yes | No | No | Yes | No |
| Koala | Yes | No | Yes | No | No |

We now repeat the procedure,

S: [Crocodile, Dolphin, Koala]

S: [1+,2-]

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

$$\begin{aligned} Entropy(S) &= -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) \\ &= 0.91829 \end{aligned}$$

- For attribute ‘Warm-blooded’:

Values(Warm-blooded) : [Yes,No]

S = [1Y,2N]

$S_{\text{Yes}} = [0\text{Y}, 2\text{N}]$ $E(S_{\text{Yes}}) = 0$

$S_{\text{No}} = [1\text{Y}, 0\text{N}]$ $E(S_{\text{No}}) = 0$

$$\text{Gain}(S, \text{Warm-blooded}) = 0.91829 - [(2/3)*0 + (1/3)*0] = \mathbf{0.91829}$$

- For attribute ‘Fur’:

Values(Fur) : [Yes,No]

S = [1Y,2N]

$S_{\text{Yes}} = [0\text{Y}, 1\text{N}]$ $E(S_{\text{Yes}}) = 0$

$S_{\text{No}} = [1\text{Y}, 1\text{N}]$ $E(S_{\text{No}}) = 1$

$$\text{Gain}(S, \text{Fur}) = 0.91829 - [(1/3)*0 + (2/3)*1] = \mathbf{0.25162}$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



For attribute ‘Swims’:

Values(Swims) : [Yes, No]

$S = [1Y, 2N]$

$S_{Yes} = [1Y, 1N] \quad E(S_{Yes}) = 1$
 $S_{No} = [0Y, 1N] \quad E(S_{No}) = 0$

$$\text{Gain}(S, \text{Swims}) = 0.91829 - [(2/3)*1 + (1/3)*0] = \\ \mathbf{0.25162}$$

Gain(S,Warm-blooded) is maximum

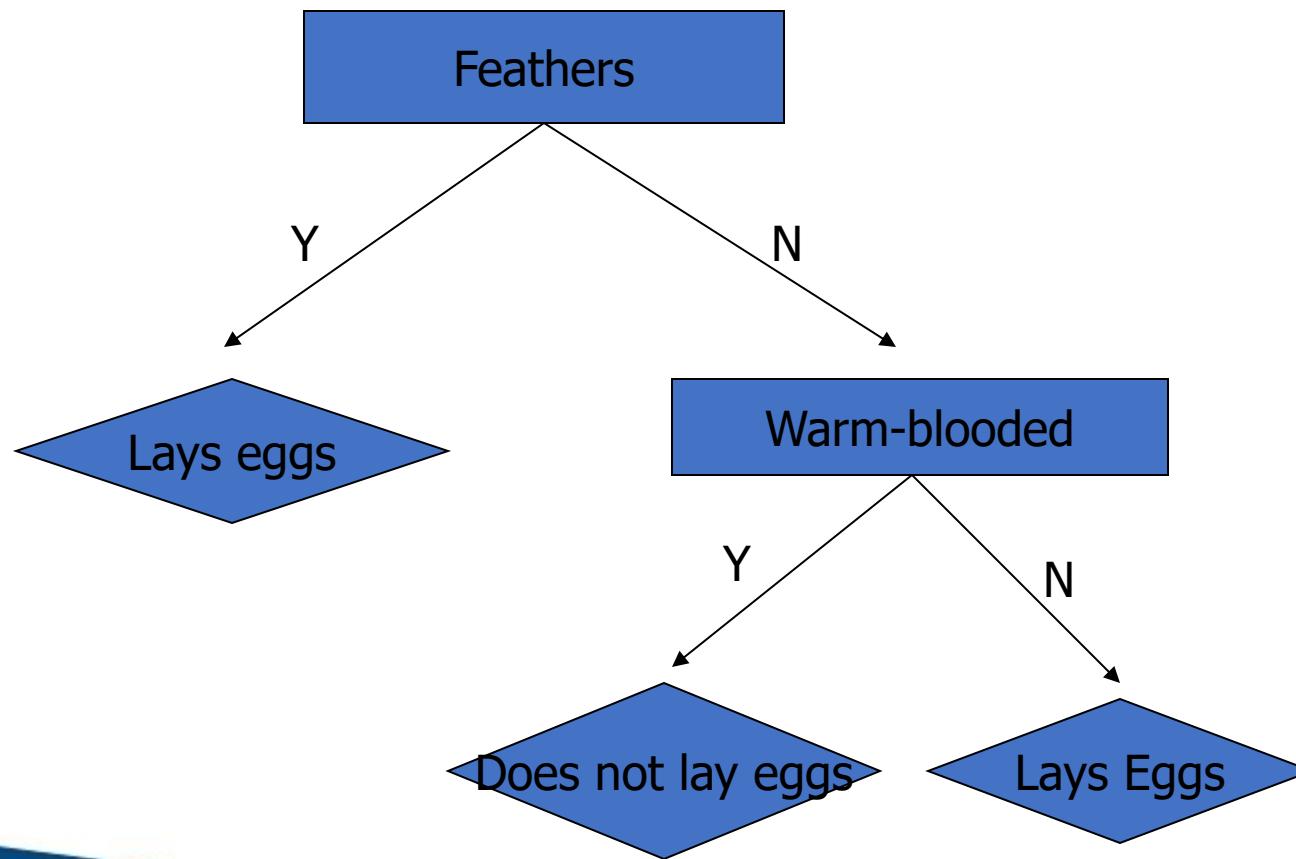


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



The final decision tree will be:



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Splitting Based on INFO...

- **Strategies to overcome the disadvantage of Information Gain**
 - Restrict the test conditions to binary splits only.
 - Used in CART
 - Modify the splitting criteria to take into account the no. of partitions produced by the attribute test condition.
 - Used in C4.5
 - **Gain ratio** determines the goodness of a split.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Splitting Based on INFO...

- Gain Ratio:

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

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

Parent Node, p is split into k partitions
n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- An attribute with more no. of splits, has high SplitINFO, hence less GainRATIO.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Splitting Criteria based on Classification Error

- Classification error at a node t :

$$Error(t) = 1 - \max_i P(i | t)$$

- Measures misclassification error made by a node.
 - Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
 - Minimum (0.0) when all records belong to one class, implying most interesting information

Examples for Computing Error

$$Error(t) = 1 - \max_i P(i | t)$$

| | |
|----|----------|
| 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 |

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

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



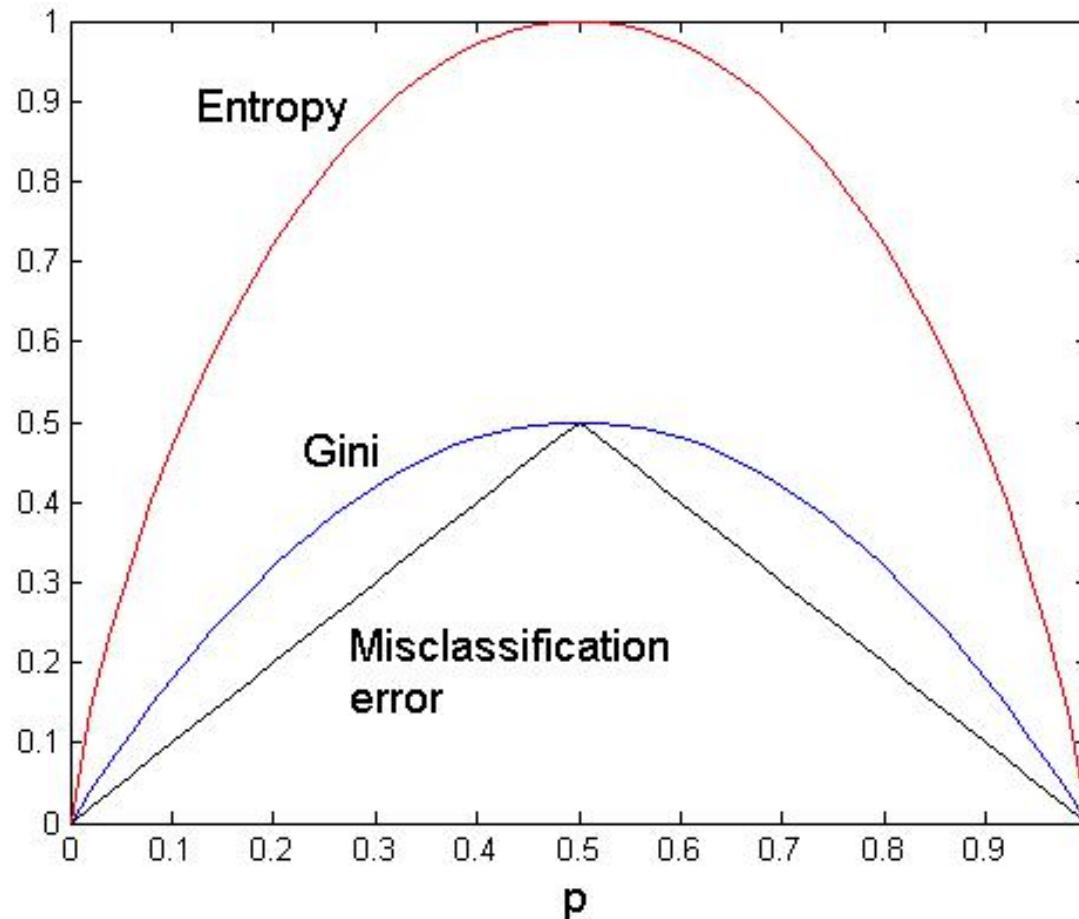
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Comparison among Splitting Criteria

Impurity measures for a 2-class problem:



Tree Induction

- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - How to specify the attribute test condition?
 - How to determine the best split?
 - **Determine when to stop splitting**



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- When the no. of records fall below a certain threshold
 - Top-down recursive partitioning approach
 - No. of records reduces down the tree.
 - Leaf nodes – difficult to make a statistically significant decision – **data fragmentation**
 - To avoid – stop growing the tree when the no. of records fall below a certain threshold.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Decision Tree Based Classification

- Characteristics of DT induction
 - Inexpensive to construct
 - Extremely fast at classifying unknown records
 - Easy to interpret for small-sized trees
 - Accuracy is comparable to other classification techniques for many simple data sets
 - Classifying a new test record is $O(w)$ – w is the maximum depth of the tree.
- Tree – pruning – reduce the size of the tree
- Large DT are susceptible to **overfitting**
- Pruning avoids overfitting and improves generalization capability.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Example: C4.5

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Model Evaluation

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Model Evaluation

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Metrics for Performance Evaluation

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

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

- a: TP (true positive)
- b: FN (false negative)
- c: FP (false positive)
- d: TN (true negative)

Metrics for Performance Evaluation...

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

- Most widely-used metric:

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



Limitation of Accuracy

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Cost in Business Intelligence

- Most business/medical applications use cost + accuracy to evaluate model quality.
- If a model classifies a customer with poor credit as low-risk category, this error is costly.
- Cost matrix shows the cost of misclassifications and benefits of correct classifications.
- Cost matrix helps to choose a model with minimum costly misclassifications.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Cost Matrix

| | | PREDICTED CLASS | | |
|--------------|-----------|-----------------|-----------|----------|
| | | C(i j) | Class=Yes | Class>No |
| ACTUAL CLASS | Class=Yes | C(Yes Yes) | C(No Yes) | |
| | Class>No | C(Yes No) | C(No No) | |

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

Computing Cost of Classification

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

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

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

If M₁ and M₂ are classification models to diagnose cancer patients.

Accuracy = 80%

Accuracy = 90%

Cost = 4255

Some Performance measures for a Classifier

Assume the data set has

P positive instances and

N negative instances

1) True Positive Rate :-

- The proportion of positive instances that are correctly classified as positive.
- TP/P
- Also called hit rate/recall/sensitivity/TP rate



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Some Performance measures for a Classifier

2) False Positive Rate :-

- The proportion of negative instances that are erroneously classified as positive.
- FP/N
- Also called False Alarm Rate/FP rate



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Some Performance measures for a Classifier.....

3) False Negative Rate :-

- The proportion of positive instances that are erroneously classified as negative.
- $1 - \text{True Positive Rate}$
- FN/P
- Also called FN rate

Some Performance measures for a Classifier.....

4) True Negative Rate :-

- The proportion of negative instances that are correctly classified as negative.
- TN/N
- Also called Specificity/TN rate

Some Performance measures for a Classifier.....

5) Precision:-

- The proportion of instances that are classified as positive that are really positive.
- $TP / (TP + FP)$
- Also called Positive Predictive Value .

Some Performance measures for a Classifier.....

6) F1 Score:-

- A measure that combines Precision and Recall.
- $$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Some Performance measures for a Classifier....

7) Accuracy/Predictive Accuracy:-

- The proportion of instances that are correctly classified.
- $(TP + TN) / (P + N)$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Some Performance measures for a Classifier....

8) Error Rate :-

- The proportion of instances that are incorrectly classified.
- $(FP + FN) / (P + N)$

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
 - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
 - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point



**PRESIDENCY
UNIVERSITY**

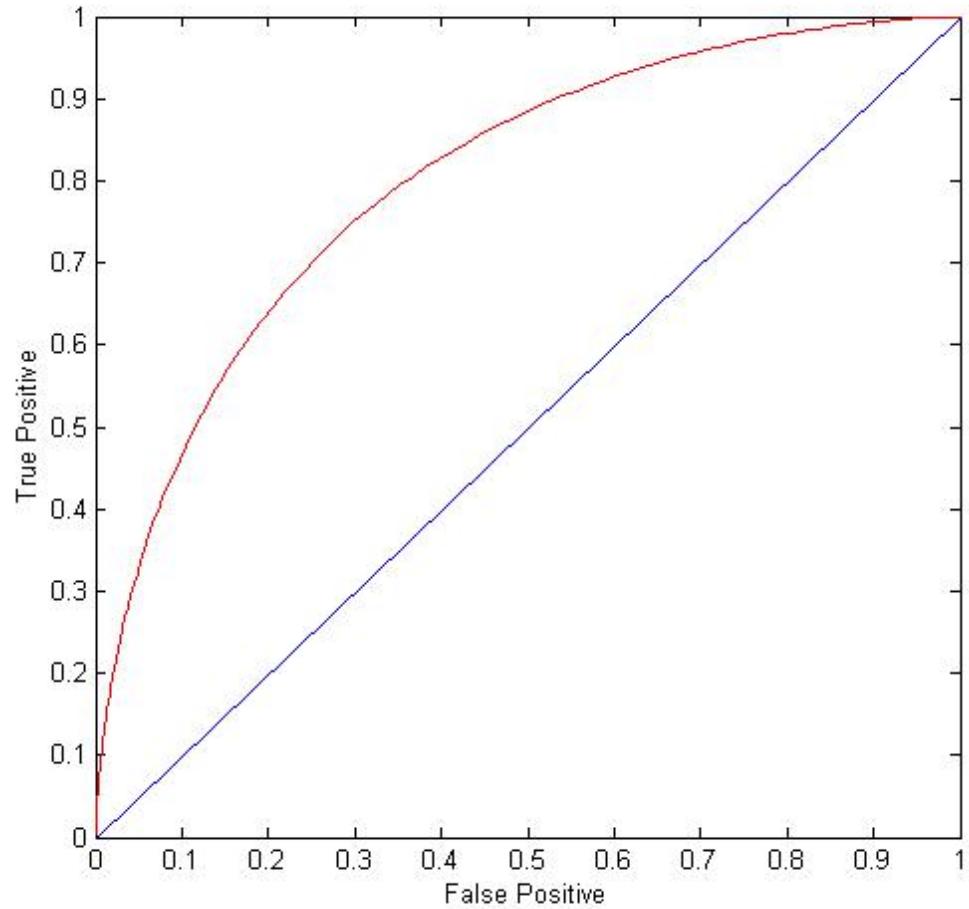
Private University Estd. in Karnataka State by Act No. 41 of 2013



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



Model Evaluation

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Methods for Performance Evaluation

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

Methods of Estimation

- **Holdout method**

- Reserve 2/3 for training and 1/3 for testing

Disadv :

1) if training set size is small, large variance of the model.

It depends on the training set size.

2) If training set size is large, then the accuracy got with smaller test size is less reliable.

- **Random Subsampling:-**

- Repeated holdout for k times.

- **Disadv:**

- No control over the no. of times each record is used for testing and training



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Methods of Estimation

- Cross Validation (CV):
 - Partition data into k disjoint subsets.
 - K-fold : train on $k-1$ partitions, test on the remaining one partition
- **Leave-one-out CV :**
 - $K = n$, where $k = \text{no. of folds}$, $n = \text{no. of examples in the data set.}$
 - In all the above methods, training records are sampled without replacement, i.e., no duplicates in training and test set.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Methods of Estimation

- **Stratified Sampling:-**

- Identify the categorical variables to divide the data set into stratum/subgroups. (Gender, Marital status, etc)
- Form stratum/subgroups.
- Within each strata, all records have same probability of being included in the sample.
- Probability for each record to be sampled differs across the strata.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Methods of Estimation

- Bootstrap Sampling :-
 - Sampling with replacement
 - Each record is equally likely to be redrawn for training and testing.
 - With total no. of records as N, a bootstrap sample of size N will include 63.2% of the records in the original data
 - Prob. of a record being chosen by a bootstrap sample is $1 - (1 - 1/N)^N = 1 - e^{-1} = 0.632$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Methods of Estimation

- Bootstrap Sampling :-
 - Sampling with replacement
 - Each record is equally likely to be redrawn for training and testing.
 - With total no. of records as N, a bootstrap sample of size N will include 63.2% of the records in the original data
 - Prob. of a record being chosen by a bootstrap sample is $1 - (1 - 1/N)^N = 1 - e^{-1} = 0.632$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Practical Issues of Classification

- **Errors of a Classification model:**
 - Training errors – errors training phase
 - Generalization error – expected error
- A good classification model should have low training error and low generalization error as well.
- **Model Overfitting:-**

When a model fits the training data too well, it can have poorer generalization error.

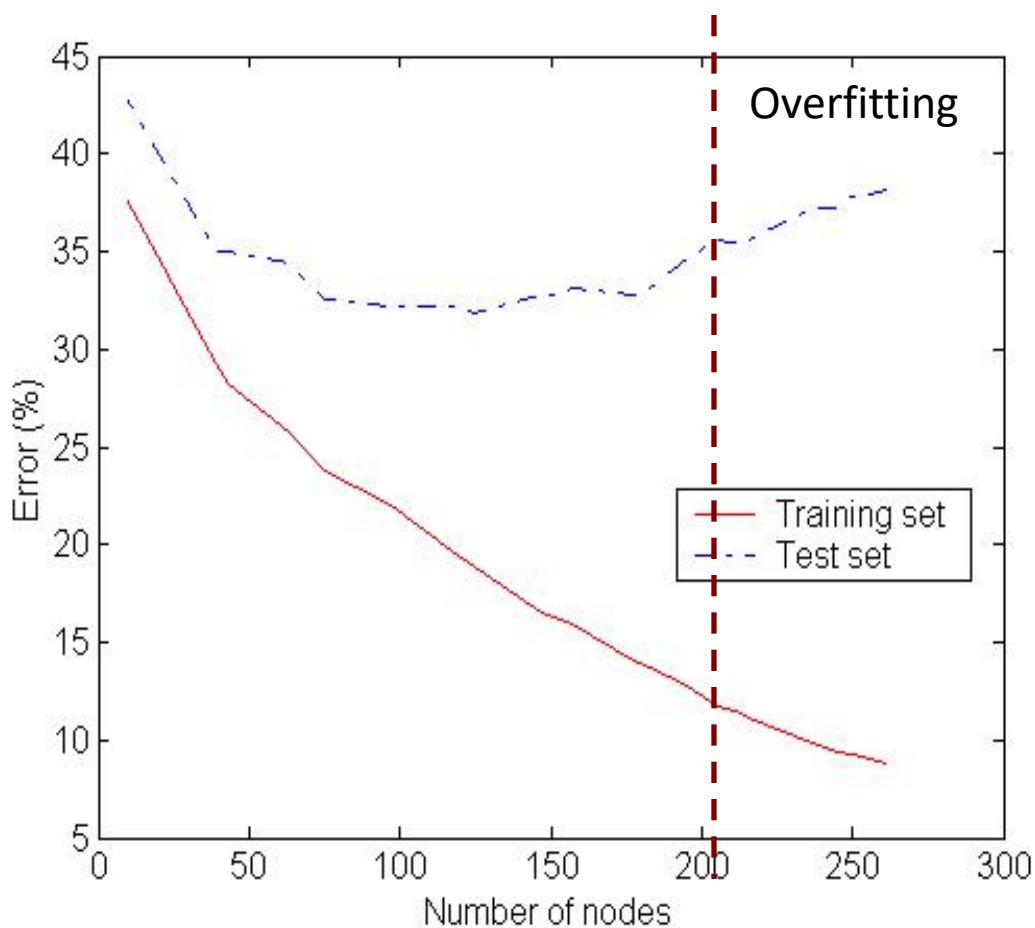


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Underfitting and Overfitting



Underfitting:

- when model is too simple, both training and test errors are large .
- when the model has yet to learn the true structure of the data.
- it performs poorly on training and test data

As the size of the tree increases, the tree has fewer training and test errors.

Overfitting:

- When the tree becomes too large, the test error increases, although training error decreases.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
 - Stop the algorithm before it becomes a fully-grown tree
 - More restrictive stopping conditions should be used
 - Stop expanding a leaf node, when the observed gain in impurity measure falls below a certain threshold.
 - Challenges? difficulty in choosing right threshold for early-termination
 - too high threshold --- underfitted models
 - too low threshold ---- insufficient to avoid overfitting.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to Address Overfitting...

- **Post-pruning**

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

Advantages:

- Pruning decision is done on a fully grown tree than on a pre-mature tree.

Disadvantages:

- additional computations to construct the full tree is wasted, when the sub-tree is pruned.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Machine Learning with WEKA

- Waikato Environment for Knowledge Analysis is a suite of machine learning software.



WEKA: the software

- Machine learning/data mining software written in Java (distributed under the GNU Public License)
- Used for research, education, and applications
- Complements “Data Mining” by Witten & Frank
- Main features:
 - Comprehensive set of data pre-processing tools, learning algorithms and evaluation methods
 - Graphical user interfaces (incl. data visualization)
 - Environment for comparing learning algorithms



**PRESIDENCY
UNIVERSITY**

Private University Encl. in Karnataka State by Act No. 41 of 2013



11/23/2020

University of Waikato

WEKA only deals with “flat” files

```
@relation heart-disease-simplified  
@attribute age numeric  
@attribute sex { female, male}  
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}  
@attribute cholesterol numeric  
@attribute exercise_induced_angina { no, yes}  
@attribute class { present, not_present}  
@data  
63,male,typ_angina,233,no,not_present  
67,male,asympt,286,yes,present  
67,male,asympt,229,yes,present  
38,female,non_anginal,?,no,not_present  
...
```



Flat file in
ARFF format

WEKA only deals with “flat” files

@relation heart-disease-simplified

@attribute age numeric

@attribute sex { female, male}

@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}

@attribute cholesterol numeric

@attribute exercise_induced_angina { no, yes}

@attribute class { present, not_present}

@data

63, male, typ_angina, 233, no, not_present

67, male, asympt, 286, yes, present

67, male, asympt, 229, yes, present

38, female, non_anginal, ?, no, not_present

numeric attribute
nominal attribute

Explorer: pre-processing the data

- Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
- Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools in WEKA are called “filters”
- WEKA contains filters for:
 - Discretization, normalization, resampling, attribute selection, transforming and combining attributes, ...



**PRESIDENCY
UNIVERSITY**

Private University Encl. in Karnataka State by Act No. 41 of 2013



11/23/2020

University of Waikato

Weka Knowledge Explorer

[Preprocess](#)[Classify](#)[Cluster](#)[Associate](#)[Select attributes](#)[Visualize](#)[Open file...](#)[Open URL...](#)[Open DB...](#)[Undo](#)[Save...](#)

Filter

[Choose](#) **None**[Apply](#)

Current relation

Relation: None

Instances: None

Attributes: None

Selected attribute

Name: None

Missing: None

Type: None

Distinct: None

Unique: None

Attributes

[Visualize All](#)

Status

Welcome to the Weka Knowledge Explorer

[Log](#)

x 0

Weka Knowledge Explorer

[Preprocess](#)[Classify](#)[Cluster](#)[Associate](#)[Select attributes](#)[Visualize](#)[Open file...](#)[Open URL...](#)[Open DB...](#)[Undo](#)[Save...](#)

Filter

[Choose](#) **None**[Apply](#)

Current relation

Relation: None

Instances: None

Attributes: None

Selected attribute

Name: None

Missing: None

Type: None

Distinct: None

Unique: None

Attributes

[Visualize All](#)

Status

Welcome to the Weka Knowledge Explorer

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose None

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: sepallength

Type: Numeric

Missing: 0 (0%)

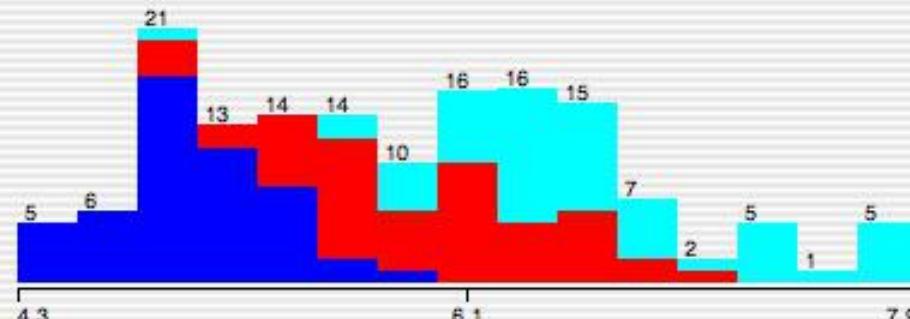
Distinct: 35

Unique: 9 (6%)

| Statistic | Value |
|-----------|-------|
| Minimum | 4.3 |
| Maximum | 7.9 |
| Mean | 5.843 |
| StdDev | 0.828 |

Colour: class (Nom)

Visualize All



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose None

Apply

Current relation

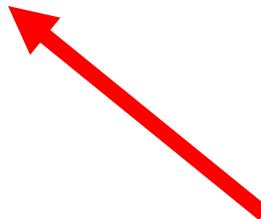
Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |



Selected attribute

Name: sepallength

Type: Numeric

Missing: 0 (0%)

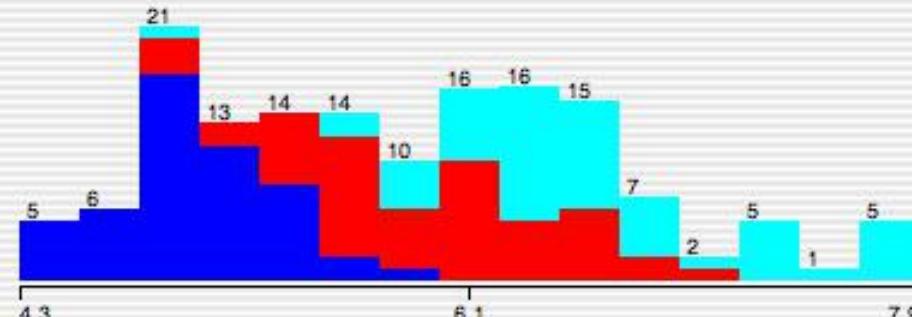
Distinct: 35

Unique: 9 (6%)

| Statistic | Value |
|-----------|-------|
| Minimum | 4.3 |
| Maximum | 7.9 |
| Mean | 5.843 |
| StdDev | 0.828 |

Colour: class (Nom)

Visualize All



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose None

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: class

Missing: 0 (0%)

Type: Nominal

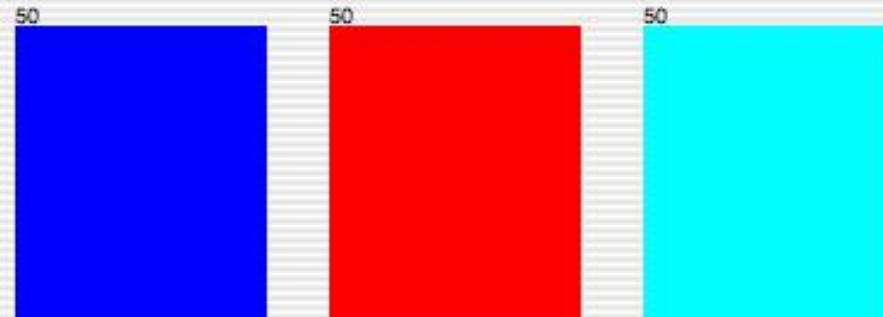
Distinct: 3

Unique: 0 (0%)

| Label | Count |
|-----------------|-------|
| Iris-setosa | 50 |
| Iris-versicolor | 50 |
| Iris-virginica | 50 |

Colour: class (Nom)

Visualize All



Status

OK

Log



Weka Knowledge Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose None

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: class

Missing: 0 (0%)

Type: Nominal

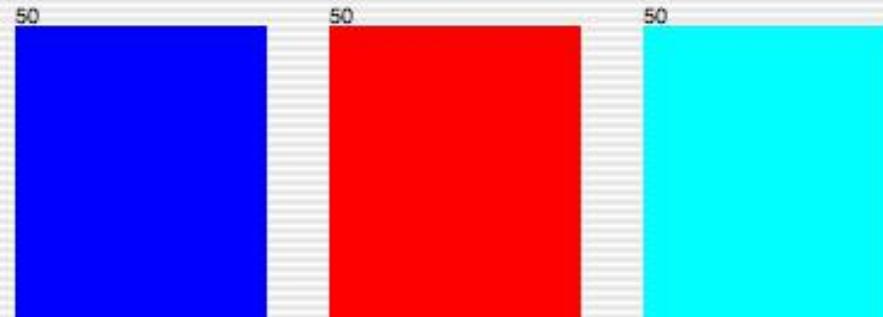
Distinct: 3

Unique: 0 (0%)

| Label | Count |
|-----------------|-------|
| Iris-setosa | 50 |
| Iris-versicolor | 50 |
| Iris-virginica | 50 |

Colour: class (Nom)

Visualize All

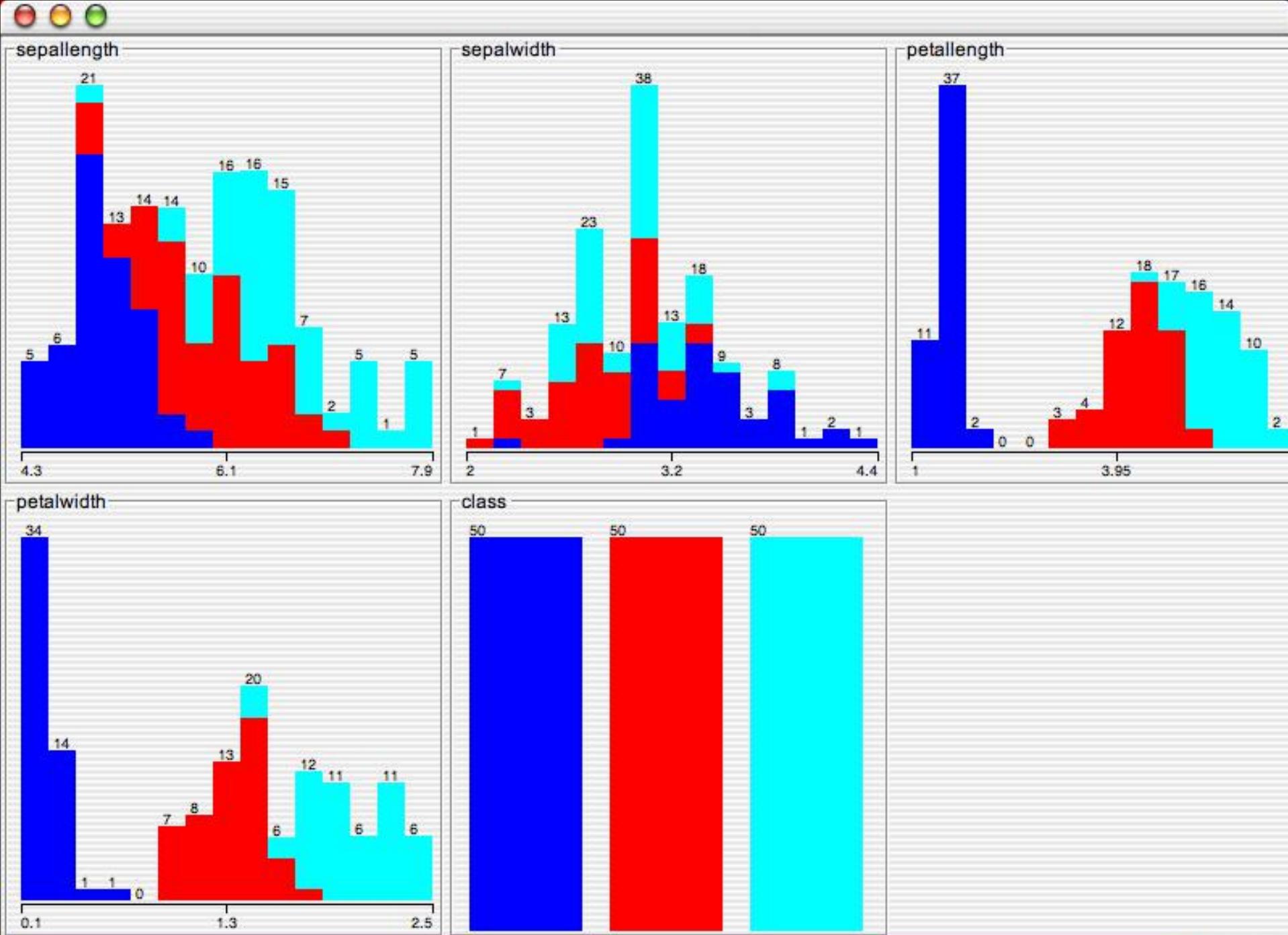


Status

OK

Log





Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose None

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Numeric

Missing: 0 (0%)

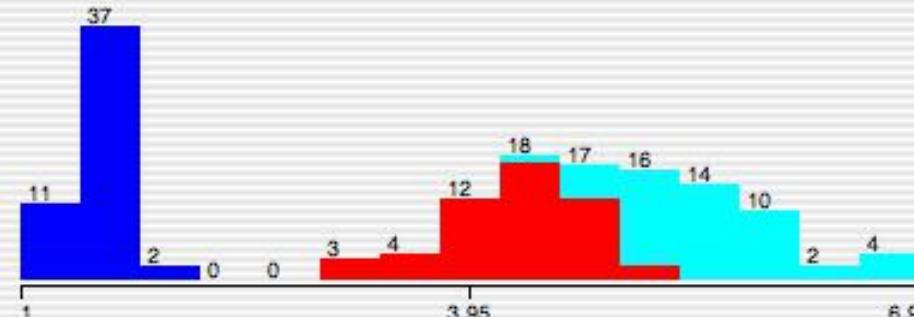
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose **None**

Apply

Current relation:

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Numeric

Missing: 0 (0%)

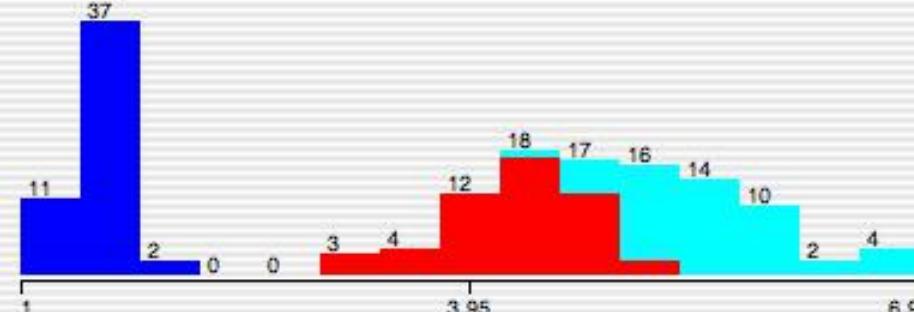
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

- weka
- filters
 - unsupervised
 - attribute
 - instance

Apply

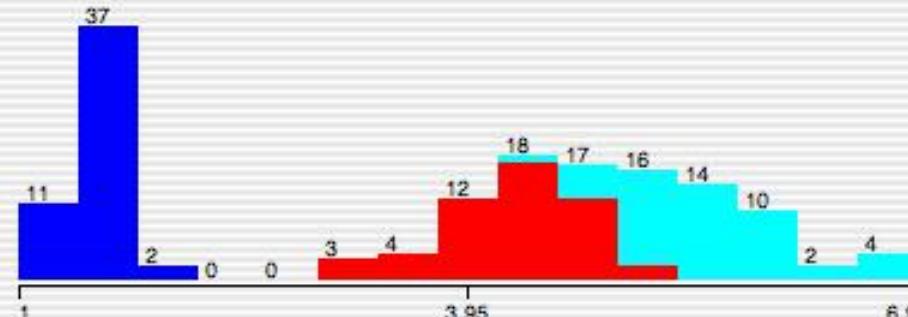
Selected attribute

Name: petallength Type: Numeric
Missing: 0 (0%) Distinct: 43 Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

- weka
- filters
 - unsupervised
 - attribute
 - instance

Apply

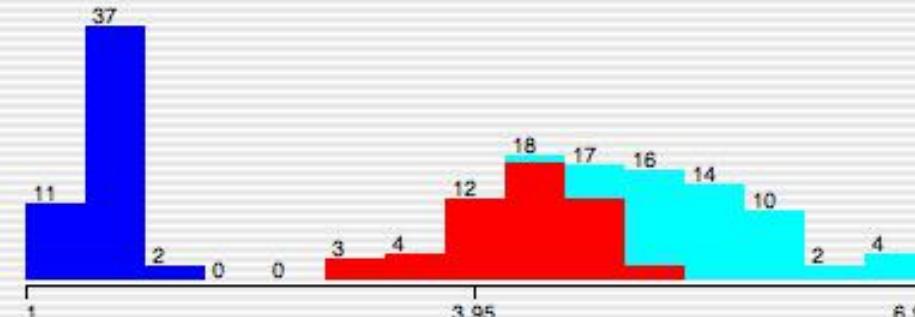
Selected attribute

Name: petallength Type: Numeric
Missing: 0 (0%) Distinct: 43 Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

weka

filters

unsupervised

attribute

Add

AddCluster

AddExpression

AddNoise

Copy

Discretize

FirstOrder

MakeIndicator

MergeTwoValues

NominalToBinary

Normalize

NumericToBinary

NumericTransform

Obfuscate

PKIDiscretize

Remove

RemoveType

Apply

Selected attribute

Name: petallength

Missing: 0 (0%) Distinct: 43

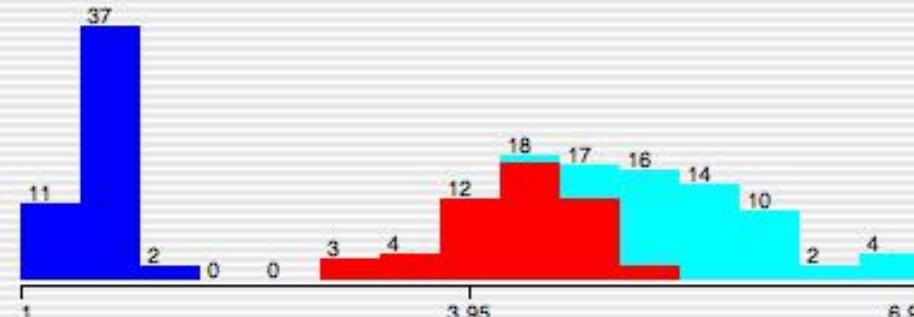
Type: Numeric

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Numeric

Missing: 0 (0%)

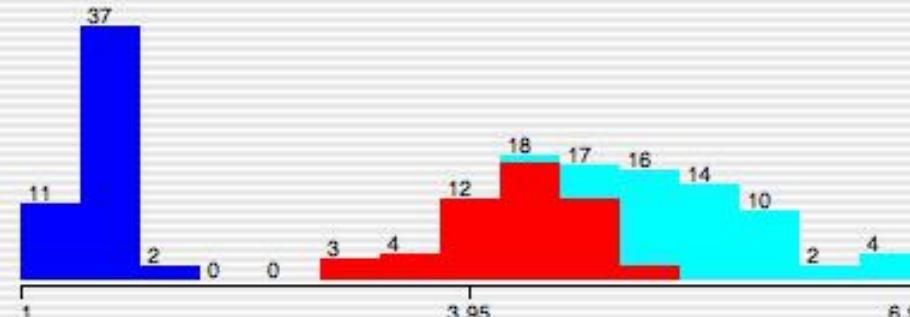
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Missing: 0 (0%)

Type: Numeric

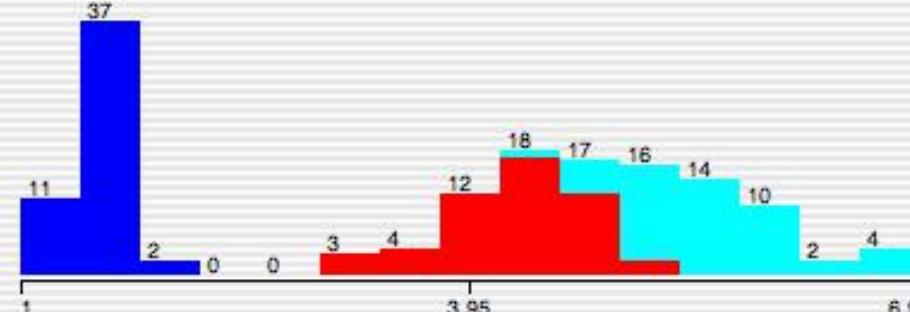
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last



weka.gui.GenericObjectEditor

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 4

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More

: Numeric

: 10 (7%)

e

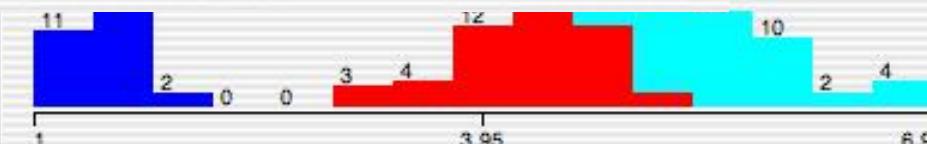
| | |
|-------------------|------------|
| attributeIndices | first-last |
| bins | 10 |
| findNumBins | False |
| invertSelection | False |
| makeBinary | False |
| useEqualFrequency | False |

Open...

Save...

OK

Cancel



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last



weka.gui.GenericObjectEditor

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 4

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More

: Numeric

: 10 (7%)

e

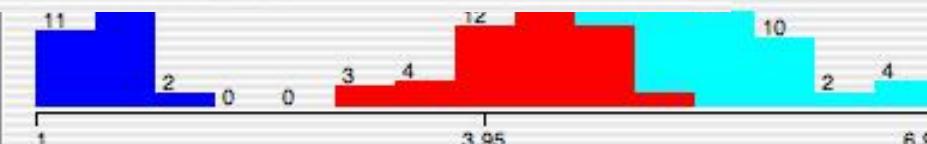
| | |
|-------------------|------------|
| attributeIndices | first-last |
| bins | 10 |
| findNumBins | False |
| invertSelection | False |
| makeBinary | False |
| useEqualFrequency | False |

Open...

Save...

OK

Cancel



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last



weka.gui.GenericObjectEditor

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 4

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More

: Numeric

: 10 (7%)

e

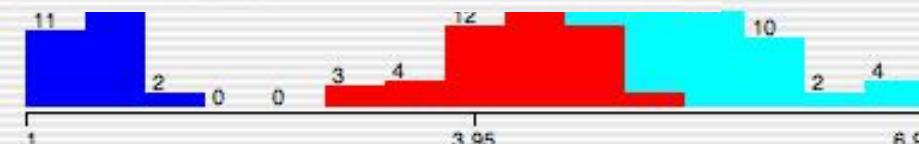
| | |
|-------------------|------------|
| attributeIndices | first-last |
| bins | 10 |
| findNumBins | False |
| invertSelection | False |
| makeBinary | False |
| useEqualFrequency | True |

Open...

Save...

OK

Cancel



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -B 10 -R first-last



weka.gui.GenericObjectEditor

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 4

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More

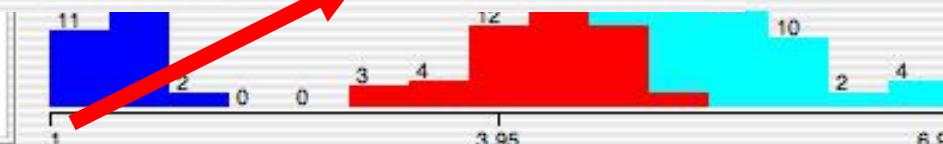
: Numeric

: 10 (7%)

e

| | |
|-------------------|------------|
| attributeIndices | first-last |
| bins | 10 |
| findNumBins | False |
| invertSelection | False |
| makeBinary | False |
| useEqualFrequency | True |

Open... Save... OK Cancel



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -F -B 10 -R first-last

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|-------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Numeric

Missing: 0 (0%)

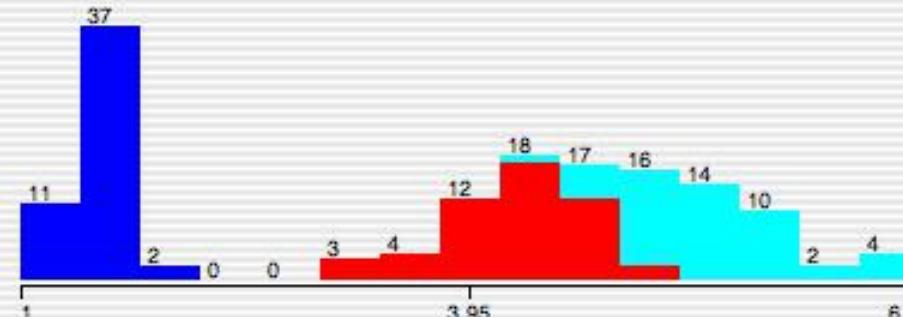
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

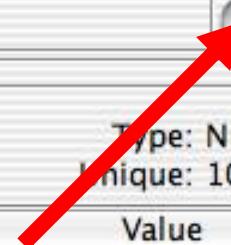
Open DB...

Undo

Save...

Filter

Choose Discretize -F -B 10 -R first-last

Apply

Current relation

Relation: iris

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Numeric

Missing: 0 (0%)

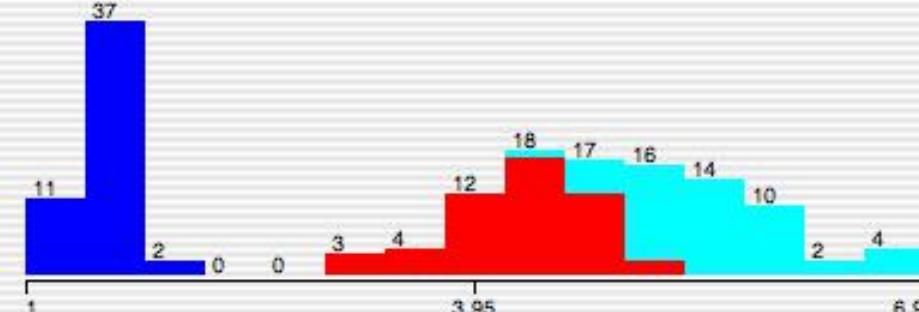
Distinct: 43

Unique: 10 (7%)

| Statistic | Value |
|-----------|-------|
| Minimum | 1 |
| Maximum | 6.9 |
| Mean | 3.759 |
| StdDev | 1.764 |

Colour: class (Nom)

Visualize All



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Undo

Save...

Filter

Choose Discretize -F -B 10 -R first-last

Apply

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Disc...

Instances: 150

Attributes: 5

Attributes

| No. | Name |
|-----|--------------------|
| 1 | sepallength |
| 2 | sepalwidth |
| 3 | petallength |
| 4 | petalwidth |
| 5 | class |

Selected attribute

Name: petallength

Type: Nominal

Missing: 0 (0%)

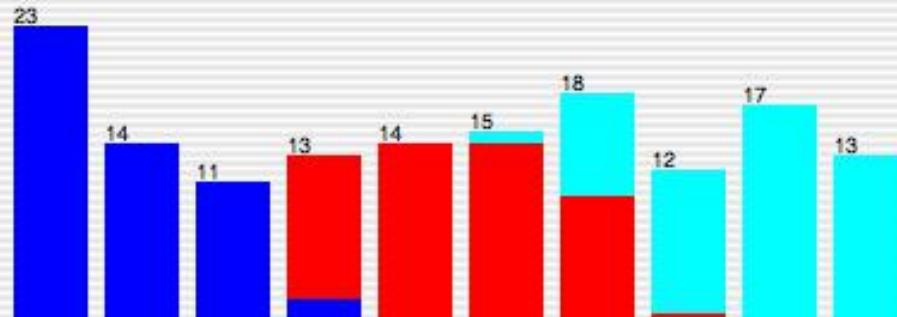
Distinct: 10

Unique: 0 (0%)

| Label | Count |
|---------------|-------|
| '(-inf-1.45]' | 23 |
| '(1.45-1.55]' | 14 |
| '(1.55-1.8]' | 11 |
| '(1.8-3.95]' | 13 |
| '(3.95-4.35]' | 14 |
| '(4.35-4.65]' | 15 |
| '(4.65-5.05]' | 18 |

Colour: class (Nom)

Visualize All



Status

OK

Log



x 0

Explorer: building “classifiers”

- Classifiers in WEKA are models for predicting nominal or numeric quantities
- Implemented learning schemes include:
 - Decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, Bayes' nets, ...
- “Meta”-classifiers include:
 - Bagging, boosting, stacking, error-correcting output codes, locally weighted learning, ...



**PRESIDENCY
UNIVERSITY**

Private University Encl. in Karnataka State by Act No. 41 of 2013



11/23/2020

University of Waikato

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose ZeroR

Test options

Use training set

Supplied test set [Set...](#)

Cross-validation Folds

Percentage split %

[More options...](#)

Classifier output

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose **ZeroR**

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

Classifier output

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

weka

classifiers

bayes

functions

lazy

meta

misc

trees

adtree

DecisionStump

Id3

j48

J48

Imt

m5

RandomForest

RandomTree

REPTree

UserClassifier

rules

ifier output

Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set **Set...**
- Cross-validation Folds 10
- Percentage split % 66

More options...

Classifier output

(Nom) class

Start**Stop**

Result list (right-click for options)

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

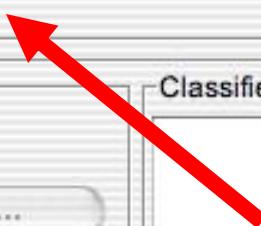
Select attributes

Visualize

Classifier

Choose

J48 -C 0.25 -M 2



Test options

 Use training set Supplied test set Cross-validation Folds Percentage split % (Nom) class

Result list (right-click for options)

Classifier output

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

weka.gui.GenericObjectEditor

Test options

- Use training set
- Supplied test set
- Cross-validation Folds 10
- Percentage split % 66

(Nom) class

Result list (right-click for options)

| | |
|---------------------|-------|
| binarySplits | False |
| confidenceFactor | 0.25 |
| minNumObj | 2 |
| numFolds | 3 |
| reducedErrorPruning | False |
| saveInstanceData | False |
| subtreeRaising | True |
| unpruned | False |
| useLaplace | False |

Status

OK



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

weka.gui.GenericObjectEditor

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

| | |
|---------------------|-------|
| binarySplits | False |
| confidenceFactor | 0.25 |
| minNumObj | 2 |
| numFolds | 3 |
| reducedErrorPruning | False |
| saveInstanceData | False |
| subtreeRaising | True |
| unpruned | False |
| useLaplace | False |

Open...

Save...

OK

Cancel



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set
- Cross-validation Folds
- Percentage split %

Classifier output

(Nom) class

Result list (right-click for options)

Status

OK



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

Classifier output

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set [Set...](#)
- Cross-validation Folds 10
- Percentage split % 66

[More options...](#)

Classifier output

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

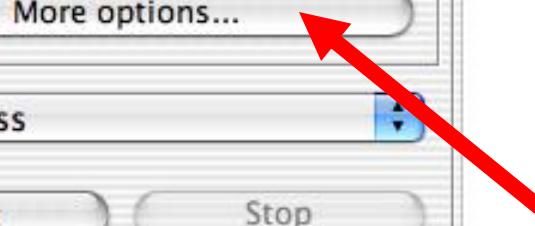
(Nom) class

Start

Stop

Result list (right-click for options)

Classifier output



Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Cross-validation Folds Percentage split % (Nom) class

Result list (right-click for options)

Classifier output

 Classifier evaluation opt Output model Output per-class stats Output entropy evaluation measures Output confusion matrix Store predictions for visualization Output text predictions on test set Cost-sensitive evaluation Random seed for XVal / % Split

Status

OK



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

Classifier output

Classifier evaluation opt

 Output model Output per-class stats Output entropy evaluation measures Output confusion matrix Store predictions for visualization Output text predictions on test set Cost-sensitive evaluation Set...

Random seed for XVal / % Split

1

OK



Status

OK

Log



Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set [Set...](#)
- Cross-validation Folds 10
- Percentage split % 66

[More options...](#)

Classifier output

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

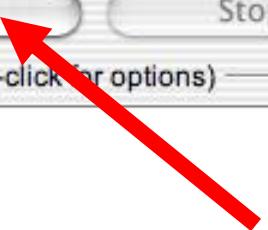
Classifier output

(Nom) class

Start

Stop

Result list (right-click for options)



This area displays the results of the classification process. It is currently empty, showing a plain white space.

Status

OK

Log



x 0

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set
- Cross-validation Folds 10
- Percentage split % 66

(Nom) class

Result list (right-click for options)

11:49:05 - trees.J48.J48

Classifier output

==== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
 Relation: iris
 Instances: 150
 Attributes: 5
 sepallength
 sepalwidth
 petallength
 petalwidth
 class

Test mode: split 66% train, remainder test

==== Classifier model (full training set) ===

J48 pruned tree

```
-----
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Number of Leaves : 5

Status

OK



x 0

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set **Set...**
- Cross-validation Folds 10
- Percentage split % 66

More options...

(Nom) class

Start**Stop**

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

==== Run information ====
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: split 66% train, remainder test

==== Classifier model (full training set) ====
J48 pruned tree

```
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Number of Leaves : 5



Status

OK

Log

x 0

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

Time taken to build model: 0.24 seconds

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

Time taken to build model: 0.24 seconds

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

Log



x 0

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

Time taken to build model: 0.24 seconds

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| | Recall | F-Measure | Class |
|-------|--------|-----------|-----------------|
| 1 | 1 | 1 | Iris-setosa |
| 1 | 0.95 | 0.95 | Iris-versicolor |
| 0.882 | 0.938 | 0.938 | Iris-virginica |

View in main window

View in separate window

Save result buffer

Load model

Save model

Re-evaluate model on current test set

Visualize classifier errors

Visualize tree

Visualize margin curve

Visualize threshold curve

Visualize cost curve

Log



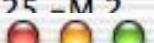
Status

OK

Classifier

Choose

J48 -C 0.25 -M 2



Weka Classifier Tree Visualizer: 11:49:05 – trees.j48.J48 (iris)

Test options

- Use training set
- Supplied test set
- Cross-validation
- Percentage split

More options

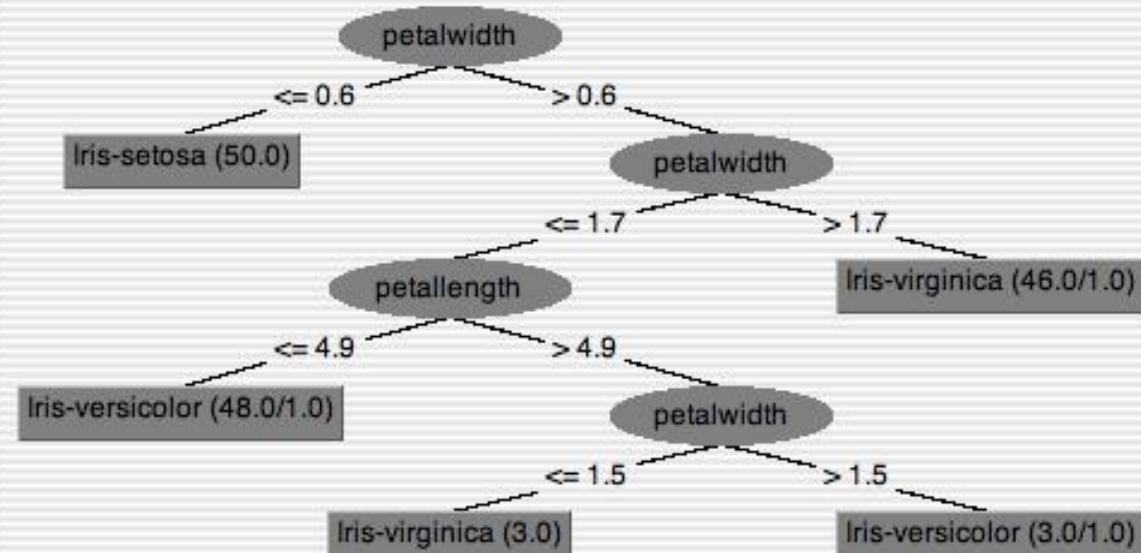
(Nom) class

Start

Result list (right-click for

11:49:05 – trees.j48.J48

Tree View



96.0784 %
3.9216 %

ass
is-setosa
is-versicolor
is-virginica

```

+-----+-----+
| a = Iris-setosa |
| 0 19 0 | b = Iris-versicolor |
| 0 2 15 | c = Iris-virginica |
+-----+-----+
  
```

Status

OK

Log



x 0

Classifier

Choose J48 -C 0.25 -M 2

Test options

 Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

Time taken to build model: 0.24 seconds

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
 Supplied test set
 Cross-validation Folds 10
 Percentage split % 66

(Nom) class

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

Time taken to build model: 0.24 seconds

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK



x 0

Classifier

- weka
- classifiers
 - bayes
 - functions
 - LeastMedSq
 - LinearRegression
 - Logistic
 - neural
 - NeuralNetwork
 - pace
 - supportVector
 - SimpleLinearRegression
 - SimpleLogistic
 - VotedPerceptron
 - Winnow
- lazy
- meta
- misc
- trees
- rules

'output'

Time taken to build model: 0.24 seconds

Evaluation on test split ===

Summary ===

| | | |
|--|-----------|-----------|
| Number of Classified Instances | 49 | 96.0784 % |
| Number of Incorrectly Classified Instances | 2 | 3.9216 % |
| Mean absolute statistic | 0.9408 | |
| Absolute error | 0.0396 | |
| Mean squared error | 0.1579 | |
| Mean absolute error | 8.8979 % | |
| Relative squared error | 33.4091 % | |
| Number of Instances | 51 | |

Detailed Accuracy By Class ===

| | FP Rate | Precision | Recall | F-Measure | Class |
|--|---------|-----------|--------|-----------|-----------------|
| | 0 | 1 | 1 | 1 | Iris-setosa |
| | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

Log



x 0

Classifier

Choose

NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

 Use training set Supplied test set [Set...](#) Cross-validation Folds 10 Percentage split % 66[More options...](#)

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

[Log](#)

Classifier

Choose

NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

 Use training set Supplied test set [Set...](#) Cross-validation Folds 10 Percentage split % 66[More options...](#)

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

Classifier output

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 49 | 96.0784 % |
| Incorrectly Classified Instances | 2 | 3.9216 % |
| Kappa statistic | 0.9408 | |
| Mean absolute error | 0.0396 | |
| Root mean squared error | 0.1579 | |
| Relative absolute error | 8.8979 % | |
| Root relative squared error | 33.4091 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.063 | 0.905 | 1 | 0.95 | Iris-versicolor |
| 0.882 | 0 | 1 | 0.882 | 0.938 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

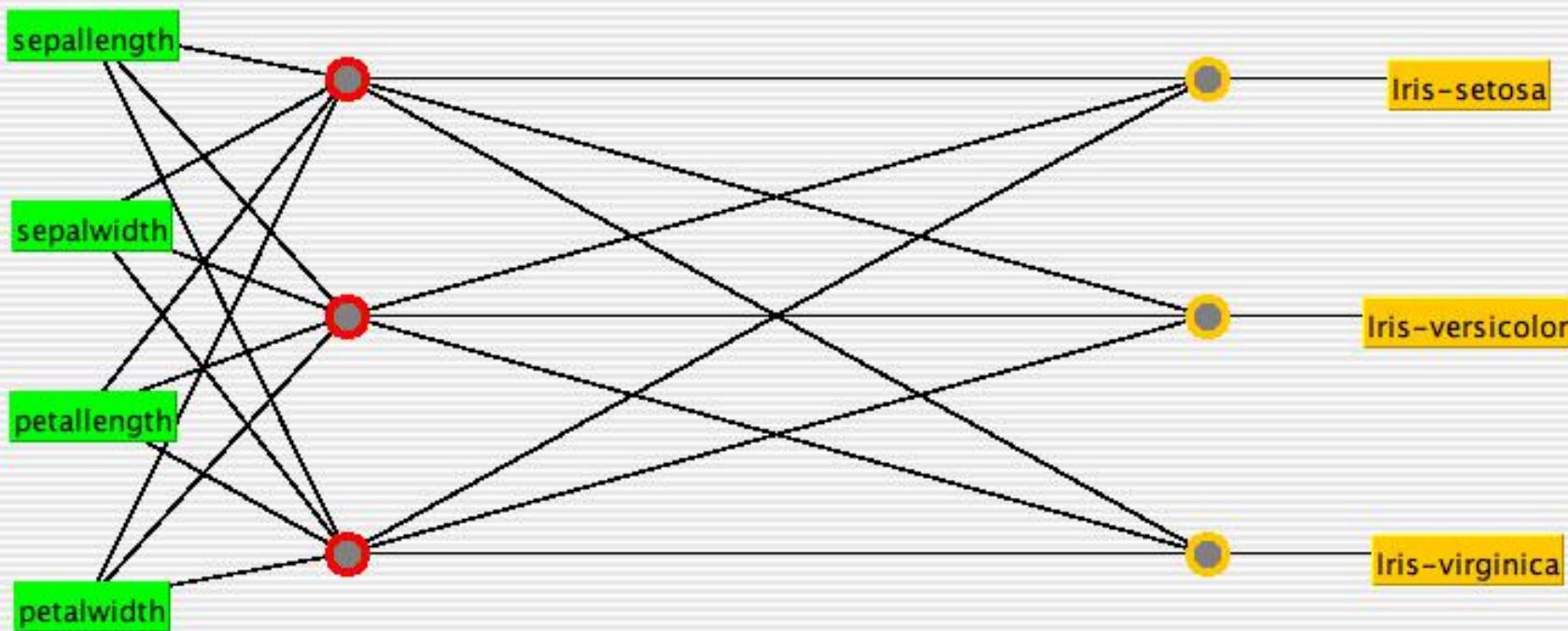
OK

Log



x 0

Neural Network



-Controls

Start

Epoch 0

Num Of Epochs 500

Accept Error per Epoch = 0

Learning Rate = 0.3

Momentum =

Building model on training data...

Preprocess

Classify

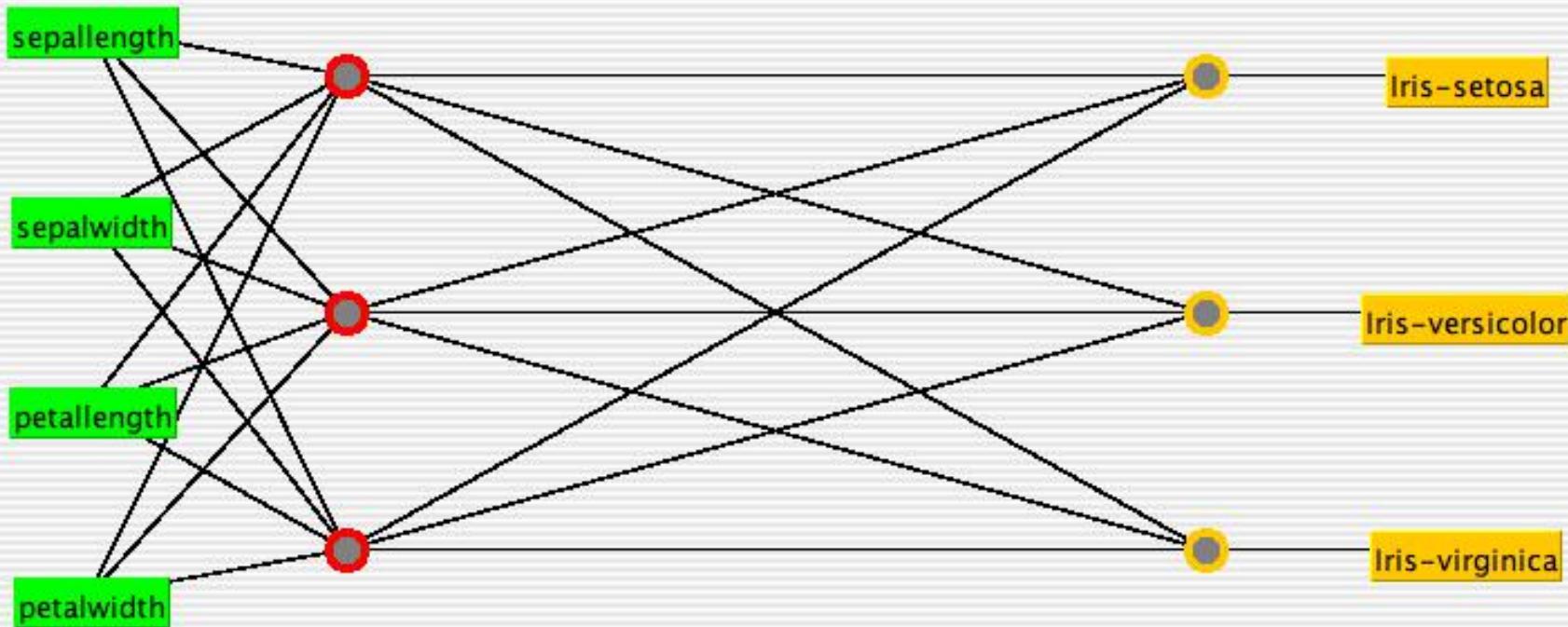
Cluster

Associate

Select attributes

Visualize

Neural Network



Controls

Start

Epoch 0

Num Of Epochs 500

Accept

Error per Epoch = 0

Learning Rate = 0.3

Momentum = 0.2

building model on training data...

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Test options

 Use training set Supplied test set [Set...](#) Cross-validation Folds 10 Percentage split % 66[More options...](#)

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

14:34:28 - functions.neural.NeuralNetwork

Classifier output

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 50 | 98.0392 % |
| Incorrectly Classified Instances | 1 | 1.9608 % |
| Kappa statistic | 0.9704 | |
| Mean absolute error | 0.0239 | |
| Root mean squared error | 0.1101 | |
| Relative absolute error | 5.3594 % | |
| Root relative squared error | 23.2952 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.031 | 0.95 | 1 | 0.974 | Iris-versicolor |
| 0.941 | 0 | 1 | 0.941 | 0.97 | Iris-virginica |

==== Confusion Matrix ===

| | | | |
|----|----|----|---------------------|
| a | b | c | <-- classified as |
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 1 | 16 | c = Iris-virginica |

Status

OK

Log



x 0

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

NeuralNetwork -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Test options

 Use training set Supplied test set

Set...

 Cross-validation

Folds 10

 Percentage split

% 66

More options...

(Nom) class

Start

Stop

Result list (right-click for options)

11:49:05 - trees.j48.J48

14:34:28 - functions.neural.NeuralNetwork

Classifier output

==== Evaluation on test split ====

==== Summary ====

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 50 | 98.0392 % |
| Incorrectly Classified Instances | 1 | 1.9608 % |
| Kappa statistic | 0.9704 | |
| Mean absolute error | 0.0239 | |
| Root mean squared error | 0.1101 | |
| Relative absolute error | 5.3594 % | |
| Root relative squared error | 23.2952 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ====

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.031 | 0.95 | 1 | 0.974 | Iris-versicolor |
| 0.941 | 0 | 1 | 0.941 | 0.97 | Iris-virginica |

==== Confusion Matrix ====

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 1 | 16 | c = Iris-virginica |

Status

OK

Log



x 0

Classifier

- weka
- classifiers
 - bayes
 - AODE
 - BayesNetK2
 - BayesNetB
 - NaiveBayes
 - NaiveBayesMultinomial
 - NaiveBayesSimple
 - NaiveBayesUpdateable
- functions
- lazy
- meta
- misc
- trees
- rules

Classifier output

```
-- Evaluation on test split ===
```

```
-- Summary ===
```

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 50 | 98.0392 % |
| Incorrectly Classified Instances | 1 | 1.9608 % |
| | | |
| Appa statistic | 0.9704 | |
| Mean absolute error | 0.0239 | |
| Root mean squared error | 0.1101 | |
| Relative absolute error | 5.3594 % | |
| Root relative squared error | 23.2952 % | |
| Total Number of Instances | 51 | |

```
-- Detailed Accuracy By Class ===
```

| | P Rate | FP Rate | Precision | Recall | F-Measure | Class |
|-----------------|--------|---------|-----------|--------|-----------|-----------------|
| Iris-setosa | 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| Iris-versicolor | 1 | 0.031 | 0.95 | 1 | 0.974 | Iris-versicolor |
| Iris-virginica | 0.941 | 0 | 1 | 0.941 | 0.97 | Iris-virginica |

```
-- Confusion Matrix ===
```

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 1 | 16 | c = Iris-virginica |

Classifier

Choose **NaiveBayes**

Test options

Use training set

Supplied test set [Set...](#)

Cross-validation Folds 10

Percentage split % 66

[More options...](#)

(Nom) class

[Start](#)

[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

14:34:28 - functions.neural.NeuralNetwork

Classifier output

==== Evaluation on test split ====

==== Summary ====

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 50 | 98.0392 % |
| Incorrectly Classified Instances | 1 | 1.9608 % |
| Kappa statistic | 0.9704 | |
| Mean absolute error | 0.0239 | |
| Root mean squared error | 0.1101 | |
| Relative absolute error | 5.3594 % | |
| Root relative squared error | 23.2952 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ====

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.031 | 0.95 | 1 | 0.974 | Iris-versicolor |
| 0.941 | 0 | 1 | 0.941 | 0.97 | Iris-virginica |

==== Confusion Matrix ====

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 1 | 16 | c = Iris-virginica |

Classifier

Choose **NaiveBayes**

Test options

Use training set

Supplied test set [Set...](#)

Cross-validation Folds 10

Percentage split % 66

[More options...](#)

(Nom) class

[Start](#)

[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

14:34:28 - functions.neural.NeuralNetwork

Classifier output

==== Evaluation on test split ====

==== Summary ====

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 50 | 98.0392 % |
| Incorrectly Classified Instances | 1 | 1.9608 % |
| Kappa statistic | 0.9704 | |
| Mean absolute error | 0.0239 | |
| Root mean squared error | 0.1101 | |
| Relative absolute error | 5.3594 % | |
| Root relative squared error | 23.2952 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ====

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 1 | 0.031 | 0.95 | 1 | 0.974 | Iris-versicolor |
| 0.941 | 0 | 1 | 0.941 | 0.97 | Iris-virginica |

==== Confusion Matrix ====

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 19 | 0 | b = Iris-versicolor |
| 0 | 1 | 16 | c = Iris-virginica |

Classifier

Choose **NaiveBayes**

Test options

 Use training set Supplied test set [Set...](#) Cross-validation Folds 10 Percentage split % 66[More options...](#)

(Nom) class

[Start](#)[Stop](#)

Result list (right-click for options)

11:49:05 - trees.j48.J48

14:34:28 - functions.neural.NeuralNetwork

14:48:05 - bayes.NaiveBayes

Classifier output

==== Evaluation on test split ===

==== Summary ===

| | | |
|----------------------------------|-----------|-----------|
| Correctly Classified Instances | 48 | 94.1176 % |
| Incorrectly Classified Instances | 3 | 5.8824 % |
| Kappa statistic | 0.9113 | |
| Mean absolute error | 0.0447 | |
| Root mean squared error | 0.1722 | |
| Relative absolute error | 10.0365 % | |
| Root relative squared error | 36.4196 % | |
| Total Number of Instances | 51 | |

==== Detailed Accuracy By Class ===

| TP Rate | FP Rate | Precision | Recall | F-Measure | Class |
|---------|---------|-----------|--------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 1 | Iris-setosa |
| 0.947 | 0.063 | 0.9 | 0.947 | 0.923 | Iris-versicolor |
| 0.882 | 0.029 | 0.938 | 0.882 | 0.909 | Iris-virginica |

==== Confusion Matrix ===

| a | b | c | <-- classified as |
|----|----|----|---------------------|
| 15 | 0 | 0 | a = Iris-setosa |
| 0 | 18 | 1 | b = Iris-versicolor |
| 0 | 2 | 15 | c = Iris-virginica |

Status

OK

Log



x 0

Instance-Based Classifiers

Set of Stored Cases

| Atr1 | | AtrN | Class |
|------|-------|------|-------|
| | | | A |
| | | | B |
| | | | B |
| | | | C |
| | | | A |
| | | | C |
| | | | B |

- Store the training records
- Use training records to predict the class label of unseen cases

Unseen Case

| Atr1 | | AtrN |
|------|-------|------|
| | | |



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

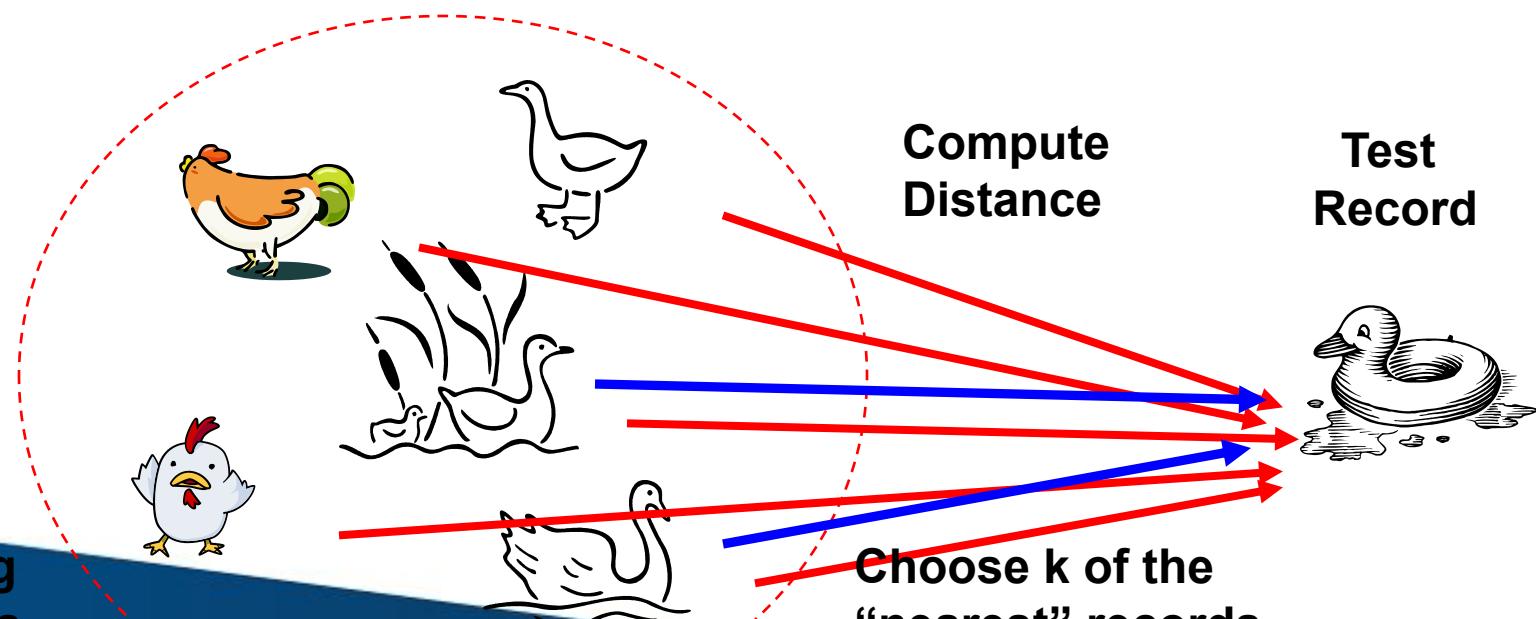


Instance Based Classifiers

- Examples:
 - Rote-learner
 - Memorizes entire training data and performs classification only if attributes of test record match one of the training examples exactly.
 - Nearest neighbor
 - Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

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

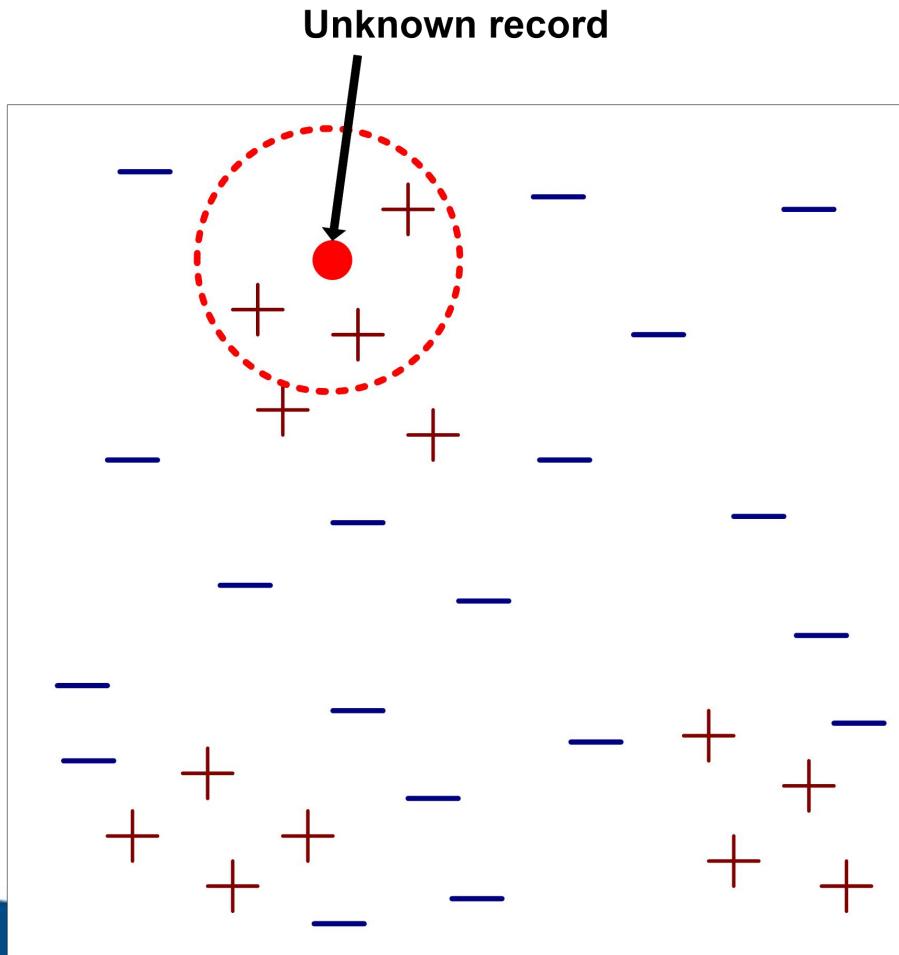


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Nearest-Neighbor Classifiers



- Requires three things
 - The set of stored records
 - Distance Metric to compute distance between records
 - The value of k , the number of nearest neighbors to retrieve
- To classify an unknown record:
 - Compute distance to other training records
 - Identify k nearest neighbors
 - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

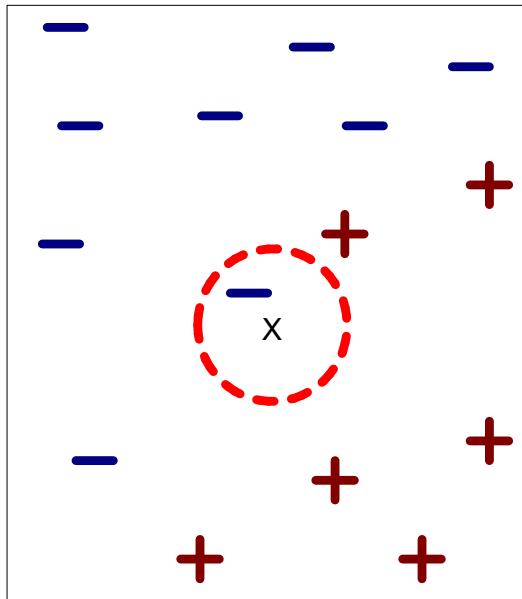


**PRESIDENCY
UNIVERSITY**

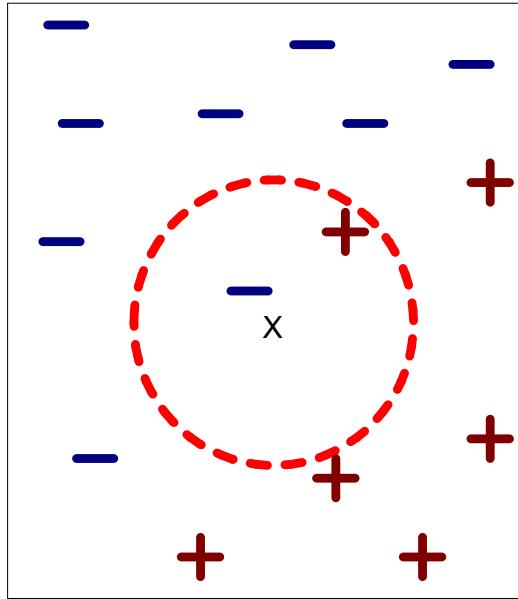
Private University Estd. in Karnataka State by Act No. 41 of 2013



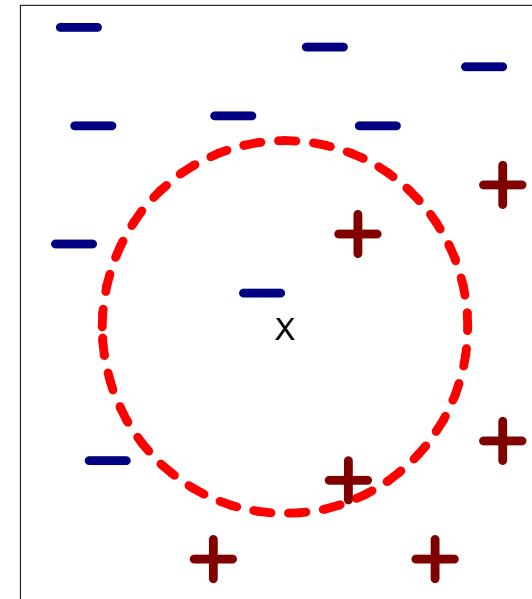
Definition of Nearest Neighbor



(a) 1-nearest neighbor



(b) 2-nearest neighbor



(c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest Neighbor Classification

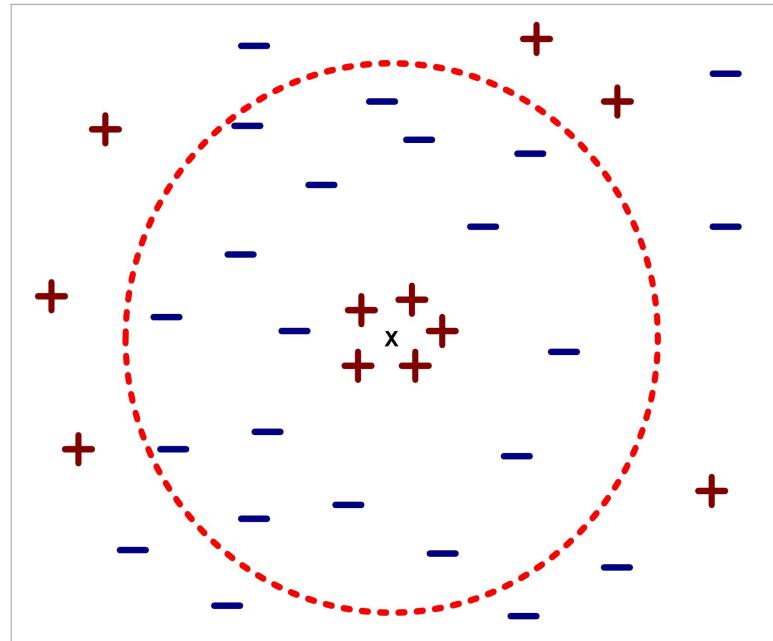
- Compute distance between two points:
 - Euclidean distance

$$d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
 - take the majority vote of class labels among the k-nearest neighbors
 - Weigh the vote according to distance
 - weight factor, $w = 1/d^2$

Nearest Neighbor Classification...

- Choosing the value of k:
 - If k is too small, susceptible to overfitting, due to noise points in the training data.
 - If k is too large, neighborhood may include points from other classes.



Nearest Neighbor Classification...

- Scaling issues
 - 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
 - Solution: Normalize the vectors to unit length



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Nearest Neighbor Classification...

1. Let k be the no. of nearest neighbors and \mathbf{D} be the set of training examples.
2. *for each test example $z = (x', y')$ do*
 - 2.1 compute $d(x', x)$, the distance between z and every example $(x, y) \in \mathbf{D}$.
 - 2.2 Select $\mathbf{D}_z \subseteq \mathbf{D}$, the set of k closest training examples to z .
 - 2.3 $y' = \operatorname{argmax}_{\nu} \sum_{(x_i, y_i) \in D_z} I(\nu = y_i)$
 - 2.4 *end for*

Nearest neighbor Classification...

- k-NN classifiers are lazy learners
 - It does not build models explicitly
 - Unlike eager learners such as decision tree induction and rule-based systems
 - Classifying unknown records are relatively expensive



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Bayes Classifier

- Different from DT approach
- Based on Bayes prob. Theorem
- We have a hypothesis that the given data belongs to a particular class.
- Calculate the probability of the hypothesis to be true.
- Used where the relationship between attb sets and the class label is non-deterministic.
- The class of test record cannot be predicted with certainty (although its attb values match some of the training examples)
- This may be due to some confounding factors that affect classification, not included in the analysis.

Bayes Classifier.....

- If the task is to predict whether a person is at risk for heart disease based on diet and workout frequency.
- Proper diet, enough workout frequency → less risk
- But, not true always, due to some other factors like heredity, excessive smoking and alcohol abuse.
- Result:- uncertainties into the learning problem.
- ∴ Bayes theorem is used to form a probabilistic framework for classification.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Bayes Classifier...

Notations:-

- $P(A)$ = prob. that event A will occur
- $P(A / B)$ = prob. that event A will occur, given that event B has already occurred.
- ∵ it is the conditional prob. of A
- ∵ it is based on the condition that B has already occurred.

Ex:-

- A..... Prob of a student passing course A
- B..... Prob of a student passing course B
- Then, $P(A / B)$?
- It is the prob. of the student passing A, when we know that he has passed B.

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(B | A) = \frac{P(B, A)}{P(A)}$$

$$P(A | B) = \frac{P(A, B)}{P(B)}$$

- Bayes theorem:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Bayesian Classifiers

- From where?
- $P(A/B) = \frac{P(A \& B)}{P(B)}$ 1
- $P(B/A) = \frac{P(A \& B)}{P(A)}$ 2

$\therefore \frac{1}{2}$ gives Bayes theorem

$$\frac{P(A/B)}{P(B/A)} = \frac{P(A \& B)}{P(B)} \times \frac{P(A)}{P(A \& B)}$$

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(B)}$$



Example of Bayes Theorem

- Given:
 - A doctor knows that meningitis causes stiff neck 50% of the time
 - Prior probability of any patient having meningitis is 1/50,000
 - Prior probability of any patient having stiff neck is 1/20
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2, \dots, A_n)
 - Goal is to predict class C
 - Specifically, we want to find the value of C that maximizes $P(C | A_1, A_2, \dots, A_n)$
- Can we estimate $P(C | A_1, A_2, \dots, A_n)$ directly from data?

Bayesian Classifiers

- Approach:
 - compute the posterior probability $P(C | A_1, A_2, \dots, A_n)$ for all values of C using the Bayes theorem

$$P(C | A_1 A_2 \dots A_n) = \frac{P(A_1 A_2 \dots A_n | C) P(C)}{P(A_1 A_2 \dots A_n)}$$

- Choose value of C that maximizes $P(C | A_1, A_2, \dots, A_n)$
- Equivalent to choosing value of C that maximizes $P(A_1, A_2, \dots, A_n | C) P(C)$
- How to estimate $P(A_1, A_2, \dots, A_n | C)$?



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
 - $P(A_1, A_2, \dots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j)$
 - Can estimate $P(A_i | C_j)$ for all A_i and C_j .
 - New point is classified to C_j if $P(C_j) \prod P(A_i | C_j)$ is maximal.

How to Estimate Probabilities from Data?

| Tid | Refund | Marital Status | Taxable Income | Evade |
|-----|--------|----------------|----------------|-------|
| 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 |

- Class: $P(C) = N_c/N$
 - e.g., $P(\text{No}) = 7/10$,
 $P(\text{Yes}) = 3/10$
- For discrete attributes:
$$P(A_i | C_k) = |A_{ik}| / N_c$$
 - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
 - Examples:
$$P(\text{Status}=\text{Married}|\text{No}) = 4/7$$
$$P(\text{Refund}=\text{Yes}|\text{Yes})=0$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



How to Estimate Probabilities from Data?

- For continuous attributes:
 - Discretize the range into bins
 - one ordinal attribute per bin
 - violates independence assumption
 - Two-way split: $(A < v)$ or $(A > v)$
 - choose only one of the two splits as new attribute
 - Probability density estimation:
 - Assume attribute follows a normal distribution
 - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
 - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

| Tid | Refund | Marital Status | Taxable Income | Evade |
|-----|--------|----------------|----------------|-------|
| 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 |

- Normal distribution:

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

- One for each (A_i, c_i) pair
- For (Income, Class=No):
 - If Class=No
 - sample mean = 110
 - sample variance = 2550

$$P(\text{Income}=120 | \text{No}) = \frac{1}{\sqrt{2\pi}(50.49)} e^{-\frac{(120-110)^2}{2(2550)}} = 0.00774$$

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

$$P(\text{Refund}=\text{Yes}|\text{No}) = 3/7$$

$$P(\text{Refund}=\text{No}|\text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes}|\text{Yes}) = 0$$

$$P(\text{Refund}=\text{No}|\text{Yes}) = 1$$

$$P(\text{Marital Status}=\text{Single}|\text{No}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{No}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{No}) = 4/7$$

$$P(\text{Marital Status}=\text{Single}|\text{Yes}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{Yes}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{Yes}) = 0$$

For taxable income:

If class=No: sample mean=110
sample variance=2975

If class=Yes: sample mean=90
sample variance=25

- $P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No}|\text{Class}=\text{No}) \times P(\text{Married}|\text{ Class}=\text{No}) \times P(\text{Income}=120\text{K}|\text{ Class}=\text{No}) = 4/7 \times 4/7 \times 0.0072 = 0.0024$
- $P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No}|\text{ Class}=\text{Yes}) \times P(\text{Married}|\text{ Class}=\text{Yes}) \times P(\text{Income}=120\text{K}|\text{ Class}=\text{Yes}) = 1 \times 0 \times 1.2 \times 10^{-9} = 0$

Since $P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})$

Therefore $P(\text{No}|X) > P(\text{Yes}|X)$
 $\Rightarrow \text{Class} = \text{No}$

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original : } P(A_i | C) = \frac{N_{ic}}{N_c}$$

c: number of classes

$$\text{Laplace : } P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

p: user specified parameter. Prior probability

$$\text{m - estimate : } P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

m: parameter, equivalent to the sample size.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Example of Naïve Bayes Classifier

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

A: attributes

M: mammals

N: non-mammals

$$P(A | M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A | N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A | M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A | N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

$P(A|M)P(M) > P(A|N)P(N)$

=> Mammals

| Give Birth | Can Fly | Live in Water | Have Legs | Class |
|------------|---------|---------------|-----------|-------|
| yes | no | yes | no | ? |

Naïve Bayes (Summary)

- Robust to isolated noise points. They are averaged when estimating conditional prob.
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
 - Use other techniques such as Bayesian Belief Networks (BBN)



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Similarity and Dissimilarity

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Similarity/Dissimilarity for Objects with Single Attribute

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

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

Table 5.1. Similarity and dissimilarity for simple attributes

Dissimilarities between Data Objects with multiple Numeric attributes

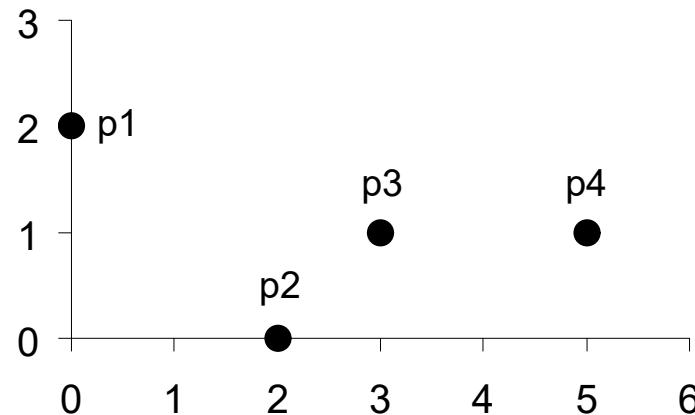
- Euclidean Distance

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

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

- Standardization is necessary, if scales differ.

Euclidean Distance



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

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

Distance Matrix

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance. Given two objects p and q

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

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

Minkowski Distance: Examples

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



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 |

Common Properties of a Distance

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

Common Properties of a Similarity

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

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

Similarity Between Binary Vectors

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

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

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

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

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

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

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

J = number of 11 matches / number of not-both-zero attributes values

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



SMC versus Jaccard: Example

$p = 1000000000$

$q = 0000001001$

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

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

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

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

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

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



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



SMC & JC

- SMC - Counts both presences and absences equally. Used for objects with symmetric binary attributes.
- Can be used to find students who answered similarly in a test – true/false questions
- JC is used to handle objects with assymetric binary attributes.
- **Ex: In a TDB:**
- No. of products not purchased is far more than purchased
- SMC would say all transactions are very similar.

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

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

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

- Example:

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$
$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

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

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

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

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

$\text{Cos}(x,y) = 0$ indicates both are dissimilar

$\text{Cos}(x,y) = 1$ indicates both are similar



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Extended Jaccard Coefficient (Tanimoto)

- Variation of JC
- Used for document data
- Reduces to Jaccard for binary attributes

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

Correlation

- Correlation measures the linear relationship between objects (-1 to +1)

- Pearson's correlation between x and y is

$$\text{corr}(x, y) = \frac{\text{covariance}(x, y)}{S.D(x) * S.D(y)}$$

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

$$S.D(x) = S_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$$

$$S.D(y) = S_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \bar{y})^2}$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Pearson's Correlation

- If correlation between two variables x and y is -1, they are negatively correlated.
 - If one increases, the other decreases and vice versa.
- If correlation between two variables x and y is +1, they are positively correlated.
 - Either both increase or both decrease.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.
1. For the k^{th} attribute, compute a similarity, s_k , in the range $[0, 1]$.
 2. Define an indicator variable, δ_k , for the k_{th} attribute as follows:
$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$
 3. Compute the overall similarity between the two objects using the following formula:

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

Using Weights to Combine Similarities

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

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

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

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers
- Improves the classification accuracy
- Predicted output of the base classifiers is combined by majority voting
- Build different experts and let them vote.

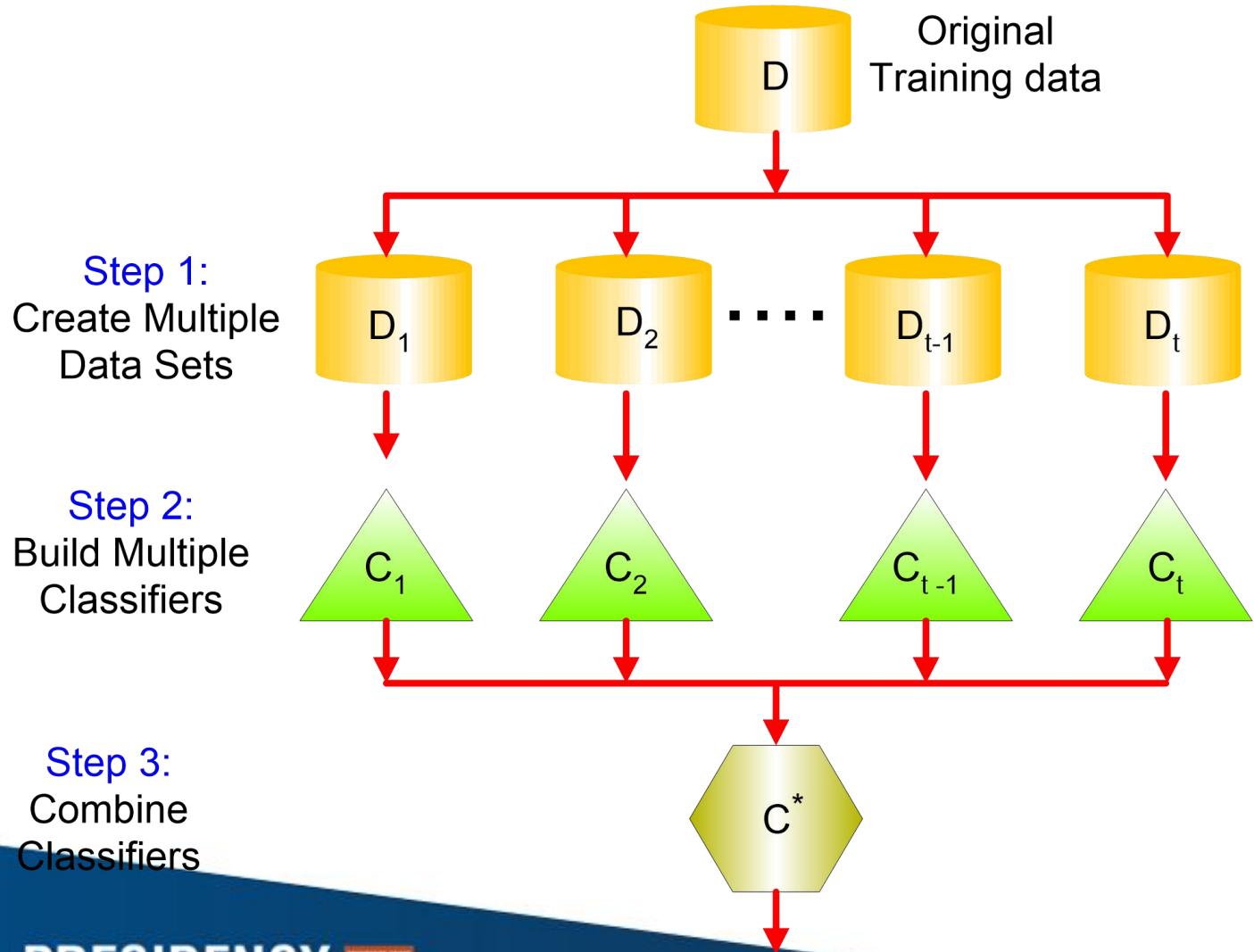


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



General Idea



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
 - Bagging }
 - Boosting }
 - **By choosing a subset of features**
 - ◆ Each training set has a subset of features chosen randomly. Base classifiers
 - ◆ Used when data set has more redundant features
 - ◆ Ex: Random forest....decision tree is base classifier
 - **By manipulating the class labels**
 - Class labels are partitioned randomly into two disjoint subsets A_0 and A_1



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Examples of Ensemble Methods by manipulating the class labels

- Instances $\in A_0$ labeled as class 0
- Instances $\in A_1$ labeled as class 1
- Relabeled examples train base classifier
- Class relabeling, model building are repeated for several iterations.
- Each iteration builds a base classifier.
- ∴ an ensemble of base classifiers is obtained.
- When a test record is presented, each base classifier C_i gives its predicted output.
- if $C_i = 0$... All classes $\in A_0$ receives a vote
- if $C_i = 1$... All classes $\in A_1$ receives a vote



Examples of Ensemble Methods

- repeated for each C_i
- Votes are tallied
- Test record is assigned to the class with highest vote.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



General Procedure for ensemble methods

1. Let D The original training data, k ... no. of base classifiers, T Test data
2. For $i = 1$ to k do
 1. Create training set D_i from D
 2. Build a base classifier C_i from D
3. End for
4. For each test record $x \in T$ do
5. $C^*(x) = \text{vote}(C_1(x), C_2(x), \dots, C_k(x))$
6. Each C_i returns its class prediction.
7. End for



PRESIDENCY
UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013



Bagging

- Sampling with replacement
- Bootstrap samples D_i , each with 63% of original data

| Original Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------------------|---|---|----|----|---|---|----|----|---|----|
| Bagging (Round 1) | 7 | 8 | 10 | 8 | 2 | 5 | 10 | 10 | 5 | 9 |
| Bagging (Round 2) | 1 | 4 | 9 | 1 | 2 | 3 | 2 | 7 | 3 | 2 |
| Bagging (Round 3) | 1 | 8 | 5 | 10 | 5 | 5 | 9 | 6 | 3 | 7 |

- Build classifier on each bootstrap sample
- Each sample has probability $(1 - 1/n)^n$ of being selected in each D_i

Example of Bagging

- Refer notes for a numerical example.
- Data Set used to construct an ensemble of bagging classifiers

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
 - Initially, all N records are assigned equal weights, $1/N$
 - Unlike bagging, weights may change at the end of boosting round
 - With each boosting sample, a classifier is induced(iteratively) and is used to classify all training examples.
 - Misclassified examples are assigned more weights for the next round.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

| Original Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------------------|---|---|---|----|---|---|---|----|---|----|
| Boosting (Round 1) | 7 | 3 | 2 | 8 | 7 | 9 | 4 | 10 | 6 | 3 |
| Boosting (Round 2) | 5 | 4 | 9 | 4 | 2 | 5 | 1 | 7 | 4 | 2 |
| Boosting (Round 3) | 4 | 4 | 8 | 10 | 4 | 5 | 4 | 6 | 3 | 4 |

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds
- Final ensemble is an aggregate of the base classifiers got from each boosting round.

How Boosting Works?

- Weights are assigned to each training tuple.
- A series of k classifiers is iteratively learnt
- After a classifier M_i is learnt, the weights are updated to allow the subsequent classifier M_{i+1} , to pay more attention to the training tuples that were misclassified by M_i .
- The final M^* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Basic Idea

- Suppose there are just 5 training examples {1, 2, 3, 4, 5}
- Initially each example has 0.2 (1/5) probability, of being sampled.
- If the boosting samples for the first round are {2,4,4,3,2}, a base classifier is built from this.
- Suppose 2,3,5 are correctly predicted by this classifier and 1,4 are wrongly predicted:
 - Weight of 1,4 is increased
 - Weight of 2,3,5 is decreased.
- Second round of boosting , again 5 samples, but now 1,4 are more likely to be sampled.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Boosting

- Is an iterative procedure.
- The distribution of training examples are adaptively changed, so that the base classifiers in the next iteration, focus more on examples that are wrongly predicted in the previous iteration.
- Boosting assigns a weight to each example.
- Weights are adaptively changed at the end of each boosting round.
- Weights assigned to the training examples are used in the following ways:-



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Boosting

1. To draw a set of bootstrap samples from the original data
2. Can be used by the base classifier to learn a model that is biased towards higher weight examples.

Steps:-

1. Initially wt of all examples are same $1/N$.
2. A sample is drawn as per the sampling distbn of the training examples to get a new training set.
3. A classifier is induced from this training set.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Boosting

4. All examples of the original data are classified using this classifier.
5. Wrongly classified examples ... increase in weight

Correctly classified examples decrease in weight.

So, wrongly classified examples will be focussed more in subsequent iterations.

6. Repeat steps 2 to 5 for k times($k = \text{no. of base classifiers}$)



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Boosting

7. As boosting round proceeds, wrongly classified examples become more prevalent.
8. Final ensemble is got by aggregating base classifiers got from each boosting round.

Several implementations of the boosting algorithm have been developed. They all differ in terms of

- 1) How the weights of the examples are updated.
- 2) How the predictions of the base classifiers are combined.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Example: AdaBoost

1. Let $\{(x_j, y_j) \mid j = 1, 2, 3, 4, \dots, N\}$ is a set of N training examples.

- Base classifiers: C_1, C_2, \dots, C_T

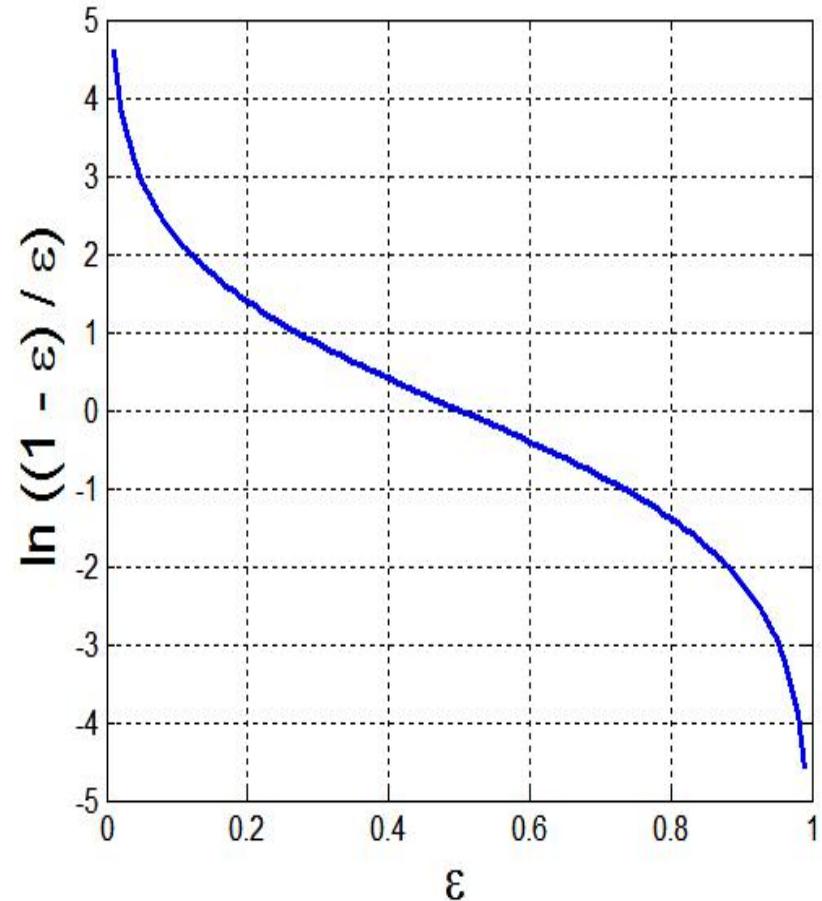
2. Error rate of a base classifier:

$$\varepsilon_i = \frac{1}{N} \sum_{j=1}^N w_j \delta(C_i(x_j) \neq y_j)$$

where $\delta(P) = 1$ if P is true
= 0 otherwise

- Importance of a classifier:

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_i}{\varepsilon_i} \right)$$



Example: AdaBoost

3. α_i is also used to update the weight of training examples.

Weight update:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

where Z_j is the normalization factor

Z_j is used to ensure that $\sum w_i^{j+1} = 1$

- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to $1/n$ and the resampling procedure is repeated



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Ada Boost

4. Instead of majority voting, the prediction made by each C_j is weighted according to α_j . The weighted average of this is the final ensemble.

Classification

$$C^*(x) = \arg \max_y \sum_{j=1}^T \alpha_j \delta(C_j(x) = y)$$

[Refer book for the algorithm
Refer transparencies for a numerical example]

Class Imbalance Problem

- In many real time data sets the class distbn is imbalanced.
- Correct classfn of the rare class is of great value, than the correct classfn of the majority class.
- ∵ accuracy can't be used to evaluate the classifiers.
- ∵ other evaluation metrics based on ROC are used.

Alternative metrics:-

Rare class +ve class

Majority class..... -ve class.

TPR, TNR, FPR, FNR, Recall, Precision and F_1 measure

Multi-class problem

- Binary classifiers can be extended to handle multiclass problems.
- Ex:- $Y = \{y_1, y_2, \dots, y_k\}$ where $y_i \in Y$ is the set of classes of the input data.
- **Approach 1:- (one-against-rest) (1-r)**
- Multiclass problem – K binary problems.
- K iterations K binary classifiers are built.
- In iteration i,
 - $y_i \in Y$ are taken as +ve class examples .
 - all other classes are taken as -ve examples.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Multi-class problem

- **Approach 2:-** (one-against-one) (1 – 1)
- $\frac{k(k-1)}{2}$ binary classifiers are built.
- Each classifier distinguishes a pair of classes (y_i, y_j)
- Instances $\notin (y_i, y_j)$ are ignored.

Classifying a test instance:-

- In both (1-r) and (1-1), the predictions made by all base classifiers are combined by majority voting.

Multi-class problem

- Ex:- $Y = \{y_1, y_2, y_3, y_4\}$. Suppose a test instance is classified as (+, -, -, -) by (1-r)approach. What is its predicted class?

| Base Classifier | +ve example | -ve examples | vote |
|-----------------|-------------|---------------|---------------|
| B_1 | y_1 | $y_2 y_3 y_4$ | y_1 |
| B_2 | y_2 | $y_1 y_3 y_4$ | $y_1 y_3 y_4$ |
| B_3 | y_3 | $y_1 y_2 y_4$ | $y_1 y_2 y_4$ |
| B_4 | y_4 | $y_1 y_2 y_3$ | $y_1 y_2 y_3$ |



PRESIDENCY
UNIVERSITY



Private University Estd. in Karnataka State by Act No. 41 of 2013

Humanity voting class of test instance is v.

Using 1-1 approach:-

- If the test instance is classified as shown:

| | | | | | | |
|-------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Binary pair of classes | + : y_1 | + : y_1 | + : y_1 | + : y_2 | + : y_2 | + : y_3 |
| | - : y_2 | - : y_3 | - : y_4 | - : y_3 | - : y_4 | - : y_4 |
| Classfn | + | + | - | + | - | + |

- Votes:- $y_1 = 2$, $y_2 = 1$, $y_3 = 1$, $y_4 = 2$
- \therefore test record is classified as y_1 or y_4 depending on the tie breaking procedure.

Error-correcting output coding:-

- Using Hamming distance – distance of two bit strings is the no. of bits that differ.

| Class | Code word |
|-------|---------------|
| y_1 | 1 1 1 1 1 1 1 |
| y_2 | 0 0 0 0 1 1 1 |
| y_3 | 0 0 1 1 0 0 1 |
| y_4 | 0 1 0 1 0 1 0 |

- Each class is encoded by a 7-bit code-word.
- Each bit b_i of the code word is used to train a binary classifier.
- If a test instance is classified as {0,1,1,1,1,1,1} by all binary classifiers.

Error-correcting output coding:-

- Hamming distance between test code word and y_1 is 1, with remaining classes is 3.
- \therefore test instance is classified as y_1 .



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Sampling

- Common approach for selecting a subset of data objects to be analyzed
 - Select only *some* instances for the training set, instead of *all* of them
- 1. Sampling without replacement - as each item is sampled, it is removed from the population
- 2. Sampling with replacement - the same object/instance can be picked more than once

Ensemble Methods

- Currently using one *single classifier* induced from training data as our model, to predict class of test instance
- What if we used *multiple decision trees*?
- Motivation: committee of experts working together are likely to better solve a problem than a single expert
 - ▣ But no “group think”: each model should make predictions independently of other models in the ensemble
- In practice: methods work surprisingly well, usually greatly improve decision tree accuracy

Ensemble Characteristics

1. Build *multiple* models from the same training data by creating each model on a *modified* version of the training data.
2. Make a final, ensemble prediction by aggregating the predictions of the individual models
 - Classification prediction: Let each model have a vote on the correct class prediction. Assign the class with the most votes.
 - Regression prediction: Measure of central tendency (mean or median)

Rationale

- How can an ensemble method improve a classifier's performance?
 - Assume we have 25 binary classifiers
 - Each has error rate: $\varepsilon = 0.35$
1. If all 25 classifiers are identical:
 - ▣ They will vote the same way on each test instance
 - ▣ Ensemble error rate: $\varepsilon = 0.35$

Rationale

- How can an ensemble method improve a classifier's performance?
 - Assume we have 25 binary classifiers
 - Each has error rate: $\varepsilon = 0.35$
2. If all 25 classifiers are independent (errors are uncorrelated):
- Ensemble method only makes a wrong prediction if more than half of the base classifiers predict incorrectly.

$$e_{\text{ensemble}} = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

6% much less than 35%

Rati

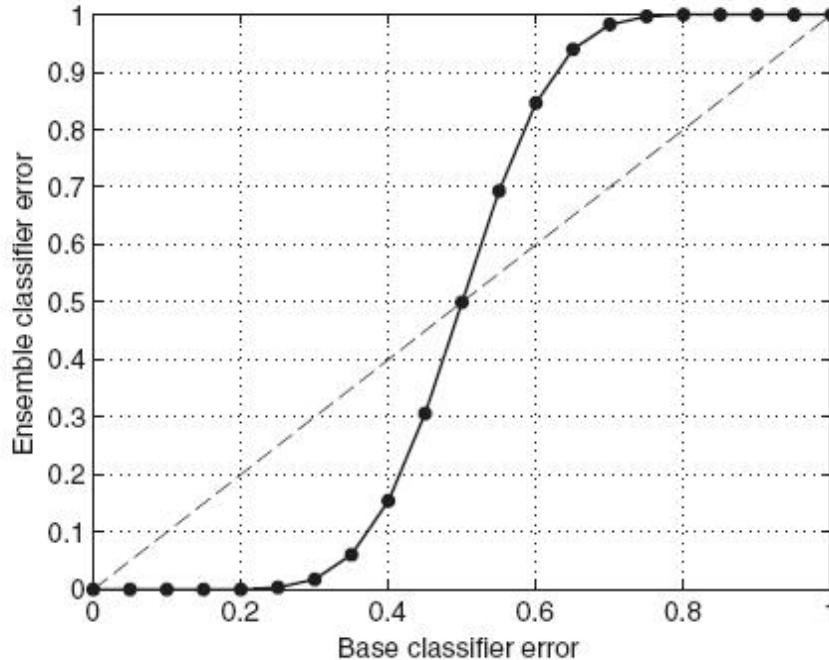
- In practice, difficult to have base classifiers than are completely independent.
- Ensemble methods have been shown to improve classification accuracies even when there is some correlation.

- Conditions necessary for an ensemble classifier to perform better than a single classifier:
 1. Base classifiers should be *independent* of each other
 2. Base classifiers should not do worse than a classifier doing random guessing
 - Example: for two-class problem, base classifier error rate: $\epsilon < .5$

Error Rate Comparison

Ensemble method performs worse than single base classifier when $\varepsilon > 0.5$

Comparison of 25 base classifiers when error rate is varied



- Dashed diagonal line: when base classifiers are identical
- Solid curve: when base classifiers are independent

Figure 5.30. Comparison between errors of base classifiers and errors of the ensemble classifier.

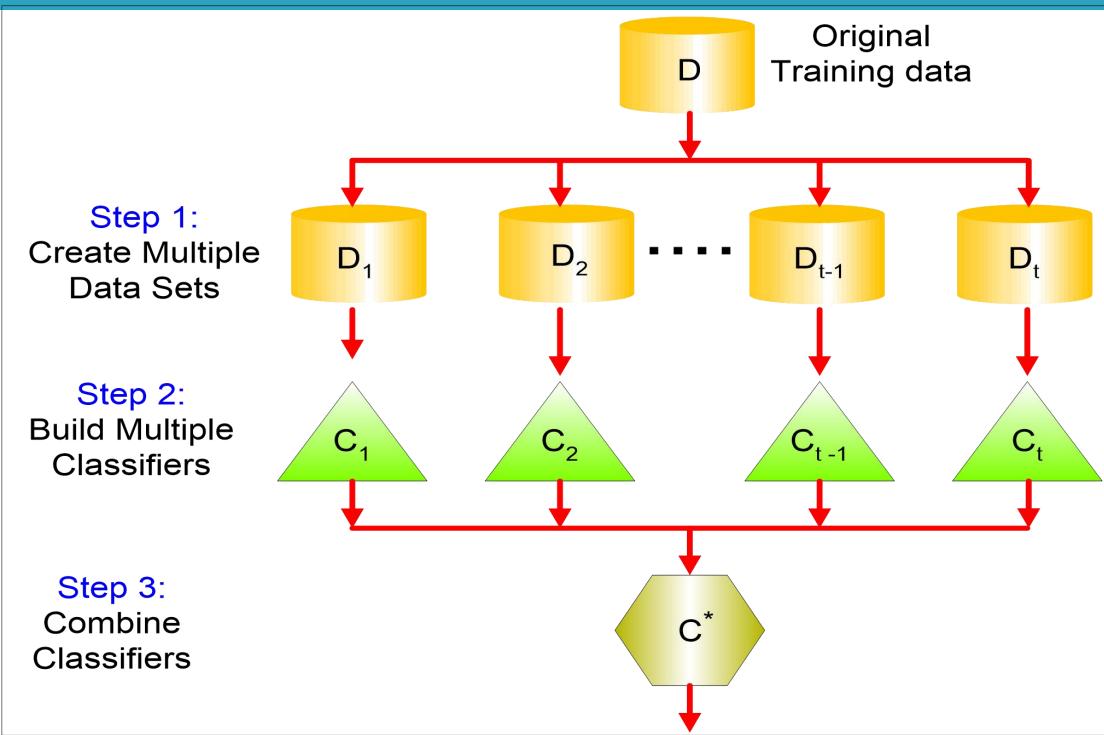
Bootstrap

- What is the chance that a particular instance will not be picked for the training set?
- Instance has $(1/n)$ probably of being picked each time...
 - So a $1 - (1/n)$ probably of not being picked each time

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

- 36.8% chance a particular instance of not being picked
- Training set will end up with approximately 63.2% of the original instances (for large enough n)

Methods for Constructing an Ensemble Classifier



Methods for Constructing an Ensemble Classifier

1. By manipulating the training set.
2. By manipulating the input features.
3. By manipulating the class labels.
4. By manipulating the learning algorithm.

Methods for Constructing an Ensemble Classifier

1. By manipulating the training set.

- Multiple training sets created by resampling original data
- Resample according to some sampling distribution
 - Example: equal probability
 - Example: weighted
 - Determines how likely it is that an example will be selected for training.
- Classifier built from each training set.
- Ensemble methods that manipulate the training set:
 1. Bagging
 2. Boosting

Methods for Constructing an Ensemble Classifier

2. By manipulating the input features.

- Subset of input features is chosen at random from overall collection of features
- Each training set has different feature set
- Ensemble method that manipulates input features:
 1. Random Forest
 - Works well with datasets that contain highly redundant features

Methods for Constructing an Ensemble Classifier

3. By manipulating the class labels.

- Used when large number of classes
- Transform into many binary class problems
- Training Approach:
 1. Randomly partition class labels into two disjoint subsets A_0 (class 0) and A_1 (class 1)
 2. Train a base classifier based on this class reassignment.
 3. Repeat multiple times, once for each base classifier.
- Testing Approach:
 1. Each base classifier predicts test instance with its respective binary class subset
 2. All classes in subset receive a vote
 3. Class with highest count wins

Methods for Constructing an Ensemble Classifier

4. By manipulating the learning algorithm.

- ... so that applying algorithm on same training data may result in different models
- How to introduce randomness into decision tree induction?
 - ▣ Instead of choose best splitting attribute at each node, randomly choose one of top k attributes for splitting

Ensemble Methods

- Ensemble methods work best with unstable classifiers, base classifiers that are sensitive to minor perturbations in the training set.
 - Example: decision trees
 - Unstable classifiers have *high variability*.

Decision Tree Model Ensembles

- 
1. Boosting
 2. Bagging
 3. Random Forests

Bagging

On average, each D_i will contain 63% of original training data.
Probability of sample being selected for D_i : $1 - (1 - (1/N)^N)$

- Converges to: $1 - 1/e = 0.632$

- Ensemble method that “manipulates the training set”
- Action: repeatedly sample with replacement according to uniform probability distribution
 - Every instance has equal chance of being picked
 - Some instances may be picked multiple times; others may not be chosen
- Sample Size: same as training set
- D_i : each bootstrap sample
- Footnote: also called bootstrap aggregating

Consequently, every bootstrap sample will be missing some of the instances from the dataset so each bootstrap sample will be different and this means that models trained on different bootstrap samples will also be different

Bagging Algorithm

Model Generation:

- Let n be the number of instances in the training data.
- For each of t iterations:
 - Sample n instances with replacement from training data.
 - Apply the learning algorithm to the sample.
 - Store the resulting model.

Classification:

- For each of the t models:
 - Predict class of instance using model.
- Return class that has been predicted most often.

Now going to apply bagging and create many decision stump base classifiers.

Bagging Example

Dataset: 10 instances

Predictor Variable: x

Target Variable: y

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

- Decision Stump: one-level binary decision tree
- Splitting condition will be $x \leq k$, where k is the split point
 - Best splits: $x \leq 0.35$ or $x \leq 0.75$
 - Best accuracy: 60%

Bagging Example

- First choose how many “bagging rounds” to perform
 - Chosen by analyst
- We’ll do 10 bagging rounds in this example:

In each round, create D_i by sampling with replacement

Bagging Example

Round 1:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| X | 0.1 | 0.2 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.6 | 0.9 | 0.9 |
| Y | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |

Learn decision stump.

What stump will be learned?

If $x \leq 0.35$ then $y = 1$

If $x > 0.35$ then $y = -1$

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| X | 0.1 | 0.2 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.6 | 0.9 | 0.9 |
| Y | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.8 | 0.9 | 1 | 1 | 1 |
| Y | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 |
| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 |
| Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |
| X | 0.1 | 0.1 | 0.2 | 0.4 | 0.4 | 0.5 | 0.5 | 0.7 | 0.8 | 0.9 |
| Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |
| X | 0.1 | 0.1 | 0.2 | 0.5 | 0.6 | 0.6 | 0.6 | 1 | 1 | 1 |
| Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| X | 0.2 | 0.4 | 0.5 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
| Y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| X | 0.1 | 0.4 | 0.4 | 0.6 | 0.7 | 0.8 | 0.9 | 0.9 | 0.9 | 1 |
| Y | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 |
| X | 0.1 | 0.2 | 0.5 | 0.5 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
| Y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| X | 0.1 | 0.3 | 0.4 | 0.4 | 0.6 | 0.7 | 0.7 | 0.8 | 1 | 1 |
| Y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| X | 0.1 | 0.1 | 0.1 | 0.1 | 0.3 | 0.3 | 0.8 | 0.8 | 0.9 | 0.9 |
| Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

If $x \leq 0.35$ then $y = 1$
If $x > 0.35$ then $y = -1$
If $x \leq 0.65$ then $y = -1$
If $x > 0.65$ then $y = 1$
If $x \leq 0.35$ then $y = 1$
If $x > 0.35$ then $y = -1$
If $x \leq 0.3$ then $y = 1$
If $x > 0.3$ then $y = -1$
If $x \leq 0.35$ then $y = 1$
If $x > 0.35$ then $y = -1$
If $x \leq 0.75$ then $y = -1$
If $x > 0.75$ then $y = 1$
If $x \leq 0.75$ then $y = -1$
If $x > 0.75$ then $y = 1$
If $x \leq 0.75$ then $y = -1$
If $x > 0.75$ then $y = 1$
If $x \leq 0.75$ then $y = -1$
If $x > 0.75$ then $y = 1$
If $x \leq 0.05$ then $y = -1$
If $x > 0.05$ then $y = 1$

10 bagging rounds. 10 D_i 's. 10 learned models.

Classify test instance by using each base classifier and taking majority vote.

Bagging Example

10 test instances. Let's see how each classifier votes:

100% overall ensemble method accuracy (*improvement from 60%*)

| Round | $x=0.1$ | $x=0.2$ | $x=0.3$ | $x=0.4$ | $x=0.5$ | $x=0.6$ | $x=0.7$ | $x=0.8$ | $x=0.9$ | $X=1.0$ |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 4 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 5 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 6 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 7 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 8 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 9 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sum | 2 | 2 | 2 | -6 | -6 | -6 | -8 | 0 | 0 | 0 |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| True | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

Bagging Summary

- *In Previous Example:* even though base classifier was decision stump ($\text{depth}=1$), bagging aggregated the classifiers to effectively learn a decision tree of $\text{depth}=2$
- Bagging helps to reduce variance
- Bagging does not focus on any particular instance of training data
 - less susceptible to model overfitting with noisy data
 - robust to minor perturbations in training set
- If base classifiers are stable (not much variance), bagging can degrade performance
 - Training set is $\sim 37\%$ smaller than original data

Boosting

- Boosting works by iteratively creating models and adding them to the ensemble
- Iteration stops when a predefined number of models have been added
- Each new model added to the ensemble is biased to pay more attention to instances that previous models misclassified (weighted dataset).

General Boosting Algorithm

1. Initially instances are assigned weights of $1/N$
 - Each is equally likely to be chosen for sample
2. Sample drawn *with replacement*: D_i
3. Classifier induced on D_i
4. Weights of training examples are updated:
 - Instances classified incorrectly have weights increased
 - Instances classified correctly have weights decreased

General Boosting Algorithm

- During each iteration the algorithm:
 1. Induces a model and calculates the total error, ϵ , by summing the weights of the training instances for which the predictions made by the model are incorrect.
 2. Increases the weights for the instances misclassified
 3. Decreases the weights for the instances correctly classified
 4. Calculate a confidence factor α , for the model such that α increases as ϵ decreases

$$\mathbf{w}[i] \leftarrow \mathbf{w}[i] \times \left(\frac{1}{2 \times \epsilon} \right)$$

$$\mathbf{w}[i] \leftarrow \mathbf{w}[i] \times \left(\frac{1}{2 \times (1 - \epsilon)} \right)$$

$$\alpha = \frac{1}{2} \times \log_e \left(\frac{1 - \epsilon}{\epsilon} \right)$$

Boosting Example

Boosting (Round 1)

| | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|---|
| 7 | 3 | 2 | 8 | 7 | 9 | 4 | 10 | 6 | 3 |
|---|---|---|---|---|---|---|----|---|---|

- Suppose that *Instance #4* is hard to classify.
- Weight for this instance will be increased in future iterations, as it gets misclassified repeatedly.
- Examples not chosen in previous round (*Instances #1, #5*) also may have better chance of being selected in next round.
 - Why? Predictions in previous round are likely to be wrong since they weren't trained on.

Boosting (Round 2)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 9 | 4 | 2 | 5 | 1 | 7 | 4 | 2 |
|---|---|---|---|---|---|---|---|---|---|

Boosting (Round 3)

| | | | | | | | | | |
|---|---|---|----|---|---|---|---|---|---|
| 4 | 4 | 8 | 10 | 4 | 5 | 4 | 6 | 3 | 4 |
|---|---|---|----|---|---|---|---|---|---|

- As boosting rounds proceed, instances that are the hardest to classify become even more prevalent.

Prediction

- Once the set of models have been created the ensemble makes predictions using a weighted aggregate of the predictions made by the individual models.
- The weights used in this aggregation are simply the confidence factors associated with each model.

Boosting Algorithms

- Several different boosting algorithms exist
- Different by:
 1. How weights of training instances are updated after each boosting round
 2. How predictions made by each classifier are combined
 - Each boosting round produces one base classifier

AdaBoost

- AdaBoost is a popular boosting algorithm
- Regarding predictions of final ensemble classifier:
 - Importance of a base classifier depends on its error rate

$$\varepsilon_i = \frac{1}{N} \left[\sum_{j=1}^N \omega_j I(C_i(x_j) \neq y_j) \right]$$

$I(p)$ = 1 if predicate p is true, and 0 otherwise

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_i}{\varepsilon_i} \right)$$

C_i : Base Classifier

ε_i : error rate

α_i : importance of classifier

AdaBoost

- α_i also used to update weight of training examples after each boosting round

$$\omega_i^{(j+1)} = \frac{\omega_i^{(j)}}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

j : current round
 $j+1$: next round
 Z_j : normalization factor

$$\sum_i \omega_i^{(j+1)} = 1$$

- Increases weights of incorrectly classified instances
- Decreases weights of correctly classified instances

AdaBoost Example

Dataset: 10 instances

Predictor Variable: x

Target Variable: y

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

- Initially all instances have equal weights

AdaBoost Example

AdaBoost Example

| Initial Weights: | X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Round 1: | ω_i | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | X | 0.1 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 1 |
| | Y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

- Going to learn decision stump base classifier
- What is the model?

Model:

If $x \leq 0.75$ then $y = -1$

If $x > 0.75$ then $y = 1$

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| True Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| Pred. Y | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$\varepsilon_1 = \frac{1}{10} \left[\sum \omega_j \cdot I \right] = \frac{1}{10} \left[\sum 1 \cdot I \right] = 0.03$$

$$\text{Classifier Importance } \alpha_1 = \frac{1}{2} \ln \left(\frac{1 - .03}{.03} \right) \approx 1.738$$

AdaBoost Example

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| ω_1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Update: | $0.1 \times e^{1.738}$ | $0.1 \times e^{1.738}$ | $0.1 \times e^{1.738}$ | $0.1 \times e^{-1.738}$ |
| | 0.568 | 0.568 | 0.568 | 0.018 | 0.018 | 0.018 | 0.018 | 0.018 | 0.018 | 0.018 |

$$\Sigma = .568 \times 3 + .018 \times 7 = 1.83$$

Normalizing: (diving each by sum)

| ω_2 | 0.311 | 0.311 | 0.311 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
|------------|-------|-------|-------|------|------|------|------|------|------|------|
|------------|-------|-------|-------|------|------|------|------|------|------|------|

$$\Sigma = 1$$

| Initial | X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---------------------|------------|-------|-------|-------|------|------|------|------|------|------|------|
| Weights: | ω_i | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Round 1: | X | 0.1 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 1 |
| | Y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |
| Updated Weights: | X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| | ω_i | 0.311 | 0.311 | 0.311 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| | X | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |
| | Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

 $\Sigma=1$

Model:

If $x \leq 0.05$ then $y = -1$ If $x > 0.05$ then $y = 1$ $\Sigma=1$

- Going to learn decision stump base classifier
- What is the model?

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| True Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| Pred. Y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$$\varepsilon_2 = \frac{1}{10} [\sum \omega_j \cdot I] = \frac{1}{10} (0.01 * 4) = 0.004$$

$$\text{Classifier Importance } \alpha_2 = \frac{1}{2} \ln \left(\frac{1 - .004}{.004} \right) = 2.7587$$

AdaBoost Example

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|------------|----------------------------|----------------------------|----------------------------|--------------------------|--------------------------|--------------------------|--------------------------|---------------------------|---------------------------|---------------------------|
| ω_1 | 0.311 | 0.311 | 0.311 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| Update: | $0.311 \times e^{-2.7587}$ | $0.311 \times e^{-2.7587}$ | $0.311 \times e^{-2.7587}$ | $0.01 \times e^{2.7587}$ | $0.01 \times e^{2.7587}$ | $0.01 \times e^{2.7587}$ | $0.01 \times e^{2.7587}$ | $0.01 \times e^{-2.7587}$ | $0.01 \times e^{-2.7587}$ | $0.01 \times e^{-2.7587}$ |
| | 0.019709 | 0.019709 | 0.019709 | 0.157793 | 0.157793 | 0.157793 | 0.157793 | 0.000633 74 | 0.000633 74 | 0.000633 74 |

$$\Sigma = 0.019709 \times 3 + \mathbf{0.157793} \times 4 + 0.000633 \times 3 = 0.6922$$

Normalizing: (diving each by sum)

| ω_2 | 0.085419 | 0.085419 | 0.085419 | 0.911834 | 0.911834 | 0.911834 | 0.911834 | 0.002747 | 0.002747 | 0.002747 |
|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|

$$\Sigma = 1$$

Calculations.....



AdaBoost Example

| Round | Split Point | Left Class | Right Class |
|-------|-------------|------------|-------------|
| 1 | 0.75 | -1 | 1 |
| 2 | 0.05 | -1 | 1 |
| 3 | 0.3 | 1 | -1 |

- Going to learn decision stump base classifier
- What is the model?

Model:

If $x \leq 0.3$ then $y = 1$

If $x > 0.3$ then $y = -1$

| X | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----------|-----------|-----------|
| True Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| Pred. Y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

$$\varepsilon_1 = \frac{1}{10} \left[\sum \omega_j \cdot I \right] = \frac{1}{10} (0.002747 * 3) = 0.0008241$$

$$\text{Classifier Importance } \alpha_1 = \frac{1}{2} \ln \left(\frac{1 - 0.0008241}{0.0008241} \right) = 3.55019$$

AdaBoost Prediction

- Prediction made by each base classifier C_i is weighted by α_i

AdaBoost Example

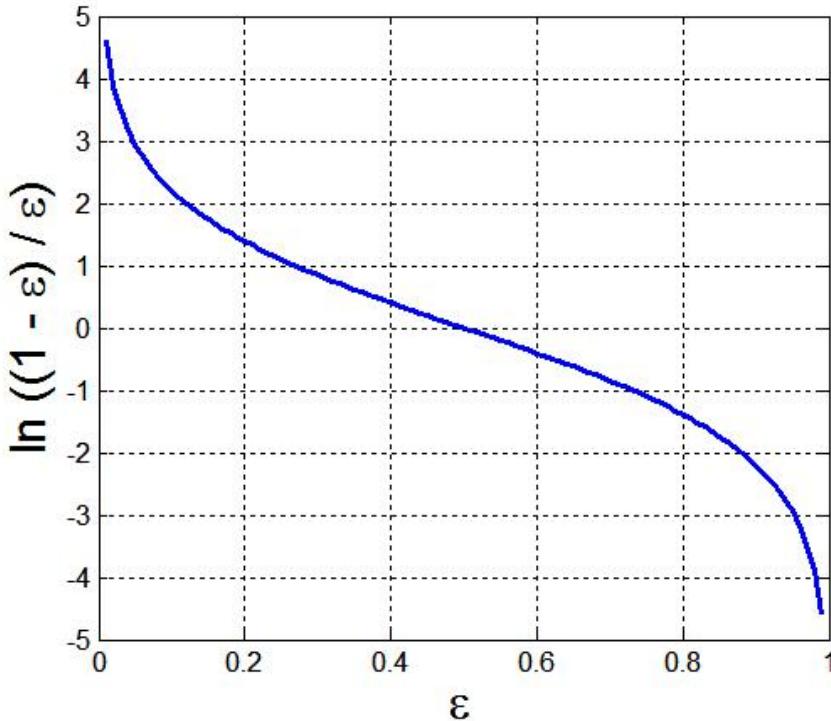
| Round | Split Point | Left Class | Right Class | α |
|-------|-------------|------------|-------------|----------|
| 1 | 0.75 | -1 | 1 | 1.738 |
| 2 | 0.05 | 1 | 1 | 2.7587 |
| 3 | 0.3 | 1 | -1 | 3.55019 |

| Round | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|-------|------|------|------|-------|-------|-------|-------|------|------|------|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| Sum | 4.57 | 4.57 | 4.57 | -2.53 | -2.53 | -2.53 | -2.53 | 0.95 | 0.95 | 0.95 |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$-1 \times 1.738 + 1 \times 2.7587 + 1 \times 3.55019 = 4.570$$

AdaBoost

- α_i has a large positive value if error rate is close to 0
- α_i has a large negative value if error rate is close to 1



Boosting Algorithm

From Tan, Alg. 5.7

Algorithm 5.7 AdaBoost algorithm.

- 1: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$. {Initialize the weights for all N examples.}
- 2: Let k be the number of boosting rounds.
- 3: **for** $i = 1$ to k **do**
- 4: Create training set D_i by sampling (with replacement) from D according to \mathbf{w} .
- 5: Train a base classifier C_i on D_i .
- 6: Apply C_i to all examples in the original training set, D .
- 7: $\epsilon_i = \frac{1}{N} \left[\sum_j w_j \delta(C_i(x_j) \neq y_j) \right]$ {Calculate the weighted error.}
- 8: **if** $\epsilon_i > 0.5$ **then**
- 9: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$. {Reset the weights for all N examples.}
- 10: Go back to Step 4.
- 11: **end if**
- 12: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$.
- 13: Update the weight of each example according to Equation 5.69.
- 14: **end for**
- 15: $C^*(\mathbf{x}) = \operatorname{argmax}_y \sum_{j=1}^T \alpha_j \delta(C_j(\mathbf{x}) = y)$.

Bagging vs. Boosting

- Boosting: Sample with nonuniform distribution
 - ▣ Unlike bagging where each instance had equal chance of being selected
 - ▣ *Boosting Motivation:* focus on instances that are harder to classify
 - *How:* give harder instances more weight in future rounds

References

- *Fundamentals of Machine Learning for Predictive Data Analytics*, 1st Edition, Kelleher et al.
- *Data Science from Scratch*, 1st Edition, Grus
- *Data Mining and Business Analytics in R*, 1st edition, Ledolter
- *An Introduction to Statistical Learning*, 1st edition, James et al.
- *Discovering Knowledge in Data*, 2nd edition, Larose et al.
- *Introduction to Data Mining*, 1st edition, Tam et al.

Bagging Round 1:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.1 | 0.2 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.6 | 0.9 | 0.9 |
| y | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |

$$x \leq 0.35 \Rightarrow y = 1$$

$$x > 0.35 \Rightarrow y = -1$$

Bagging Round 2:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|---|---|---|
| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.8 | 0.9 | 1 | 1 | 1 |
| y | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 |

$$x \leq 0.65 \Rightarrow y = 1$$

$$x > 0.65 \Rightarrow y = -1$$

Bagging Round 3:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 |
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

$$x \leq 0.35 \Rightarrow y = 1$$

$$x > 0.35 \Rightarrow y = -1$$

Bagging Round 4:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.1 | 0.1 | 0.2 | 0.4 | 0.4 | 0.5 | 0.5 | 0.7 | 0.8 | 0.9 |
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

$$x \leq 0.3 \Rightarrow y = 1$$

$$x > 0.3 \Rightarrow y = -1$$

Bagging Round 5:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|---|---|---|
| x | 0.1 | 0.1 | 0.2 | 0.5 | 0.6 | 0.6 | 0.6 | 1 | 1 | 1 |
| y | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$x \leq 0.35 \Rightarrow y = 1$$

$$x > 0.35 \Rightarrow y = -1$$

Bagging Round 6:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x | 0.2 | 0.4 | 0.5 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
| y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$x \leq 0.75 \Rightarrow y = -1$$

$$x > 0.75 \Rightarrow y = 1$$

Bagging Round 7:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x | 0.1 | 0.4 | 0.4 | 0.6 | 0.7 | 0.8 | 0.9 | 0.9 | 0.9 | 1 |
| y | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 |

$$x \leq 0.75 \Rightarrow y = -1$$

$$x > 0.75 \Rightarrow y = 1$$

Bagging Round 8:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x | 0.1 | 0.2 | 0.5 | 0.5 | 0.5 | 0.7 | 0.7 | 0.8 | 0.9 | 1 |
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$x \leq 0.75 \Rightarrow y = -1$$

$$x > 0.75 \Rightarrow y = 1$$

Bagging Round 9:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|---|---|
| x | 0.1 | 0.3 | 0.4 | 0.4 | 0.6 | 0.7 | 0.7 | 0.8 | 1 | 1 |
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

$$x \leq 0.75 \Rightarrow y = -1$$

$$x > 0.75 \Rightarrow y = 1$$

Bagging Round 10:

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.1 | 0.1 | 0.1 | 0.1 | 0.3 | 0.3 | 0.8 | 0.8 | 0.9 | 0.9 |
| y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$$x \leq 0.05 \Rightarrow y = -1$$

$$x > 0.05 \Rightarrow y = 1$$

| Round | x=0.1 | x=0.2 | x=0.3 | x=0.4 | x=0.5 | x=0.6 | x=0.7 | x=0.8 | x=0.9 | x=1.0 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 4 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 5 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 6 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 7 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 8 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 9 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sum | 2 | 2 | 2 | -6 | -6 | -6 | -6 | 2 | 2 | 2 |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| True Class | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

EX: Boosting Round 1:

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| x | 0.1 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 1 |
| y | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |

Boosting Round 2:

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |
| y | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Boosting Round 3:

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| x | 0.2 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 0.5 | 0.6 | 0.6 | 0.7 |
| y | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

Weights of the training records.

| Round | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2 | 0.311 | 0.311 | 0.311 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| 3 | 0.029 | 0.029 | 0.029 | 0.228 | 0.228 | 0.228 | 0.228 | 0.009 | 0.009 | 0.009 |

Decision Stump is the base classifier.

| Round | Split point | left class | Right class | α |
|-------|-------------|------------|-------------|----------|
| 1 | 0.75 | -1 | 1 | 1.738 |
| 2 | 0.05 | -1 | 1 | 2.7784 |
| 3 | 0.3 | 1 | -1 | 4.1195 |

Details of Round 1:-

Original training set:-

| x | y | wt | P(x) |
|-----|----|-----|------|
| 0.1 | 1 | 0.1 | -1 x |
| 0.2 | 1 | 0.1 | -1 x |
| 0.3 | 1 | 0.1 | -1 x |
| 0.4 | -1 | 0.1 | -1 |
| 0.5 | -1 | 0.1 | -1 |
| 0.6 | -1 | 0.1 | -1 |
| 0.7 | -1 | 0.1 | -1 |
| 0.8 | +1 | 0.1 | 1 |
| 0.9 | 1 | 0.1 | 1 |
| 1 | 1 | 0.1 | 1 |

$\therefore 3$ wrong predictions

$$\epsilon_i = \frac{1}{10} [3 \times 0.1]$$

$$= 0.03$$

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - \epsilon_i}{\epsilon_i} \right)$$

$$= \frac{1}{2} \ln (32.33)$$

$$= \frac{1}{2} \times 3.48$$

$$= 1.738$$

$$\lambda = 1.738$$

wt updation:

| <u>x</u> | <u>WP/CP</u> | <u>old wt</u> | <u>Newwt</u> | $\exp(-\lambda x) =$ <u>Normal wt</u> |
|----------|--------------|---------------|--------------------|--|
| 0.1 | x | 0.1 | 0.1×5.69 | |
| 0.2 | x | 0.1 | 0.1×5.69 | 0.569 |
| 0.3 | x | 0.1 | 0.1×5.69 | |
| 0.4 | ✓ | 0.1 | 0.1×0.176 | |
| 0.5 | ✓ | 0.1 | 0.1×0.176 | |
| 0.6 | ✓ | 0.1 | 0.1×0.176 | 0.0176 |
| 0.7 | ✓ | 0.1 | " | 0.02 |
| 0.8 | ✓ | 0.1 | " | |
| 0.9 | ✓ | 0.1 | " | |
| 1 | ✓ | 0.1 | " | |

| <u>λ</u> | <u>$\exp(\lambda)$</u> | <u>$\exp(-\lambda)$</u> |
|-----------------------------|-----------------------------------|------------------------------------|
| 1.738 | 5.69 | 0.17587 |

Normalizing the wts:

Sum of the wts: $3 \times 0.569 + 7 \times 0.0176$

$$= 1.707 + 0.1232$$

$$= 1.8302$$

Divide each wt by the sum of wts.

Combining classifiers

| Round | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|-------|------|------|------|-------|-------|-------|-------|-----|-----|----|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| Sum | 5.16 | 5.16 | 5.16 | -3.08 | -3.08 | -3.08 | 0.397 | → | | |
| Sign | 1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 |

Data Mining Cluster Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 8

Introduction to Data Mining

by

Tan, Steinbach, Kumar



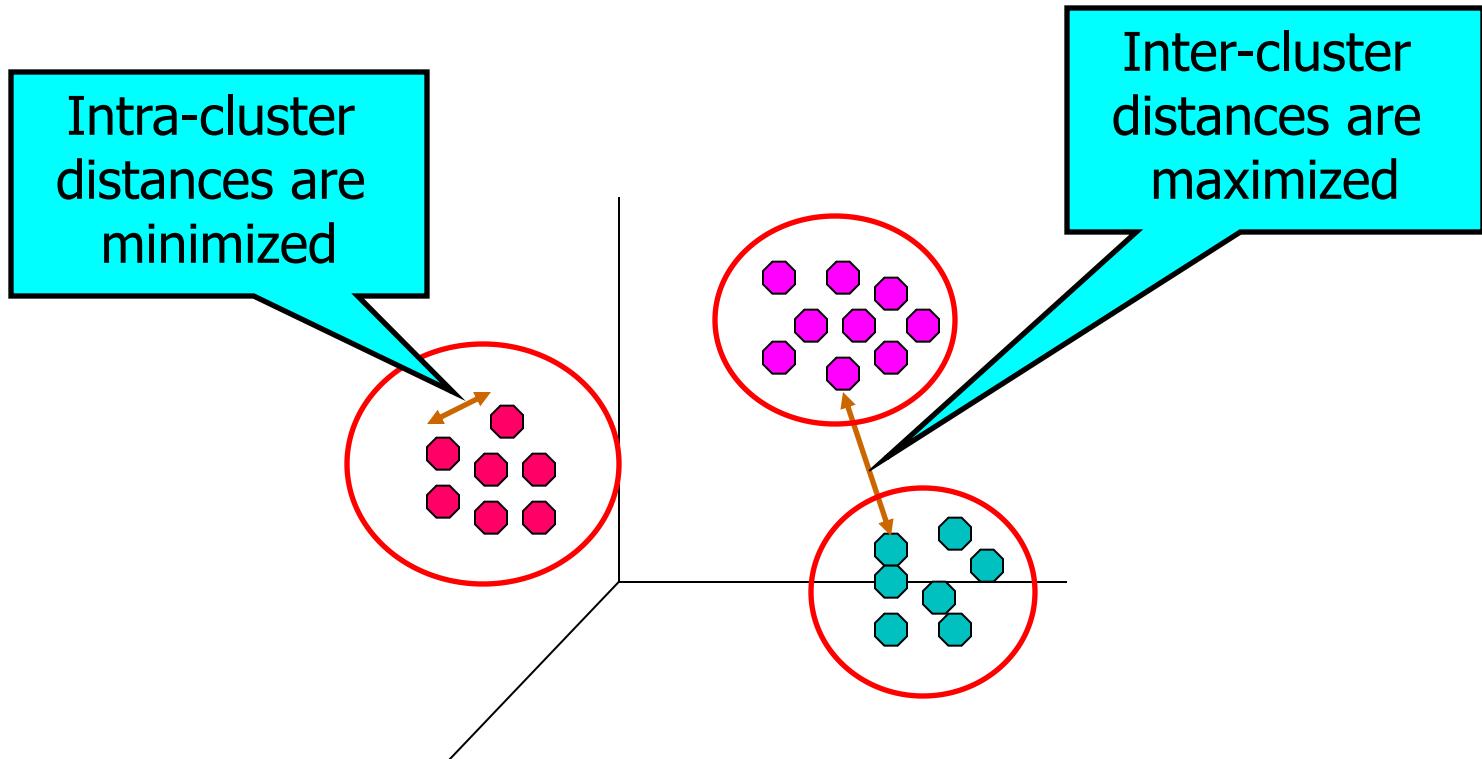
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



What is Cluster Analysis?

- 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



Clustering

- Clustering of data is a method by which large sets of data is grouped into clusters of smaller sets of similar data
- Objects in one cluster have high similarity to each other and are dissimilar to objects in other clusters
- An example of unsupervised learning
- Group objects that share common characteristics



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earthquake studies: Observed earth quake epicenters should be clustered along continent faults

What Is Good Clustering?

- A good clustering method will produce high quality clusters with
 - high intra-class similarity
 - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.



**PRESIDENCY
UNIVERSITY**

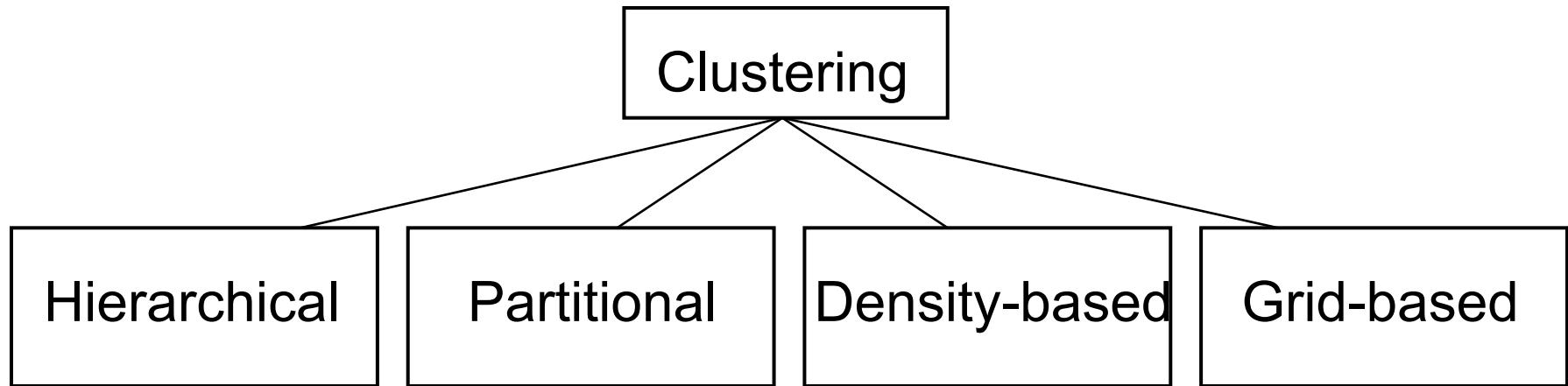
Private University Estd. in Karnataka State by Act No. 41 of 2013



Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Clustering Approaches



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Partitioning Methods

Given a DB of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$ such that

1. Each group must contain atleast one object, and
2. Each object must belong to exactly one group

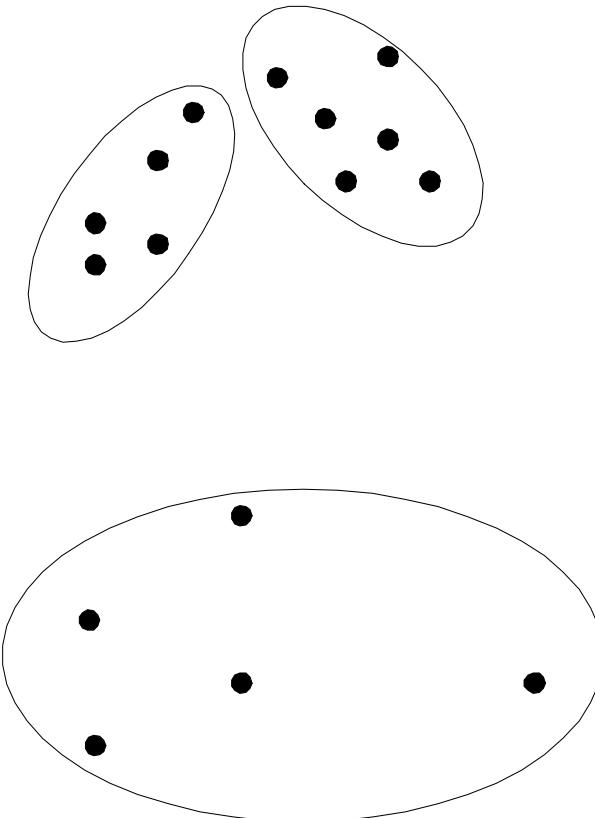
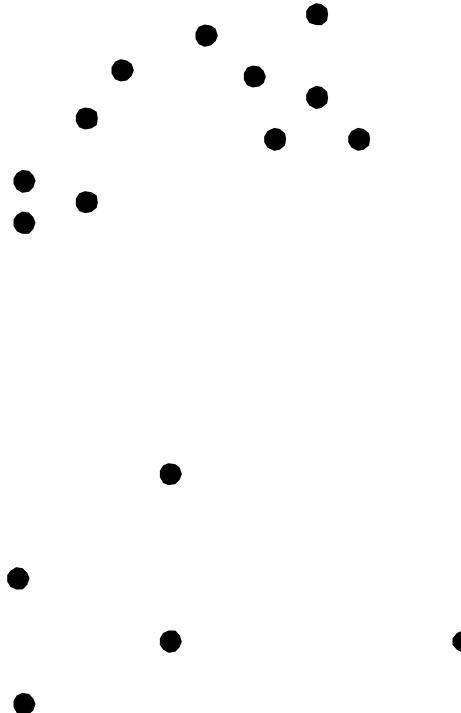


**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Partitional Clustering



A Partitional Clustering

Original Points



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Density-based Methods

- Most partitioning-based methods cluster objects based on distances between them
- Can find only spherical-shaped clusters
- Density-based clustering
- Continue growing a given cluster as long as the density in the ‘neighborhood’ exceeds some threshold.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Hierarchical Clustering

- Hierarchical partition of the objects into clusters.
- It doesn't require the no. of clusters as input.
- Need to specify a termination condition.
- A dendrogram shows how the clusters are merged/split hierarchically.

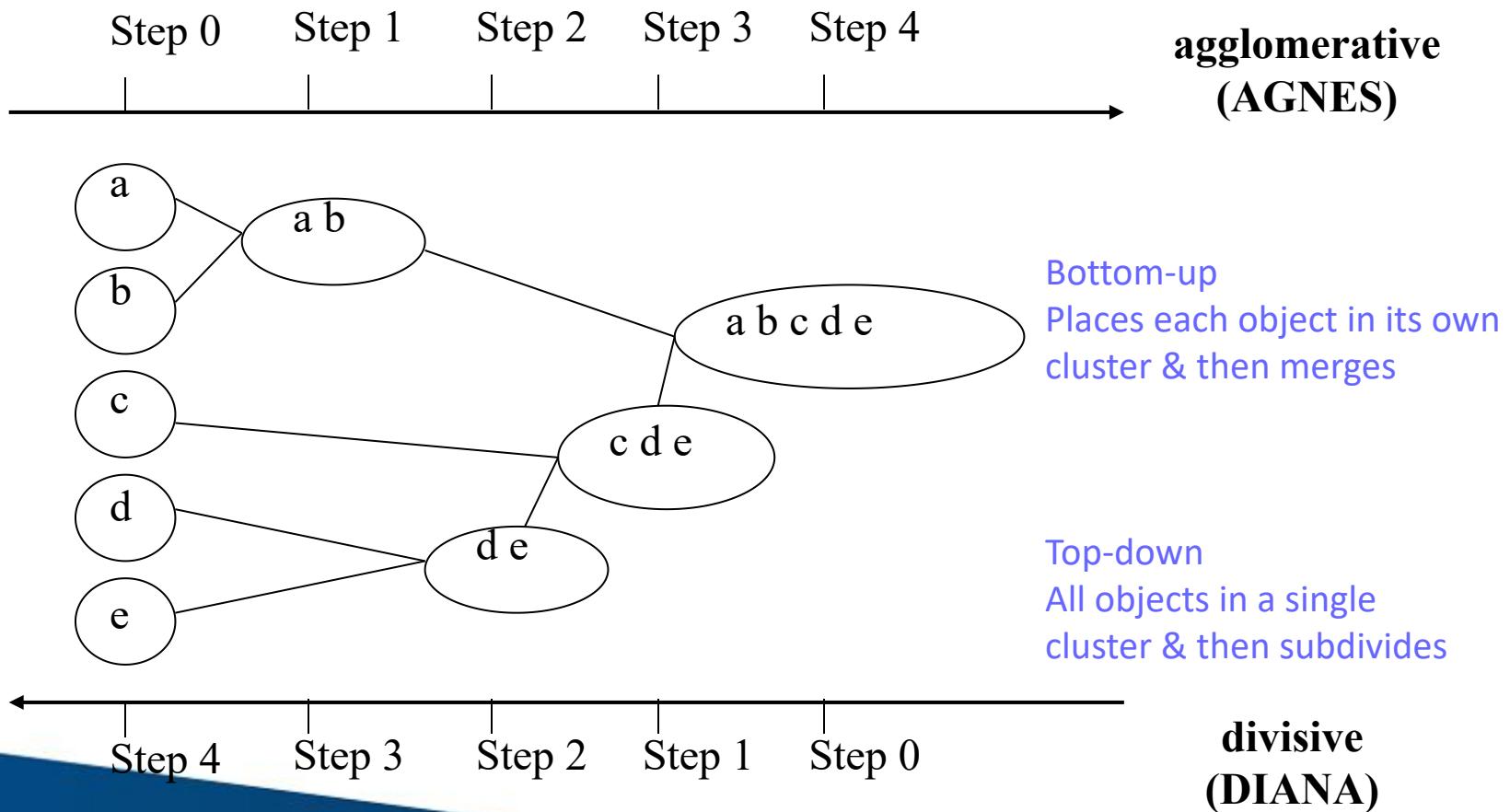
a) Agglomerative : (Bottom Up)

- Start with each object in a cluster.
- Combine similar clusters into larger and larger clusters.

b) Divisive: (Top Down)

- Start with one large cluster and split into smaller and smaller clusters.

Hierarchical Clustering

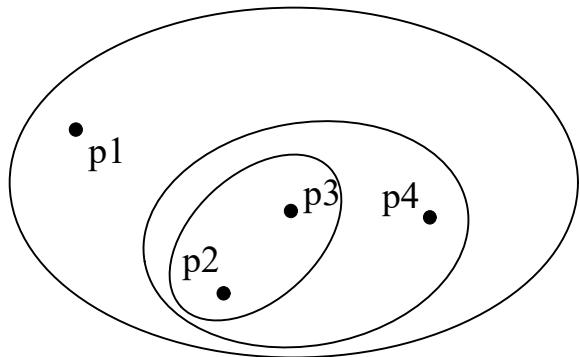


**PRESIDENCY
UNIVERSITY**

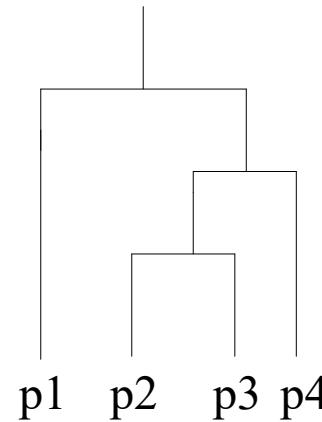
Private University Estd. in Karnataka State by Act No. 41 of 2013



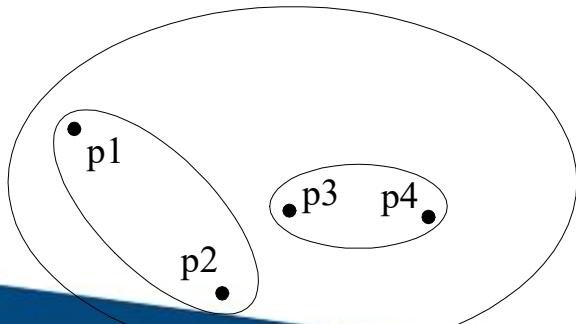
Hierarchical Clustering



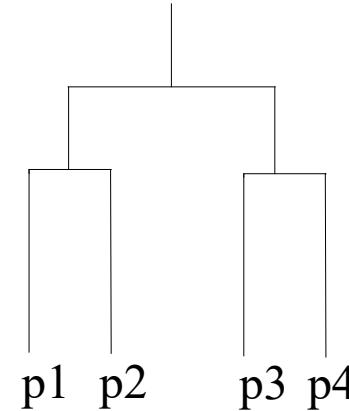
Traditional Hierarchical Clustering



Traditional Dendrogram

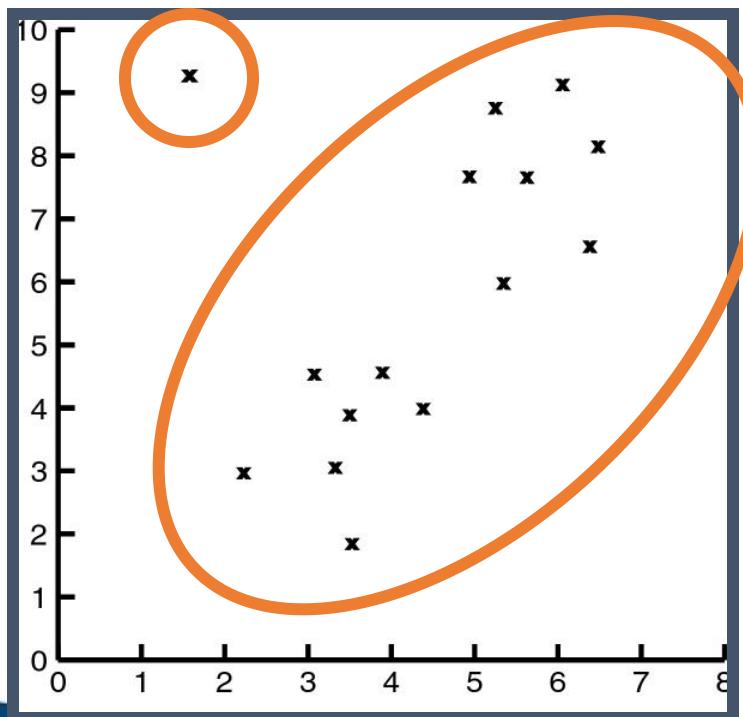


 **PRESIDENCY**
UNIVERSITY
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013



Non-traditional Dendrogram

Impact of Outliers on Clustering



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Problems with Outliers

- Many clustering algorithms take as input the number of clusters
- Some clustering algorithms find and eliminate outliers
- Statistical techniques to detect outliers



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database D of n objects into a set of k clusters
- Given a k , find a partition of k *clusters* that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



K-means Clustering

1. Select the number of clusters, k .
2. Pick k seeds randomly as centroids of the k clusters.
3. Compute the Euclidean distance of each object from each of the centroids.
4. Allocate each object to its nearest cluster.
5. Compute the new centroids of the clusters
 - Mean of the attribute values of the objects in each cluster is the cluster centroid.
6. Check if the stopping condition has been met. If yes, stop else go to step 3.
 - No change in the cluster membership.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



K-means Example 1

- For simplicity, 1-dimension objects and k=2.
 - Numerical difference is used as the distance
- Objects: 1, 2, 5, 6, 7
- K-means:
 - Randomly select 5 and 6 as centroids;
 - End of Iteration 1 : Two clusters {1,2,5} and {6,7}; meanC1=8/3, meanC2=6.5
 - End of Iteration 2 : {1,2}, {5,6,7}; meanC1=1.5, meanC2=6
 - End of Iteration 3 : no change.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



K-means Clustering – Details

- Initial centroids are often chosen randomly.
 - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
 - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is $O(n * K * I * d)$
 - n = number of points, K = number of clusters,
 I = number of iterations, d = number of attributes



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
 - For each point, the error is the distance to the nearest cluster
 - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
 - can show that m_i corresponds to the center (mean) of the cluster
- Given two clustering results, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
 - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Handling empty clusters

- Basic k-means algorithm can yield empty clusters
- Several strategies to fill in the empty cluster
 - Choose the point that contributes most to the total SSE
 - Choose a point from the cluster with the highest SSE
 - If there are several empty clusters, the above can be repeated several times.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a cluster
- An alternative is to update the centroids after each assignment of a point to a cluster (incremental approach)
 - all clusters start with a single point.
 - Each assignment updates zero or two centroids
 - Never get an empty cluster
 - More expensive
 - Introduces an order dependency

Pre-processing and Post-processing

- Pre-processing
 - Normalize the data
 - Eliminate outliers – avoid clustering points that will not cluster well.
 - Outliers are not eliminated in all clustering applications.
 - Ex: data compression – every point needs to be clustered.
 - Financial analysis – an unusually profitable customer is an interesting point.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Post-processing

- Post-processing
 - Eliminate small clusters that may represent outliers
 - Strategy to decrease the total SSE
 - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
 - Strategy to decrease k, the no. of clusters and also tries to minimize the increase in total SSE
 - Merge clusters that are ‘close’ and that have relatively low SSE – used in Ward’s method.
 - disperse a cluster – cluster that increases the total SSE the least.



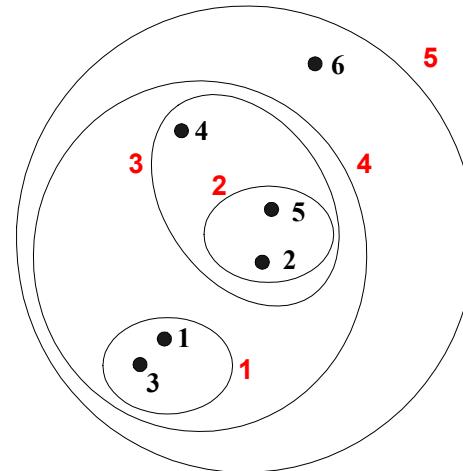
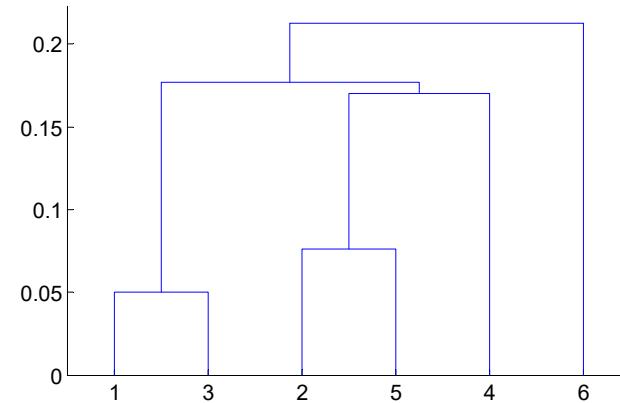
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
 - A tree like diagram that records the sequences of merges or splits



Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
 - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Hierarchical Clustering

- Two main types of hierarchical clustering
 - Agglomerative:
 - Start with the points as individual clusters
 - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
 - Divisive:
 - Start with one, all-inclusive cluster
 - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
 - Merge or split one cluster at a time

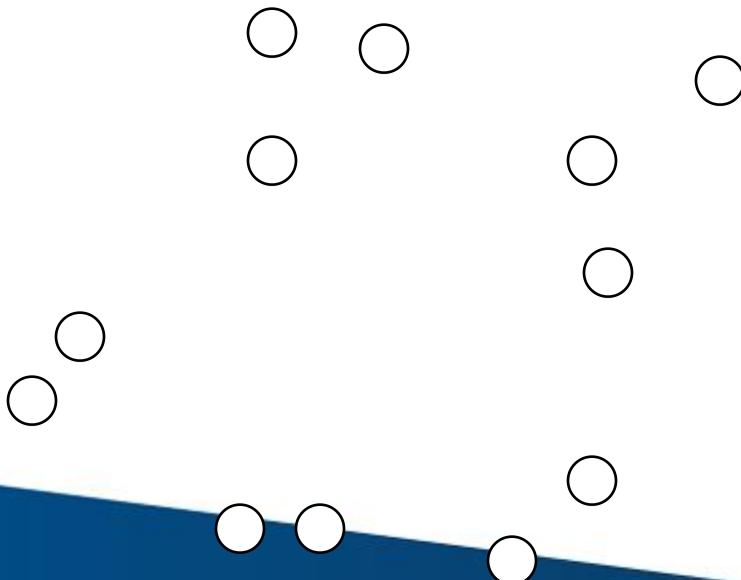


Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
 1. Compute the proximity matrix
 2. Let each data point be a cluster
 3. **Repeat**
 4. Merge the two closest clusters
 5. Update the proximity matrix
 6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
 - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

- Start with clusters of individual points and a proximity matrix



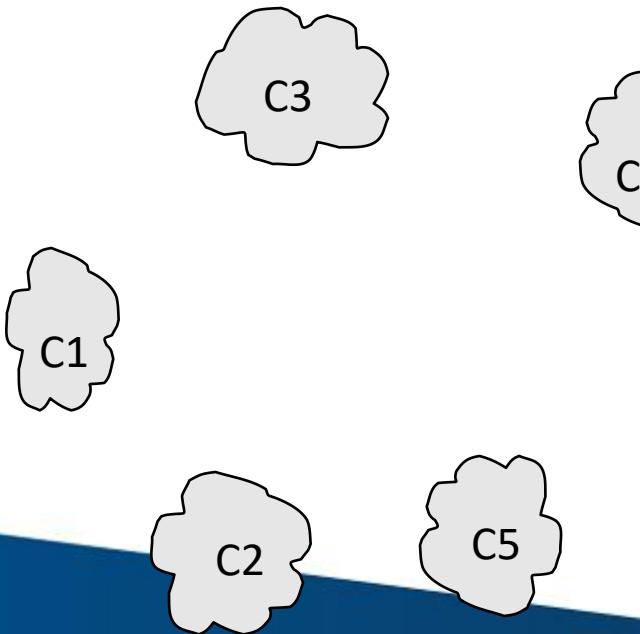
| | | | | | | |
|----|----|----|----|----|----|-----|
| | p1 | p2 | p3 | p4 | p5 | ... |
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| | | | | | | |

Proximity Matrix

p1 p2 p3 p4 ... p9 p10 p11 p12

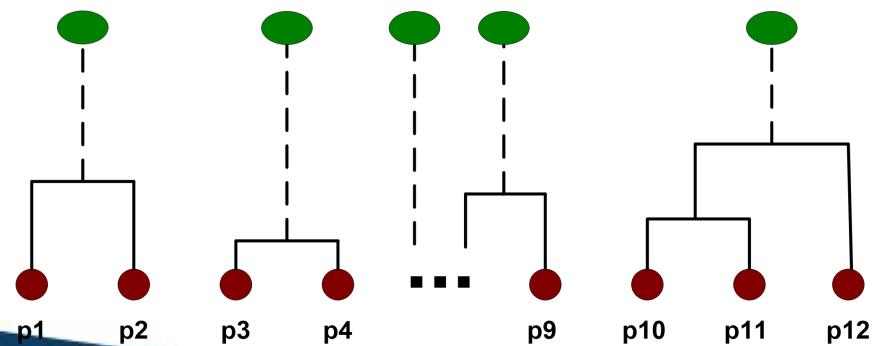
Intermediate Situation

- After some merging steps, we have some clusters



| | C1 | C2 | C3 | C4 | C5 |
|----|----|----|----|----|----|
| C1 | | | | | |
| C2 | | | | | |
| C3 | | | | | |
| C4 | | | | | |
| C5 | | | | | |

Proximity Matrix



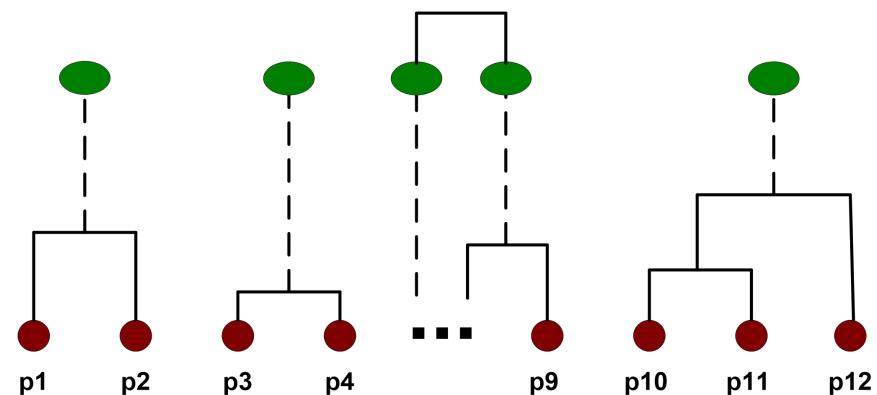
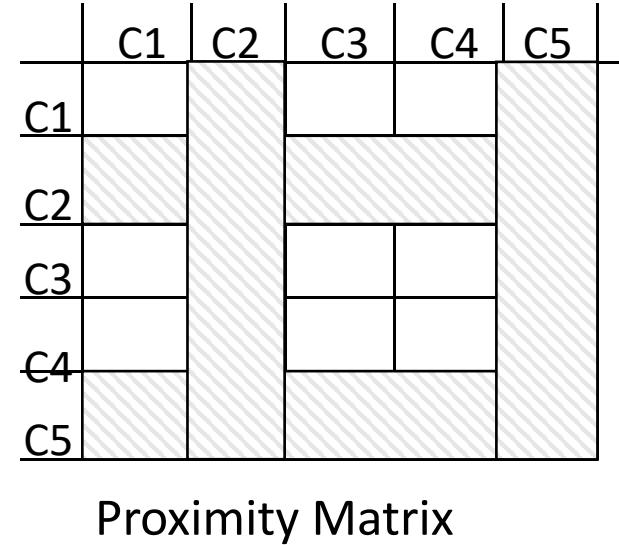
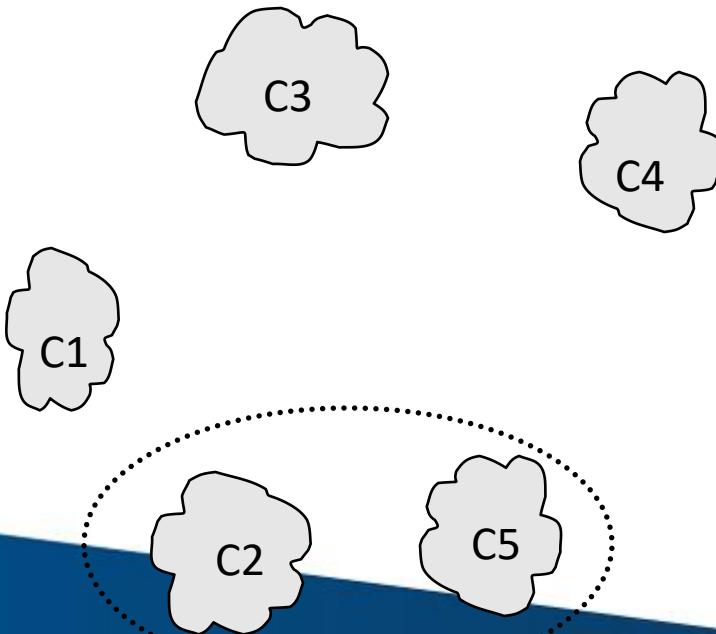
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



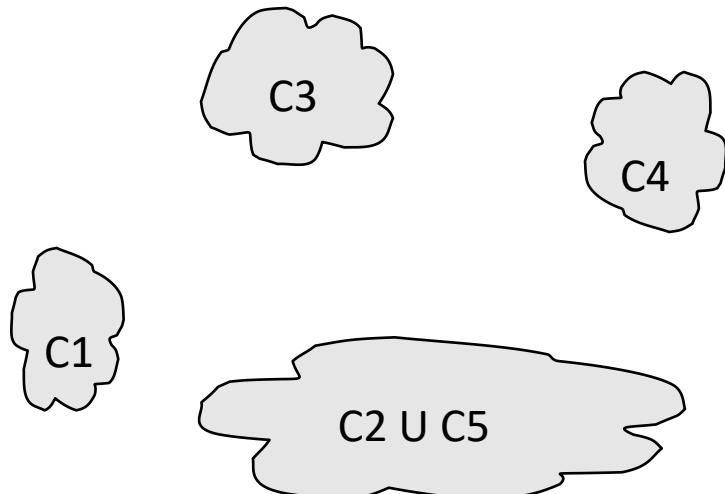
PRESIDENCY
UNIVERSITY



Private University Estd. in Karnataka State by Act No. 41 of 2013

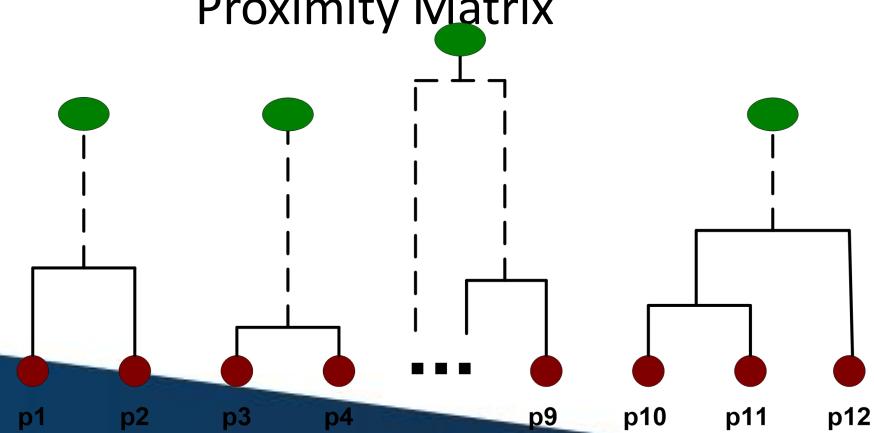
After Merging

- The question is “How do we update the proximity matrix?”



| | | C2 | | | |
|----|---------|----|---------|----|----|
| | | C1 | U C5 | C3 | C4 |
| C1 | | | ? | | |
| C2 | U C5 | ? | ? | ? | ? |
| C3 | | ? | | | |
| C4 | | ? | | | |

Proximity Matrix

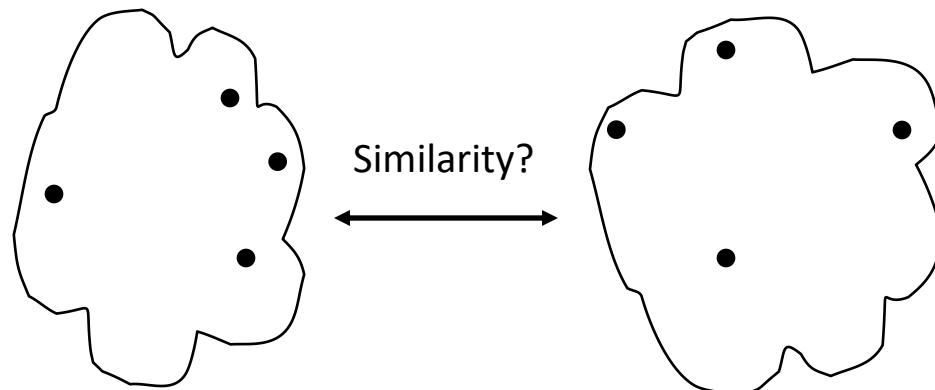


**PRESIDENCY
UNIVERSITY**



Private University Estd. in Karnataka State by Act No. 41 of 2013

How to Define Inter-Cluster Similarity



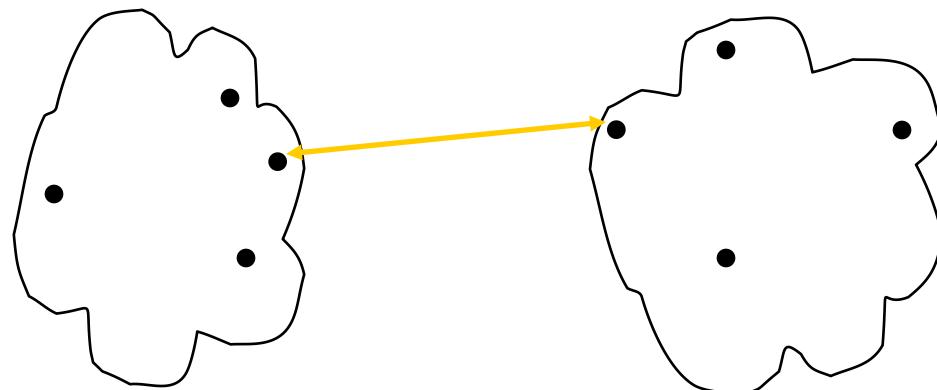
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

| | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

. Proximity Matrix

.

How to Define Inter-Cluster Similarity

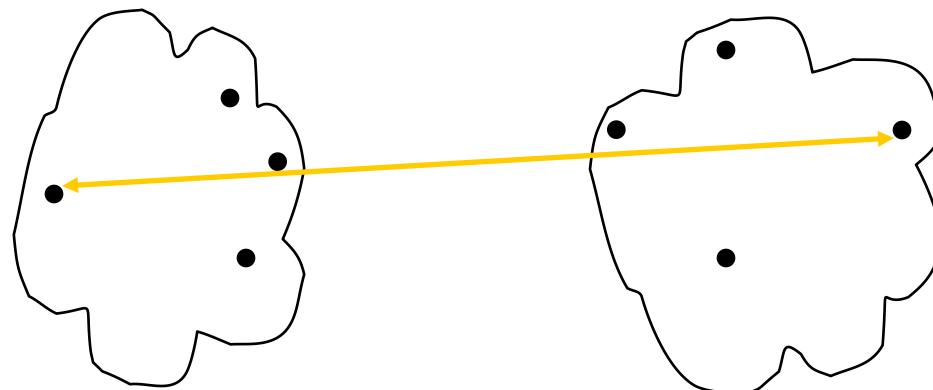


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

| | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

Proximity Matrix

How to Define Inter-Cluster Similarity

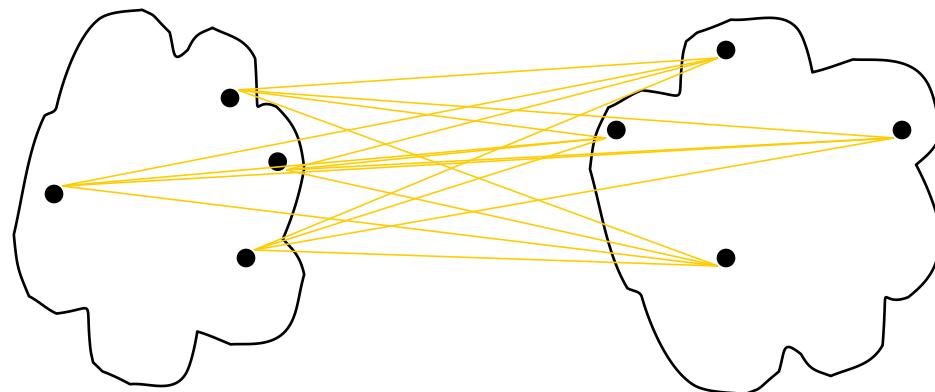


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

| | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

Proximity Matrix

How to Define Inter-Cluster Similarity

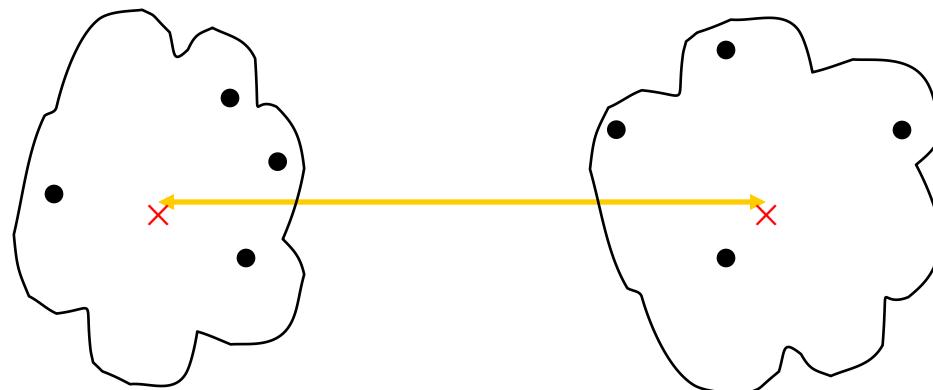


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function

| | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

Proximity Matrix

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- **Distance Between Centroids**
- Other methods driven by an objective function

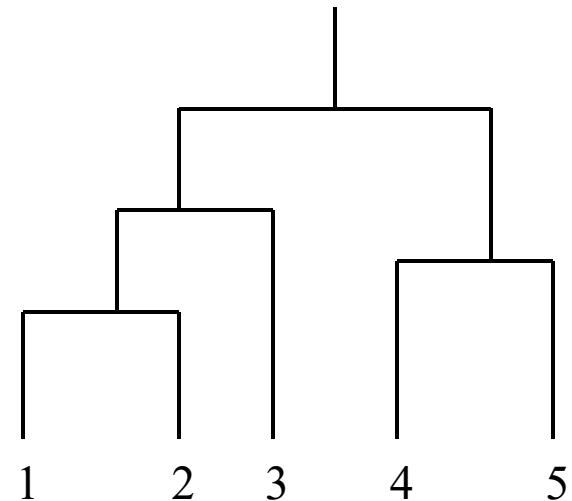
| | p1 | p2 | p3 | p4 | p5 | ... |
|----|----|----|----|----|----|-----|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

Proximity Matrix

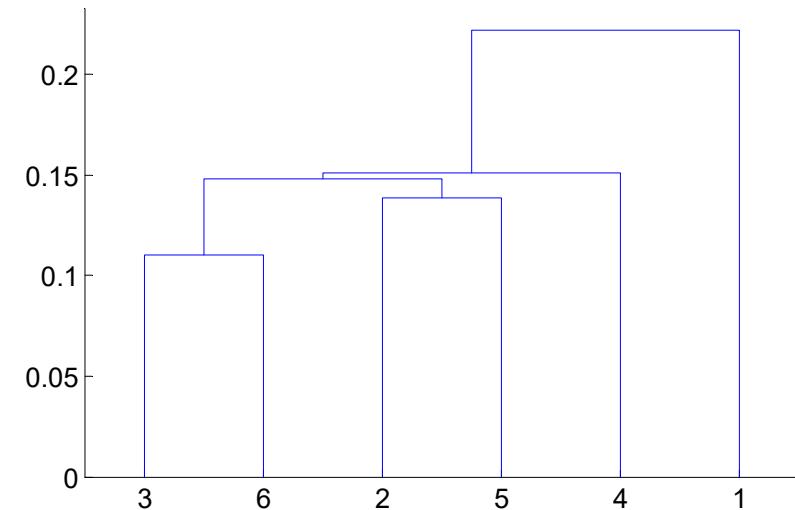
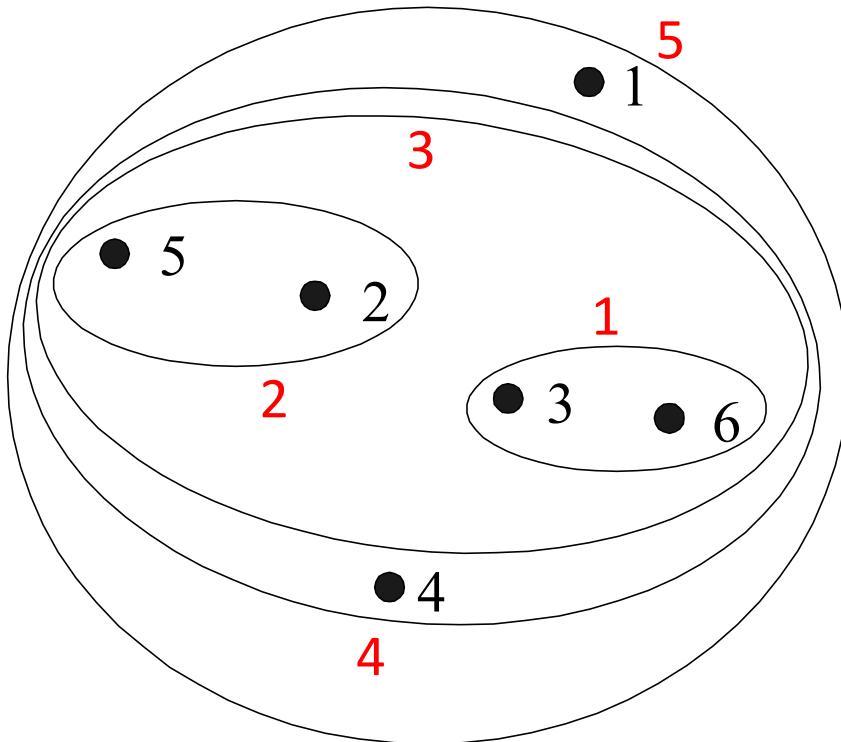
Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
 - Determined by one pair of points, i.e., by one link in the proximity graph.

| | I1 | I2 | I3 | I4 | I5 |
|----|------|------|------|------|------|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |



Hierarchical Clustering: MIN



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

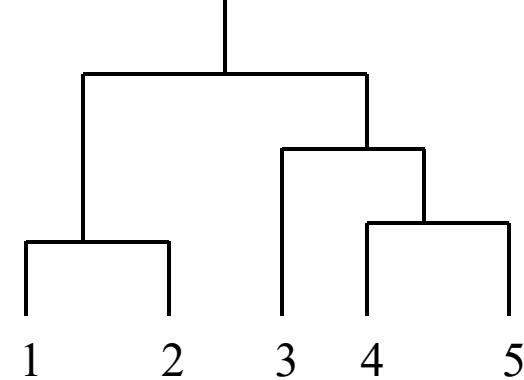


Dendrogram

Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
 - Determined by all pairs of points in the two clusters

| | I1 | I2 | I3 | I4 | I5 |
|----|------|------|------|------|------|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |

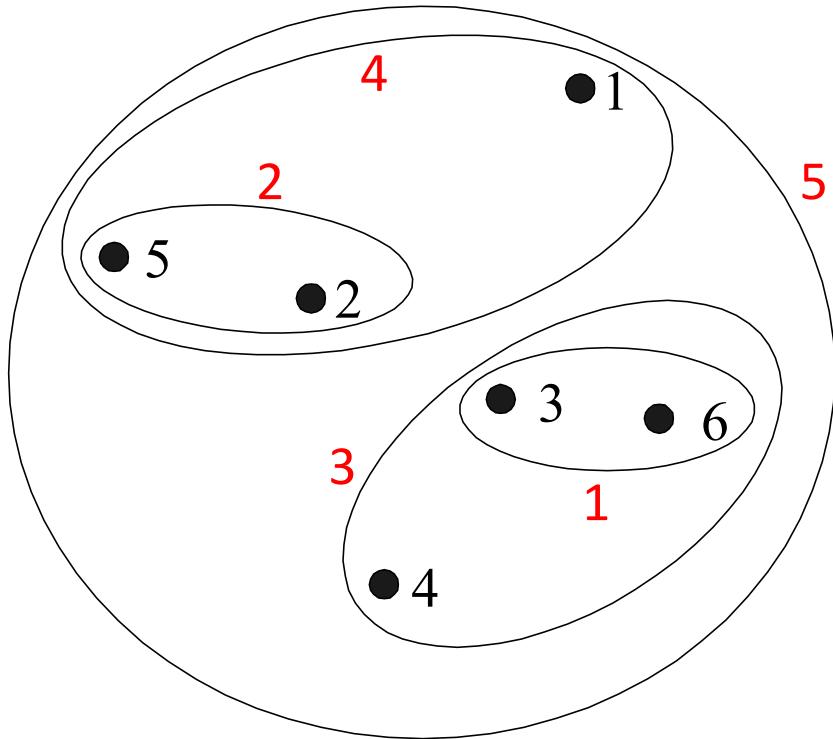


**PRESIDENCY
UNIVERSITY**

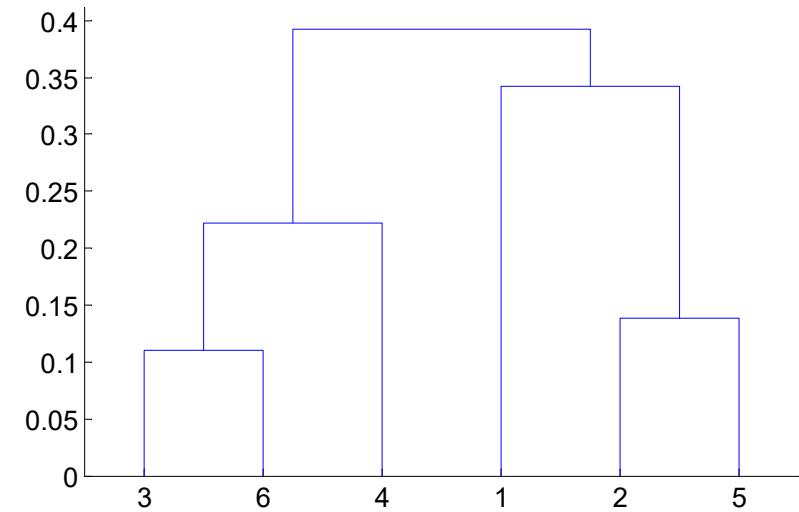
Private University Estd. in Karnataka State by Act No. 41 of 2013



Hierarchical Clustering: MAX



Nested Clusters



Dendrogram



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



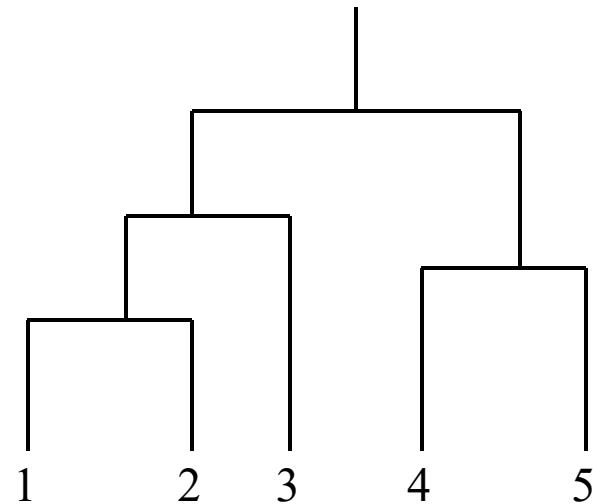
Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

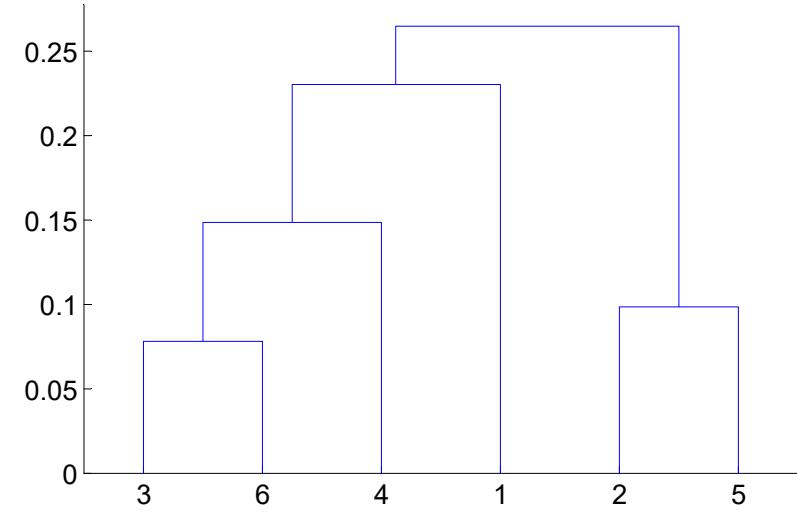
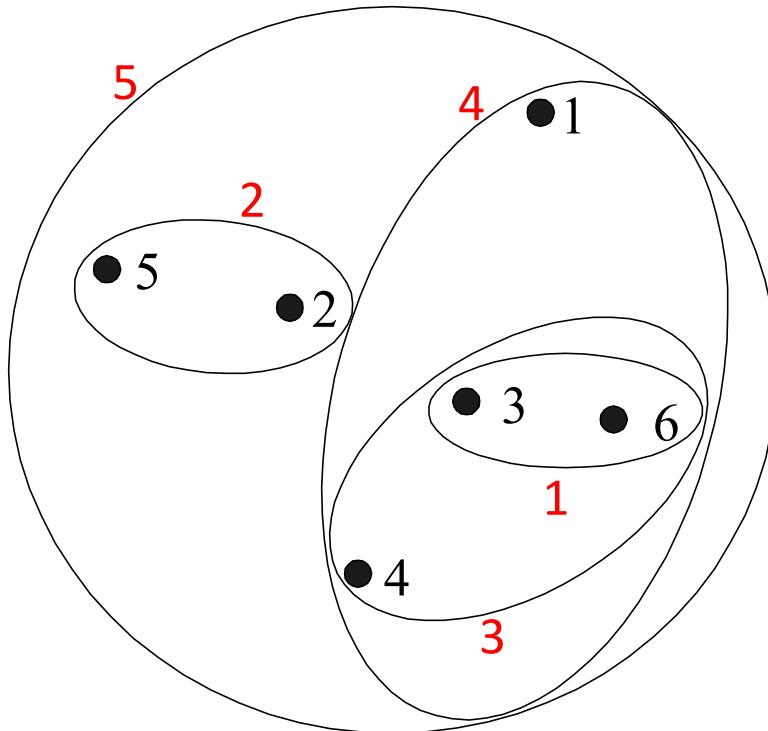
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

| | I1 | I2 | I3 | I4 | I5 |
|----|------|------|------|------|------|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |



Hierarchical Clustering: Group Average



Nested Clusters

Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
 - Less susceptible to noise and outliers
- Limitations
 - Biased towards globular clusters



**PRESIDENCY
UNIVERSITY**

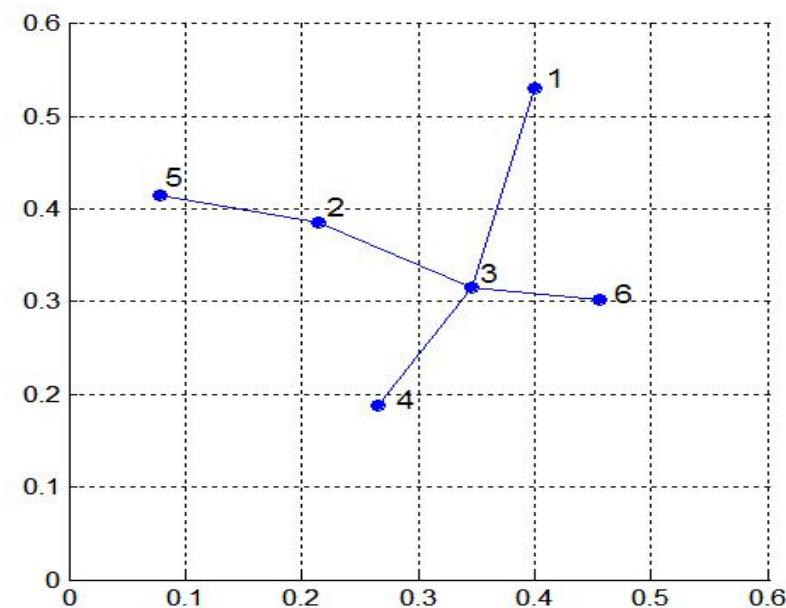
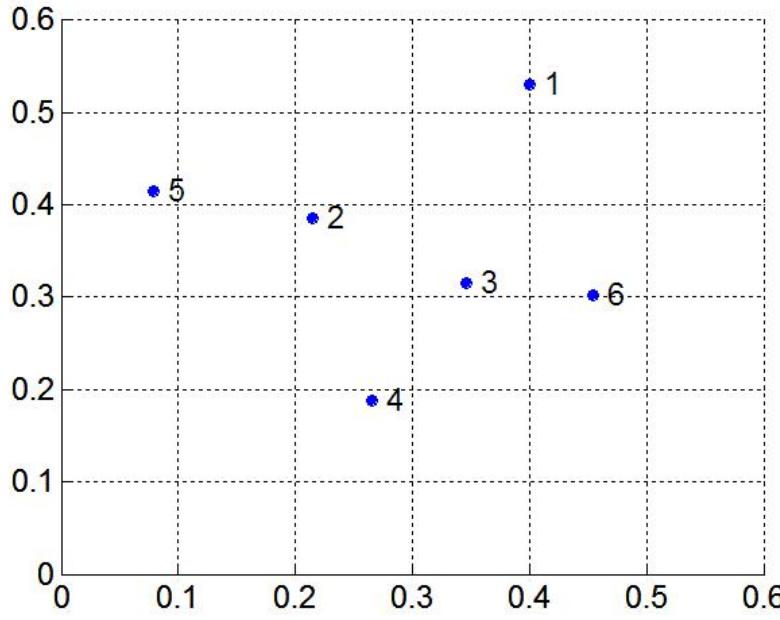
Private University Estd. in Karnataka State by Act No. 41 of 2013



MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)

- Start with a tree that consists of any point
- In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
- Add q to the tree and put an edge between p and q



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

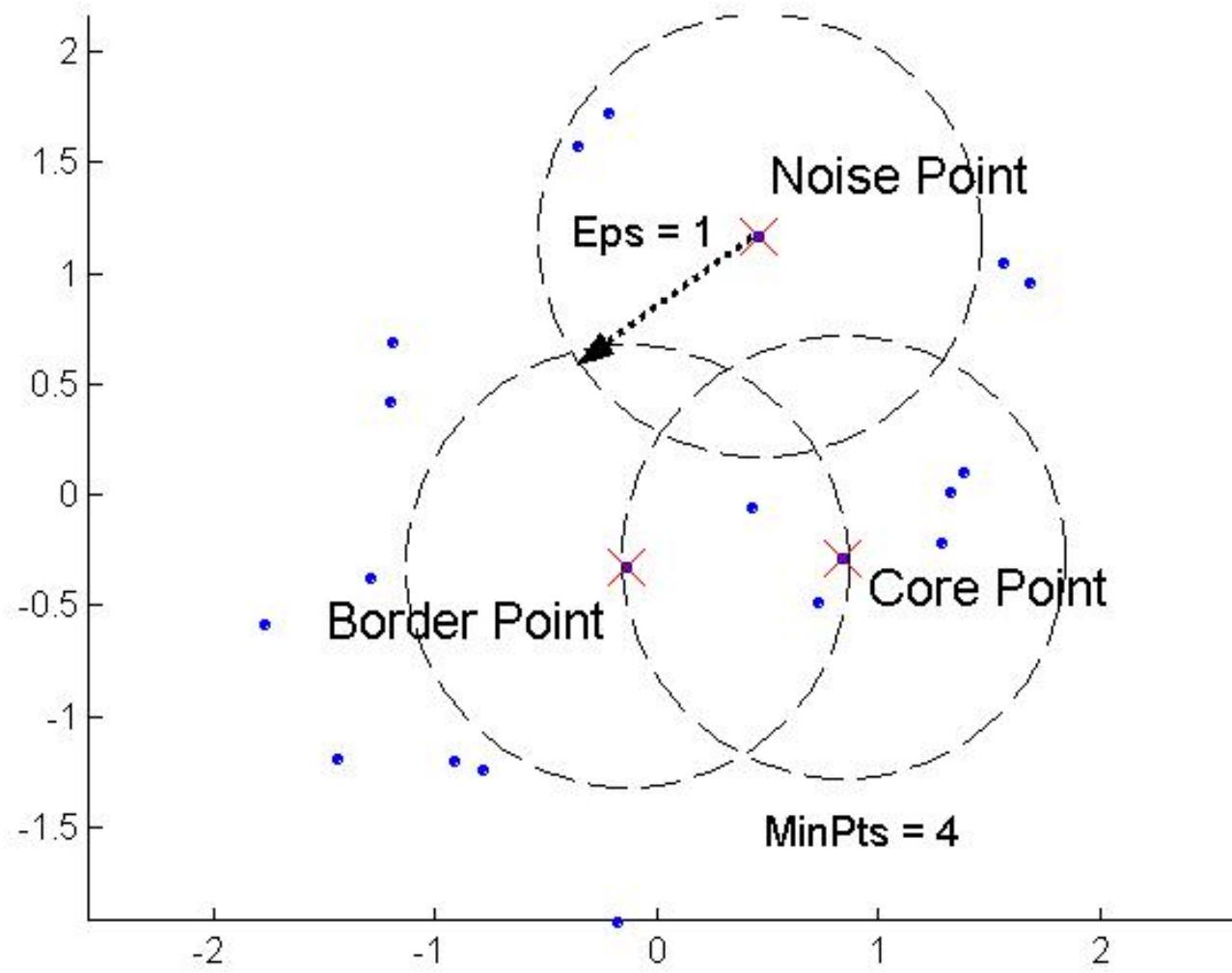
Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

- 1: Compute a minimum spanning tree for the proximity graph.
 - 2: **repeat**
 - 3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
 - 4: **until** Only singleton clusters remain
-

DBSCAN

- DBSCAN is a density-based algorithm.
 - Density = number of points within a specified radius (Eps)
 - A point is a **core point** if it has more than a specified number of points (MinPts) within Eps
 - These are points that are at the interior of a cluster
 - A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
 - A **noise point** is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013



YEARS
OF ACADEMIC
WISDOM

DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

current_cluster_label $\leftarrow 1$

for all core points **do**

if the core point has no cluster label **then**

current_cluster_label $\leftarrow \text{current_cluster_label} + 1$

 Label the current core point with cluster label *current_cluster_label*

end if

for all points in the *Eps*-neighborhood, except *ith* the point itself **do**

if the point does not have a cluster label **then**

 Label the point with cluster label *current_cluster_label*

end if

end for

end for



UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

TERMS
OF ACADEMIC
WISDOM

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
 - Accuracy, precision, recall, confusion matrix, cross validation
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
 - To avoid finding patterns in noise
 - To compare clustering algorithms
 - To compare two sets of clusters
 - To compare two clusters



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
 - **External Index (supervised):** Used to measure the extent to which cluster labels match externally supplied class labels.
 - Entropy
 - **Internal Index (unsupervised):** Used to measure the goodness of a clustering structure *without* respect to external information.
 - Sum of Squared Error (SSE)
- **Relative Index:** Used to compare two different clusterings or clusters.
 - Often an external or internal index is used for this function, e.g., SSE or entropy



Internal measures – SSE - cohesion

- SSE = average pairwise distance between the points in a cluster.
- Cluster SSE = $\sum_{x \in C_i} dist(C_i, x)^2$

$$= \frac{1}{2m_i} \sum_{x \in C_i} \sum_{y \in C_i} dist(x, y)^2$$

Where m_i is the centroid of cluster i

x and y are points in cluster i.

- measures how closely related are points in a cluster.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Internal measures – Separation

- How distinct or well separated is a cluster from other clusters.
- **Ex: cohesion :-** within cluster sum of squares

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

Separation : between clusters sum of squares

$$BSS = \sum |C_i| (m - m_i)^2$$

where $|C_i|$ is the size of cluster i

m is the overall centroid.



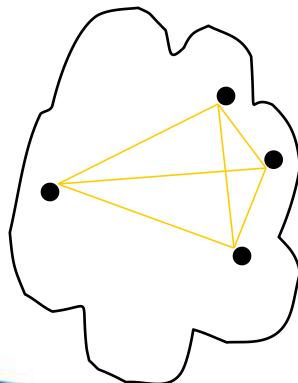
**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

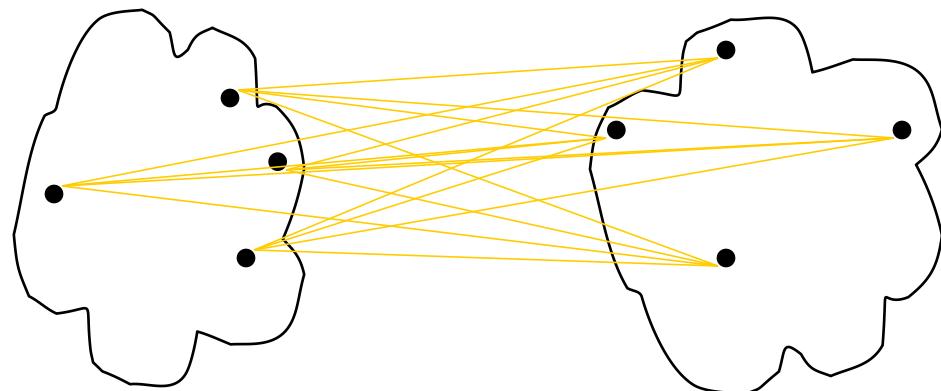


Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
 - Cluster cohesion is the sum of the weight of all links within a cluster.
 - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



separation



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



External measures: Entropy and purity

- **Entropy of a cluster C_j :-**
- Find the probability of each class i in C_j , where

$$p_{ij} = \frac{m_{ij}}{m_j}$$

where m_j = no. of values in cluster C_j .

m_{ij} = no. of values of class i in cluster C_j .

Entropy of C_j :-

$$e_i = \sum_{i=1}^L p_{ii} \log_2 p_{ii}$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Entropy of a clustering with k no. of clusters

- $e = \sum_{i=1}^k \frac{m_j}{m} e_j$

Where e_j = entropy of j^{th} cluster

m_j = size of the j^{th} cluster

m = total no. of data points.



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Purity :-

1) Purity of a cluster C_j :-

$$purity_j = \max P_{ij}$$

2) Purity of a clustering:-

$$\text{purity} = \sum_{i=1}^k \frac{m_i}{m} purity_i$$



**PRESIDENCY
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Table 5.9. K-means Clustering Results for LA Document Data Set

| Custer | Entertainment | Financial | Foreign | Metro | National | Sports | Entropy | Purity |
|--------|---------------|-----------|---------|-------|----------|--------|---------|--------|
| 1 | 3 | 5 | 40 | 506 | 96 | 27 | 1.2270 | 0.7474 |
| 2 | 4 | 7 | 280 | 29 | 39 | 2 | 1.1472 | 0.7756 |
| 3 | 1 | 1 | 1 | 7 | 4 | 671 | 0.1813 | 0.9796 |
| 4 | 10 | 162 | 3 | 119 | 73 | 2 | 1.7487 | 0.4390 |
| 5 | 331 | 22 | 5 | 70 | 13 | 23 | 1.3976 | 0.7134 |
| 6 | 5 | 358 | 12 | 212 | 48 | 13 | 1.5523 | 0.5525 |
| Total | 354 | 555 | 341 | 943 | 273 | 738 | 1.1450 | 0.7203 |

ropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij} , the ‘probability’ that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j . Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{j=1}^K \frac{m_j}{m} e_j$, where m_j is the size of cluster j , K is the number of clusters, and m is the total number of data points.

urity Using the terminology derived for entropy, the purity of cluster j , is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{j=1}^K \frac{m_j}{m} purity_j$.

Ex using Gini Index

$$G_1(S) = 1 - \sum_{i=1}^n p_i^2$$

There are 10 samples and three classes.

$$\therefore G_1(S) =$$

$$1 - \left(\frac{3}{10}\right)^2 - \left(\frac{3}{10}\right)^2 - \left(\frac{4}{10}\right)^2$$

Frequencies of these

classes are:-

$$A = 3 \quad B = 3 \quad C = 4$$

$$G_1(S) = 0.66 ; \quad \text{Round} : - \text{Find the root note}$$

) Attribute "Owns Home"

For Value = yes } \rightarrow class distribution is
 $S_1 = 5 \text{ records}$ } $A = 1, B = 2 \text{ and } C = 2$

For Value = NO } \rightarrow class distib is
 $S_2 = 5 \text{ records}$ } $A = 2, B = 1, C = 2$

$$\therefore G_1(\text{split on Owns-Home}) = \frac{n_1}{S} G_1(S_1) + \frac{n_2}{S} G_1(S_2)$$

$$\therefore G_1(\text{Owns-Home} = \text{yes}) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2 \\ = 0.64$$

$$\therefore G_1(S_2 \text{ OwnsHome} = \text{No}) = 0.64$$

$$\therefore G_1(\text{split on Owns-Home}) = \frac{5}{10} \times 0.64 + \frac{5}{10} \times 0.64 \\ = 0.64$$

2. Attrib "Married": -

S_1 = Married = yes has $A=0, B=1, C=4; n_1=5$

S_2 = Married = no; has $A=3, B=2, C=0; n_2=5$

$$G_1(y) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2 = 0.32$$

$$G_1(n) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$$

$$\therefore G_1(\text{Married}) = \frac{5}{10} \times 0.32 + \frac{5}{10} \times 0.48 \\ = \underline{\underline{0.40}}$$

3. Attrib "Gender": -

S_1 : Gender = Male has $A=0, B=3, C=0; n_1=3$

S_2 : Gender = Female has $A=3, B=0, C=4; n_2=7$

$$G_1(\text{Male}) = 1 - \left(\frac{3}{3}\right)^2 - (0)^2 = 0$$

$$G_1(\text{Female}) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.511$$

$$\therefore G_1(\text{Gender}) = \frac{3}{10} \times 0 + \frac{7}{10} \times 0.511 = \underline{\underline{0.358}}$$

4. Attrib "Employed": -

S_1 : Employed = yes; has $A=3, B=1, C=4; n_1=8$

S_2 : Employed = no; has $A=0, B=2, C=0; n_2=10$

$$G_1(\text{yes}) = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{1}{8}\right)^2 - \left(\frac{4}{8}\right)^2 = 0.594$$

$$G_1(\text{no}) = 1 - \left(\frac{2}{10}\right)^2 = 0$$

$$\therefore G_1(\text{Employed}) = \frac{8}{10} \times 0.594 + \frac{2}{10} \times 0 = \underline{\underline{0.475}}$$

5) Attr "Credit Rating"

S_1 : Credit Rating = A ; has A=2, B=1, C=2; $n_1 = 5$

S_2 : Credit Rating = B ; has A=1, B=2, C=2; $n_2 = 5$

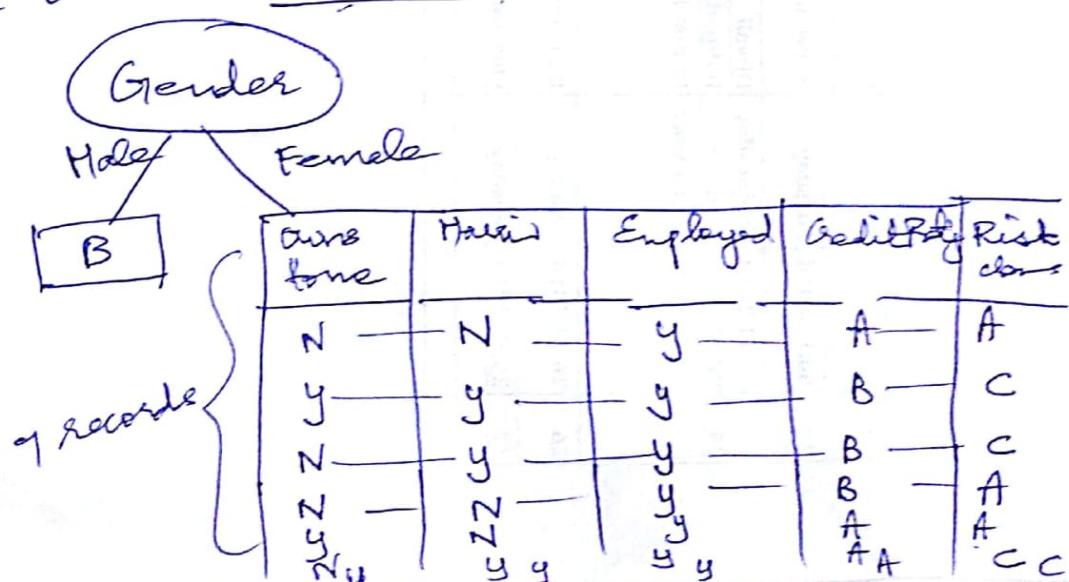
$$G(A) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.64$$

$$G(B) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.64 = G(A)$$

$$\therefore G(\text{Credit Rating}) = \frac{5}{10} \times 0.64 + \frac{5}{10} \times 0.64 \\ = \underline{\underline{0.64}}$$

| Attr | Gini index before Split | Gini index after split | Gain |
|---------------|----------------------------|---------------------------|--------------|
| Owes Home | 0.66 | 0.64 | 0.02 |
| Married | 0.66 | 0.40 | 0.26 |
| <u>Gender</u> | 0.66 | 0.358 | <u>0.302</u> |
| Employed | 0.66 | 0.495 | 0.185 |
| Credit Rating | 0.66 | 0.64 | 0.02 |

\therefore Gender is the root node.



Round 2:

| Owns Home | Hired | Employed | Credit Rating | Risk class |
|-----------|-------|----------|---------------|------------|
| N | N | Y | A | A |
| Y | Y | Y | B | C |
| N | Y | Y | B | C |
| N | N | Y | B | A |
| Y | N | Y | A | A |
| N | Y | Y | A | C |
| Y | Y | Y | A | C |

$$S=7 ; A=3 \quad B=0 \quad C=4$$

$$\therefore G(S) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.511$$

i) Attrib "Owns Home":

$$S_1: \text{Owes-Home} = Y \text{ has } A=1 ; C=2 ; n_1=3$$

$$S_2: \text{Owes-Home} = N \text{ has } A=2 \quad C=2 \quad n_2=4$$

$$G(S_1) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 1 - 0.11 - 0.44 \\ = 0.45$$

$$G(S_2) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 1 - 0.25 - 0.25 \\ = 0.5$$

$$\therefore G(\text{Owes-Home}) = \frac{3}{7} \times 0.45 + \frac{4}{7} \times 0.5 \\ = 0.193 + 0.7286 = 0.479$$

2) Attrib "Married":

s_1 : Married = yes ; has A = 0 ; C = 4 ; $n_1 = 4$

s_2 : Married = no ; has A = 3 ; C = 0 ; $n_2 = 3$

$$G(s_1) = G(s_2) = 0$$

$$\therefore G(\text{Married}) = \frac{4}{7} \times 0 + \frac{3}{7} \times 0 = 0$$

3) Attrib "Employed":

s_1 : Employed = y has A = 3 C = 4 ; $n_1 = 7$

s_2 : Employed = n ; $n_2 = 0$

$$\therefore G(s_1) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.511$$

$$\therefore G(\text{Employed}) = \frac{7}{7} \times 0.511 = \underline{\underline{0.511}}$$

A) Attrib "Credit Rating":

s_1 : Credit Rating = A has A = 2 C = 2 $n_1 = 4$

s_2 : Credit Rating = B has A = 1 C = 2 $n_2 = 3$

$$G(s_1) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

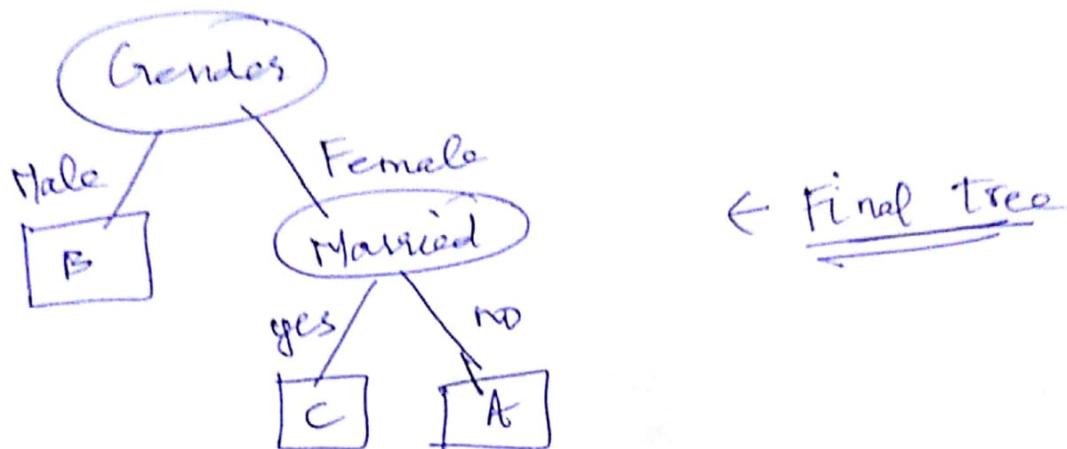
$$G(s_2) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.45$$

$$\begin{aligned} \therefore G(\text{Credit Rating}) &= \frac{4}{7} \times 0.5 + \frac{3}{7} \times 0.45 \\ &= 0.286 + 0.193 = \underline{\underline{0.479}} \end{aligned}$$

| Attrib | Gini Index before split | Gini Index after split | Gain |
|---------------|-------------------------|------------------------|-------|
| Owns Home | 0.511 | 0.479 | 0.032 |
| Married | 0.511 | 0 | 0.511 |
| Employed | 0.511 | 0.511 | 0 |
| Credit Rating | 0.511 | 0.479 | 0.032 |

∴ Married is the best split

∴ Tree is



Ex: for cluster validity.

Assume we have a document collection D of 900 documents from three topics (or 3 classes) Science, Sports and Politics. Each class has 300 documents. Each document in D is labelled with one of the topic (class). These documents are grouped into 3 clusters. Measure the effectiveness of the clustering.

| Cluster | Science | Sports | Politics | Total |
|---------|---------|--------|----------|-------|
| 1 | 250 | 20 | 10 | 280 |
| 2 | 20 | 180 | 80 | 280 |
| 3 | 30 | 100 | 210 | 340 |
| Total | 300 | 300 | 300 | |

Solution:

Step 1: calculate the total in each cluster.

$$\text{cluster 1} \rightarrow 250 + 20 + 10 = 280$$

Step 2: Find out the probability of each cluster.

| Cluster | Science | Sports | Politics | Purity |
|---------|---------|--------|----------|--------|
| 1 | 0.893 | 0.0714 | 0.035 | 0.893 |
| 2 | 0.0714 | 0.643 | 0.286 | 0.643 |
| 3 | 0.0882 | 0.2941 | 0.6176 | 0.617 |
| Total | 300 | 300 | 300 | 0.711 |

$$C_1: \text{Prob(Science)} = 250/280 = 0.893, \text{Prob(Politics)} = 0.0357$$

①

Step 3: Calculation of Purity, by considering the Maximum probability. = Max(Prob).

$$C_1 : \text{Max}(0.893, 0.0714, 0.035) = 0.893$$

Similarly for all other clusters.

Step 4: Purity of the clustering: $\sum_{i=1}^3 \frac{m_i}{m}$ Purity.

$$= \left[\frac{280}{900} \times 0.893 + \frac{280}{900} \times 0.643 + \frac{340}{900} \times 0.617 \right]$$

$$= 0.277 + 0.2000 + 0.2330$$

$$\Rightarrow 0.711$$

Step 5: Calculate the Entropy of each cluster.

$$C_1 : - \left(\frac{250}{280} \log_2 \frac{250}{280} + \frac{30}{280} \log_2 \frac{30}{280} + \frac{10}{280} \log_2 \frac{10}{280} \right)$$

$$= -(0.893 \log_2 0.893 + 0.0714 \log_2 0.0714 + 0.035 \log_2 0.035)$$

$$= -(0.14579 + 0.271886 - 0.1692745)$$

$$\Rightarrow 0.587$$

Similarly for all the clusters calculate the entropy.

| Prob | Science | Sports | Politics | Entropy | Purity |
|------|---------|--------|----------|---------|--------|
| 1 | 0.893 | 0.0714 | 0.035 | 0.587 | 0.893 |
| 2 | 0.0714 | 0.643 | 0.286 | 1.198 | 0.643 |
| 3 | 0.088 | 0.294 | 0.617 | 1.257 | 0.617 |
| | | | | 1.0201 | 0.711 |

$$\left(\frac{20}{280} \log_2 \frac{20}{28} + \frac{180}{280} \log_2 \frac{180}{280} + \frac{80}{280} \log_2 \frac{80}{280} \right)$$

$$(0.071 \log_2 0.071 + 0.642 \log_2 0.642 + 0.285 \log_2 0.285) \\ - (-0.27093 - 0.41046 - 0.51612) = 1.198 //$$

$$\left(\frac{30}{340} \log_2 \frac{30}{340} + \frac{100}{340} \log_2 \frac{100}{340} + \frac{210}{340} \log_2 \frac{210}{340} \right)$$

$$-(0.0882 \log_2 0.0882 + 0.294 \log_2 0.294 + 0.617 \log_2 0.617) \\ = -(-0.30897 - 0.5192 - 0.4298) \\ \Rightarrow \cancel{1.257}; \quad 1.257 //$$

Step 5: Calculate the entropy of clustering.

$$= \sum_{j=1}^K \frac{m_j}{m} e_j$$

$$= \frac{280}{900} \times 0.587 + \frac{280}{900} \times 1.198 + \frac{340}{900} \times 1.257$$

$$= 0.1826 + 0.3727 + 0.4748$$

$$= 1.0301 //$$

Clustering

k-means Ex :- 1

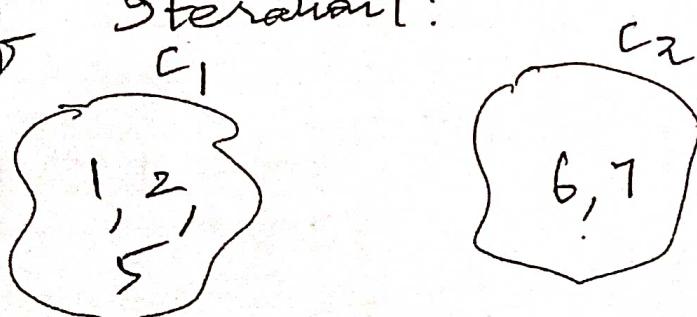
For simplicity, consider 1-7 objects and k=2
objects: 1, 2, 5, 6, 7

Randomly choose 5 and 6 as centroids

| <u>Initial Objects</u> | <u>Dist from c1</u> | <u>Dist from c2</u> | <u>Allocated to cluster</u> |
|------------------------|---------------------|---------------------|----------------------------------|
| 1 | 4 | 5 | c ₁ |
| 2 | 3 | 3 | c ₁ or c ₂ |
| 5 | 0 | 1 | c ₁ |
| 6 | 1 | 0 | c ₂ |
| 7 | 2 | 1 | c ₂ |

Assume absolute value of difference as the distance measure.

End of 1st iteration:



$$\text{Mean of } c_1 = 8/3 = \underline{\underline{2.67}} \quad \text{Mean of } c_2 = \underline{\underline{6.5}}$$

Ques 2: Centroid of cluster 1 = 2.67 Centroid of cluster 2 = 6.5

| Object | Dist from c_1 | Dist from c_2 | Allocated to cluster |
|--------|-----------------|-----------------|----------------------|
| 1 | <u>1.67</u> | 5.5 | c_1 |
| 2 | <u>0.67</u> | 4.5 | c_1 |
| 5 | 2.53 | <u>1.5</u> | c_2 |
| 6 | 4.67 | <u>0.5</u> | c_2 |
| 7 | 3.53 | <u>0.5</u> | c_2 |

c_1 c_2

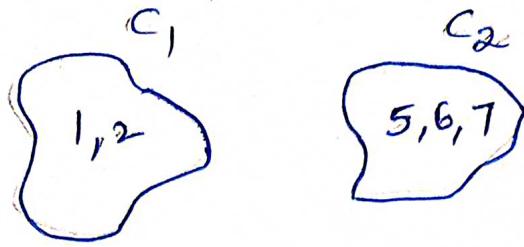
$$\text{Mean of } c_1 = 1.5 \quad \text{Mean of } c_2 = 6$$

Ques 3: Centroid of $c_1 = 1.5$ Centroid of $c_2 \} = 6$

| Obj | Dist from c_1 | Dist from c_2 | Allocated to cluster |
|-----|-----------------|-----------------|----------------------|
| 1 | <u>0.5</u> | 5 | c_1 |
| 2 | <u>0.5</u> | 4 | c_1 |
| 5 | 3.5 | <u>1</u> | c_2 |
| 6 | 4.5 | <u>0</u> | c_2 |
| 7 | 5.5 | <u>1</u> | c_2 |

Pg : ②

End of Qtn 3



\therefore No change in cluster membership,
solution converges.

Example - 2:

Consider the data about students given in the table below. Group them into three clusters, assuming s_1 , s_2 and s_3 as initial seeds/centroids.

| Student | Age | Mark 1 | Mark 2 | Mark 3 |
|----------|-----|--------|--------|--------|
| s_1 | 18 | 73 | 75 | 57 |
| s_2 | 18 | 71 | 85 | 75 |
| s_3 | 23 | 70 | 70 | 52 |
| s_4 | 20 | 55 | 55 | 55 |
| s_5 | 22 | 85 | 86 | 87 |
| s_6 | 19 | 91 | 90 | 89 |
| s_7 | 20 | 70 | 65 | 60 |
| s_8 | 21 | 53 | 56 | 59 |
| s_9 | 19 | 82 | 82 | 60 |
| s_{10} | 47 | 75 | 76 | 77 |

Pg : (3)

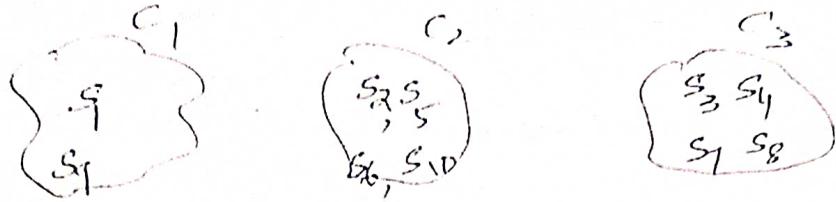
The three initial Seeds

| Student | Age | Mark1 | Mark2 | Mark3 |
|---------|-----|-------|-------|-------|
| s_1 | 18 | 73 | 75 | 57 |
| s_2 | 18 | 79 | 85 | 75 |
| s_3 | 23 | 70 | 70 | 52 |

Iteration 1:

| | | | | | List from | | | Allocate to cluster |
|----------|----|----|----|----|-----------|-------|-------|------------------------|
| c_1 | 18 | 73 | 75 | 57 | c_1 | c_2 | c_3 | |
| s_1 | 18 | 73 | 75 | 57 | 0 | 34 | 18 | c_1 |
| s_2 | 18 | 79 | 85 | 75 | 34 | 0 | 52 | c_2 |
| s_3 | 23 | 70 | 70 | 52 | 18 | 52 | 0 | c_3 |
| s_4 | 20 | 55 | 55 | 55 | 42 | 76 | 36 | c_3 |
| s_5 | 22 | 85 | 86 | 87 | 57 | 23 | 67 | c_2 |
| s_6 | 19 | 91 | 90 | 89 | 66 | 32 | 82 | c_2 |
| s_7 | 20 | 70 | 65 | 60 | 18 | 46 | 16 | c_3 |
| s_8 | 21 | 53 | 56 | 59 | 44 | 74 | 40 | c_3 |
| s_9 | 19 | 82 | 82 | 60 | 20 | 22 | 36 | c_1 |
| s_{10} | 47 | 75 | 76 | 71 | 32 | 44 | 60 | c_2 |

F5 : (4)



∴ New Centroids of:

$$C_1 = \text{Average of } S_1 \text{ and } S_9$$

$$C_2 = \text{Average of } S_2, S_5, S_6, S_{10}, S_{16}, S_{19}$$

$$C_3 = \text{Average of } S_3, S_4, S_7, S_8$$

| \times Cluster Centroid | Age | Mark1 | Mark2 | Mark3 |
|------------------------------|------|-------|-------|-------|
| C_1 | 18.5 | 77.5 | 78.5 | 58.5 |
| C_2 | 26.5 | 82.5 | 84.3 | 82.0 |
| C_3 | 21 | 61.5 | 61.5 | 56.5 |

Iterations:- - Find the distance of each object from the new cluster centroids:-

| | Dist from | | | | | |
|-------|-----------|-------|-------|------|---------|------------|
| | C_1 | C_2 | C_3 | | Alloted | |
| C_1 | 18.5 | 77.5 | 78.5 | 58.5 | | to cluster |
| C_2 | 26.5 | 82.5 | 84.5 | 62 | | |
| C_3 | 21 | 61.5 | 61.5 | 56.5 | | |

| | | | | | | | | |
|----------|----|----|----|----|----|------|----|-------|
| S_1 | 18 | 73 | 75 | 51 | 10 | 52.3 | 28 | C_1 |
| S_2 | 18 | 71 | 85 | 75 | 25 | 19.8 | 62 | C_2 |
| S_3 | 23 | 70 | 70 | 52 | 27 | 60.3 | 23 | C_3 |
| S_4 | 20 | 55 | 55 | 55 | 51 | 90.3 | 16 | C_3 |
| S_5 | 22 | 85 | 86 | 81 | 41 | 13.8 | 74 | C_2 |
| S_6 | 19 | 91 | 90 | 89 | 56 | 28.8 | 92 | C_2 |
| S_7 | 20 | 70 | 65 | 60 | 24 | 60.3 | 46 | C_3 |
| S_8 | 21 | 53 | 56 | 59 | 50 | 86.3 | 17 | C_3 |
| S_9 | 19 | 82 | 82 | 60 | 10 | 32.3 | 46 | C_1 |
| S_{10} | 41 | 75 | 76 | 77 | 52 | 41.3 | 74 | C_2 |

C_1
 S_1
 S_9

C_2
 $S_2 S_5$
 $S_6 S_{10}$

C_3
 $S_3 S_4$
 $S_7 S_8$

∴ No change in clusters. Stop.

KNN CLASSIFICATION EXAMPLE

| S NO | NAME | AGE | GENDER | HOBBY |
|------|--------|-----|--------|----------|
| 1 | AKASH | 32 | MALE | MUSIC |
| 2 | ANU | 22 | FEMALE | GAMES |
| 3 | RIYA | 16 | FEMALE | MUSIC |
| 4 | RITHU | 27 | FEMALE | PAINTING |
| 5 | AKSHAY | 26 | MALE | MUSIC |
| 6 | RENU | 20 | FEMALE | GAMES |
| 7 | BINUSH | 19 | MALE | PAINTING |

ASHA - 21 - FEMALE - ?

| S NO | NAME | AGE | GENDER | DISTANCE | HOBBY |
|------|--------|-----|--------|----------|----------|
| 1 | AKASH | 32 | 0 | 11.04 | MUSIC |
| 2 | ANU | 22 | 1 | 1 | GAMES |
| 3 | RIYA | 16 | 1 | 5 | MUSIC |
| 4 | RITHU | 27 | 1 | 6 | PAINTING |
| 5 | AKSHAY | 26 | 0 | 5.09 | MUSIC |
| 6 | RENU | 20 | 1 | 1 | GAMES |
| 7 | BINUSH | 19 | 0 | 2.236 | PAINTING |

Consider,

male - 0, female - 1

test data : ASHA - 21 - FEMALE (1) - ?

ASHA - 21 - FEMALE (1) - GAMES

Euclidean distance = $\sqrt{(d_1-d_2)^2}$

Example of numerical on KNN classification

| Name | Acid durability | Strength | Class |
|----------|-----------------|----------|-------|
| Type - 1 | 7 | 7 | Bad |
| Type - 2 | 7 | 4 | Bad |
| Type - 3 | 3 | 4 | Good |
| Type - 4 | 1 | 4 | Good |

Test data \rightarrow Acid durability = 3

$$\text{Strength} = ?$$

$$\text{Class} = ?$$

| Name | Acid durability | Strength | Class | Distance |
|----------|-----------------|----------|-------|----------------------------------|
| Type - 1 | 7 | 7 | Bad | $\sqrt{(7-3)^2 + (7-7)^2} = 4$ |
| Type - 2 | 7 | 4 | Bad | $\sqrt{(7-3)^2 + (7-4)^2} = 5$ |
| Type - 3 | 3 | 4 | Good | $\sqrt{(3-3)^2 + (7-4)^2} = 3$ |
| Type - 4 | 1 | 4 | Good | $\sqrt{(3-1)^2 + (7-4)^2} = 3.6$ |

Write the Rank for the records based on the distance between them.

Type Acid Strength class distance Rank

dura
bility

Type-1 7 7 Bad 4 3

-2 7 4 Bad 5 4

-3 3 4 Good 3 1

-4 1 4 Good 3.6 2

For.

K = 1

Test data belongs to class "Good".

K = 2

Test data belongs to class "Good".

$p = \frac{1}{2}(F+D) + \frac{1}{2}(E+F)$. $\text{Ans} F$

$e = \frac{1}{2}(p) + \frac{1}{2}(E+F)$. $\text{Ans} + \frac{1}{2}$

$\bar{s} = \frac{1}{2}(p) + \frac{1}{2}(E+F)$. $\text{Ans} + \frac{1}{2}$

$p+s = \frac{1}{2}(p) + \frac{1}{2}(F+E)$. $\text{Ans} + \frac{1}{2}$

Based above all four above will always
be correctly classified (most likely) all the

Clustering Using Single linkage:-

Given the Mutual distance of 4 objects below, cluster them using hierarchical agglomerative single linkage clustering.

| | O_1 | O_2 | O_3 | O_4 |
|-------|-------|-------|-------|-------|
| O_1 | 0 | | | |
| O_2 | 1 | 0 | | |
| O_3 | 11 | 2 | 0 | |
| O_4 | 5 | 3 | 4 | 0 |

Step 1:

Merge O_1 and O_2 . Update distance matrix

| | $O_{1,2}$ | O_3 | O_4 |
|-----------|-----------|-------|-------|
| $O_{1,2}$ | 0 | | |
| O_3 | 2 | 0 | |
| O_4 | 3 | 4 | 0 |



$$\text{dist}(O_{1,2} \text{ and } O_3) = \min [\text{dist}(O_1, O_3), \text{dist}(O_2, O_3)]$$

$$= \min(11, 2) = 2$$

$$\text{dist}(O_{1,2} \text{ and } O_4) = \min [\text{dist}(O_1, O_4), \text{dist}(O_2, O_4)]$$

$$= \min(5, 3) = 3 \quad \text{Pg: ①}$$

Step 2: Merge $O_{1,2}$ and O_3 .

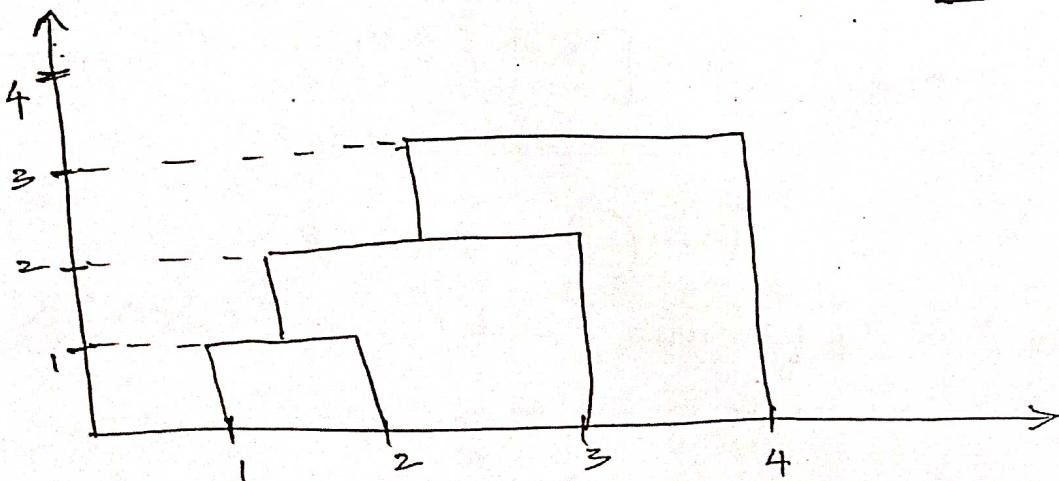
Update the distance matrix.

$$\begin{array}{cc} & O_{1,2,3} \quad O_4 \\ O_{1,2,3} & \left(\begin{array}{cc} 0 & \\ 0 & (3) \end{array} \right) \\ O_4 & \quad 0 \end{array}$$

$$\begin{aligned} \text{Dist}(O_{1,2,3} \text{ and } O_4) &= \min [\text{Dist}(O_{1,2} \text{ and } O_4), \\ &\quad \text{Dist}(O_3 \text{ and } O_4)] \\ &= \min [3, 4] \\ &= 3 \end{aligned}$$

Step 3: Merge $O_{1,2,3}$ and O_4 into one large cluster.

Dendrogram



Given

| | O_1 | O_2 | O_3 | O_4 |
|-------|-------|-------|-------|-------|
| O_1 | 0 | | | |
| O_2 | 1 | 0 | | |
| O_3 | 11 | 2 | 0 | |
| O_4 | 5 | 3 | 4 | 0 |

Step 1: Merge O_1 and O_2 . Update $\frac{\text{inter-cluster distance}}{n}$
matrix using complete linkage

| | $O_{1,2}$ | O_3 | O_4 |
|-----------|-----------|-------|-------|
| $O_{1,2}$ | 0 | | |
| O_3 | 11 | 0 | |
| O_4 | 5 | (4) | 0 |

$$\text{dist}(O_{1,2} \text{ and } O_3) = \max(\text{dist}(O_1, O_3) \text{ and } \text{dist}(O_2, O_3)) \\ = \max(11, 2) = 11$$

$$\text{dist}(O_{1,2} \text{ and } O_4) = \max(\text{dist}(O_1, O_4) \text{ and } \text{dist}(O_2, O_4))$$

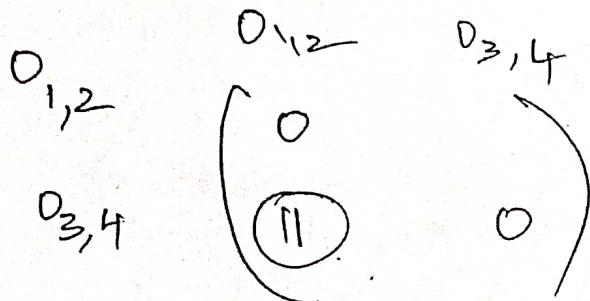
Pg. ③

$$= \max(5, 3) = 5$$

Step 2:

Merge O_3 and O_4

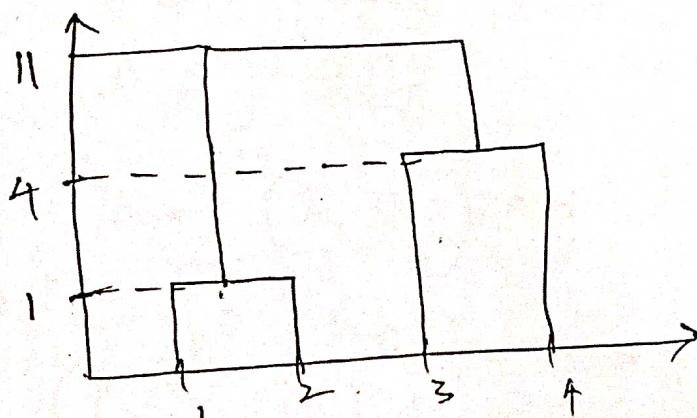
Update inter-cluster Dist. matrix



$$\begin{aligned} \text{Dist}(O_{1,2} \text{ and } O_{3,4}) &= \max(\text{Dist}(O_{1,2} \text{ and } O_3), \\ &\quad \text{Dist}(O_{1,2} \text{ and } O_4)) \\ &= \max(11, 5) \end{aligned}$$

$$= 11$$

Step 3: Merge $O_{3,4}$ and $O_{1,2}$ into one cluster



Dendrogram

Pg: 7

Given,

| | O_1 | O_2 | O_3 | O_4 |
|-------|-------|-------|-------|-------|
| O_1 | 0 | | | |
| O_2 | 1 | 0 | | |
| O_3 | 11 | 2 | 0 | |
| O_4 | 5 | 3 | 4 | 0 |

Step 1: Merge O_1 and O_2 .

Updated distance matrix

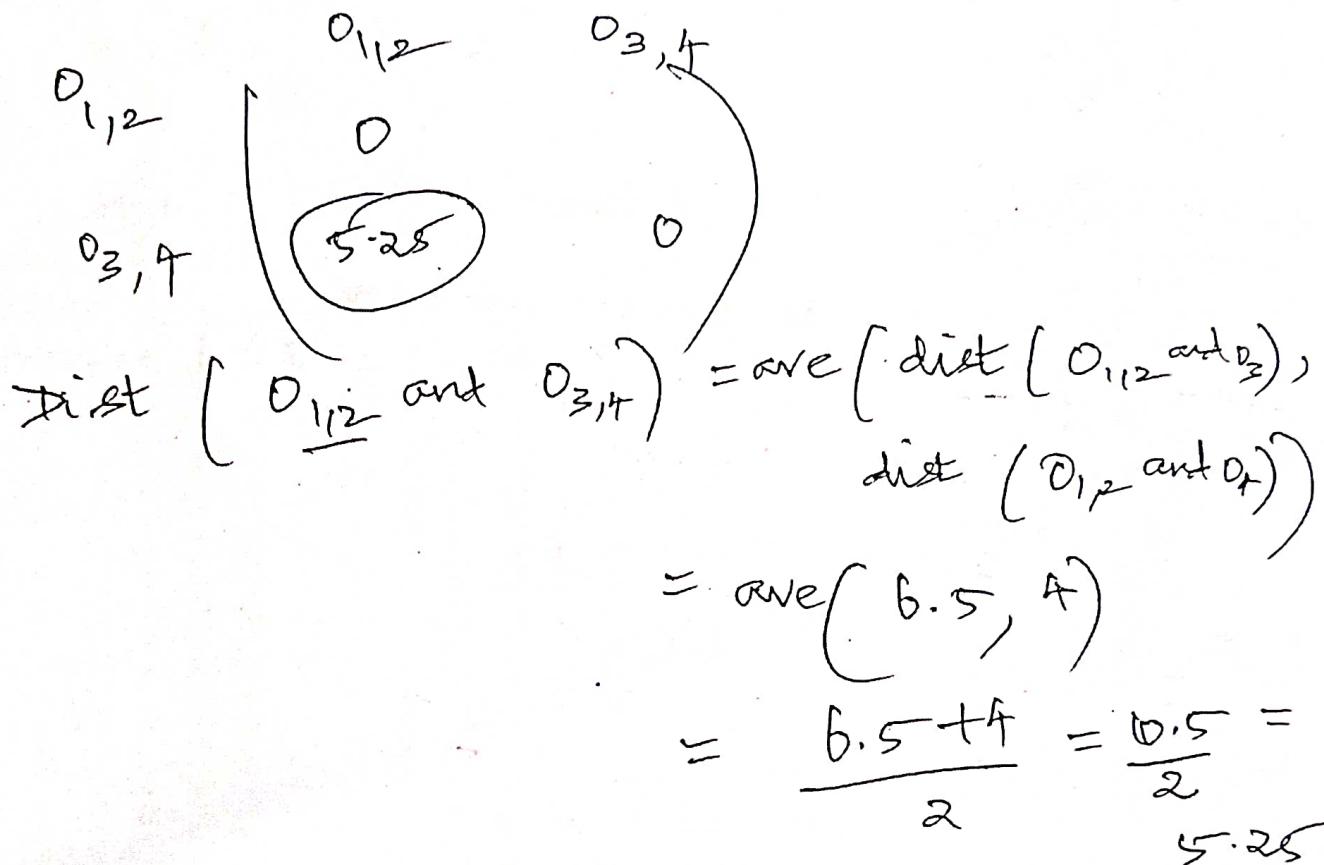
| | $O_{1,2}$ | O_3 | O_4 |
|-----------|-----------|-------|-------|
| $O_{1,2}$ | 0 | | |
| O_3 | 6.5 | 0 | |
| O_4 | 4 | 4 | 0 |

$$\text{Dist}(O_{1,2} \text{ and } O_3) = \text{average}(\text{Dist}(O_1, O_3) \text{ and } \text{Dist}(O_2, O_3)) \\ = \text{ave}(11, 2) = \frac{11+2}{2} = 6.5$$

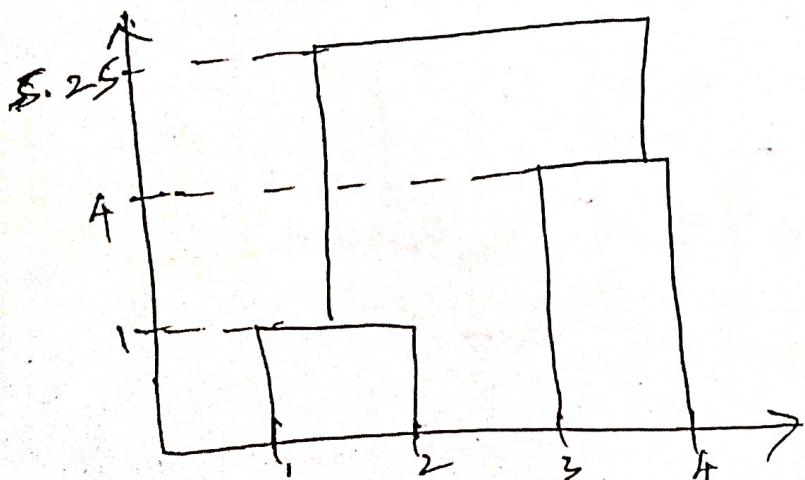
$$\text{Dist}(O_{1,2} \text{ and } O_4) = \text{ave}(\text{Dist}(O_1, O_4) \text{ and } \text{Dist}(O_2, O_4)) \\ = \text{ave}(5, 3) = \frac{5+3}{2} = 4.$$

Step3: Merge O_2 and O_4 }
 Merge $O_{1,2}$ and O_3 } same dist

∴ Updated distance matrix



Step4: Merge $O_{1,2}$ and $O_{3,4}$ into one large cluster.



Dendrogram

Pg. 6