CHAPTER-4

Classification, Prediction & Clustering

Based on CO4

Students should be able to Compare and evaluate different data mining techniques like classification, prediction and clustering.

Outline:

- Basic Concepts
- Classification methods:
  - Bayesian Classification: Naïve Bayes' Classifier.
  - Decision Tree Induction: Attribute Selection Measures, Tree pruning.
- Prediction: Structure of Regression models, Simple Linear Regression, Multiple Linear Regressions.
- Model Evaluation & Selection: Accuracy and Error measures, Holdout, Random Sampling, Cross Validation, Bootstrap.
- Clustering : Distance Measure, Partitioning Methods-(K-Means, K-Mediods), Hierarchical Methods-(Agglomerative, Divisive)

Basic Concept of Classification

University Question

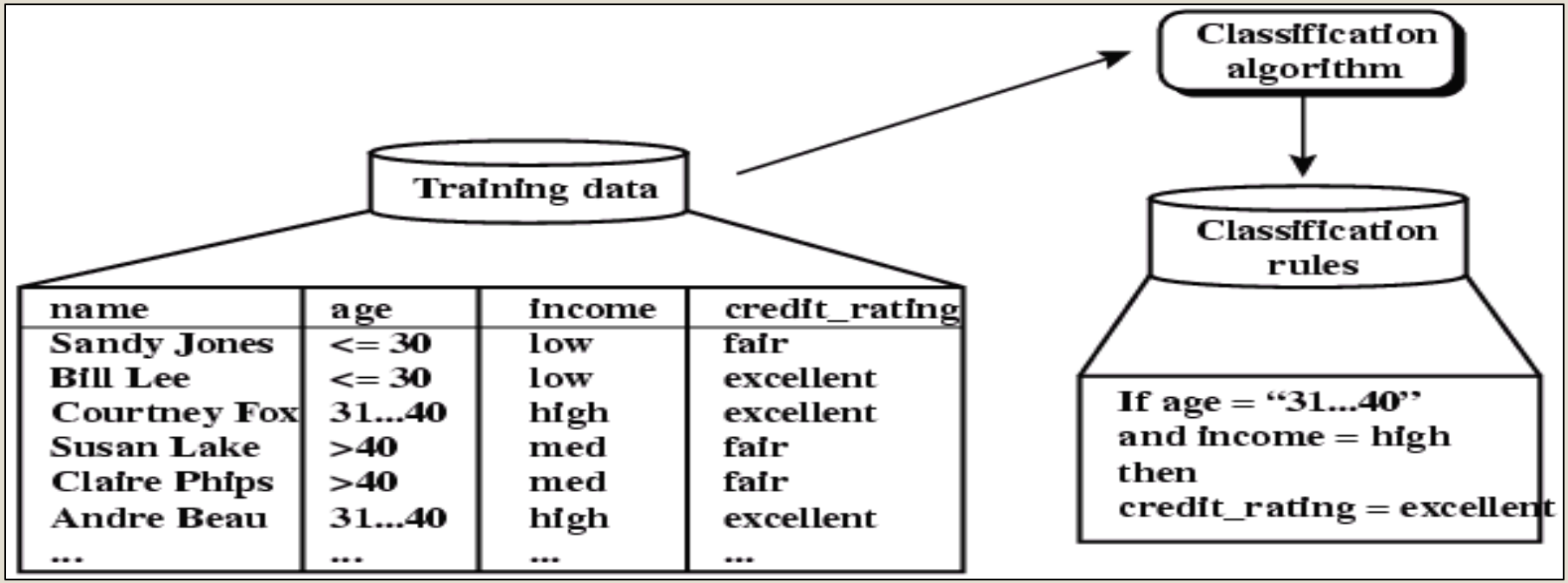
1. **What Is Classification? What Is The Issue In Classification? (May 2010)**

Classification (May 2010, May 2012, Dec 17)

- ***Definition:*** Classifies data (constructs a model) based on the training set and the values (class Labels) in a classifying attribute and uses it in classifying new data.
- Maps data into predefined groups or classes.
- **Example:** *Teacher's classify student's grades as A, B, C, D based on their marks.*
- **Typical Applications:**
  - **Credit card approval, Medical Diagnosis etc.**
- **Classification Techniques:**
  - **Bayes Classification**
  - **Decision Tree**

Classification Process: Learning Phase/ Model Construction

1. **Learning/ Model Construction**
   - Training data are analyzed by a classification algorithm
   - Each tuple is assumed to belong to a predefined class as determined by the **class label attribute here credit_rating**
   - The set of tuples used for model construction is called as **training set.**

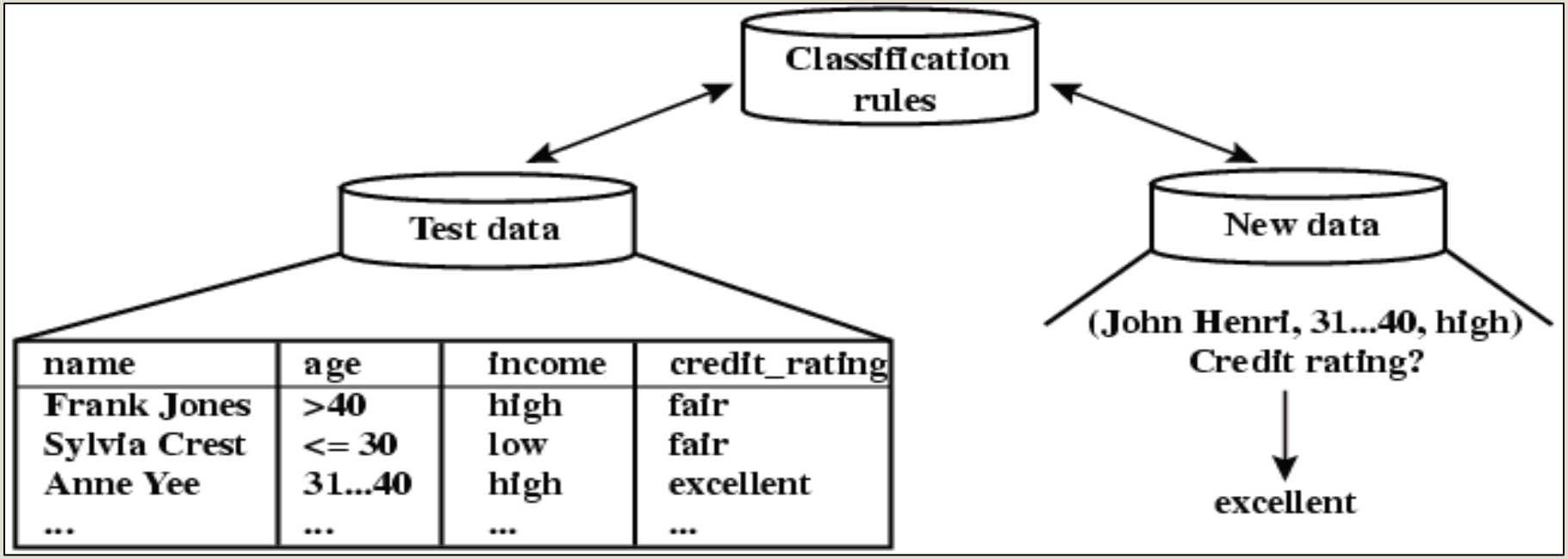


name	age	income	credit_rating
Sandy Jones	<= 30	low	fair
Bill Lee	<= 30	low	excellent
Courtney Fox	31...40	high	excellent
Susan Lake	>40	med	fair
Claire Phips	>40	med	fair
Andre Beau	31...40	high	excellent
...	...	...	...

Classification Process: Model Usage/ Testing Phase

2. Testing / Model Usage

- Test data are used to estimate the accuracy of the classification rules
- If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples



name	age	income	credit_rating
Frank Jones	>40	high	fair
Sylvia Crest	<= 30	low	fair
Anne Yee	31...40	high	excellent
...	...	...	...

Supervised vs. Unsupervised Learning

Classification consists of assigning a class label to a set of unclassified cases.

1. **Supervised Classification**

   The set of possible classes is known in advance. New data is classified based on the training set

2. **Unsupervised Classification**

   Set of possible classes is not known. After classification we can try to assign a name to that class. Unsupervised classification is called **clustering.**

Supervised Classification contd.

- The input data, also called the training set, consists of multiple records each having multiple attributes or features.
- Each record is tagged with a class label.
- *The objective of classification is to analyze the input data and to develop an accurate description or model for each class using the features present in the data.*
- This model is used to classify test data for which the class descriptions are not known.

Naïve Bayesian Theorem

1. Numerical on Classification based on Statistical Based Algorithm. **(May 2010)**
2. Why naïve Bayesian Classification is called "naïve"? Briefly outline the major ideas of navies Bayesian Classification. **(Dec 16, May 18)**

Things We'd Like to Do- Bayes' Theorem

- Spam Classification
  - Given an email, predict whether it is spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has disease X or not
- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

Bayes Theorem

- **Bayes Theorem**, also known as Bayes Rule is a useful tool for calculating conditional probabilities.
- All probabilities are between 0 and 1. For any proposition a $0 \leq P(a) \leq 1$ P(true)=1, P(false)=0

1. Before the evidence is obtained; **prior probability**
   - *P(a)* the prior probability that the proposition is true
   - *P(cavity)=0.1*
2. After the evidence is obtained; **posterior probability**
   - *P(a| b)*
   - The probability of 'a' given that all we know is 'b'
   - *P(cavity| toothache)=0.8*

Axioms of Probability

- Kolmogorov's axioms, first published in German 1933.
- The probability of disjunction is given by

$$\boxed{P(a \vee b) = P(a) + P(b) - P(a \wedge b)}$$

- Product rule

$$\boxed{\begin{aligned} P(a \wedge b) &= P(a \mid b)P(b) \\ P(a \wedge b) &= P(b \mid a)P(a) \end{aligned}}$$

So, we can write P(a|b)P(b)= P(b|a).P(a)

Bayes Theorem

- Given a hypothesis h and data D which bears on the hypothesis:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- *P(h)*: independent probability of *h*: *prior probability* (past or earlier)
- *P(D)*: independent probability of *D*
- *P(D|h)*: conditional probability of *D* given h: *likelihood* (possibility) [**P(Sunny|P_Yes)] =2/9**
- *P(h|D)*: conditional probability of *h* given *D*: *posterior probability* (next or later) [**P(P_Yes |Outlook=Overcast, temp=Hot)]**

Maximum A Posterior : $P(h|D)$ ie P(P_Yes |Outlook_Overcast, temp=Hot)

- Based on Bayes Theorem, we can compute the *Maximum A Posterior* (MAP) hypothesis for the data
- We are interested in the best hypothesis for some space H given observed training data D.

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}}\, P(h \mid D)$$

$$= \underset{h \in H}{\operatorname{argmax}} \frac{P(D \mid h)P(h)}{P(D)}$$

$$= \underset{h \in H}{\operatorname{argmax}}\, P(D \mid h)P(h)$$

H: set of all hypothesis.

Note that we can drop P(D) as the probability of the data is constant (and independent of the hypothesis).

Example 1: Play-tennis example: estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14
P(n) = 5/14

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p)= 6/9	P(normal|n) =1/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

Play-Tennis example: Classifying X

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- An unseen sample

X=<Outlook, Temperature, Humidity, Windy>

with values **X = <rain, hot, high, false>**

- **P(p|X) =P(X|p)·P(p)** =P(rain|p)·P(hot|p)· P(high|p)· P(false|p)· P(p) =3/9·2/9·3/9·6/9·9/14 = **0.010582**
- **P(n|X)=P(X|n)·P(n)**=P(rain|n)· P(hot|n)· P(high|n)· P(false|n)· P(n) = 2/5·2/5·4/5·2/5·5/14 = **0.018286**
- So given the data Sample X , is classified in class **n** (don't play) i.e **P(n|X)..** ***X belongs to class don't play***

Example 2: Buying Computer Example :Estimating $P(x_i|C)$ May 17

Apply the Naive Bayes classifier algorithm for buys computer classification a classify the tuple X=(age=''young'', income=''medium'', student=''yes'' and credit-rating=''fair'')

Id	Age	Income	Student	Credit-rating	buys computer
1	young	high	no	fair	no
2	young	high	no	good	no
3	middle	high	no	fair	yes
4	old	medium	no	fair	yes
5	old	low	yes	fair	yes
6	old	low	yes	good	no
7	middle	low	yes	good	yes
8	young	medium	no	fair	no
9	young	low	yes	fair	yes
10	old	medium	yes	fair	yes
11	young	medium	yes	good	yes
12	middle	medium	no	good	yes
13	middle	high	yes	fair	yes
14	old	medium	no	good	no

P(H|D)=P(D|H).P(H)

- P(Buy|X)=P(X|H).P(H)

=P(Young|yes). P(Medium|yes). P(Styes|yes). P(fair|yes). P(yes)= (2/9). (4/9).(6/9).(6/9).(9/14)

= 0.02822

- P(NotBuy|X)=P(X|H).P(H)

=P(Young|No). P(Medium|No). P(Styes|No). P(fair|No). P(No)= (3/5).(2/5). (1/5).(2/5).(5/14) =0.006857

As the probability of Buy computer is the highest the Data X belongs to the class label Buys computer.

Example 2: Buying Computer Example :Estimating $P(x_i|C)$

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data sample
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

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

Example 2: Buying Computer Example :Estimating $P(x_i|C)$

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

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

Example 3: Naive Bayesian Classifier

Apply Statistical based algorithm to obtain the actual probabilities of each event to classify the new tuple as a tall. Use the following data: (May 2010)

Person ID	Name	Gender	Height	Class
1	Kristina	Female	1·6 m	Short
2	Jim	Male	2 m	Tall
3	Maggi	Female	1·9 m	Medium
4	Marya	Female	2·1 m	Tall
5	Stephanie	Female	1·7 m	Short
6	Bob	Male	1·85 m	Medium
7	Catherine	Female	1·6 m	Short
8	Dave	Male	1·7 m	Short
9	Wilson	Male	2·2 m	Tall

Sample:

Gender	HEIGHT	class
F	0 to 1.7	short
M	>1.9 to infinity	tall
F	>1.7 to 1.9	medium
F	>1.9 to infinity	tall
F	0 to 1.7	short
M	>1.7 to 1.9	medium
F	0 to 1.7	short
M	0 to 1.7	short
M	>1.9 to infinity	tall

0 to 1.7 : Short
>1.7 to 1.9 : Medium
>1.9 to infinity: tall

Naive Bayesian Classifier- *To classify new tuple as a tall*

- Given a training set, we can compute the probabilities **X = <Adam, Male, 1.95m>**

Gender	Short	Medium	Tall
Male	1/4	1/2	2/3
Female	3/4	1/2	1/3

Height	Short	Medium	Tall
[0,1.6]	2/4	0	0
[1.6, 1.7]	2/4	0	0
[1.7,1.8]	0	0	0
[1.8,1.9]	0	2/2	0
[1.9,2.0]	0	0	1/3
[2.0,infinity]	0	0	2/3

P(Short) = 4/9
P(Medium) = 2/9
P(Tall)= 3/9

To classify new tuple as a tall : Classifying X

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- An unseen sample **X=<Name, Gender, Height>**

  with values **X = <Adam, Male,1.95m>**
- **P(short|X) =P(X|Short)·P(short)** =P(Male|short)·P(1.95|short)· P(short) =1/4*0*4/9= **0**
- **P(Medium|X)=P(X|Medium)·P(medium)**=P(Male|medium)·P(1.95|medium)· P(medium) =1/2*0*2/9= **0**
- **P(Tall|X)=P(X|Tall)·P(Tall)**=P(Male|Tall)·P(1.95|Tall)· P(Tall) =2/3*1/3*3/9= 0.66* 0.33 *0.333=**0.0738**
- So given the data Sample X , is classified in class **Tall** i.e **P(Tall|X)..** ***X belongs to class tall as it has highest probability.***

Naïve Bayesian Classifier: Comments

- **Advantages**
  - Easy to implement
  - Good results obtained in most of the cases
- **Disadvantages**
  - Assumption: Class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.

    Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

Decision Tree

1. **Explain ID3 algorithm. What are the pros and cones of it? (Dec 2011)**
2. **Numerical on ID3 (May 2011, MAY 2013)**

Decision Tree

- A decision tree is a flow chart like tree structure, where
  - **Each *internal node* denotes a *test* on an attribute**
  - **Each *branch* denotes an *outcome* of the test**
  - **Each *leaf node* represent a *class***
- In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree.

Decision Tree: ID3 (Iterative Dichotomiser 3)

- A mathematical algorithm for building the decision tree.
- **Invented by J. Ross Quinlan in 1979**.
- Uses **Information Theory invented by Shannon in 1948**.
- Builds the tree from the top down, with no backtracking.
- Information Gain is used to select the most useful attribute for classification.
- *Main aim is to choose attribute which is having the highest information gain and then split it to reach the leaf node..*

Choosing Attributes

- The order in which attributes are chosen determines how complicated the tree is.
- ID3 uses information theory to determine the most informative attribute.
- A measure of the information content of a message is the inverse of the probability of receiving the message:

$$information(M) = 1/probability(M)$$

- Taking logs (base 2) makes information correspond to the number of bits required to encode a message:

$$information(M) = -log_2(probability(M))$$

Entropy

- Different messages have different probabilities of arrival.
- Overall level of uncertainty (termed entropy) is:

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

- A completely homogeneous sample has entropy of 0. (all yes or all no)
- An equally divided sample has entropy of 1.
- Entropy(s) = - p+log2 (p+) -p-log2 (p-) for a sample of negative and positive elements.
- Example: S is a collection of 14 examples with 9 YES and 5 NO examples then

  Entropy(S) = - (9/14) Log_2 (9/14) - (5/14) Log_2 (5/14)

  = 0.940

Calculating Entropy with Code

- Most programming languages and calculators do not have a log2 function.
- Use a conversion factor
- Take log function of 2, and divide by it.
- Example: log10(2) = .301
- Then divide to get log2(n):
- log10(9/14) / .301 = -0.1918/0.301=log2(3/5)

  = -0.6374

Information Gain (IG)

- The information gain is based on the decrease in entropy after a dataset is split on an attribute.
- Which attribute creates the most homogeneous branches?

1. First the entropy of the total dataset is calculated.
2. The dataset is then split on the different attributes.
3. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split.
4. The resulting entropy is subtracted from the entropy before the split.
5. The result is the Information Gain, or decrease in entropy.
6. The attribute that yields the largest IG is chosen for the decision node.

Information Gain (cont'd)

7. A branch set with entropy of 0 is a leaf node.
8. Otherwise, the branch needs further splitting to classify its dataset.
9. The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Construct a decision tree for the below example: (Numerical Example)

- Suppose we want ID3 to decide whether the weather is amenable to playing baseball. Over the course of 2 weeks, data is collected to help ID3 build a decision tree. The target classification class is "should we play baseball?" which can be yes or no.
- The weather attributes can have the following values:
  - Outlook = { sunny, overcast, rain }
  - Temperature = {hot, mild, cool }
  - Humidity = { high, normal }
  - Wind = {weak, strong }

Weather Data: Play or not Play?

Class

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Calculate Entropy for Set S containing 14 Examples:9 +ve and 5 -ve

9 Play samples
5 Don't
A total of 14.

1. First the entropy of the total dataset is calculated.

Entropy(S) = - (9/14) Log_2 (9/14) - (5/14) Log_2 (5/14)= 0.940

2. The dataset is then split on the different attributes.

Example: Outlook, Temperature, Humidity, Wind

3. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split.
4. The resulting entropy is subtracted from the entropy before the split.

- Example: **Information Gain(s, Wind)= Entropy(s)- ((8/14) * Entropy(weak)- (6/14) * Entropy(Strong))= Entropy(s)- Entropy(Wind)**

5. The result is the Information Gain, or decrease in entropy.

Wind	Play ball
Weak	No
Strong	No
Weak	Yes
Weak	Yes
Weak	Yes
Strong	No
Strong	Yes
Weak	No
Weak	Yes
Weak	Yes
Strong	Yes
Strong	Yes
Weak	Yes
Strong	No

Total 14 training instances

Wind= Strong (6)

Yes-3
No-3

Wind= Weak(8)

Yes-6
No-2

For Strong & Weak Winds:

Entropy(Strong) = - (3/6) Log2 (3/6) - (3/6) Log2 (3/6)
= 1.00

Entropy(Weak) = - (6/8) Log2 (6/8) - (2/8) Log2 (2/8)
= 0.811

Entropy (wind)= (6/14)*Entropy(Strong)+(8/14)*Entropy(Weak)
=0.89215

Information Gain(s, Wind) = Entropy(s)- Entropy(Wind)=0.940- 0.89215

Information Gain(s, Wind) =0.048

Outlook	Playball
Sunny	No
Sunny	No
Overcast	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rain	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rain	No

Total 14 training Instances

Outlook = Sunny(5) — Yes-2, No-3

Outlook = Overcast (4) — Yes-4

Outlook = Rain (5) — Yes-3, No-2

For Sunny, Overcast, Rain Outlook:

Entropy(Sunny) = $-(2/5)\,\mathrm{Log}_2\,(2/5) - (3/5)\,\mathrm{Log}_2\,(3/5) = 0.9709$

Entropy(Overcast) = $-(4/4)\,\mathrm{Log}_2\,(4/4) - (0/4)\,\mathrm{Log}_2\,(0/4) = 0$

Entropy(Rain) = $-(3/5)\,\mathrm{Log}_2\,(3/5) - (2/5)\,\mathrm{Log}_2\,(2/5) = 0.9709$

Entropy (outlook)=
(5/14)*Entropy(Sunny)+(4/14)*Entropy(Overcast)
+(5/14)*Entropy(Rain)=0.69353

Information Gain(s, Outlook)
= Entropy(s)- Entropy (outlook)= 0.940-0.69353
=Information Gain(s, Outlook) =0.2467

Temperature	Play ball
Hot	No
Hot	No
Hot	Yes
Mild	Yes
Cool	Yes
Cool	No
Cool	Yes
Mild	No
Cool	Yes
Mild	Yes
Mild	Yes
Mild	Yes
Hot	Yes
Mild	No

Total 14 training Instances

Temp = Hot(4) — Yes-2, No-2

Temp = Mild(6) — Yes-4, No-2

Temp = Cool(4) — Yes-3, No-1

For Hot, Mild, Cool Temperature:

Entropy(Hot) = $-(2/4)\ \mathrm{Log}_2\ (2/4) - (2/4)\ \mathrm{Log}_2\ (2/4) = 1$

Entropy(Mild) = $-(4/6)\ \mathrm{Log}_2\ (4/6) - (2/6)\ \mathrm{Log}_2\ (2/6) = 0.91829$

Entropy(Cool) = $-(3/4)\ \mathrm{Log}_2\ (3/4) - (1/4)\ \mathrm{Log}_2\ (1/4) = 0.8112$

Entropy (Temperature)= -(4/14)*Entropy(Hot)+(6/14)*Entropy(Mild) +(4/14)*Entropy(Cool)=0.91106

Information Gain(s, Temperature)

= Entropy(s)- Entropy (Temperature) = 0.940-0.91106

=Information Gain(s, Temperature) =0.0292

Humidity	Play ball
High	No
High	No
High	Yes
High	Yes
Normal	Yes
Normal	No
Normal	Yes
High	No
Normal	Yes
Normal	Yes
Normal	Yes
High	Yes
Normal	Yes
High	No

Total 14 training instances

Humidity= High (7) — Yes-3, No-4

Humidity= Normal (7) — Yes-6, No-1

For High & Normal Humidity:

Entropy(High) = - (3/7) Log2 (3/7) - (4/7) Log2 (4/7)
= 0.98522

Entropy(Normal) = - (6/7) Log2 (6/7) - (1/7) Log2 (1/7)
= 0.59167

Entropy (Humidity)=
(7/14)*Entropy(High)+(7/14)*Entropy(Normal) =0.7884

Information Gain(s, Humidity)
= Entropy(s)- Entropy (Humidity)= 0.940-0.7884

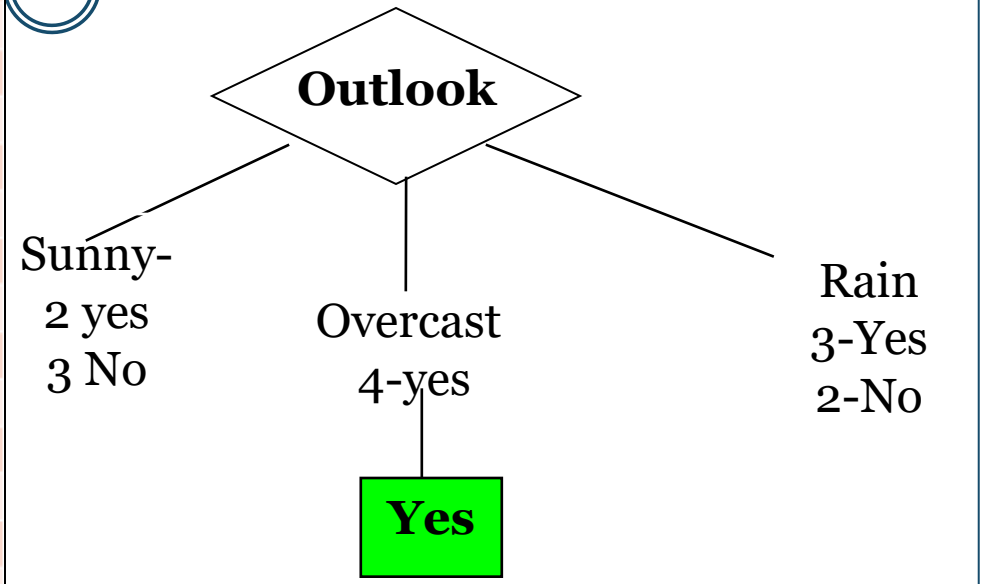
Information Gain(s, Humidity) =0.1518

7. The attribute that yields the largest IG is chosen for the decision node. A branch set with entropy of 0 is a leaf node.
8. Otherwise, the branch needs further splitting to classify its dataset.
9. The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Information Gain(s, Wind) =0.048
Information Gain(s, Outlook) =0.2467
Information Gain(s, Temperature) =0.0292
Information Gain(s, Humidity) =0.1518

Outlook-(Sunny, Overcast, Rain)

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



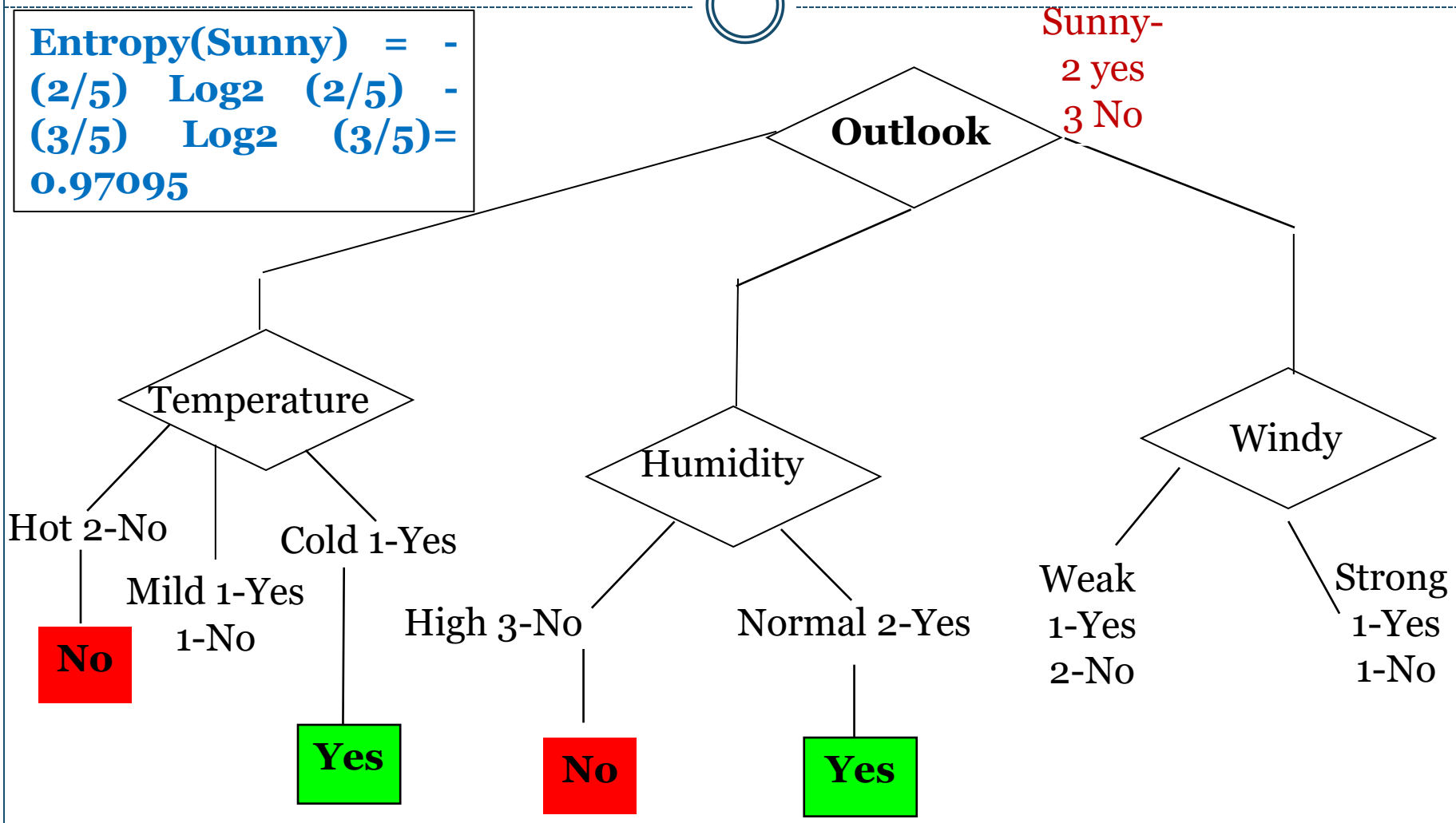
Entropy(Sunny) = - (2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095

Entropy(Overcast) = - (4/4) Log2 (4/4) - (0/4) Log2 (0/4)= 0

Entropy(Rain) = - (3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095

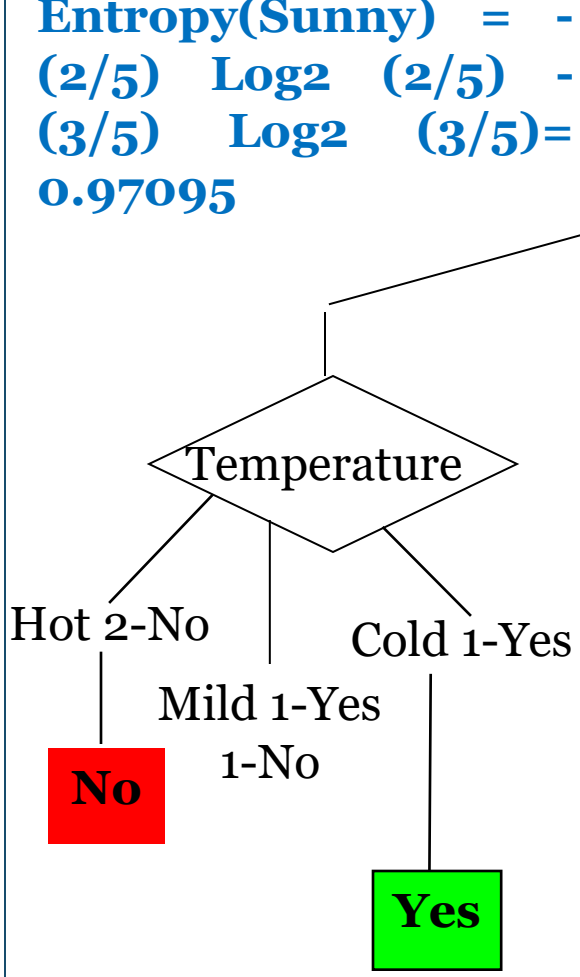
Outlook-(Sunny): Temp, Humidity, Windy

Entropy(Sunny) = -(2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095



Outlook-(Sunny): Temp

Entropy(Sunny) = -(2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095



Entropy(Hot) = -(0/2) Log_2 (0/2) - (2/2) Log_2 (2/2)= 0

Entropy(Mild) = -(1/2) Log_2 (1/2) - (1/2) Log_2 (1/2)= 1

Entropy(Cool) = -(1/1) Log_2 (1/1) - (0/1) Log_2 (0/1)= 0

Entropy (Temperature)= (2/5)*Entropy(Hot)+(2/5)*Entropy(Mild) +(1/5)*Entropy(Cool)=0.4

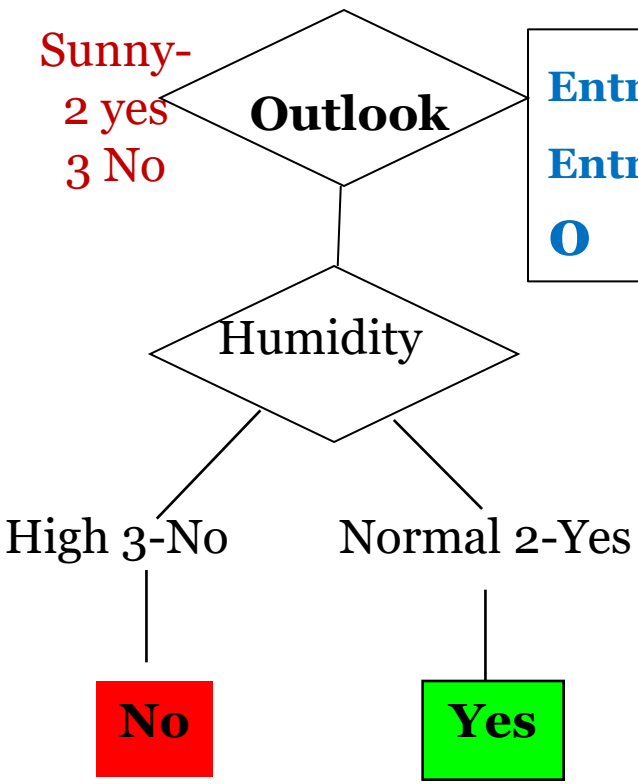
Information Gain(s, Temperature)

= Entropy(sunny)- Entropy (Temperature) = 0.940-0.4

=Information Gain(s, Temperature) =0.5709

Outlook-(Sunny): Humidity

Entropy(Sunny) = - (2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095



Entropy(High) =-(0/3) Log_2 (0/3) - (3/3) Log_2 (3/3)= 0

Entropy(Normal) =-(2/2) Log_2 (2/2) - (0/2) Log_2 (0/2)= 0

Entropy (Humidity)= (3/5)*Entropy(High)+(2/5)*Entropy(Normal) =0

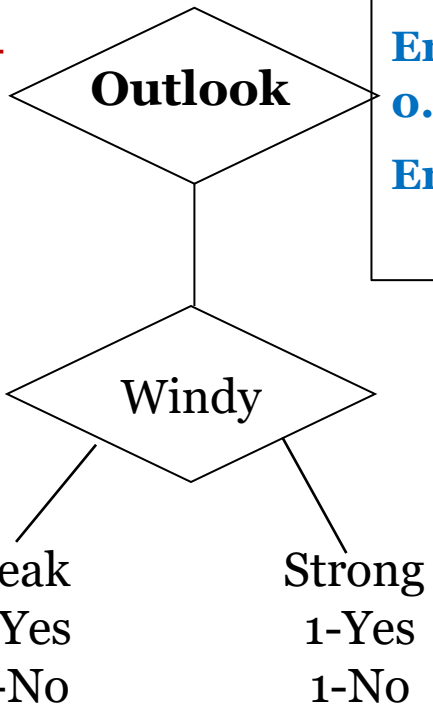
Information Gain(s, Humidity)
= Entropy(sunny)- Entropy (Humidity) = 0.940-0
=Information Gain(s, Humidity) =0.940

Outlook-(Sunny): Windy

Entropy(Sunny) = -(2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095

Information Gain(s, Wind) =0.0199
Information Gain(s, Temperature) =0.5709
Information Gain(s, Humidity) =0.9709

Sunny-
2 yes
3 No



Entropy(Weak) = -(1/3) Log_2 (1/3) - (2/3) Log_2 (2/3)= 0.91829

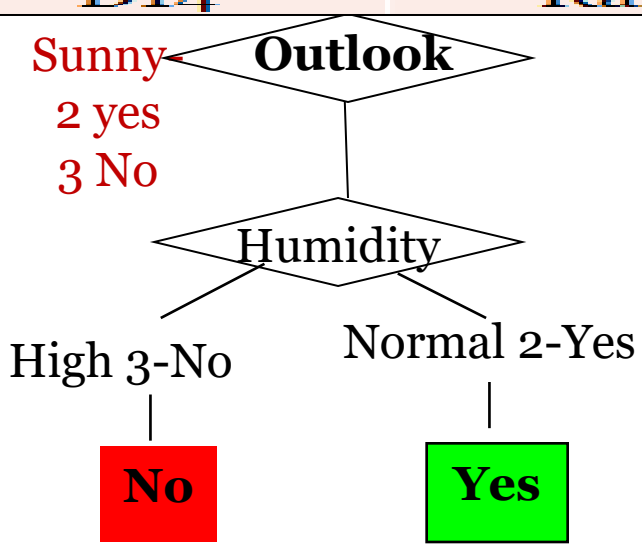
Entropy(Strong) = -(1/2) Log_2 (1/2) - (1/2) Log_2 (1/2)= 1

Entropy (Windy)= (3/5)*Entropy(Weak)+(2/5)*Entropy(Strong) =0.95097

Information Gain(s, Temperature)
= Entropy(sunny)- Entropy (Windy) = 0.940-0.95
=Information Gain(s, Temperature) =0.0199

Outlook-(Sunny): Humidity-High

Day	Outlook	Humidity	Play ball
D1	Sunny	High	No
D2	Sunny	High	No
D3	Overcast	High	Yes
D4	Rain	High	Yes
D5	Rain	Normal	Yes
D6	Rain	Normal	No
D7	Overcast	Normal	Yes
D8	Sunny	High	No
D9	Sunny	Normal	Yes
D10	Rain	Normal	Yes
D11	Sunny	Normal	Yes
D12	Overcast	High	Yes
D13	Overcast	Normal	Yes
D14	Rain	High	No

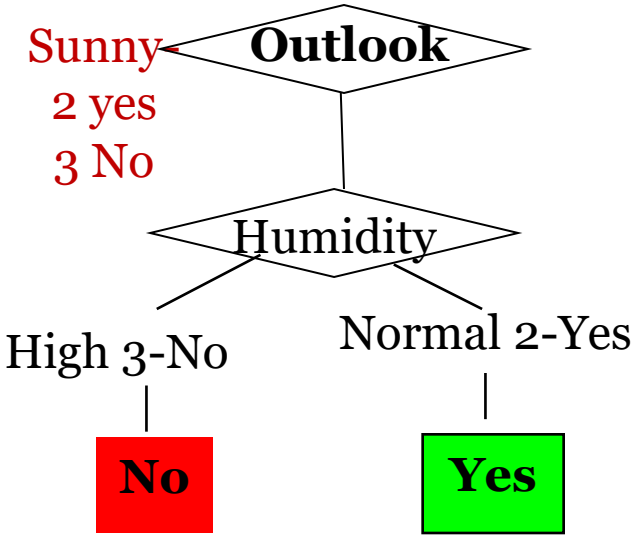


Entropy(Normal) =-(2/2) Log_2 (2/2) - (0/2) Log_2 (0/2)= 0

A branch set with entropy of 0 is a leaf node.

Outlook-(Sunny): Humidity-Normal

Day	Outlook	Humidity	Play ball
D1	Sunny	High	No
D2	Sunny	High	No
D3	Overcast	High	Yes
D4	Rain	High	Yes
D5	Rain	Normal	Yes
D6	Rain	Normal	No
D7	Overcast	Normal	Yes
D8	Sunny	High	No
D9	Sunny	Normal	Yes
D10	Rain	Normal	Yes
D11	Sunny	Normal	Yes
D12	Overcast	High	Yes
D13	Overcast	Normal	Yes
D14	Rain	High	No

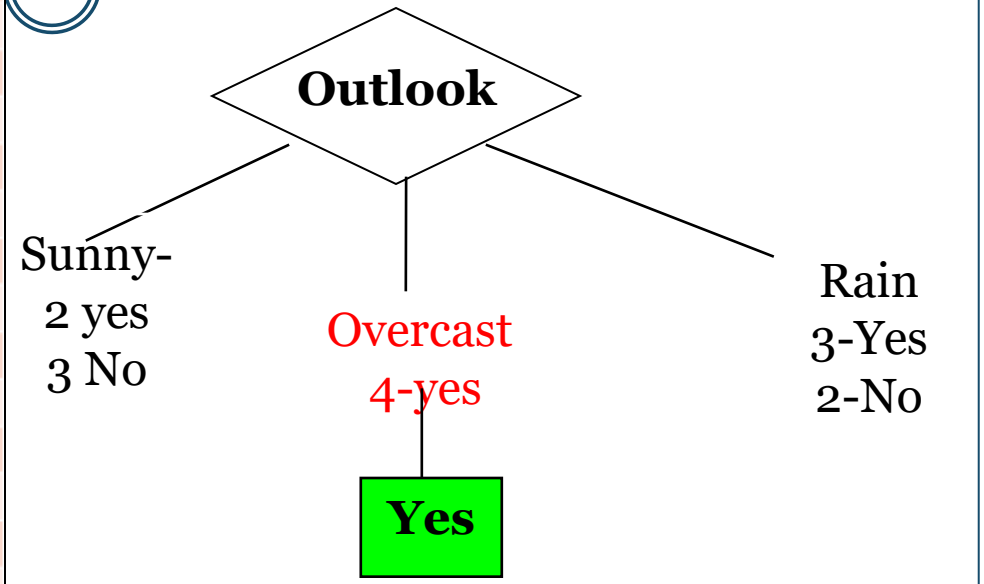


Entropy(High) =-(0/3) Log_2 (0/3) - (3/3) Log_2 (3/3)= 0

A branch set with entropy of 0 is a leaf node.

Outlook-(Sunny, Overcast, Rain)

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Entropy(Sunny) = - (2/5) Log2 (2/5) - (3/5) Log2 (3/5)= 0.97095

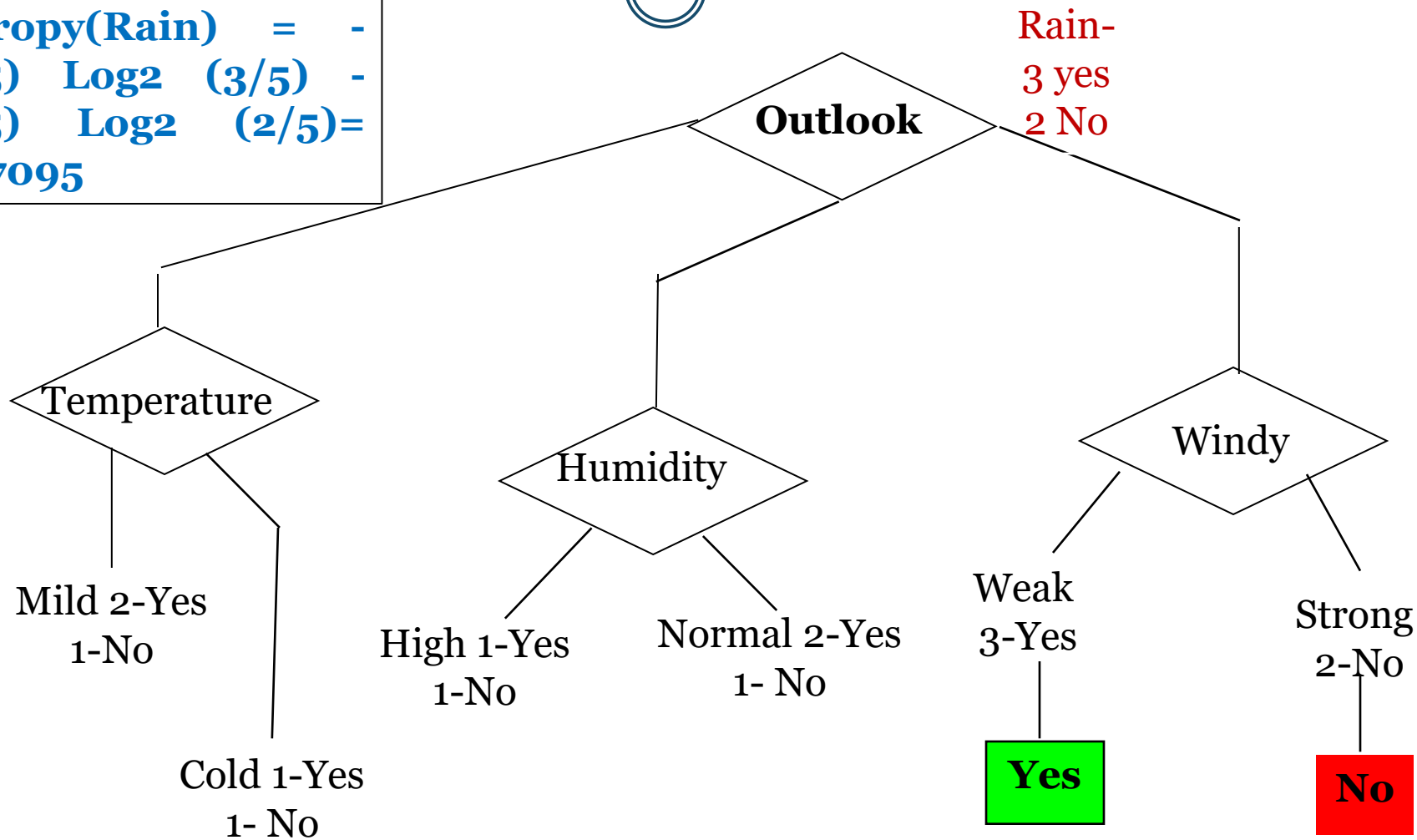
Entropy(Overcast) = - (4/4) Log2 (4/4) - (0/4) Log2 (0/4)= 0

(A branch set with entropy of 0 is a leaf node)

Entropy(Rain) = - (3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095

Outlook-(Rain): Temp, Humidity, Windy

Entropy(Rain) = -(3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095



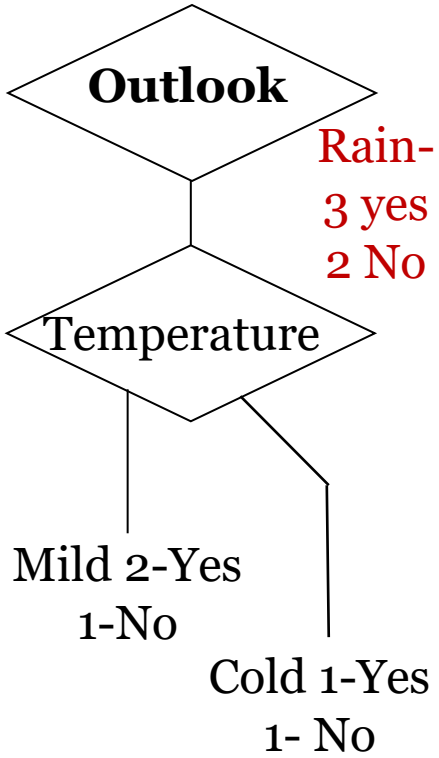
Outlook-(Rain): Temp

Entropy(Rain) = -(3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095

Entropy(Hot) = 0

Entropy(Mild) = -(2/3) Log_2 (2/3) - (1/3) Log_2 (1/3)= 0.91829

Entropy(Cold) = -(1/2) Log_2 (1/2) - (1/2) Log_2 (1/2)= 1



Entropy (Temperature)= (3/5)*Entropy(Mild)+(2/5)*Entropy(cold) =0.95097

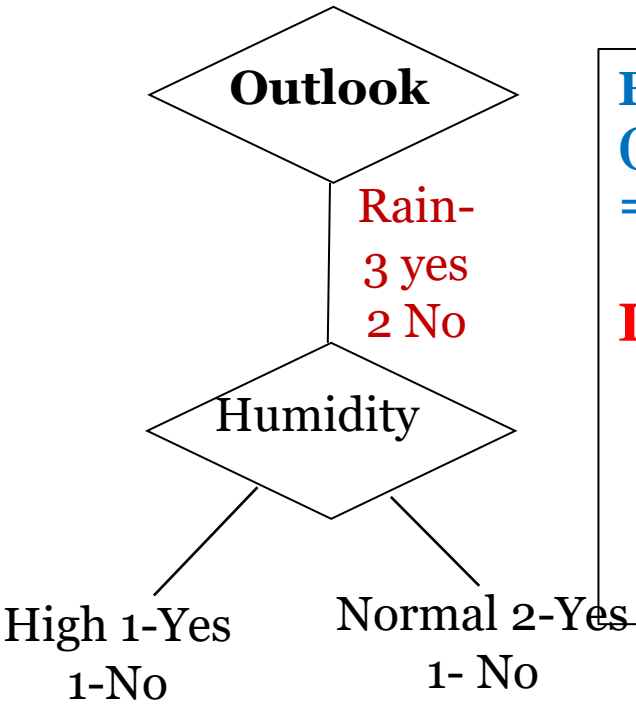
Information Gain(s, Temperature) = Entropy(Rain)- Entropy (Temperature) = 0.97095-0.95097 =Information Gain(s, Temperature) =0.0199

Outlook-(Rain): Humidity

Entropy(Rain) = -(3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095

Entropy(High) = -(1/2) Log_2 (1/2) - (1/2) Log_2 (1/2)= 1

Entropy(Normal) = -(2/3) Log_2 (2/3) - (1/3) Log_2 (1/3)= 0.91829

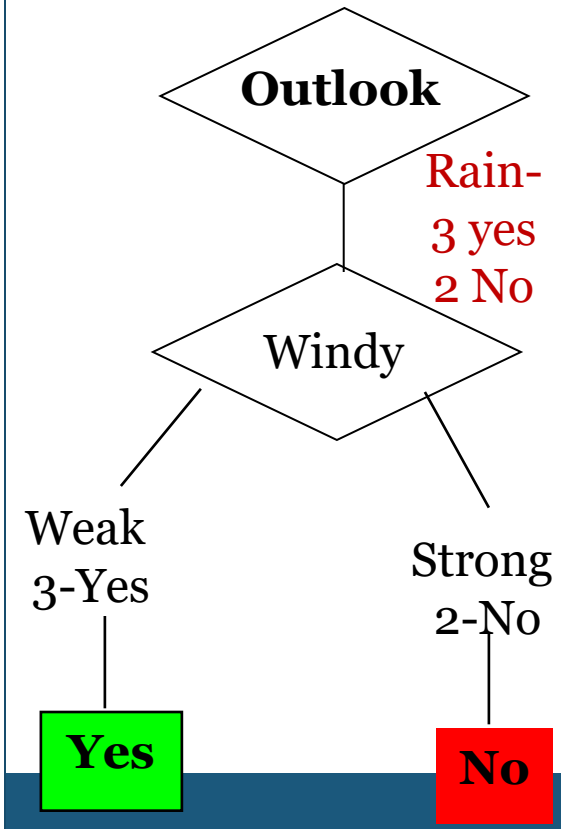


Entropy (Humidity)= (2/5)*Entropy(High)+(3/5)*Entropy(Normal) =0.95097

Information Gain(s, Humidity)
= Entropy(Rain)- Entropy (Humidity)
= 0.97095-0.95097
=Information Gain(s, Temperature)
=0.0199

Outlook-(Rain): Windy

Entropy(Rain) = -(3/5) Log2 (3/5) - (2/5) Log2 (2/5)= 0.97095



Entropy(Weak) = -(3/3) Log_2 (3/3) - (0/3) Log_2 (0/3)= 0

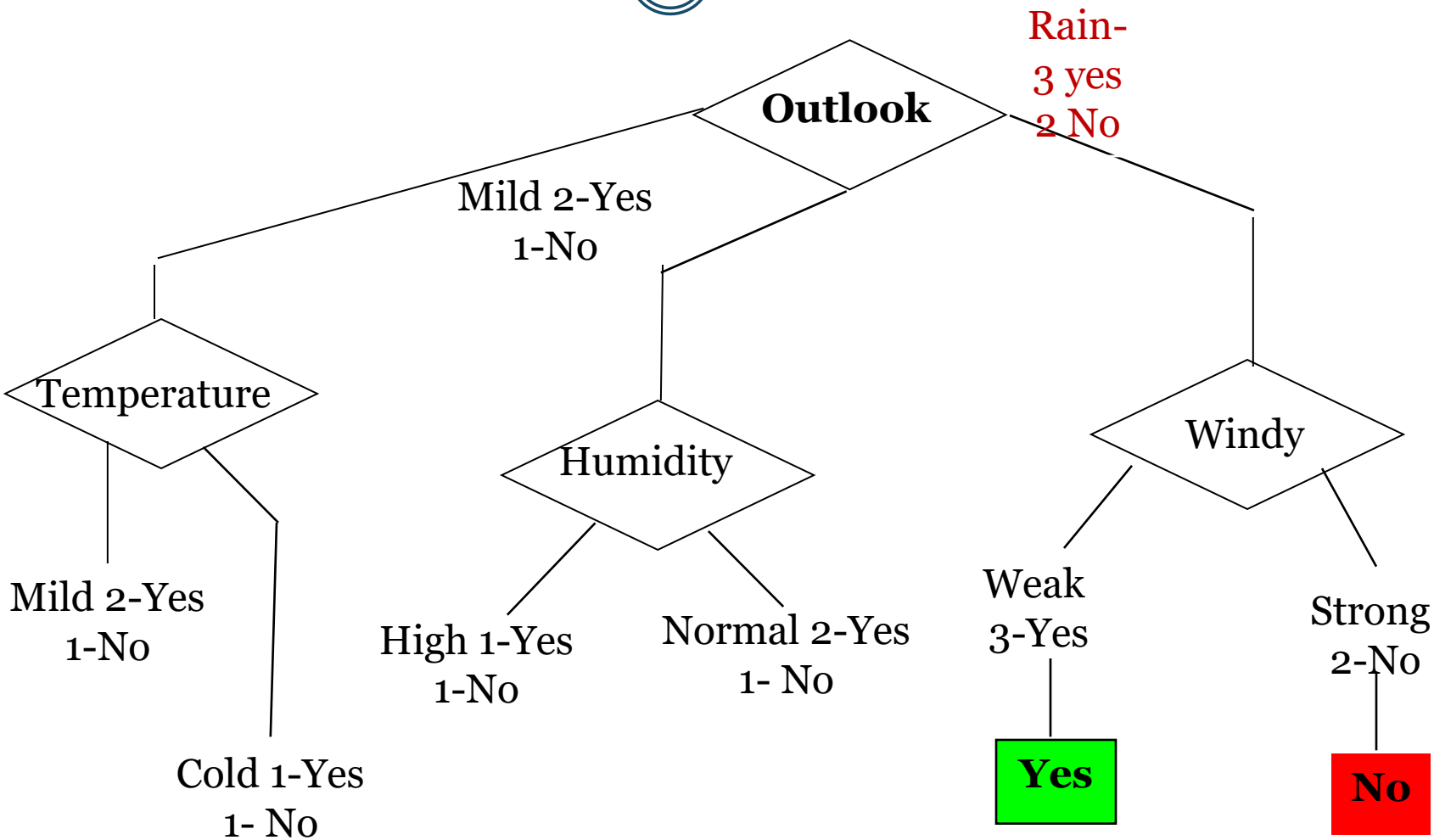
Entropy(Strong) = -(0/2) Log_2 (0/2) - (2/2) Log_2 (2/2)= 0

A branch set with entropy of 0 is a leaf node.

Entropy (Windy)= (3/5)*Entropy(weak)+(2/5)*Entropy(Strong) =0

Information Gain(s, Humidity)
= Entropy(Rain)- Entropy (Windy) = 0.97095-0
=Information Gain(s, Temperature) =0.97095

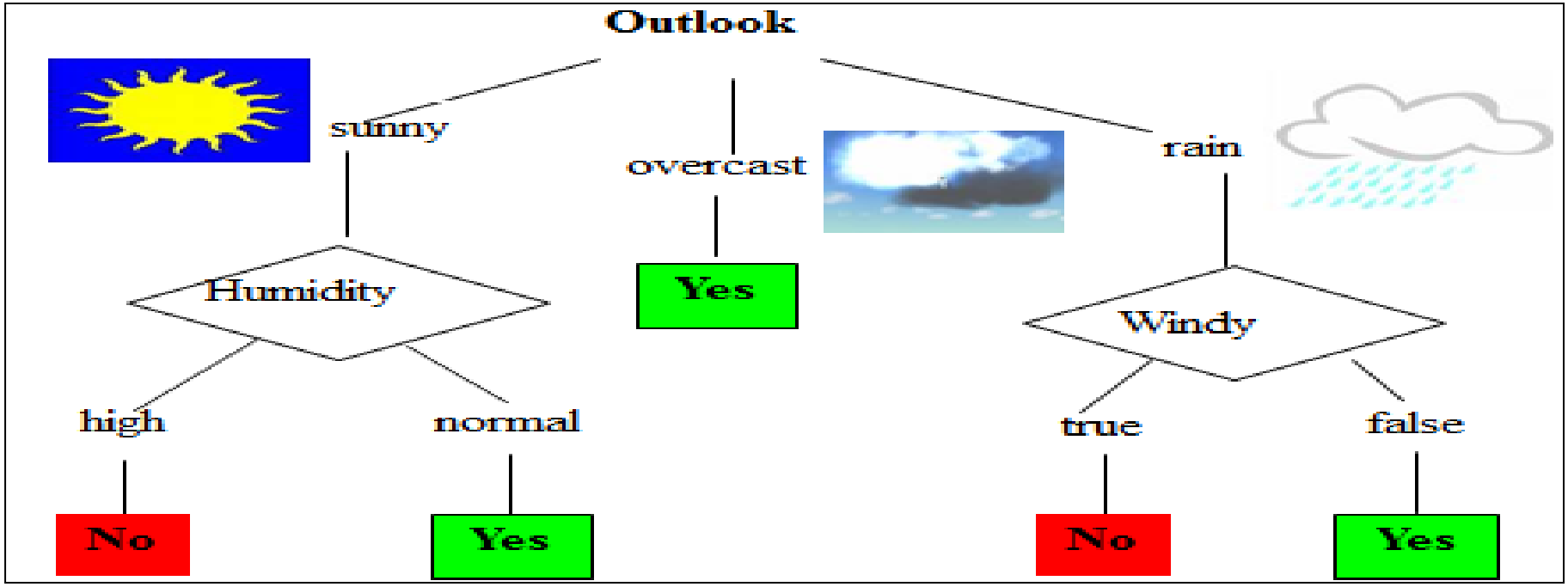
Information Gain(s, Windy) =0.97095
Information Gain(s, Temperature) =0.0199
Information Gain(s, Humidity) =0.0199



Final Decision Tree:

The Decision tree can also be expresses in rule format:

- If outlook= Sunny And Humidity= High Then Play ball= No
- If outlook= Sunny And Humidity= Normal Then Play ball=Yes
- If outlook= Overcast Then Play ball= Yes
- If outlook= Rain And Windy= Strong Then Play ball= No
- If outlook= Rain And Windy= Weak Then Play ball= No



ID3 Algorithm:

ID3 (Examples, Target_Attribute, Attributes)

Examples are the training examples. Target_Attribute is the attribute whose value is to be predicted by the tree. Attributes is the list of attributes which may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a root node for the tree
- IF all examples are positive, Return the single-node tree Root, with label = +
- If all examples are negative, Return the single-node tree Root, with label = -
- If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples
- Otherwise Begin
    - A ← The Attribute that best classifies examples
    - Decision Tree attribute for Root ← A
    - For each positive value, vi, of A,
        - Add a new tree branch below Root, corresponding to the test A = vi
        - Let Examples(vi), be the subset of examples that have the value vi for A
        - If Examples(vi) is empty
            - Then below this new branch add a leaf node with label = most common target value in the examples
            - Else below this new branch add the subtree ID3 (Examples(vi), Target_Attribute, Attributes – {A})
- End
- Return Root

Advantages of using ID3

1. Understandable prediction rules are created from the training data.
2. Builds the fastest tree.
3. Builds a short tree.
4. Only need to test enough attributes until all data is classified.
5. Finding leaf nodes enables test data to be pruned, reducing number of tests.
6. Whole dataset is searched to create tree.

Disadvantages of using ID3

1. Data may be over-fitted or over-classified, if a small sample is tested.
2. Only one attribute at a time is tested for making a decision.
3. Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

1. Numerical on ID3- **MAY 2013, May 16, Dec 17, May 18**

A simple example from the stock market involving only discrete ranges has Profit as categorical attribute, with values {up, down}. and the training data is:

AGE	COMPETITION	TYPE	PROFIT
Old	Yes	software	Down
Old	No	software	Down
Old	No	hardware	Down
Mid	Yes	software	Down
Mid	Yes	hardware	Down
Mid	No	hardware	Up
Mid	No	software	Up
New	Yes	software	Up
New	No	hardware	Up
New	No	software	Up

Apply the decision tree algorithm and show the generated rules.

Dec 17

Define classification. Discuss the issues in classification. A simple example from the stock market involving only discrete ranges has profit as categorical attribute, with values { Up, Down and the training data is

AGE	COMPETITION	TYPE	PROFIT
Old	Yes	software	Down
Old	No	software	Down
Old	No	hardware	Down
Mid	Yes	software	Down
Mid	Yes	hardware	Down
Mid	No	hardware	Up
Mid	No	software	Up
New	Yes	software	Up
New	No	hardware	Up
New	No	software	Up

Stock Market Example.

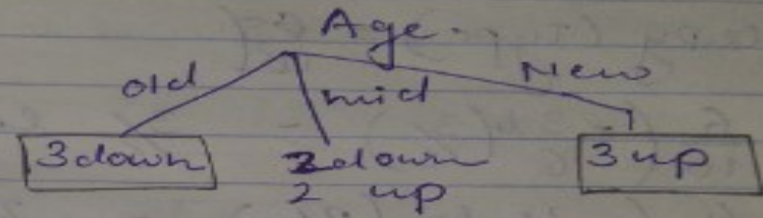
Entropy (Age|Mid)

$$= \left(\frac{4}{10}\right)\left[\left(-\frac{2}{4}\right)\ln\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right)\ln\left(\frac{2}{4}\right)\right]$$

$$= 0.4\left[-(0.5)\ln(0.5) - (0.5)\ln(0.5)\right]$$

$$= 0.4\left[-0.5 \times -1 - 0.5 \times -1\right]$$

$$= 0.4\left[0.5 + 0.5\right] = 0.4.$$



decision Rule :-

i) if Age is old profit is down.

ii) if Age is new profit is up.

IG = 1 − Entropy = 1 − 0.4 = 0.6

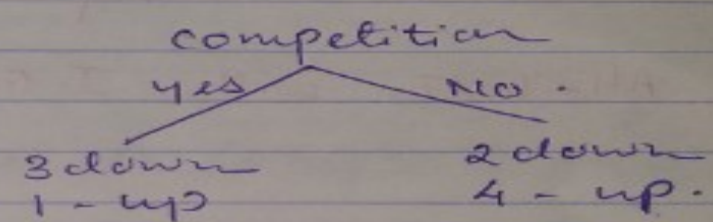
Entropy (competition).

$$= \frac{4}{10}\left[\frac{-1}{4} \times \ln\left(\frac{1}{4}\right) - \left(\frac{3}{4}\right) \times \ln\left(\frac{3}{4}\right)\right]$$

$$+ \frac{6}{10}\left[\left(\frac{4}{6}\right) \times \ln\left(\frac{4}{6}\right) - \left(\frac{2}{6}\right) \times \ln\left(\frac{2}{6}\right)\right]$$

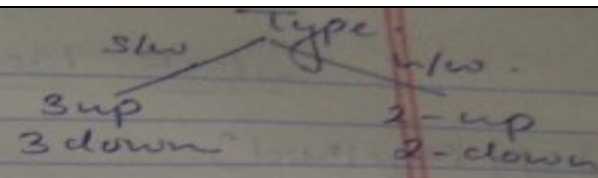
$$= \frac{4}{10}\left[1 + 0.311\right] + \frac{6}{10}\left[0.33 + 0.389\right]$$

$$= \frac{4}{10}(0.5245) + \frac{6}{10}(0.4316) = 0.956$$



IG = 1 − Entropy = 1 − 0.956 = 0.044

Entropy (Type)



Entropy (Type) = $\frac{6}{10}\Big($

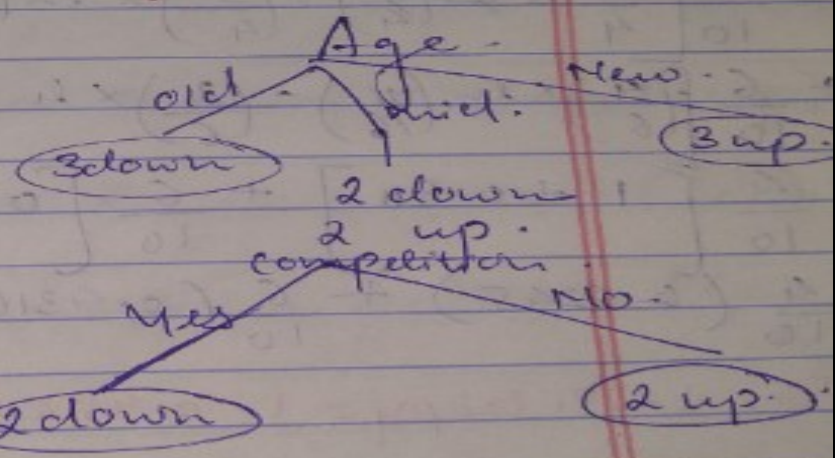
$$= \frac{6}{10}\left(-\frac{3}{6}\ln\left(\frac{3}{6}\right) - \frac{3}{6}\ln\left(\frac{3}{6}\right)\right)$$

$$+ \frac{4}{10}\left(-\frac{2}{4}\ln\left(\frac{2}{4}\right) - \frac{2}{4}\ln\left(\frac{2}{4}\right)\right)$$

$$= \frac{6}{10}\left[1\right] + \frac{4}{10}\left[1\right] = 0.6 + 0.4 = 1$$

I.G = 1 − entropy = 1 − 1 = 0

∴ Attribute with I.G = 0.4 Age is selected —



New Rules :—

3) if Age is mid and competition is yes then profit is down.

4) if Age is mid and competition is No then profit is up.

Example 2: on ID3- Dec 2015

Q4. (a) Construct a decision tree for the following set of samples. Write any two decision rules obtained from the tree. Classify a new sample with (gender = " Female", height = " 1.92m") [6+2+2]

Person ID	Gender	Height	Class
1	Female	1.6m	Short
2	Male	2m	Tall
3	Female	1.9m	Medium
4	Female	2.1m	Tall
5	Female	1.7m	Short
6	Male	1.85m	Medium
7	Female	1.6m	Short
8	Male	1.9m	Short
9	Male	2.2m	Tall

Height Class Example.

$$\text{Entropy}(S) = \left(\frac{4}{9}\right)\left(\log_2(4/9)\right) - \left(\frac{3}{9}\right)\left(\log_2(3/9)\right) - \left(\frac{2}{9}\right)\left(\log_2(2/9)\right)$$

$$= 0.5199 + 0.5283 + 0.482$$

$$= \underline{\underline{1.530}}.$$

① Entropy of gender =

gender

5 female

3 Short
1 - Med
1 - Tall

4 male

1 - Short
1 - Med
2 - Tall.

$$\frac{5}{9}\left[\left(-\tfrac{3}{5}\right)\log_2\left(\tfrac{3}{5}\right) - \left(\tfrac{1}{5}\right)\log_2\left(\tfrac{1}{5}\right) - \left(\tfrac{1}{5}\right)\log_2\left(\tfrac{1}{5}\right)\right] +$$

$$\frac{4}{9}\left[\left(-\tfrac{1}{4}\right)\cdot\log_2\left(\tfrac{1}{4}\right) - \left(\tfrac{1}{4}\right)\log_2\left(\tfrac{1}{4}\right) - \left(\tfrac{2}{4}\right)\log_2\left(\tfrac{2}{4}\right)\right].$$

$$= \frac{5}{9}\left[0.442 + 0.464 + 0.464\right] + \frac{4}{9}\left[0.5 + 0.5 + 0.5\right].$$

$$= 0.761 + 0.666 = 1.427$$

$$\boxed{\text{Information gain} = 1.530 - 1.427 = \underline{\underline{0.1034}}}$$

(2) Entropy of (Height) =

Height

- less than = 1.7 — 4 Short
- ≥ 1.7 & ≤ 1.85 — 1 Medium
- ≥ 1.85 to < 2 — 1 Medium, 1 Tall
- ≥ = 2 — 2 Tall

Entropy of height $= \frac{2}{9}\left[\left(\frac{1}{2}\right)\log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right)\log_2\left(\frac{1}{2}\right)\right]$.

$= \frac{2}{9}\left[0.5 + 0.5\right] = \frac{2}{9}(1) = 0.22$.

$\boxed{\text{Information gain} = 1.530 - 0.22 = \underline{\underline{1.307}}}$

So <u>height</u> <u>is selected</u> as first attribute :-

Height

- less than = 1.7 — 4 Short
- ≥ 1.7 & ≤ 1.85 — 1 Medium
- ≥ 1.85 to < 2 — 1 Medium, 1 Tall
  - Gender
    - F — 1 Medium
    - M — 1 Tall.
- ≥ = 2 — 2 Tall

Based on this Tree following are the decision rules:-

1. If height <= 1.7 then class is short.
2. If height > 1.7 and height <= 1.85 then class is medium.
3. If height > 1.85 and height < 2 and gender = F then class is medium.
4. If height > 1.85 and height < 2 and gender = M then class is tall.
5. If height >= 2 then class is tall.

for new sample -> (gender = Female and height = 1.92) lies into the class medium.

Example 2 Solution: Decision Rules

- **Decision Rules:**
  1. If Height<=1.7 then Class(short)
  2. If Height >1.7 & height<=1.85 then Class(Medium)
  3. If Height >1.85 and height <2 and Gender=F then Class(Medium)
  4. If Height >1.85 and height <2 and Gender=M then Class(Tall)
  5. If Height >=2 then Class(Tall)

- New Sample is Gender=F and Height= 1.92
  - As per the decision rules defined this sample will fall under the class=Medium
  - Sample<Gender=f, Height=1.92, Class=Medium>

3. Numerical on ID3- **MAY 2011**

Using the table given below, create a classification model using decision true technique. Indicate how to utilize the model to estimate the risk category of the customer with (**Credit-History** – bad, **Debt** – high, **Collateral** – none, **Income** – (15-35k)).

Sr. No.	Debt	Collateral	Income	Credit - History	Risk
1	high	none	0-15 k	bad	high risk
2	high	none	15-35 k	unknown	high risk
3	low	none	15-35 k	unknown	Moderate risk
4	low	none	0-15 k	unknown	high risk
5	low	none	over 35 k	unknown	low risk
6	low	adequate	over 35 k	unknown	low risk
7	low	none	0-15 k	bad	high risk
8	low	adequate	over 35 k	bad	Moderate risk
9	low	none	over 35 k	good	low risk
10	high	adequate	over 35 k	good	low risk
11	high	none	0-15 k	good	high risk
12	high	none	15-35 k	good	Moderate risk.

MU asked Questions

May 2019

a) The following table consists of training data from an employee database. The data have been generalized. For example, "31: : : 35" for *age* represents the age range of 31 to 35. For a given row entry, *count* represents the number of data tuples having the values for *department*, *status*, *age*, and *salary* given in that row. **[10]**

department	*status*	*age*	*salary*	*count*
sales	senior	31 ... 35	46K ... 50K	30
sales	junior	26 ... 30	26K ... 30K	40
sales	junior	31 ... 35	31K ... 35K	40
systems	junior	21 ... 25	46K ... 50K	20
systems	senior	31 ... 35	66K ... 70K	5
systems	junior	26 ... 30	46K ... 50K	3
systems	senior	41 ... 45	66K ... 70K	3
marketing	senior	36 ... 40	46K ... 50K	10
marketing	junior	31 ... 35	41K ... 45K	4
secretary	senior	46 ... 50	36K ... 40K	4
secretary	junior	26 ... 30	26K ... 30K	6

Let *status* be the class label attribute.

i. How would you modify the basic decision tree algorithm to take into consideration the *count* of each generalized data tuple (i.e., of each row entry)?

ii. Use your algorithm to construct a decision tree from the given data.

MU asked Questions

Dec 2019

a) Why is tree pruning useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning? [10]

May 2019

b) Why is *tree pruning* useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning? Given a decision tree, you have the option of (i) *converting* the decision tree to rules and then pruning the resulting rules, or (ii) *pruning* the decision tree and then converting the pruned tree to rules. What advantage does (i) have over (ii)? **[10]**

Problems of Decision tree

- Decision trees are made to classify the item set.
- While classifying we meet with 2 problems

1. Underfitting .
2. Overfitting .

Problems of Decision tree

- Underfitting problem arises when both the "training errors and test errors are large"
- This happens when the developed model is made very simple.
- Overfitting problem arises when "training errors are small but test errors are large"

Problems of Decision tree- Over fitting

- Overfitting results in decision trees that are more complex than necessary.
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records.
- Need new ways for estimating errors.

How to address Overfitting??---------- Tree Pruning is the solution

Pruning and its type

- The process of adjusting Decision Tree to minimize "misclassification error" is called pruning .
- Pruning can be done in 2 ways

  1. Prepruning.
  2. Postpruning.

1. Pre Pruning

- Prepruning is the halting of subtree construction at some node after checking some measures.
- These measures can be Information gain, Gini index,etc.
- If partitioning the tuple at a node would result in a split that falls below a prespecified threshold, then pruning is done.
- Early stopping- Pre-pruning may stop the growth process prematurely.

2. Post Pruning

- Grow decision tree to its entirety.
- Trim the nodes of the decision tree in a bottom-up fashion.Postpruning is done by replacing the node with leaf.
- If error improves after trimming, replace sub-tree by a leaf node.

Reduce error pruning

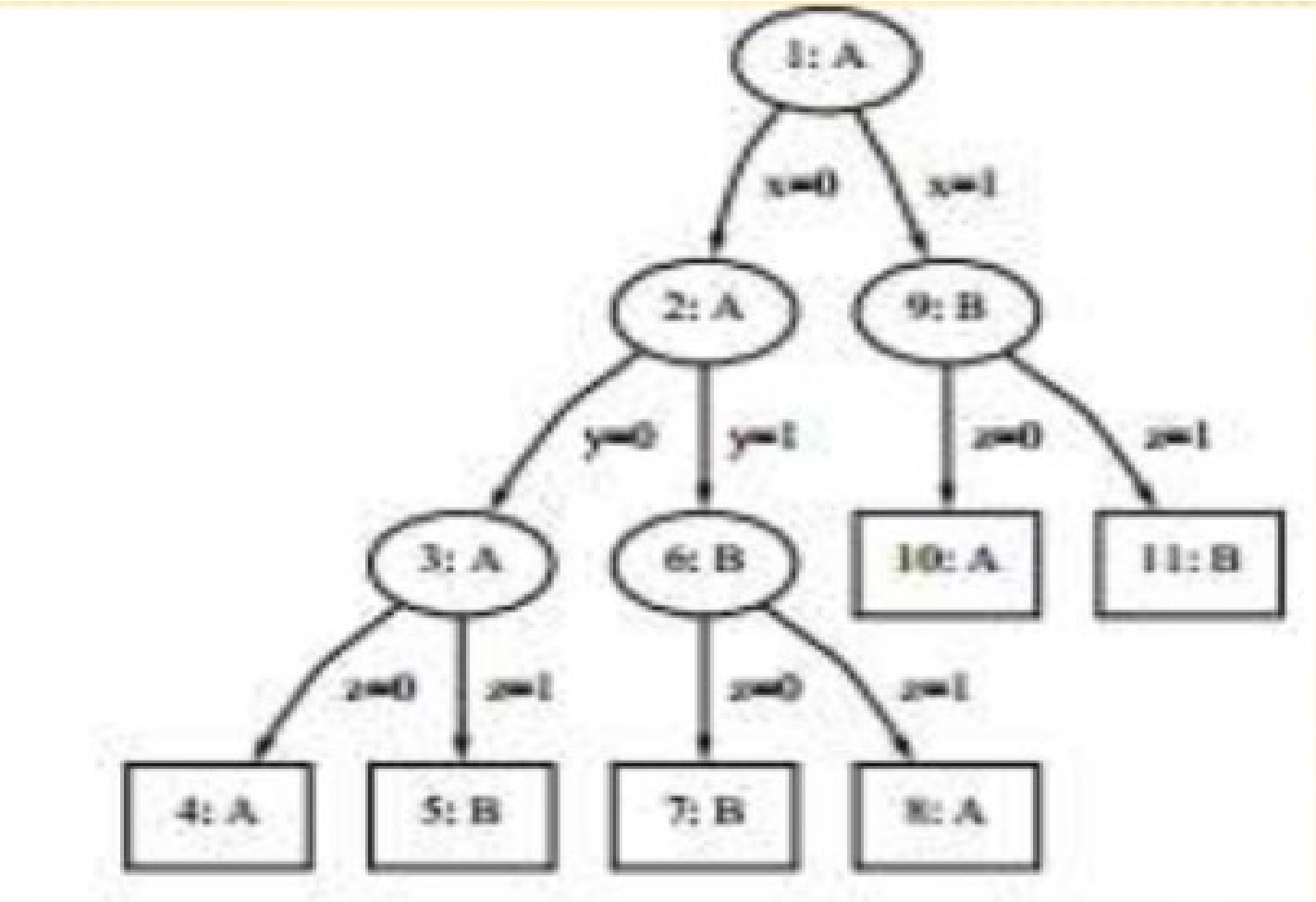
- The idea is to hold out some of the available instances—the "pruning set" after the tree is built.
- Prune the tree until the classification error on these independent instances starts to increase.
- These pruning set are not used for building the decision tree, they provide a less biased estimate of its error rate on future instances than the training data.
- Reduced error pruning is done in bottom up fashion.
- Criteria:

If error of parent is lesser than its child then prune the tree else not .

i.e if Parent (error)< Child(error) then "Prune"

else don't Prune

Example



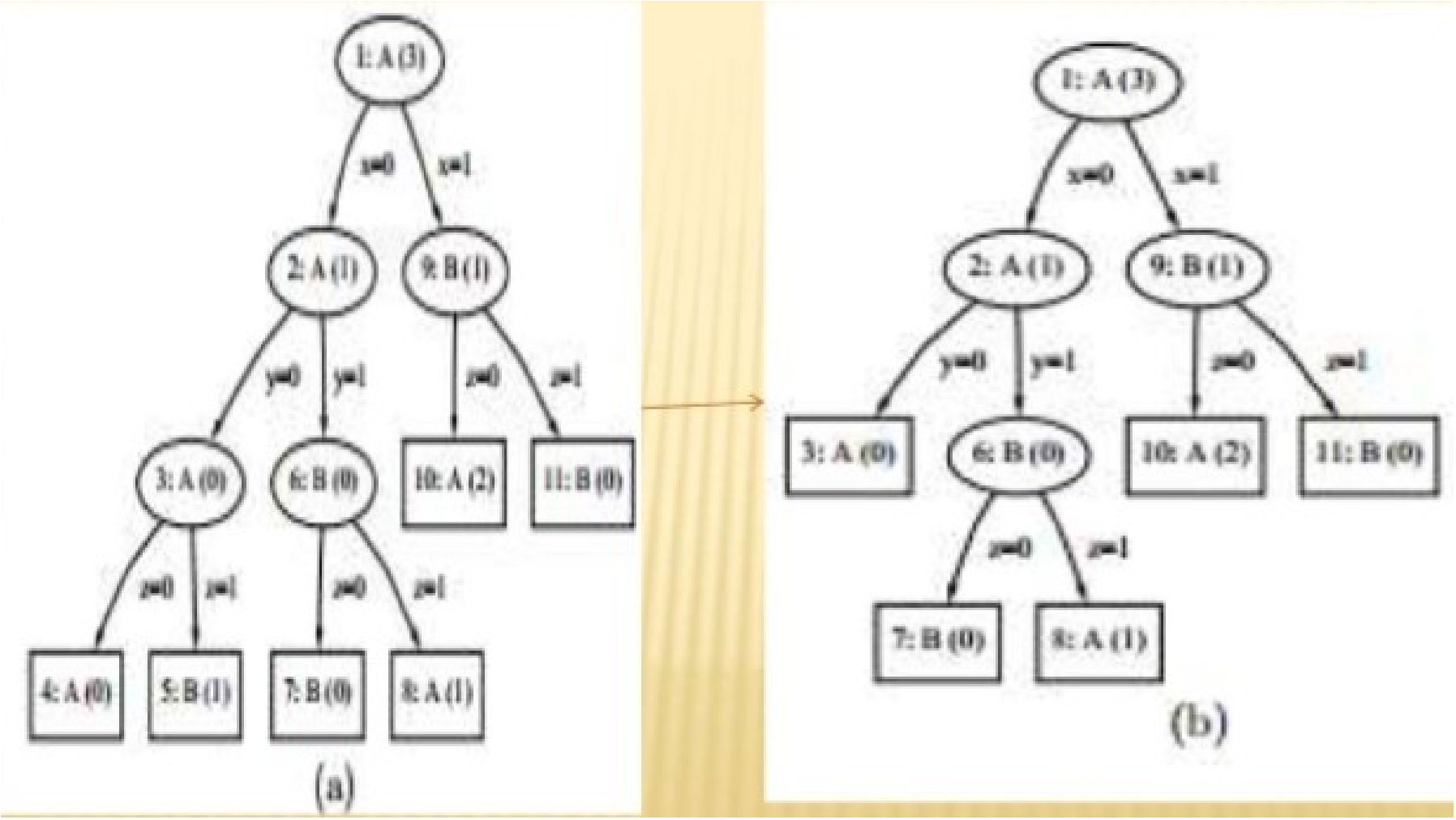
Steps to prune

- In each tree, the number of instances in the pruning data that are misclassified by the individual nodes are given in parentheses.
- Assuming that the tree is traversed left-to-right.
- The pruning procedure first considers for removal the subtree attached to node 3.
- Because the subtree's error on the pruning data (1 error) exceeds the error of node 3 itself (0errors), node 3 is converted to a leaf.
- Next, node 6 is replaced by a leaf for the same reason

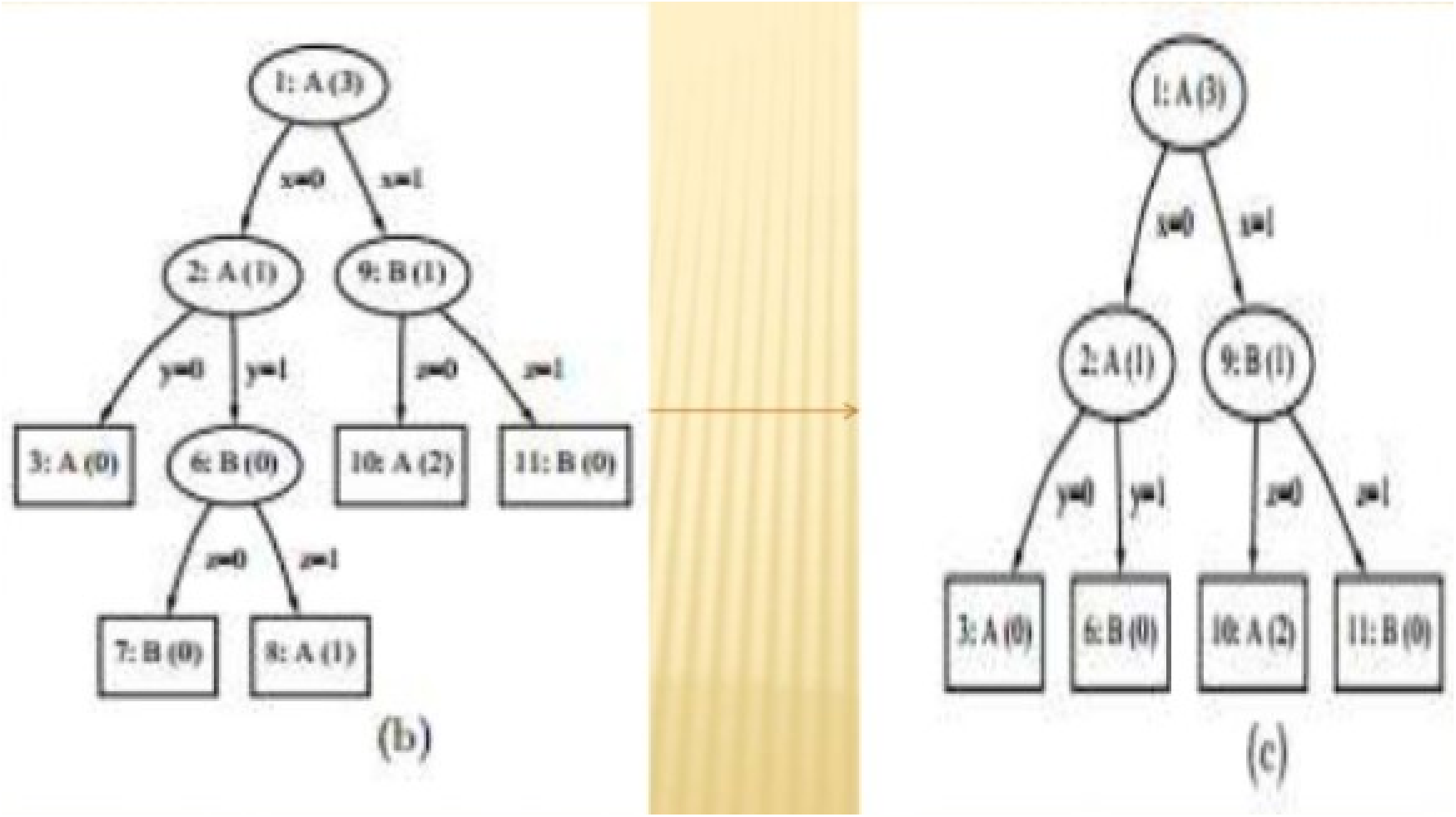
Steps to prune

- Having processed both of its successors, the pruning procedure then considers node 2 for deletion. However, because the subtree attached to node 2 makes fewer mistakes (0 errors) than node 2 itself (1 error), the subtree remains in place.
- Next, the subtree extending from node 9 is considered for pruning, resulting in a leaf
- In the last step, node 1 is considered for pruning, leaving the tree unchanged.

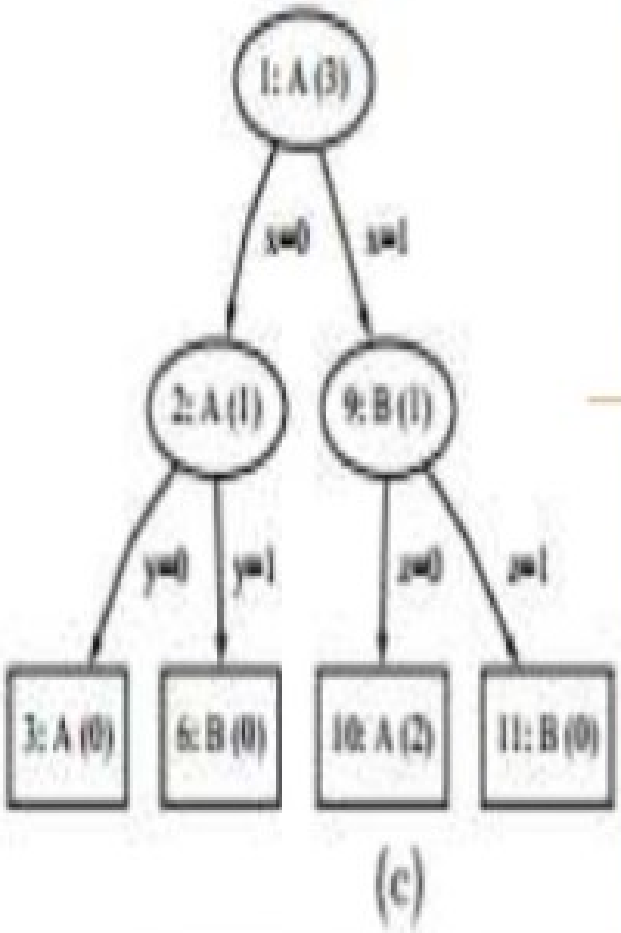
Example


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
z=0
z=1
z=0
z=1
4: A (0)
5: B (1)
7: B (0)
8: A (1)
(a)
1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
z=0
z=1
7: B (0)
8: A (1)
(b)

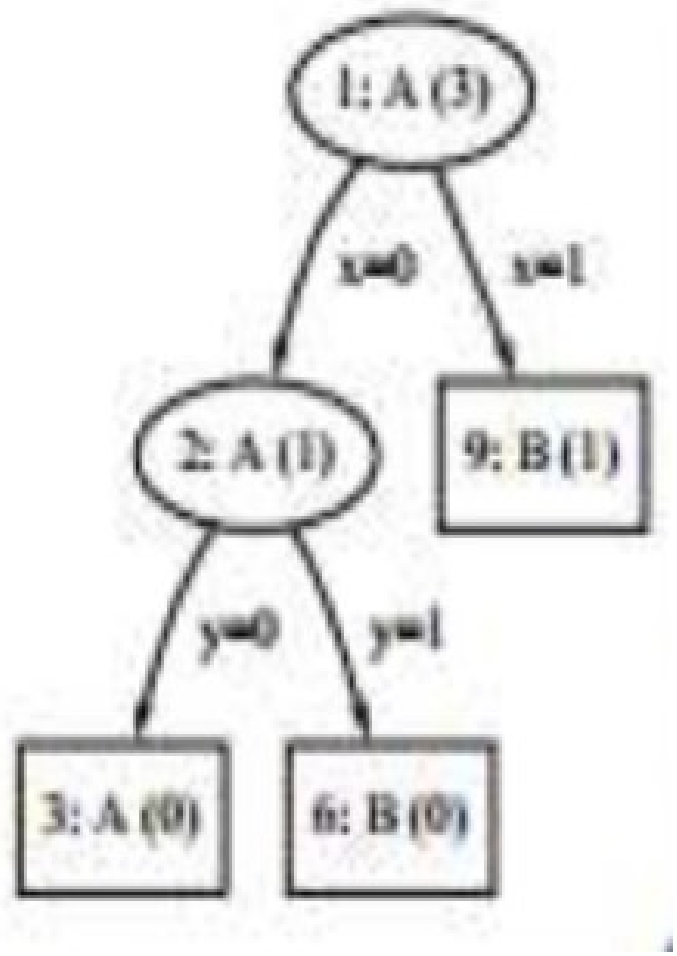
Example


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
z=0
z=1
7: B (0)
8: A (1)
(b)
1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
(c)

Example


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
y=0
y=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)

(c)


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
3: A (0)
6: B (0)

Pre pruning vs Post pruning

- Prepruning is faster than post pruning since it don't need to wait for complete construction of decision tree.

- But still Post-pruning is preferable to pre-pruning because of "interaction effect".
- These are the efects which arise after interaction of several attributes.
- Prepruning suppresses growth by evaluating each attribute individually, and so might overlook effects that are due to the interaction of several attributes and stop too early. Post-pruning, on the other hand, avoids this problem because interaction effects are visible in the fully grown tree.

Issues in Classification (May 10, Dec 17)

- **Accuracy**
  - Classifier accuracy: predicting class label
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values in training Set
- **Scalability**: Efficiency in disk-resident databases
- **Interpretability**
  - understanding and insight provided by the model
- Other measures, e.g., **goodness of rules**, such as decision tree size or compactness of classification rules

Classification v/s Prediction(May 2010)

Sr. no	Classification	Prediction
1.	Classification is a major type of prediction problem where classification is used to predict discrete or nominal values	Prediction can be viewed as the construction and use of a model to assess the class of an unlabeled sample.
2.	Classification is the use of prediction to predict class labels	It is used to assess the values or value ranges of an attribute that a given sample is likely to have.
3.	*E.g. Group patients based on their known medical data and treatment outcome*	*E.g. If a Classification model is used to predict the treatment outcome for a new patient, then it would be a prediction.*

Regression Model

Define linear, non linear multiple regressions. Plan regression model for disease Development with respect to change in weather parameters. Dec 16, 10 marks

Regression model

- Relation between variables where changes in some variables may "explain" or possibly "cause" changes in other variables.
- **Explanatory variables** are termed the **independent** variables and the **variables to be explained** are termed the **dependent** variables.
- *Regression model estimates the nature of the relationship between the independent and dependent variables.*
  - Change in dependent variables that results from changes in independent variables, i.e.. size of the relationship.
  - Strength of the relationship.
  - Statistical significance of the relationship.

Examples

- Price of a product and quantity produced or sold:
  - **Quantity sold affected by price**. Dependent variable is quantity of product sold – independent variable is price.
  - **Price affected by quantity offered for sale.** Dependent variable is price – independent variable is quantity sold.
- Dependent variable is retail price of gasoline in Regina – independent variable is the price of crude oil.
- Dependent variable is employment income – independent variables might be hours of work, education, occupation, sex, age, region, years of experience, unionization status, etc.

Bivariate or simple linear regression

- x is the independent variable
- y is the dependent variable
- The regression model is

$$y = \beta_0 + \beta_1 x + \varepsilon$$

- The model has two variables, the independent or explanatory variable x, and the dependent variable y, the variable whose variation is to be explained.
- The relationship between x and y is a linear or straight line relationship.
- Two parameters to estimate – the slope of the line β_1 and the y-intercept β_0 (where the line crosses the vertical axis).
- ε is the unexplained, random, or error component.

Types of Regression Model

- Linear regression attempts to find the mathematical relationship between variables.
- If outcome is straight line then it is considered as **linear model** and if it is curved line, then it is a **non linear model**.
- **Different types of Regression Model are:**

1. A *simple regression* model: Consider a model with only one independent variable.
   - $Y=\beta_0 + \beta_1 X$
2. A *multiple regression* model: a model with multiple independent variables.
   - $Y=\beta_0 + \beta_1 X + \beta_2 Y + \beta_3 Z$

Example Bivariate and multivariate models

Bivariate or simple regression model

(Education) $x \longrightarrow y$ (Income)

Multivariate or multiple regression model

(Education) x_1

(Sex) x_2

(Experience) x_3

(Age) x_4

y (Income)

Model with simultaneous relationship

Price of wheat $\longleftrightarrow$ Quantity of wheat produced

Example Bivariate and multivariate models

Example:

The income and education of a person are related. It is expected that, on an average, higher level of education provides higher income. So a simple linear regression model can be expressed as

$$\text{income} = \beta_0 + \beta_1 \text{ education} + \varepsilon .$$

Not that β_1 reflects the change is income with respect to per unit change is education and β_0 reflects the income when education is zero as it is expected that even an illiterate person can also have some income.

Further this model neglects that most people have higher income when they are older than when they are young, regardless of education. So β_1 will over-state the marginal impact of education. If age and education are positively correlated, then the regression model will associate all the observed increase in income with an increase in education. So better model is

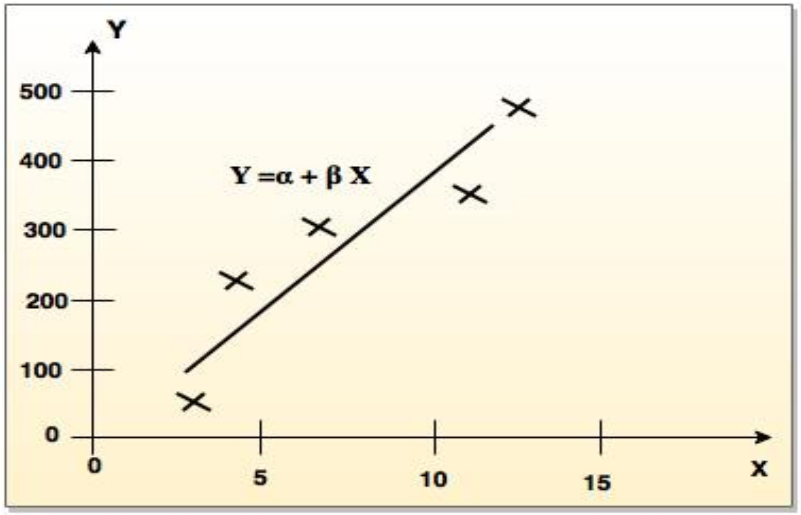
$$\text{income} = \beta_0 + \beta_1 \text{ education} + \beta_2 \text{ age} + \varepsilon .$$

Linear Regression Model

1. Linear regression

- It is simplest form of regression. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observe the data.
- Linear regression attempts to find the mathematical relationship between variables.
- If outcome is straight line then it is considered as linear model and if it is curved line, then it is a non linear model.
- The relationship between dependent variable is given by straight line and it has only one independent variable.

  Y = α + B X
- Model **'Y'**, is a linear function of **'X'**.
- The value of 'Y' increases or decreases in linear manner according to which the value of 'X' also changes.



Linear Regression

Linear Regression Example

SUBJECT	AGE X	GLUCOSE LEVEL Y
1	43	99
2	21	65
3	25	79
4	42	75
5	57	87
6	59	81

Linear Regression Example

SUBJECT	AGE X	GLUCOSE LEVEL Y	XY	X^2	Y^2
1	43	99	4257	1849	9801
2	21	65	1365	441	4225
3	25	79	1975	625	6241
4	42	75	3150	1764	5625
5	57	87	4959	3249	7569
6	59	81	4779	3481	6561
Σ	247	486	20485	11409	40022

Step 1: Find linear regression equation Y=a +b x (a:intercept, b:Slope)

From the above table, Σx = 247, Σy = 486, Σxy = 20485, Σx2 = 11409, Σy2 = 40022. n is the sample size (6, in our case).

Use the following equations to find a and b.

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Find a:

((486 × 11,409) – ((247 × 20,485)) / 6 (11,409) – 247^2)
484979 / 7445
=**65.14**

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Find b:

(6(20,485) – (247 × 486)) / (6 (11409) – 247^2)
(122,910 – 120,042) / 68,454 – 247^2
2,868 / 7,445
= **.385225**

a = **65.1416**
b = **.385225**

Step 2: Replace values of a and b in equation Y=a +b x (a:intercept, b:Slope)

a = **65.1416**

b = **.385225**

Insert the values into the equation.

$y' = a + bx$

$\mathbf{y' = 65.14 + .385225x}$

MU Asked Questions

Dec 2019

b) Develop a model to predict the salary of college graduates with 10 years of work experience using linear regression. [10]

Years of experience (x)	Salary in $100 (y)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Multiple Regression Model

2. Multiple regression model

- Multiple linear regression is an extension of linear regression analysis.
- It uses two or more independent variables to predict an outcome and a single continuous dependent variable.

$$Y = a_0 + a_1 X_1 + a_2 X_2 + \ldots\ldots\ldots + a_k X_k + e$$

where,

'Y' is the response variable.

$X_1 + X_2 + X_k$ are the independent predictors.

'e' is random error.

a_0, a_1, a_2, a_k are the regression coefficients.

Plan regression model for disease Development with respect to change in weather parameters? Dec 2016

- Considering disease (D) dependent on 3 variables humidity (x), Wind (y) and Temperature (z).
  - Humidity (high, Normal)
  - Wind(Weak, Strong)
  - Temp (Hot, Mild, Cold)
    - Following is a multivariate regression model as $D=\beta_0 + \beta_1 X + \beta_2 Y + \beta_3 Z$

Classifier Accuracy

1. **How Is The Classifier Accuracy Determined And Explains Its Various Types? (May 2012)**

Classifier Accuracy

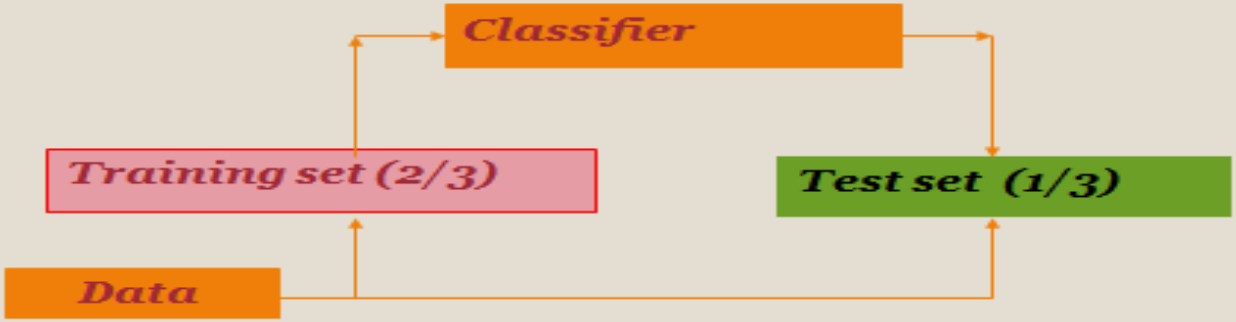
How Is The Classifier Accuracy Determined And Explains Its Various Types? (May 2012)

- ***Accuracy of a classifier M**, acc(M) is the percentage of test set tuples that are correctly classified by the model M.*
- **Basic Concepts:**

1. **Success:** Instance(record) class is predicted correctly.
2. **Error:** Instance class for a record is predicted incorrectly.
3. **Error Rate:** A percentage of errors made over the whole set of instances(records=5) used for testing. E.g. 1/5 = 20 %
4. **Predictive Accuracy:** A percentage of well classified data in the testing data set. E.g. 4/5= 80%
5. **Test Set:** A set of instances that have not been used in the training process.

Estimating Classifier Accuracy: 1. Holdout

- **Holdout:** Reserve a certain amount of data for testing and use the remainder for training (usually 1/3 for testing, 2/3 for training)



  - **Problem:** The Samples might not be representative. For example, some classes might be represented with very few instances or even with no instances at all.
  - **Solution: Stratification**-Sampling for training and testing within classes. This ensures that each class is represented with approximately equal proportions in both subsets. Its an **Advanced version of balancing the data.**

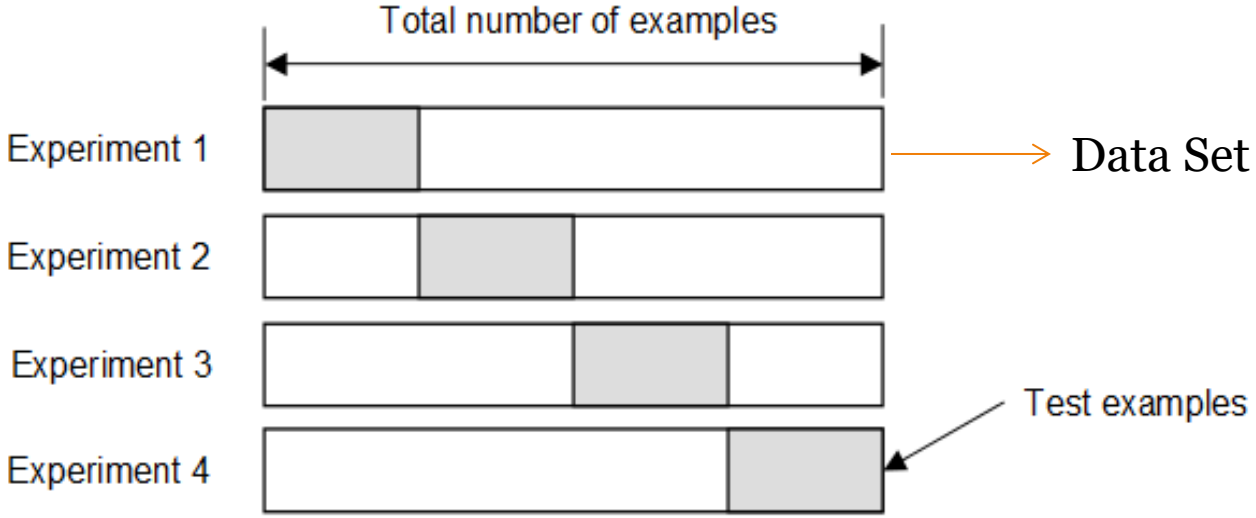
The hold-out method is usually used when we have thousands of instances, including several hundred instances from each class.

Estimating Classifier Accuracy: 2. Repeated Holdout Method (Random Sampling)

- Holdout estimate can be made more reliable by repeating the process with different subsamples.
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification).
  - The error rates on the different iterations are averaged to yield an overall error rate.
- This is called the *Repeated holdout method.*
- **Still not optimum**: The different test sets overlap, but we would like all our instances from the data to be tested at least ones.
- Can we prevent overlapping?

Estimating Classifier Accuracy: 3. *k*-Fold Cross-Validation

- **Create a K-fold partition of the the dataset**
  - For each of K experiments, use K-1 folds for training and the remaining one for testing



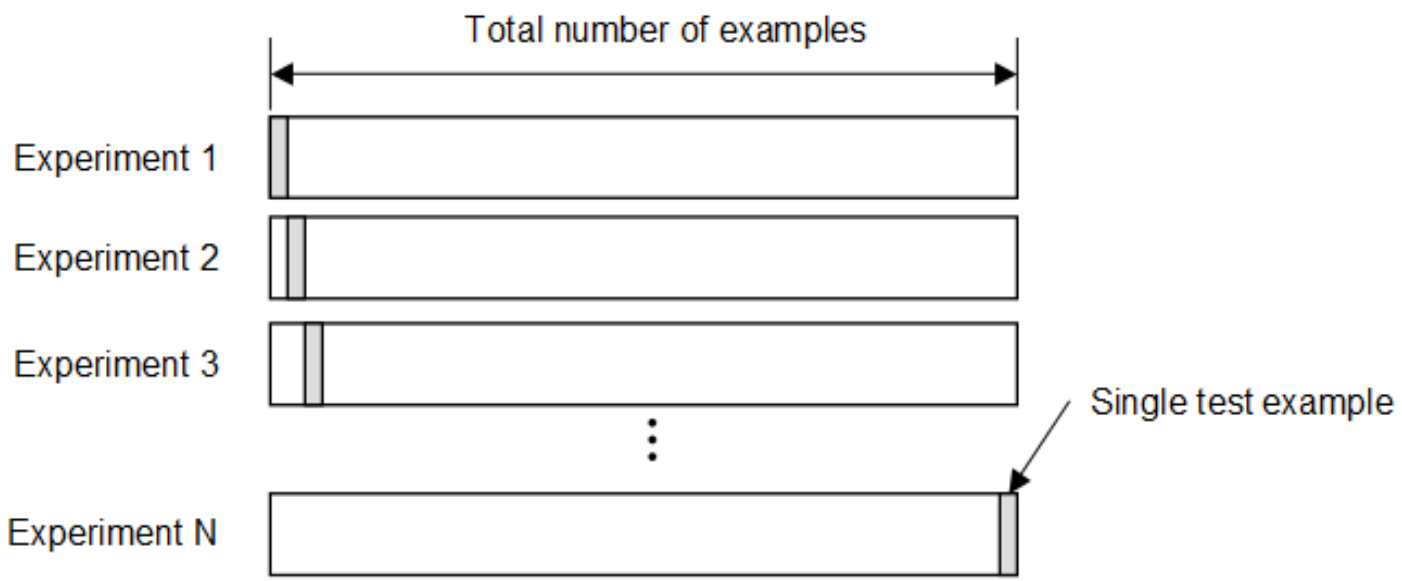
- **K-Fold Cross validation is similar to Random Subsampling**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing

More on Cross-Validation

- **Standard method for evaluation:** *Stratified 10-fold cross-validation.*
- **Why 10?** Extensive experiments have shown that this is the best choice to get an accurate estimate.
- Stratification reduces the estimate's variance.
- **Even better:** *Repeated stratified cross-validation:*
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (Reduces the variance/ Difference).

Estimating Classifier Accuracy: 4. Leave-One-Out Cross-Validation

- **Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples**
  - For a dataset with N examples, perform N experiments
  - For each experiment use N-1 examples for training and the remaining example for testing



Estimating Classifier Accuracy: 5. Bootstrap Method

- **Cross validation uses sampling *without replacement:***
  - The same instance, once selected, can not be selected again for a particular training/test set
- **The bootstrap uses sampling with replacement to form the training set:**
  - **Training Set:** A dataset of *n* instances is sampled *with replacement* n times to form a training set of *n* instances;
  - **Test Set:** Use the instances from the original dataset that don't occur in the new training set for testing.
- Bootstrapping is the best error estimation for small datasets.

MUMBAI UNIVERSITY EXAM QUESTIONS

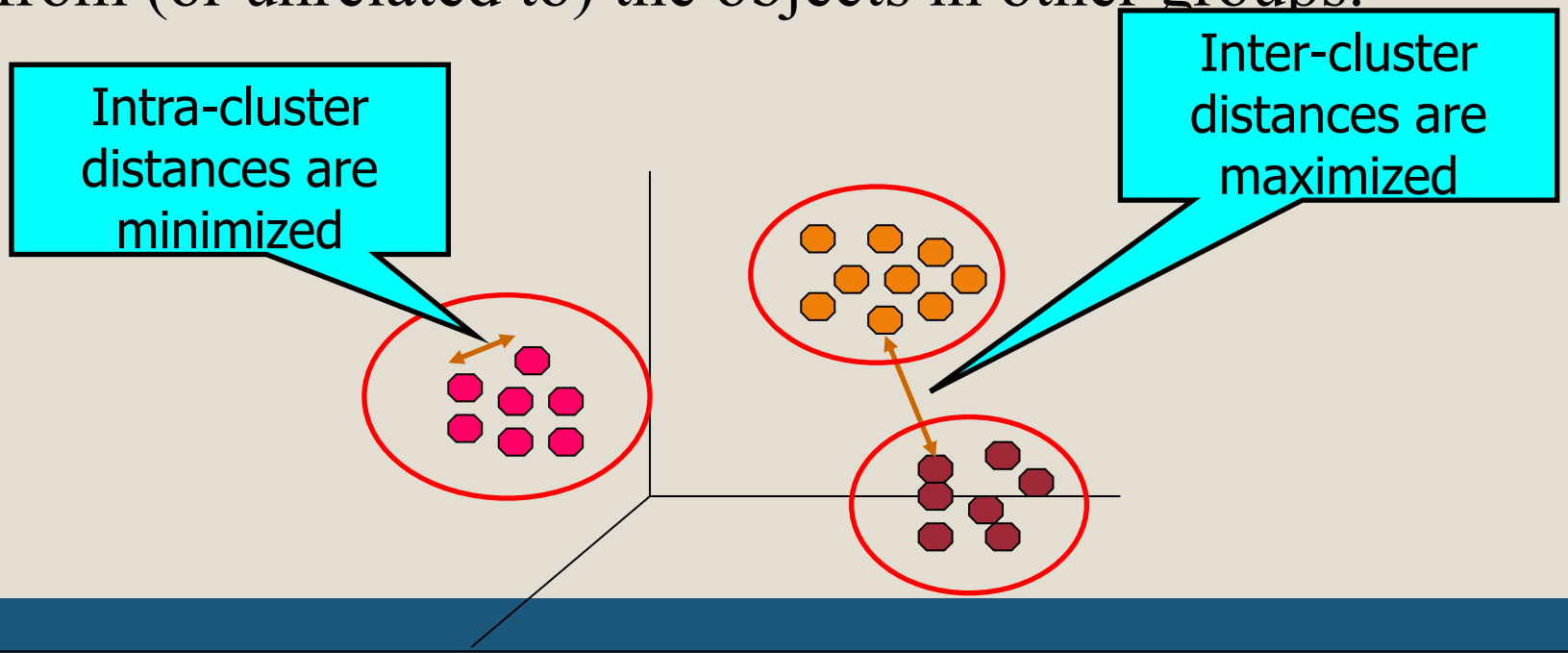
Clustering

1.Explain What Is Meant By Clustering. State And Explain The Various Types With Suitable Example For Each.
(May 2012)

2. Give five examples of applications that can use clustering. Describe any one clustering algorithm with the help of an example
(Dec 2012)

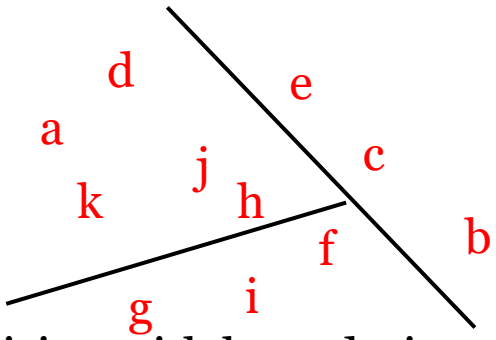
Definition

- Clustering is "the process of organizing objects into groups whose members are similar in some way".
- 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.



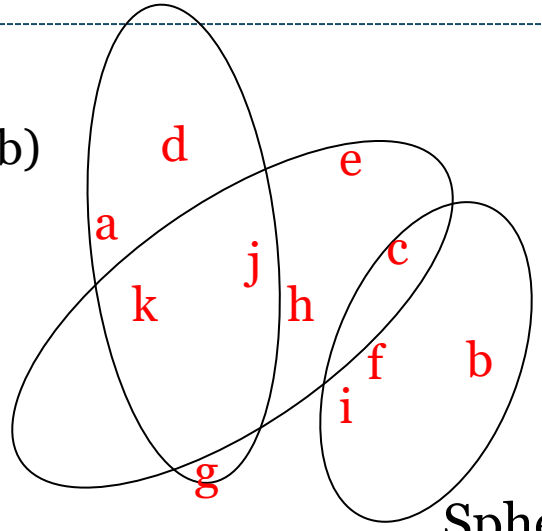
Different ways of representing clusters

(a)



Division with boundaries

(b)



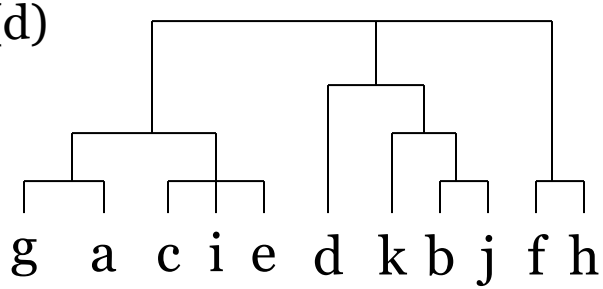
Spheres

(c)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
…			

Probabilistic

(d)

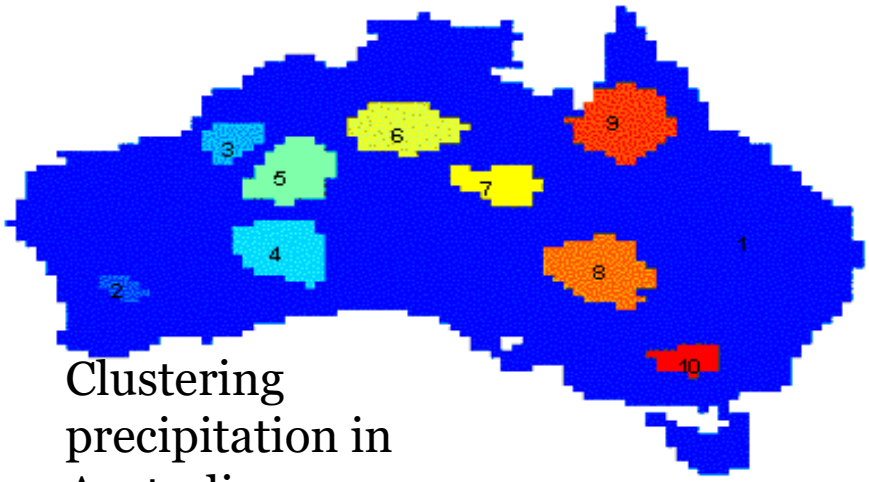


Dendrograms

Applications of Cluster Analysis

- **Understanding**
  1. Group related documents for browsing,
  2. group genes and proteins that have similar functionality,
  3. group stocks with similar price fluctuations
- **Summarization**
  4. Reduce the size of large data sets

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



Clustering precipitation in Australia

Some more examples of Clustering Applications

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

What is not Cluster Analysis?

- **Supervised classification**
  - Have class label information
- **Simple segmentation**
  - Dividing students into different registration groups alphabetically, by last name
- **Results of a query**
  - Groupings are a result of an external specification
- **Graph partitioning**
  - Some mutual relevance and synergy, but areas are not identical

Try to look for similarities and differences within a data set and group similar rows together into clusters or segments

A Clustering Example

Income: High
Children:1
Car: Luxury

Cluster 1

Income: Low
Children:0
Car:Compact

Cluster 2

Car: Sedan and
Children:3
Income: Medium

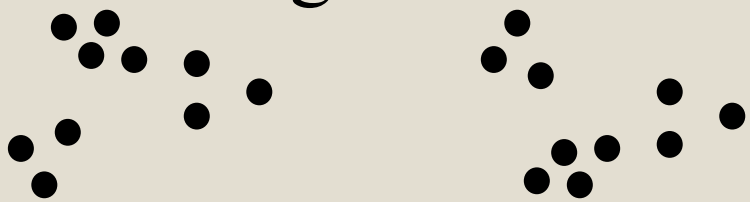
Cluster 3

Income: Medium
Children:2
Car: Truck

Cluster 4

Clustering is ambiguous

- There is no correct or incorrect solution for clustering.



How many clusters?

Six Clusters

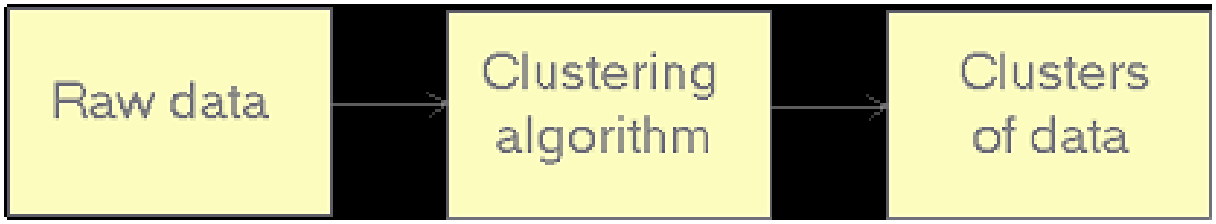
Two Clusters

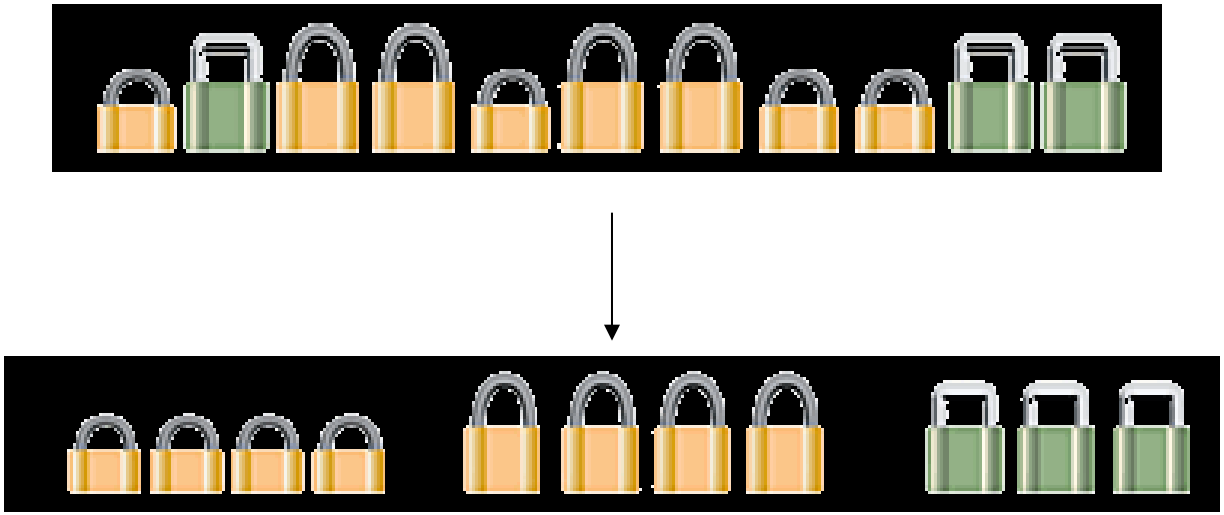
Four Clusters

Challenges faced

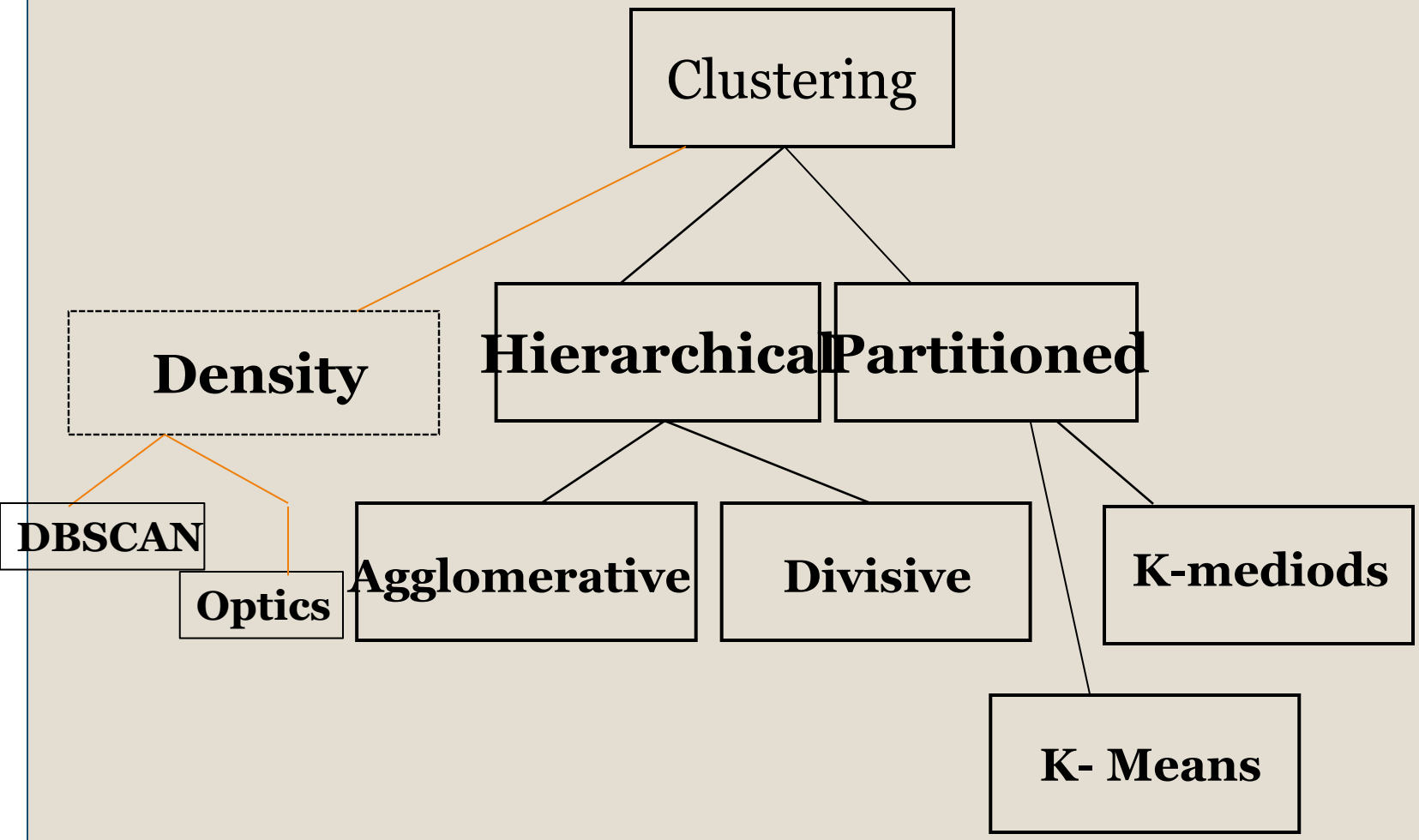
- Scalability
- Ability to deal with different types of attributes
- Noise & Outliers
- Complex shapes and types of data
- Incremental clustering and insensitivity to the order of input records
- High dimensionality
- Constraint-based clustering
- Interpretability and usability

Clustering





Clustering Approaches

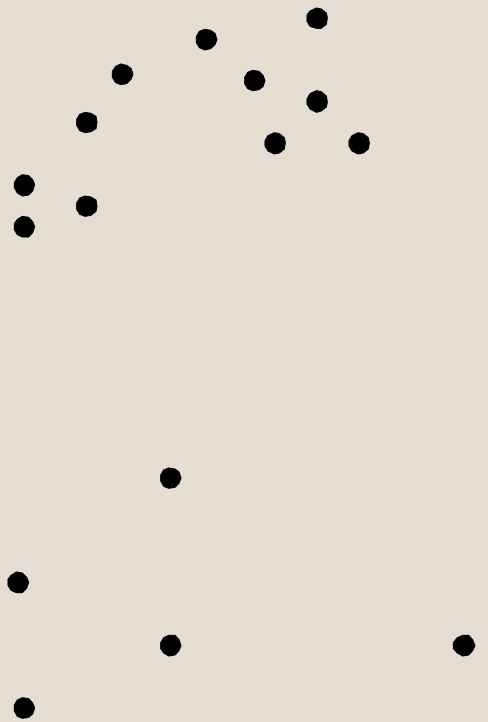


Partitioning Clustering

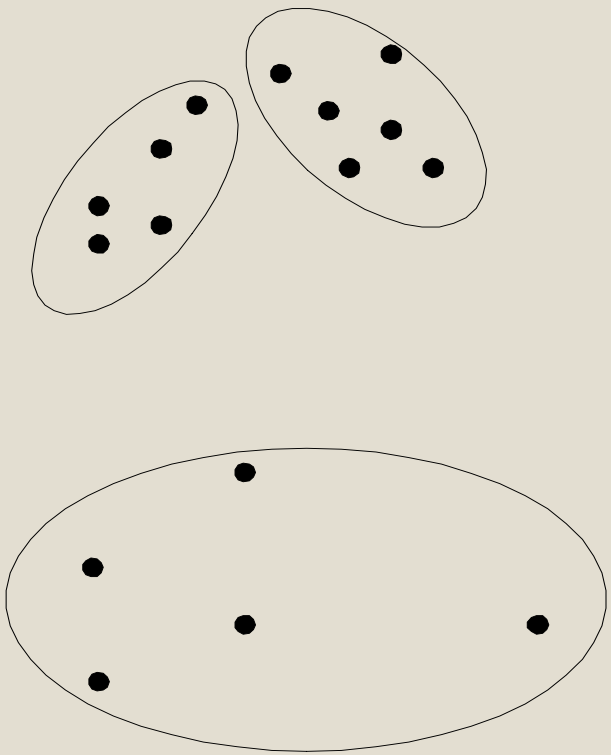
K-MEANS, K-MEDIODS

I. Partition Clustering

Explain partitioning methods for clustering. (Dec 2010)



Original Points



A Partitional Clustering

- Finding all clusters at once
- A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

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/tally all partitions
  - **Heuristic methods**: *k-means* and *k-medoids* algorithms
    1. **k-means :** Each cluster is represented by the center of the cluster
    2. **k-medoids or PAM (Partition around medoids):** Each cluster is represented by one of the objects in the cluster

K-Means

Partitioning: K-Mean Algorithm

Write short note on K-Means Clustering. (May 2013)

- Each cluster is represented by the mean value of the objects in the cluster
- ***Input***: Set of objects *(n)*, no of clusters (*k*)
- ***Output*** : Set of *k* clusters
- **Algorithm (K-Means Array):**

1. Partition objects into *k* nonempty subsets. And Randomly assign data to each cluster.
2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
3. Assign each object to the cluster with the nearest seed point .
4. Go back to Step 2, stop when no more new assignment

Example 1

Given: {2,3,6,8,9,12,15,18,22} Assume k=3.

- Solution:
  - Randomly partition given data set:
    - K1 = 2,8,15 mean = 8.3
    - K2 = 3,9,18 mean = 10
    - K3 = 6,12,22 mean = 13.3
  - Reassign
    - K1 = 2,3,6,8,9 mean = 5.6
    - K2 = mean = 0
    - K3 = 12,15,18,22 mean = 16.75

- Reassign
  - K1 = 3,6,8,9 mean = 6.5
  - K2 = 2 mean = 2
  - K3 = 12,15,18,22 mean = 16.75
- Reassign
  - K1 = 6,8,9 mean = 7.6
  - K2 = 2,3 mean = 2.5
  - K3 = 12,15,18,22 mean = 16.75
- Reassign
  - K1 = 6,8,9 mean = 7.6
  - K2 = 2,3 mean = 2.5
  - K3 = 12,15,18,22 mean = 16.75
- STOP

Example 2

What is clustering? Explain K-means clustering algorithm. Suppose the data for clustering is {2,4,10,12,3,20,30,11,25} consider K=2, Cluster the given data using above algorithm. **(Dec 2010)**

1. Given {2,4,10,12,3,20,30,11,25} Assume number of cluster i.e. k=2.

- Solution:
  - Randomly partition given data set:
    - K1 = 2,10,3,30,25 mean = 14
    - K2 = 4,12,20,11 mean = 11.75
  - Reassign
    - K1 = 20,25,30 mean = 25
    - K2 = 2,3,4,10,11,12 mean = 7
  - Reassign
    - K1 = 20,25,30 mean = 25
    - K2 = 2,3,4,10,11,12 mean = 7

So the final answer is K1= {2,3,4,10,11,12}, K2={20,25,30}

K-means : Advantage & Disadvantage

Advantages

- K-means is relatively scalable and efficient in processing large data sets
- The computational complexity of the algorithm is O(nkt)

  n: the total number of objects

  k: the number of clusters

  t: the number of iterations

  Normally: k<<n and t<<n {cluster<< objects & iterations<objects}

Disadvantage

- Can be applied only when the mean of a cluster is defined
- Users need to specify k
- K-means is not suitable for discovering clusters with non convex shapes or clusters of very different size
- It is sensitive to noise and outlier data points (can influence the mean value)

Questions on Partitioning method-K-Means

1. What Is K-means Clustering? Confer The K-means Algorithm With The Following Data For Two Clusters. Data Set {10,4,2,12,3,20,30,11,25} . **(May2010)**
2. Explain K-means clustering algorithm. Suppose the data for clustering is {2,4,10,12,3,20,30,11,25} consider K=2, Cluster the given data using above algorithm. **(Dec 2010)**
3. Write short note on K-Means Clustering. **(May 2013)**
4. **May 2016**

   Explain K-Means clustering algorithm? Apply K-Means algorithms for the following data set with two clusters. Data Set = {1, 2, 6, 7, 8, 10, 15, 17, 20} [10]

5. **May 2017**

   Explain K-Means clustering algorithm? Apply K-Means algorithms for the following Data set with two clusters. [10]
   Data Set = { 15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65}

Explain K-Means clustering algorithm? Apply K-Means algorithms for the following data set with two clusters. Data Set = {1, 2, 6, 7, 8, 10, 15, 17, 20} [10]

- D={1,2,6,7,8,10,15,17,20}
  - K1={2,7,10,17}
  - K2={1,6,8,15,20}
- Iteration 1: m1=9, m2=10
  - K1 (9)={1,2,6,7,8)
  - K2(10)={10,15,17,20}
- Iteration 2: m1=4.8, m2=15.5
  - K1(4.8)= {1,2,6,7,8,10)
  - K2(15.5)={15,17,20}
- Iteration 3: m1=5.6, m2=17.3
  - k1(5.6)={1,2,6,7,8,10}
  - k2 (17.3)={15,17,20}
- Iteration 4: m1=5.6, m2=17.3
  - k1(5.6)={1,2,6,7,8,10}
  - k2 (17.3)={15,17,20}

K-Means (Graph)

K-Means (graph)

- **Step1:** Form *k* centroids, randomly
- **Step2:** Calculate distance between centroids and each object
  - Use Euclidean's law to determine min distance:

$$d(A,B) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

- **Step3:** Assign objects based on min distance to *k* clusters
- **Step4:** Calculate centroid of each cluster using

$$C = \left(\frac{x_1+x_2+\ldots x_n}{n}, \frac{y_1+y_2+\ldots y_n}{n}\right)$$

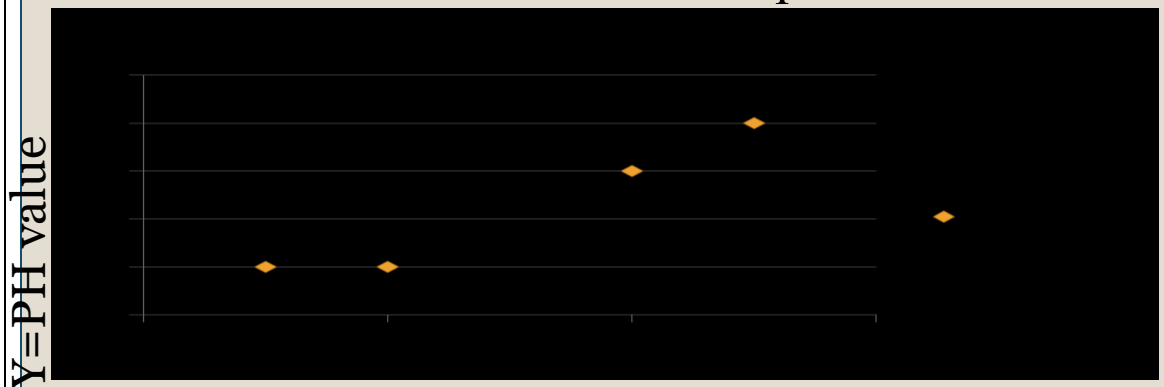
  - Go to step 2.
  - Repeat until no change in centroids.

Example 1

- There are four types of medicines and each have two attributes, as shown below. Find a way to group them into 2 groups based on their features.

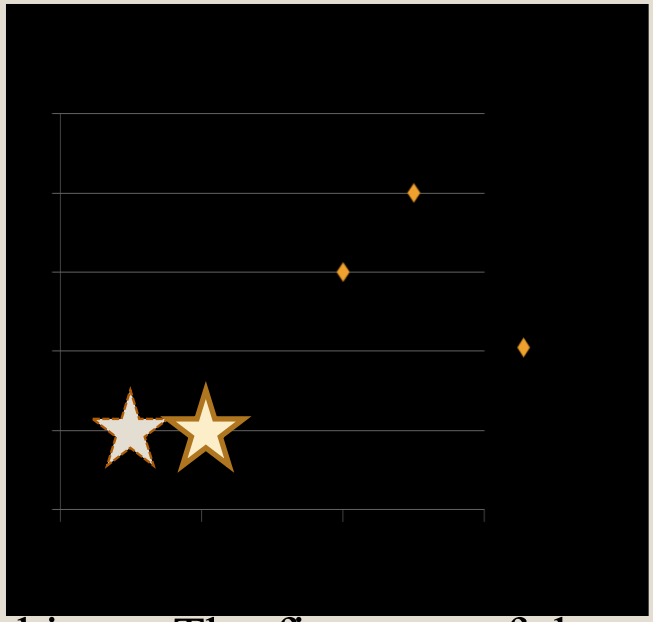
Medicine	Weight (X)	pH (Y)
A	1	1
B	2	1
C	4	3
D	5	4

- **Solution:**
  - Each medicine represents one point with 2 attributes (X,Y) that we can represent it as coordinate in an attribute space.



Solution Contd..

1. **Initial Value of centroids:** Suppose we use medicine A and Medicine B as the first centroids. Let c1 and c2 denote the coordinate of the centroids, then c1=(1,1) and c2=(2,1)
2. **Objects-Centroids Distance:** We calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is



- D =

0	1	3.61	5	**G1**
1	0	2.83	4.24	**G2**
A	B	C	D	

- Each column in the distance matrix symbolizes the objects. The first row of the distance matrix corresponds to *the distance of each object to the first centroid* and the second row is *the distance of each object to the second centroid. For e.g. distance from medicine c=(4,3) to 1st centroid c1=(1,1) is* $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

$= \sqrt{(4-1)^2 + (3-1)^2} = 3.61$, *and its distance to the 2nd centroid c2=(2,1) is*

$= \sqrt{(4-2)^2 + (3-1)^2} = 2.83$

Solution Contd..

3. Objects Clustering: we assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B, C and D is assigned to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$G^o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

4. Iteration-1, determine centroid: Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in c1=(1,1). Group 2 now has 3 members , thus the centroid is the average co-ordinate among the 3 members:

$$C2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right) = (3.6, 2.6)$$

5. Iteration-1, Objects-Centroid Distance: Next step is to compute the distance of all objects to the new centroid. Similar to step 2, we have distance matrix at iteration 1 is:

$$D = \begin{array}{c} \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{array}{l} \mathbf{G1} \\ \mathbf{G2} \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array}$$

Solution Contd..

6. Iteration 1, Objects Clustering: Similar to step 3, we assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B is moved to group 1, C and D is assigned to group 2.

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

7. Iteration-2, determine centroid: Now we repeat step 4 to calculate the new centroid co-ordinate based on the clustering of previous iteration. Group 1 and Group 2 both has two members, thus the new centroid are:

$$C1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(\frac{3}{2}, \frac{2}{2}\right) = (1.5, 1) \qquad C1 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(\frac{3}{2}, \frac{2}{2}\right) = (4.5, 3.5)$$

8. Iteration-2, Objects-Centroid Distance: Next step is to compute the distance of all objects to the new centroid. Similar to step 2, we have distance matrix at iteration 1 is:

$$D = \begin{array}{c} \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array} \begin{array}{l} \mathbf{G1} \\ \mathbf{G2} \\ \\ \end{array}$$

Solution Contd..

9. Iteration 2, Objects Clustering: Similar to step 3, we assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B is moved to group 1, C and D is assigned to group 2.

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

We obtain result that $G^2 = G^1$. Comparing the grouping of last iteration and this iteration reveals that the object does not move anymore. Thus the computation of the K-Mean clustering has reached its stability and no more iteration is needed. We get the final grouping as the results.

Questions on Partitioning method-K-Means Graph Implement as Exp-5

Find clusters using *k*-means clustering algorithm if we have several objects [10] (4 types of medicines) and each object have two attributes or features as shown in the table below. The goal is to group these objects into *k*=2 group of medicine based on the two features (pH and weight index). **May 2018**

Object	Attribute 1(X) Weight Index	Attribute 2 (Y) pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

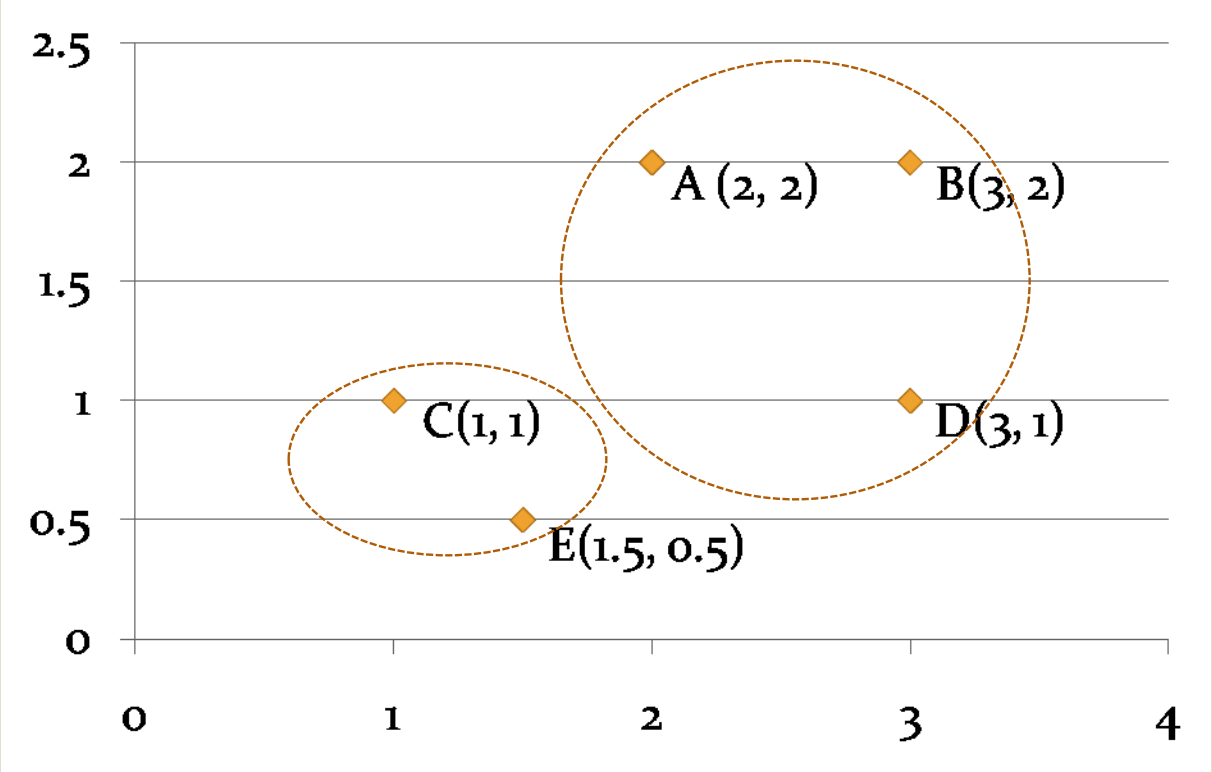
May 2019

Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points are: A_1 (2, 10), A_2 (2, 5), A_3 (8, 4), B_1 (5, 8), B_2 (7, 5), B_3 (6, 4), C_1 (1, 2), C_2 (4, 9). **[10]**

The distance function is Euclidean distance. Suppose initially we assign A_1, B_1, and C_1 as the center of each cluster, respectively. Use the *k-means* algorithm to show only (i) The three cluster centers after the first round of execution (ii) The final three clusters.

Example 2

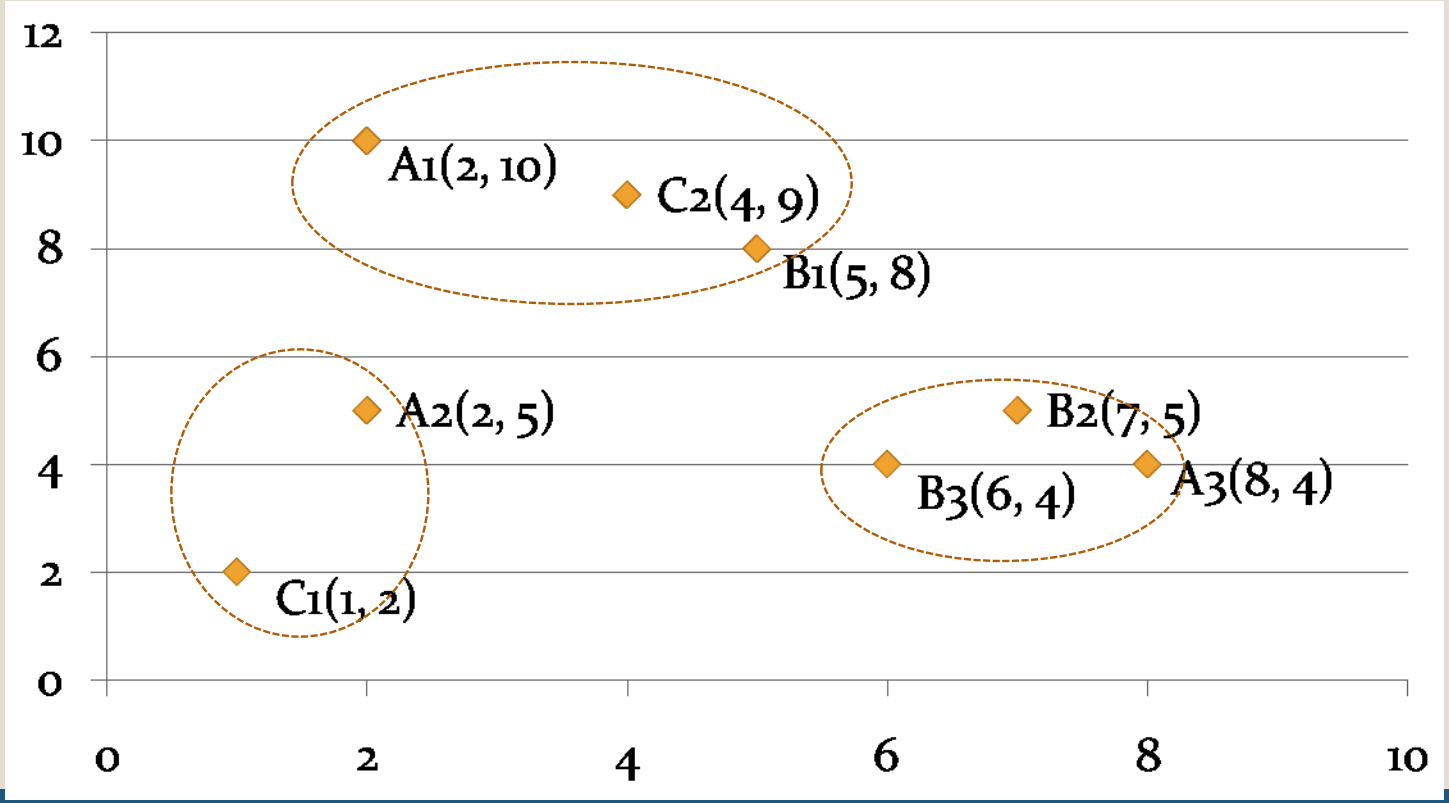
- Use K-means algorithm to create two clusters. Given:



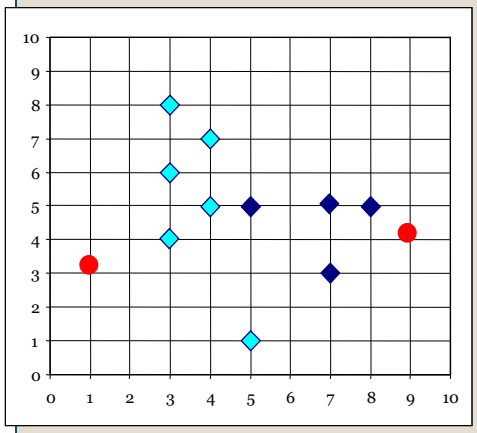
Example 3

.

Group the below points into 3 clusters



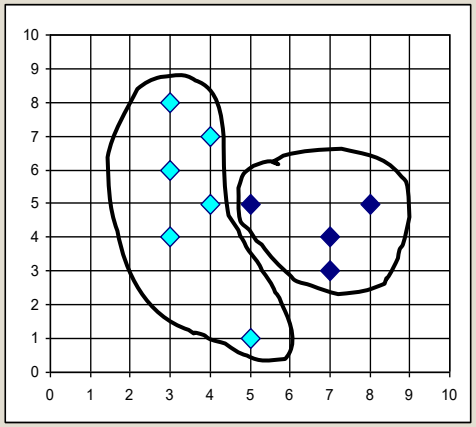
The *K-Means* Clustering Method



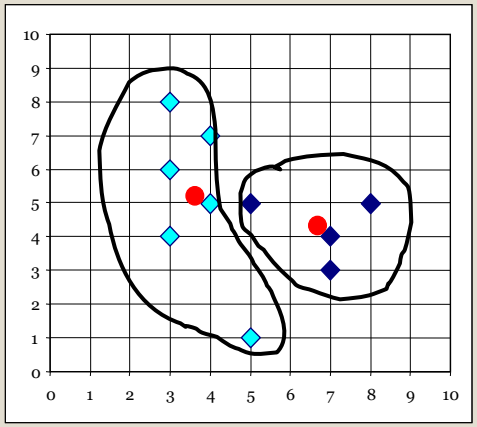
K=2

Arbitrarily choose K object as initial cluster center

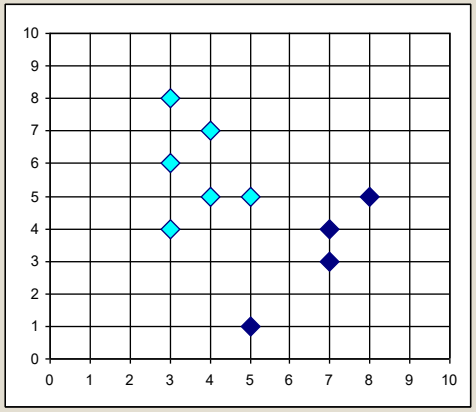
Assign each objects to most similar center



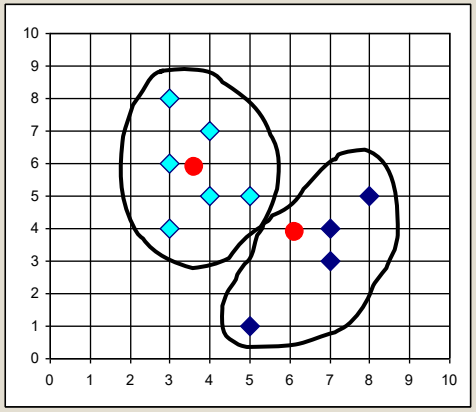
Update the cluster means



reassign



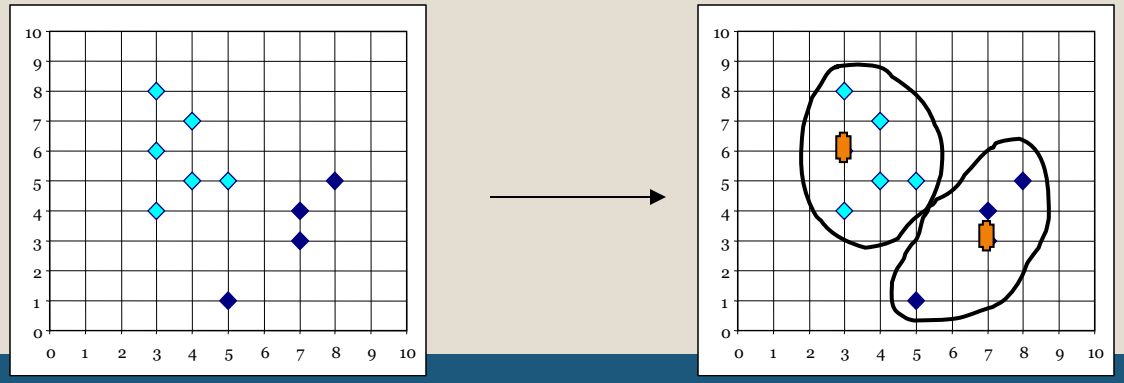
Update the cluster means



reassign

What is the problem of k-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- **K-Medoids:** Instead of taking the **mean** value of the object in a cluster as a reference point, **Medoids** can be used, which is the **most centrally located** object in a cluster.



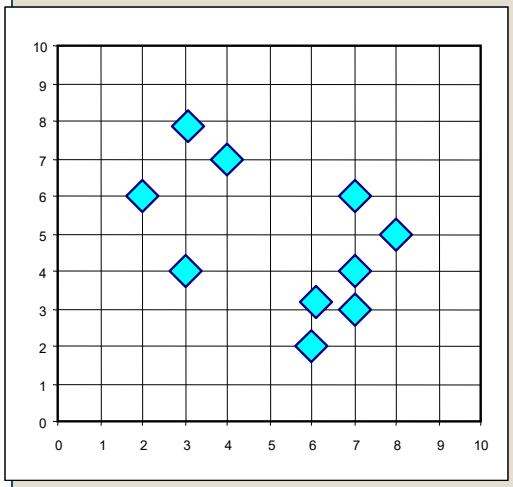
The K-Mediods Clustering Method

- Find *representative* objects, called Medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - Starts from an initial set of Medoids and iteratively replaces one of the Medoids by one of the non-Medoids if it improves the total distance of the resulting clustering
  - PAM works effectively for small data sets, but does not scale well for large data sets
  - PAM is more robust than k-means in the presence of noise and outliers.

K-Mediods

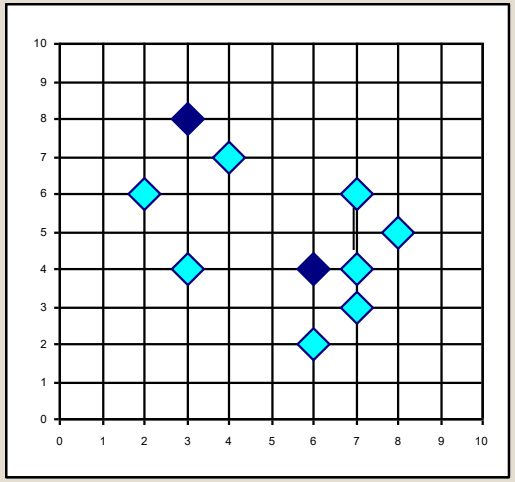
Typical k-medoids algorithm (PAM)

Total Cost = 20

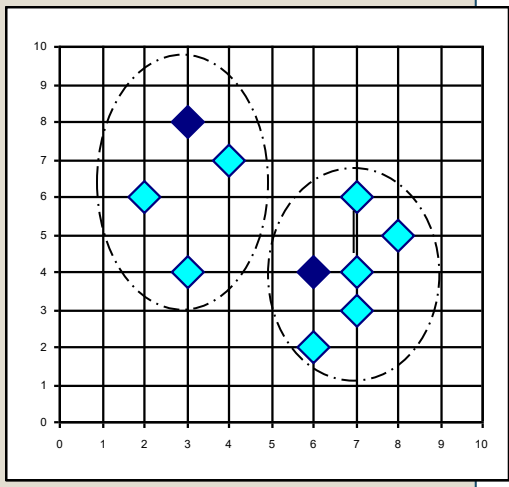


K=2

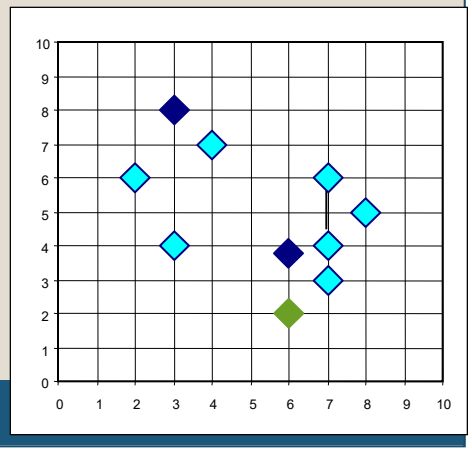
Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids

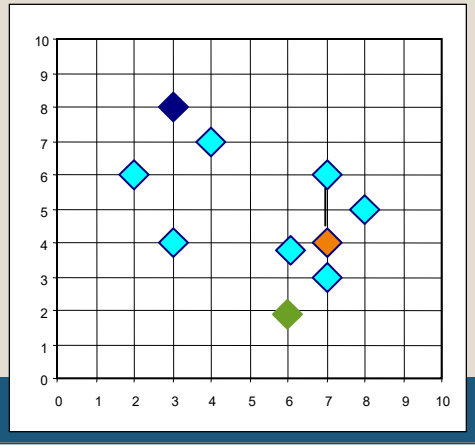


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

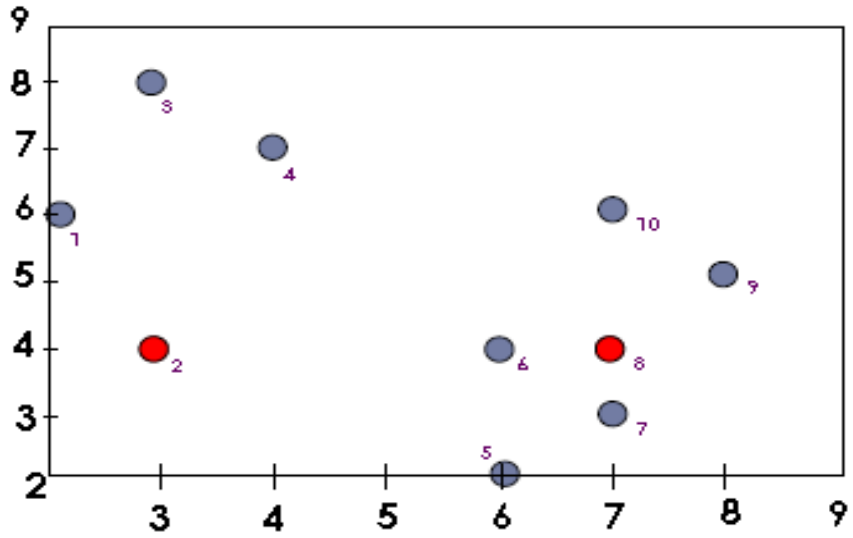
K-MedoidS (PAM) Algorithm

- Also called Partitioning Around Medoids.
  1. Given k
  2. Randomly pick k instances as initial Medoid
  3. Assign each data point to the nearest Medoid x
  4. Calculate the objective function
     - the sum of dissimilarities of all points to their nearest Medoids. (Absolute-Error Criterion / Manhattan)
  5. Randomly select an point y
  6. Swap x by y if the swap reduces the objective function
  7. Repeat (3-6) until no change

Step 1: K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Goal: create two clusters

Choose randmly two medoids

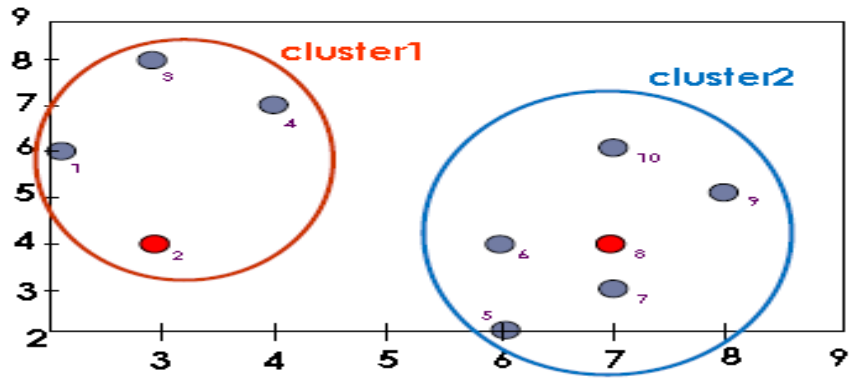
$O_2 = (3,4)$
$O_8 = (7,4)$

Objects 2 and 8 are the selected representative objects initially.(Random Selection)

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→ Assign each object to the closest representative object

→ Using L1 Metric (Manhattan), we form the following clusters

Cluster1 = $\{O_1, O_2, O_3, O_4\}$

Cluster2 = $\{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

- **Calculate the distance of every object with respect to selected object 2 and 8.**

|O1-O2|= |2-3|+|6-4|= 3 **|O1-O8|**= |2-7|+|6-4|= 4
|O3-O2|= |3-3|+|8-4|= 4 **|O3-O8|**= |3-7|+|8-4|= 8
|O4-O2|= |4-3|+|7-4|= 4 **|O4-O8|**= |4-7|+|7-4|= 6

- **Find the closest representative object with respect to the selected object.**

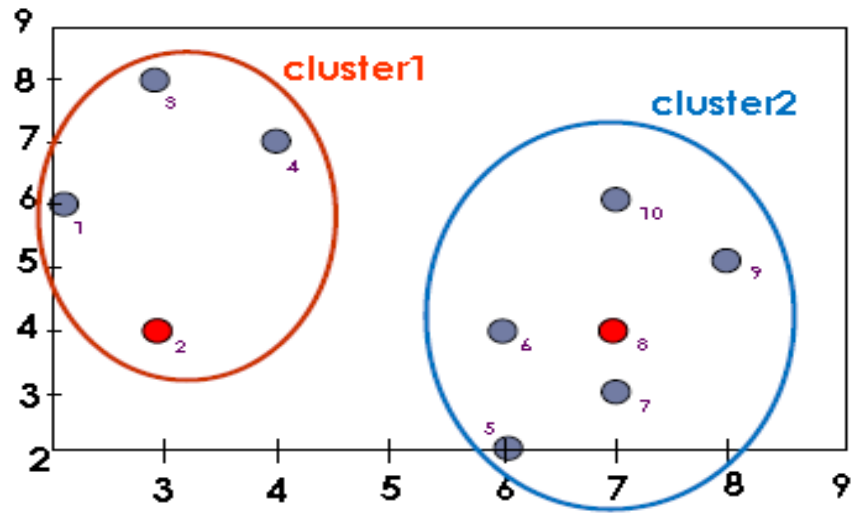
If **p = 1, L_1** is the *Manhattan (or city block)* distance:

$$L_1(x, y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_d - y_d| = \sum_{i=1}^{d} |x_i - y_i|$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the absolute error criterion [for the set of Medoids (O2,O8)]

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

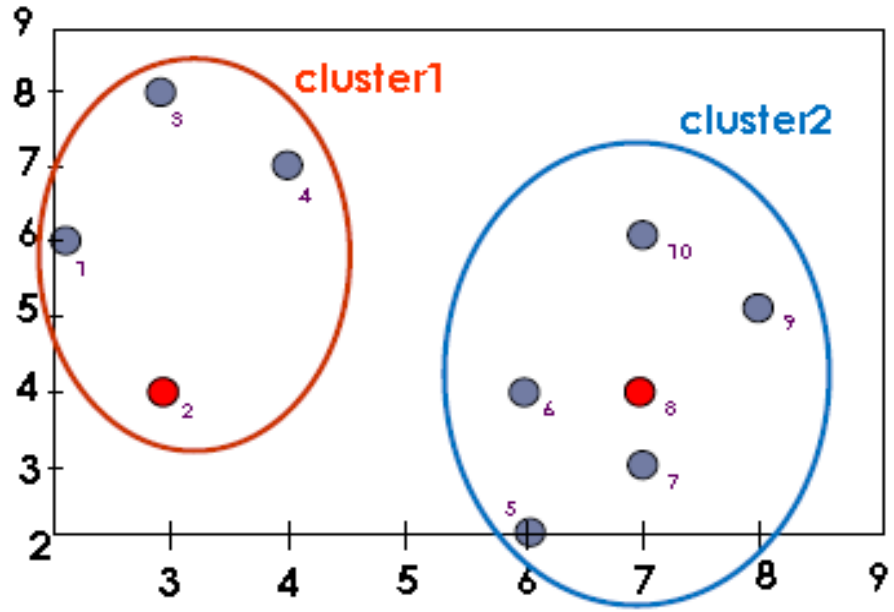
$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

- Calculate the average value of minimal dissimilarity.
- Cost= average value of minimal dissimilarity= 20

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→The absolute error criterion **[for the set of Medoids (O2,O8)]**

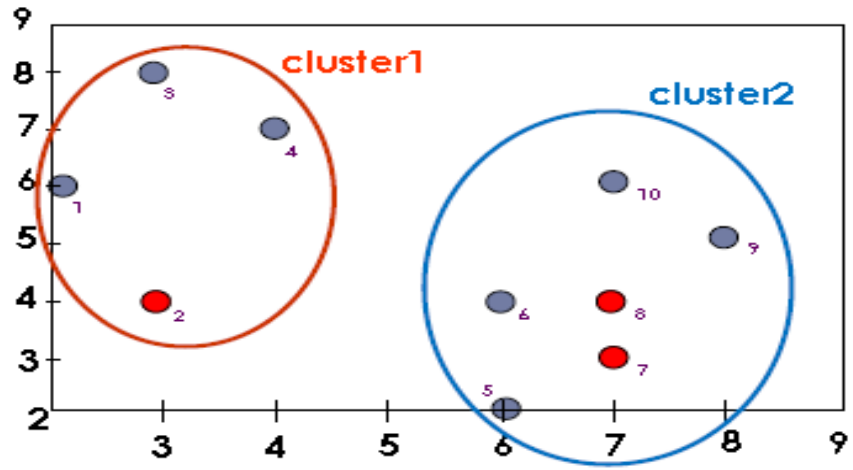
$$E = (3+4+4)+(3+1+1+2+2) = 20$$

This is the cost / average value of minimal dissimilarity

Step 2: K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

→Swap O8 and O7

→Compute the absolute error criterion [for the set of Medoids (O2,O7)]

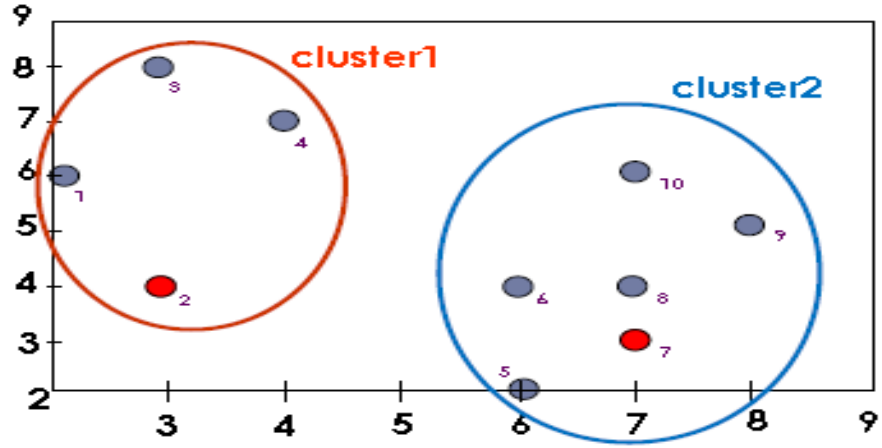
$$E = (3+4+4)+(2+2+1+3+3) = 22$$

Step 2: 1. Select two objects randomly
2. Objects O2 and O7(Swapped O8 with O7)
3. Repeat the step 1
4. Calculate the average minimal dissimilarity

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the cost function

Absolute error [for O_2,O_7] – Absolute error [O_2,O_8]

$$S = 22 - 20$$

S> 0 ⇒ it is a bad idea to replace O_8 by O_7

Step 3: Calculation of Swapping Cost= New cost-Old cost=22-20=2

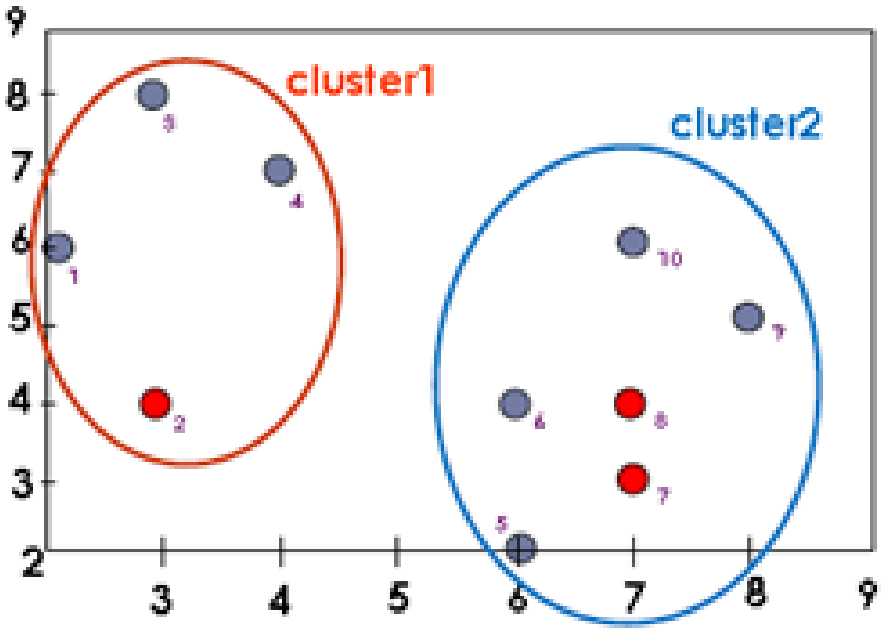
If Swapping Cost < 0, replace Medoid with new selected object

Else **Step 4:** *Repeat Step 2 and 3 until Swapping cost<=0*

K-Medoids Method

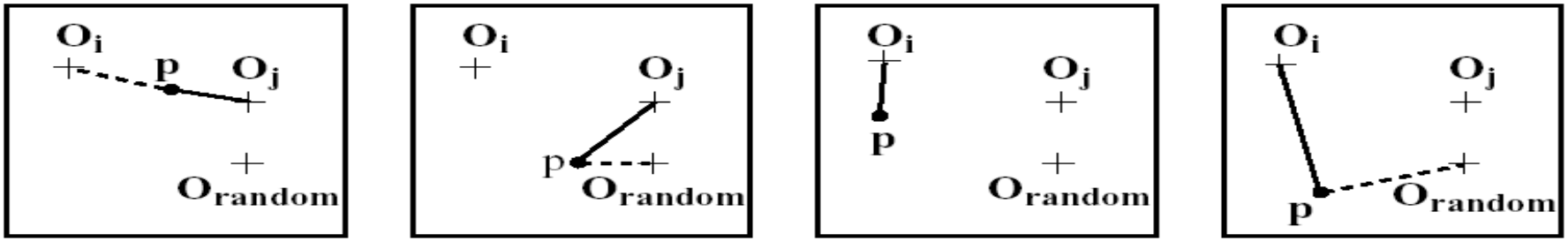
Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



- In this example, changing the medoid of cluster 2 did not change the assignments of objects to clusters.

Four Cases: Evaluating Cost of Swapping Medoids



1. Reassigned to O_i 2. Reassigned to O_{random} 3. No change 4. Reassigned to O_{random}

Case 1 p belongs to medoid o_j. If o_j is replaced by o_{random} as a medoid and p is closest to some other medoid o_i, then p is reassigned to o_i.

Case 2 p belongs to medoid o_j. If o_j is replaced by o_{random} as a medoid and p is closest to o_{random}, then p is reassigned to o_{random}.

Case 3 p belongs to medoid o_i, $i \neq j$. If o_j is replaced by o_{random} as a medoid and p is still closest to o_i, then the assignment doesn't change.

Case 4 p belongs to medoid o_i, $i \neq j$. If o_j is replaced by o_{random} as a medoid and p is closest to o_{random}, then p is reassigned to o_{random}.

Distance measures

What are the different ways of finding the distance between two clusters? **(May 2011)**

From the scientific and mathematical point of view, distance is defined as a quantitative degree of how far apart two objects are.

Synonyms for distance include dissimilarity.

Type	Possible operations	Example variable	Example values
Nominal (E.g. Map Color)	==	Major subject	Computer science Mathematics Physics
Ordinal (E.g. Post Rank)	==, <, >	Degree	Bachelor Master Doctor
Interval (E.g. Weight, height)	==, <, >, -	Temperature	10 °C 20 °C 10 °F
Ratio (E.g. Growth of bacteria, length, counts)	==, <, >, -, /	Blood count	13.3 [12-16] normal

Definition of distance *metric*

A distance function is metric if the following conditions are met for all data points x, y, z:

1. All distances are non-negative:

   $$d(x, y) \geq 0$$

2. Distance to point itself is zero:

   $$d(x, x) = 0$$

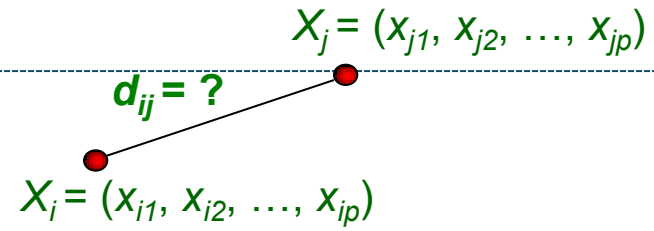
3. All distances are symmetric:

   $$d(x, y) = d(y, x)$$

4. Triangular inequality:

   $$d(x, y) \leq d(x, z) + d(z, y)$$

Common distance metrics



1. Minkowski distance

$$d(i,j) = \sqrt[q]{\underbrace{|x_{i1} - x_{j1}|^q}_{1^{st}\text{ dimension}} + \underbrace{|x_{i2} - x_{j2}|^q}_{2^{nd}\text{ dimension}} + \ldots + \underbrace{|x_{ip} - x_{jp}|^q}_{p^{th}\text{ dimension}}}$$

2. Euclidean distance

q = 2

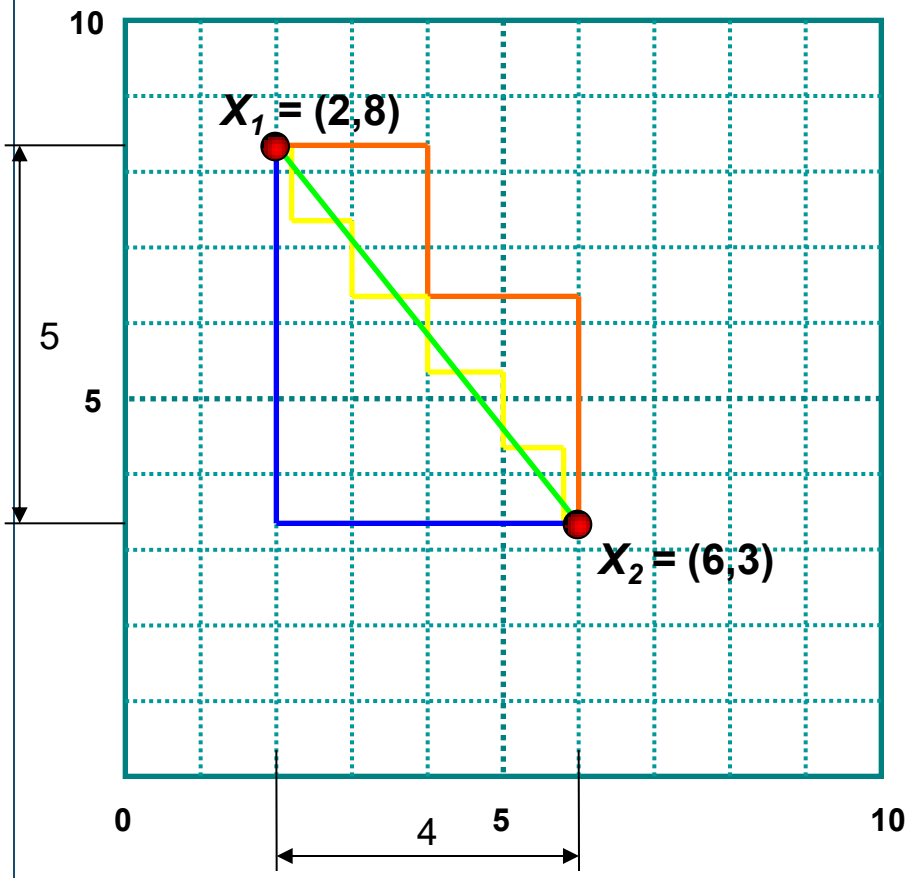
$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

3. Manhattan distance / City Block

q = 1

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

Distance metrics example



2D example

$x_1 = (2,8)$

$x_2 = (6,3)$

Euclidean distance

$$d(1,2) = \sqrt{|2-6|^2 + |8-3|^2} = \sqrt{41}$$

Manhattan distance

$$d(1,2) = |2-6| + |8-3| = 9$$

Hierarchical Clustering

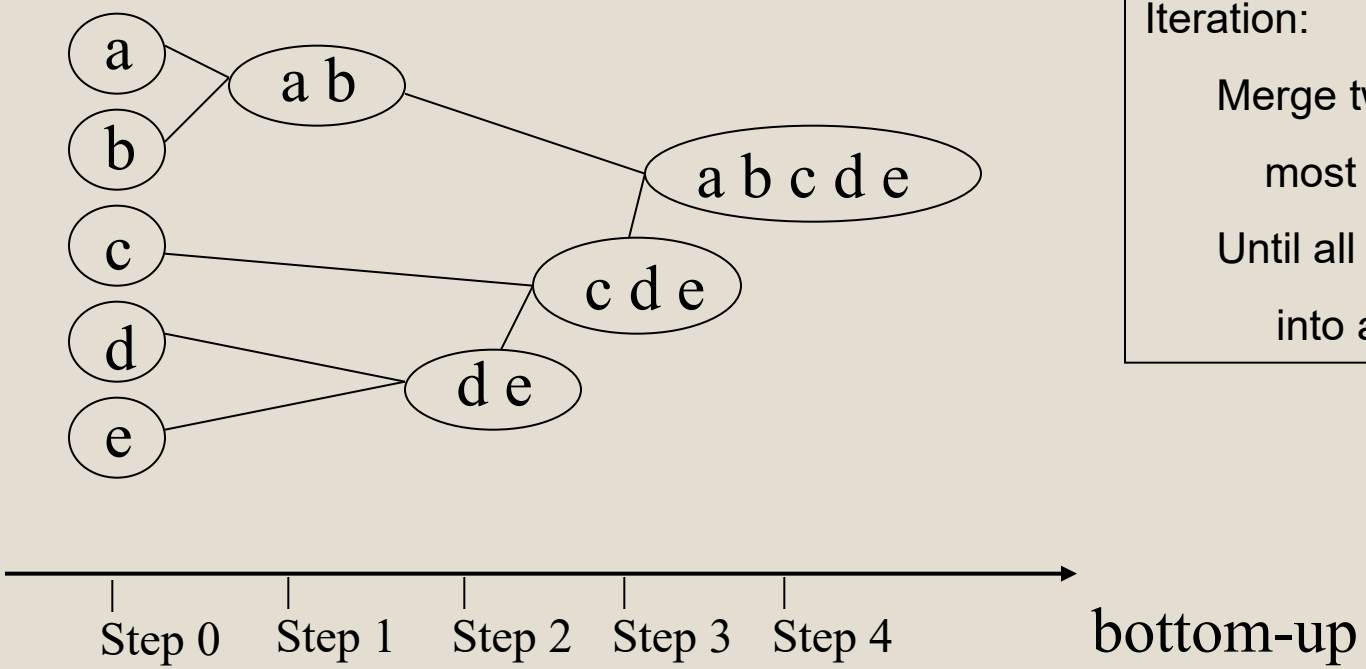
II. Hierarchical Clustering (May 17, May 18)

- Clusters are created in levels actually creating sets of clusters at each level.

1. ***Agglomerative***
   - Initially each item in its own cluster
   - Iteratively clusters are merged together
   - Bottom Up
2. ***Divisive***
   - Initially all items in one cluster
   - Large clusters are successively divided
   - Top Down

Hierarchical Clustering

- Agglomerative approach



Initialization:

Each object is a cluster

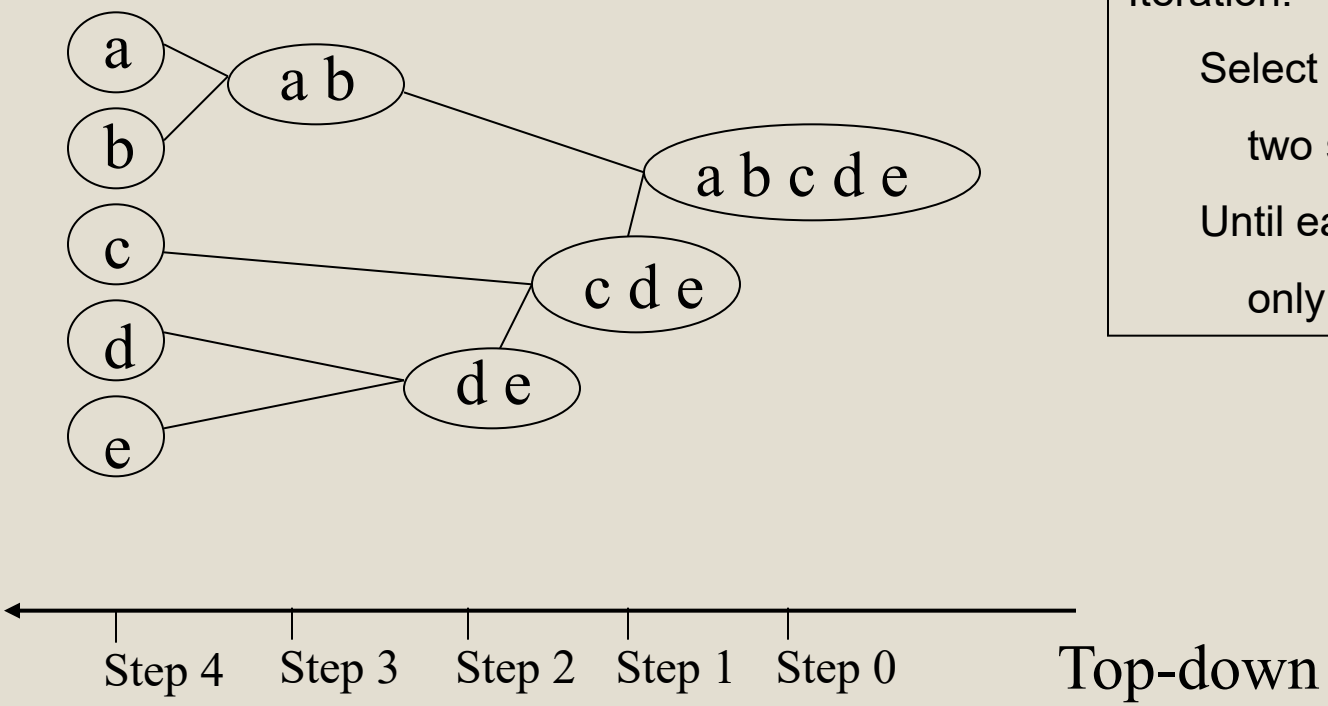
Iteration:

Merge two clusters which are most similar to each other;

Until all objects are merged into a single cluster

Hierarchical Clustering

- Divisive Approaches



Initialization:

All objects stay in one cluster

Iteration:

Select a cluster and split it into two sub clusters

Until each leaf cluster contains only one object

Agglomerative Hierarchical Clustering

1. Agglomerative hierarchical clustering

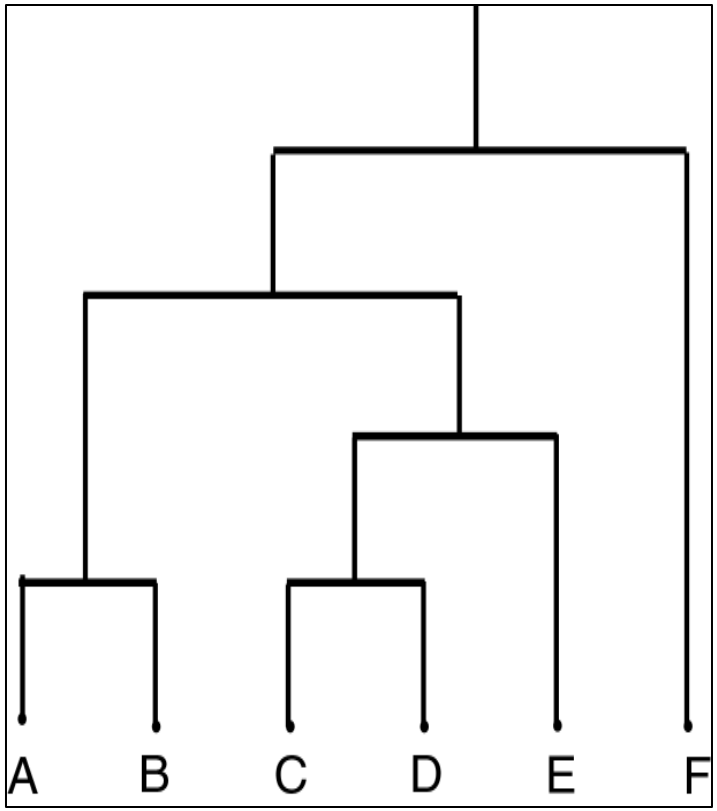
- This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
- Most hierarchical clustering methods belong to this category.

Agglomerative Algorithm (Single Link)

- **Step1:** Make each object as a cluster
- **Step2:** Calculate the Euclidean distance from every point to every other point. i.e., construct a Distance Matrix
- **Step3:** Identify two clusters with shortest distance.
  - Merge them
  - Go to Step 2
  - Repeat until all objects are in one cluster

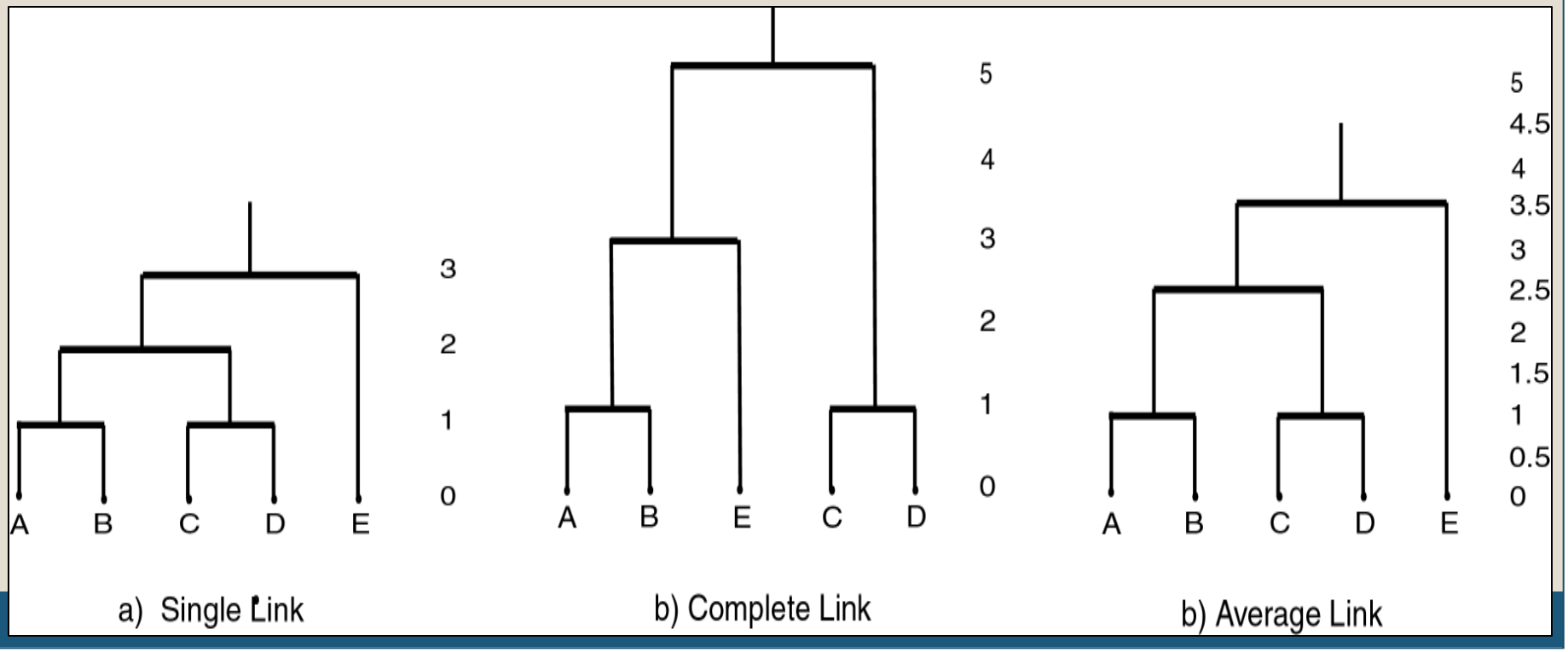
Dendrogram

- ***Dendrogram:*** a tree data structure which illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - Leaf – individual clusters
  - Root – one cluster
- A cluster at level i is the union of its children clusters at level i+1.



Hierarchical Algorithms

1. Single Link
2. Complete Link
3. Average Link



Hierarchical clustering: 1. Agglomerative example

- Let's now see a simple example: a hierarchical clustering of distances in kilometers between some Italian cities. The method used is single-linkage.
- **Input distance matrix** (L = 0 for all the clusters):

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

Hierarchical clustering: 1. Agglomerative example

- The nearest pair of cities is MI and TO, at distance 138. These are merged into a single cluster called "MI/TO". The level of the new cluster is L(MI/TO) = 138 and the new sequence number is m = 1.
- Then we compute the distance from this new compound object to all other objects.
- **In single link clustering the rule is that the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object.** So the distance from "MI/TO" to RM is chosen to be 564, which is the distance from MI to RM, and so on.

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

After merging MI with TO we obtain the following matrix: L(MI/TO)

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

Hierarchical clustering: 1. Agglomerative example

- Min d(i,j) = d(NA,RM) = 219 => merge NA and RM into a new cluster called NA/RM
  L(NA/RM) = 219 m = 2

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

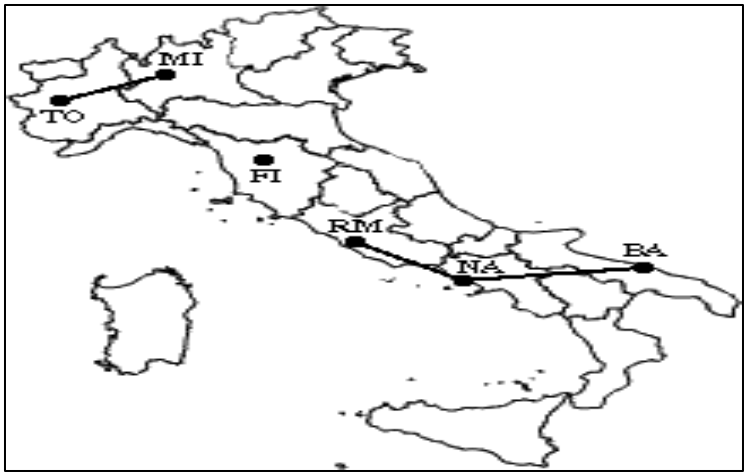
After merging NA with RM we obtain the following matrix: L(NA/RM) = 219
m = 2

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

Hierarchical clustering: 1. Agglomerative example

- min d(i,j) = d(BA,NA/RM) = 255 => merge BA and NA/RM into a new cluster called BA/NA/RM
  L(BA/NA/RM) = 255 m = 3

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0



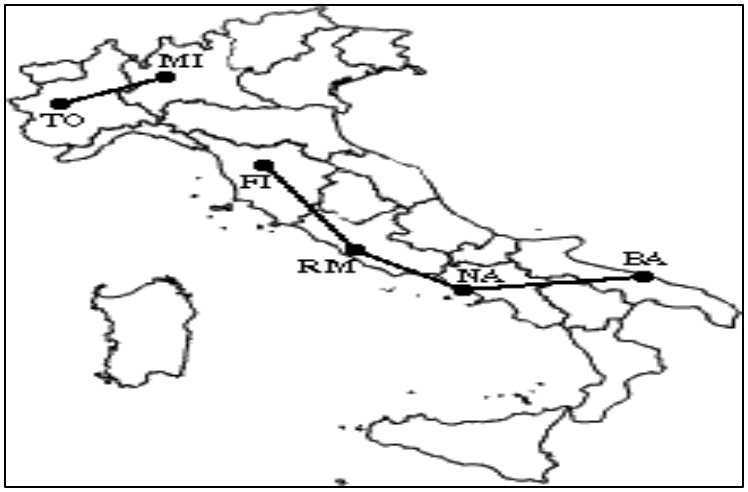
After merging BA with NA/RM we obtain the following matrix: L(BA/NA/RM) = 255 m = 3

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

Hierarchical clustering: 1. Agglomerative example

- min d(i,j) = d(BA/NA/RM,FI) = 268 => merge BA/NA/RM and FI into a new cluster called BA/FI/NA/RM L(BA/FI/NA/RM) = 268 m = 4

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0



After merging BA with NA/RM we obtain the following matrix: L(BA/FI/NA/RM) = 268 m = 4

	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0

Hierarchical clustering: 1. Agglomerative example

- Finally, we merge the last two clusters at level 295.
- The process is summarized by the following hierarchical tree:

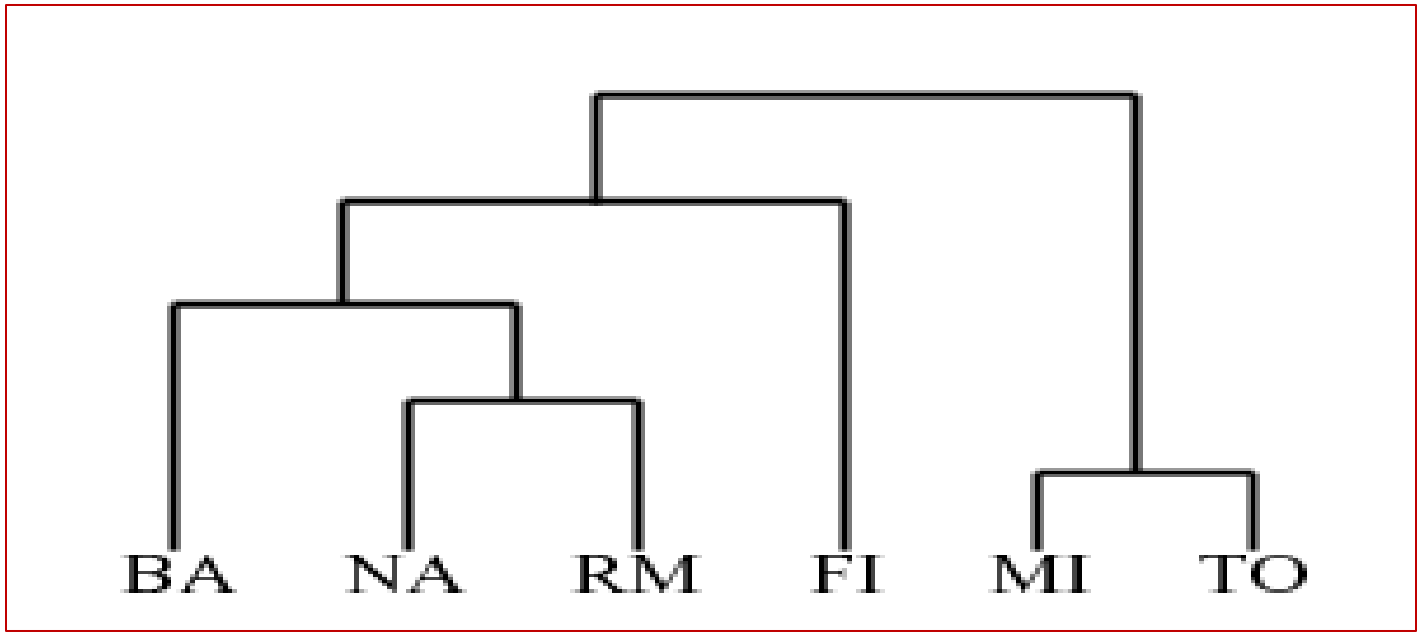


Fig: Dendrogram using link approach

Hierarchical clustering

Cost functions

Single link: the smallest distance between vectors in clusters *i* and *j*:

$$d(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

Complete-link: the largest distance between vectors in clusters *i* and *j*: (farthest-neighbor clustering algorithm)

$$d(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$

Average link: the average distance between vectors in clusters *i* and *j*:

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$$

Agglomerative – Single Link Algorithm

1. What is clustering technique? Discuss the agglomerative algorithm using following data and plot a dendrogram using link approach. The following figure contains sample data items indicating the distance between the elements. (May 2010)

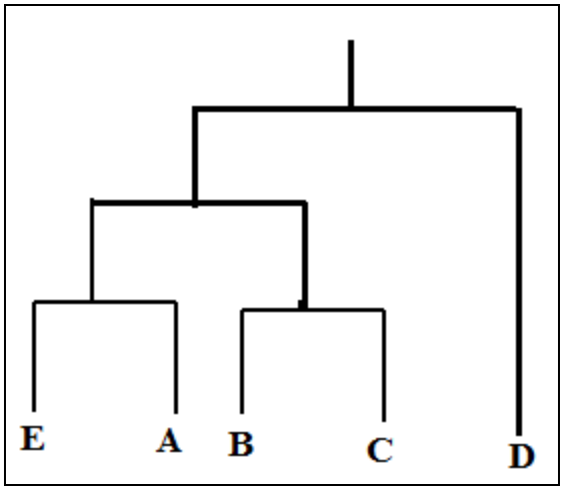
Item	E	A	C	B	D
E	0	1	2	2	3
A	1	0	2	5	3
C	2	2	0	1	6
B	2	5	1	0	3
D	3	3	6	3	0

Step 1: Merge E and A as it is having minimum distance. **(we get E/A)**

Step 2: Merge B and C as it is having minimum distance. **(we get B/C)**

Step 3: Merge E/A and B/C as it is having minimum distance. **(we get E/A/B/C)**

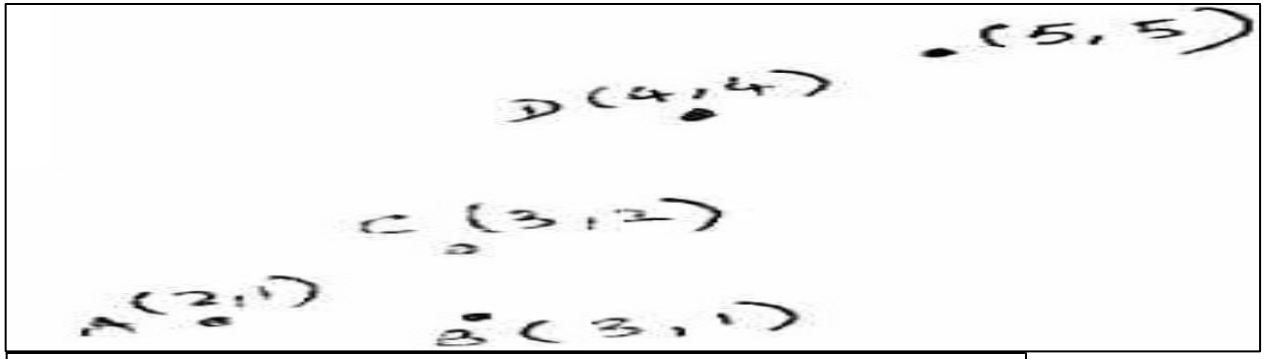
Step 4: Merge E/A/B/C and D as it is last cluster. **(we get E/A/B/C/D)**



Resulting Dendrogram

Agglomerative – Single Link Algorithm- using Adjacent Matrix

2. Consider the data set given. Create the adjacency matrix. Use single link agglomerative technique to cluster the given data. Draw the dendogram. (May 2011)



Using the Euclidean Distance we have

$$d(A,B) = \sqrt{|2-3|^2 + |1-1|^2} = 1$$

A(2,1)
B(3,1)
C(3,2)
D(4,4)
E(5,5)

	A	B	C	D	E
A	0.00	1.00	1.41	3.60	5.00
B	1.00	0.00	1.00	3.16	4.47
C	1.41	1.00	0.00	2.24	3.60
D	3.60	3.16	2.24	0.00	1.41
E	5.00	4.47	3.60	1.41	0.00

Dec 2016

What is Clustering Technique? Discuss the Agglomerative algorithm with the following data and plot a Dendrogram using single link approach. The table below comprises sample data items indicting the distance between the elements.

Item	E	A	C	B	D
E	0	1	2	2	3
A	1	0	2	5	3
C	2	2	0	1	6
B	2	5	1	0	3
D	3	3	6	3	0

DEC 2017

Consider the data given below. Create adjacency matrix. Apply single link [10] algorithm to cluster the given data set and draw the dendogram

Object	Attribute 1 (X):	Attribute 2 (Y):
A	2	2
B	3	2
C	1	1
D	3	1
E	1.5	0.5

DEC 2019

a) Show the dendrogram created by the complete link clustering algorithm for the given set of points. [10]

	A	B
P1	2	4
P2	8	2
P3	9	3
P4	1	5
P5	8.5	1

MAY 2019

b) Differentiate between simple linkage, average linkage and complete linkage algorithms. [10]
Use complete linkage algorithm to find the clusters from the following dataset.

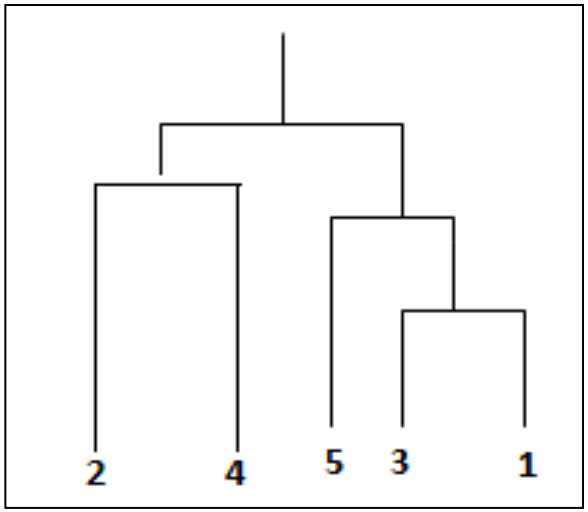
X	4	8	15	24	24
Y	4	4	8	4	12

Agglomerative – Complete Link Algorithm

1. Discuss the agglomerative algorithm using following data and plot a dendrogram using Complete link approach. The following figure contains sample data items indicating the distance between the elements.

	1	2	3	4	5
1	1.00	0.90	0.10	0.65	0.20
2	0.90	1.00	0.70	0.60	0.50
3	0.10	0.70	1.00	0.40	0.30
4	0.65	0.60	0.40	1.00	0.80
5	0.20	0.50	0.30	0.80	1.00

Step 1: Merge 1 and 3 as it is having minimum distance (0.1). **(we get 1/3).While merging consider maximum.**
Step 2: Merge 1/3 and 5 as it is having minimum distance (0.3). **(we get 1/3/5)**
Step 3: Merge 4 and 2 as it is having minimum distance (0.6). **(we get 4/2)**
Step 4: Merge 1/3/5 and 4/2 as it is last cluster. **(we get 1/3/5/4/2)**



Resulting Dendrogram

Agglomerative – Average Link Algorithm

3. Apply Agglomerative Hierarchical Clustering and draw single link and average link dendrogram for the following distance matrix. (May 2013)

	A	B	C	D	E
A	0	2	6	10	9
B	2	0	3	9	8
C	6	3	0	7	5
D	10	9	7	0	4
E	9	8	5	4	0

Step 1: Merge A and B as it is having minimum distance (2). **(we get A/B); while merging consider average.**
Step 2: Merge D and E as it is having minimum distance (4). **(we get D/E)**
Step 3: Merge A/B and C as it is having minimum distance (4.5). **(we get A/B/C)**
Step 4: Merge A/B/C and D/E as it is last cluster. With distance 7.5 **(we get A/B/C/D/E)**

Using Average Link

Comparison between Single, complete & average link

1. **Single Link**

   Advantage: Can handle non-elliptical shapes

   Dis-Advantage:

   - Sensitive to noise and outliers
   - It produces long, elongated clusters

2. **Complete Link**

   Advantage: Less susceptible to noise and outliers

   Dis-Advantage:

   - Tends to break large clusters
   - Biased towards globular clusters

3. **Average Link**

   - **Advantage:** Less susceptible to noise and outliers

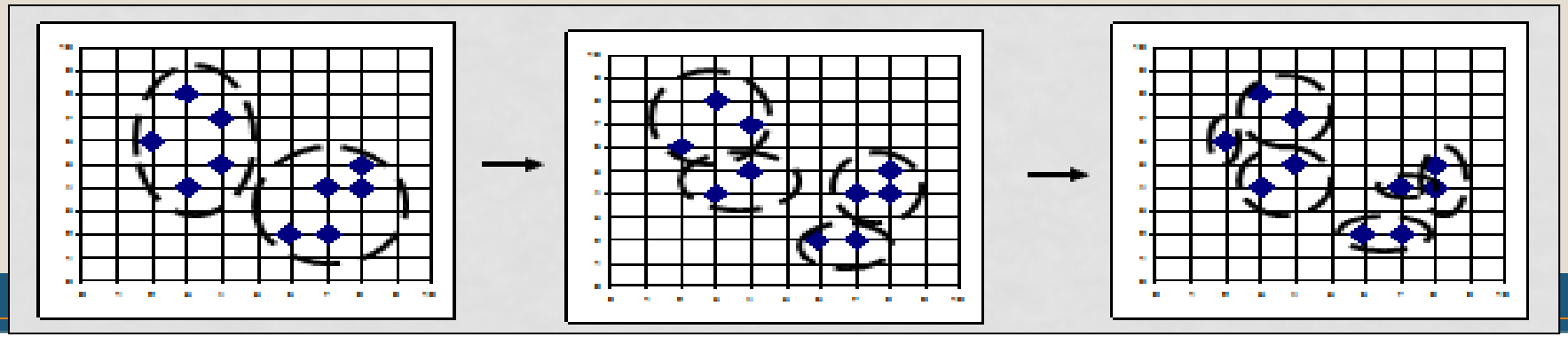
   Dis-Advantage: Biased towards globular clusters

Divisive Hierarchical Clustering

2. Divisive Hierarchical Clustering

Divisive Analysis- (DIANA)

- This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster.
- It subdivides the clusters into smaller and smaller pieces, until **each object form a cluster on its own** or until **it satisfies certain termination conditions**, *such as a desired number of cluster or the diameter of each cluster is within a certain threshold.*



DIANA- Explored

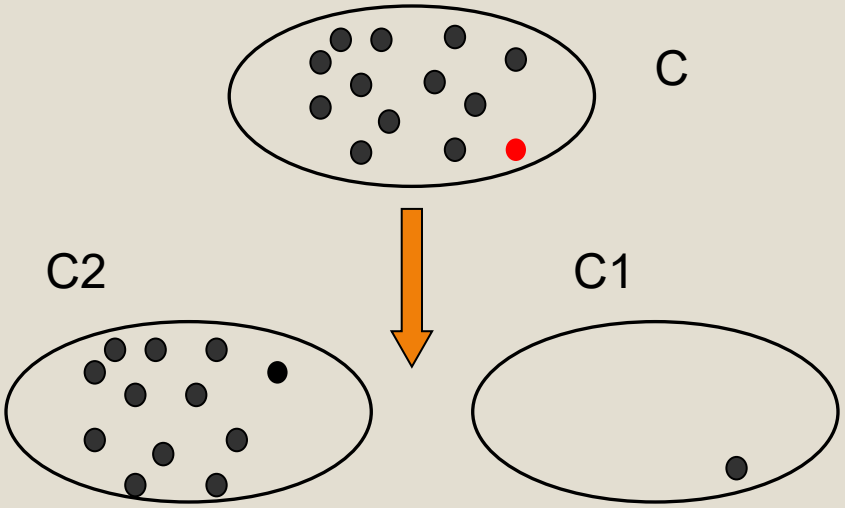
- First, all of the objects form one cluster.
- The cluster is split according to some principle, such as the minimum Euclidean distance between the closest neighboring objects in the cluster.
- The cluster splitting process repeats until, eventually, each new cluster contains a single object or a termination condition is met.

Splitting Process of DIANA

Intialization:

1. Choose the object O_h which is most dissimilar to other objects in C.

2. Let C1={O_h}, C2=C-C1.



Splitting Process of DIANA (Cont'd)

Iteration:

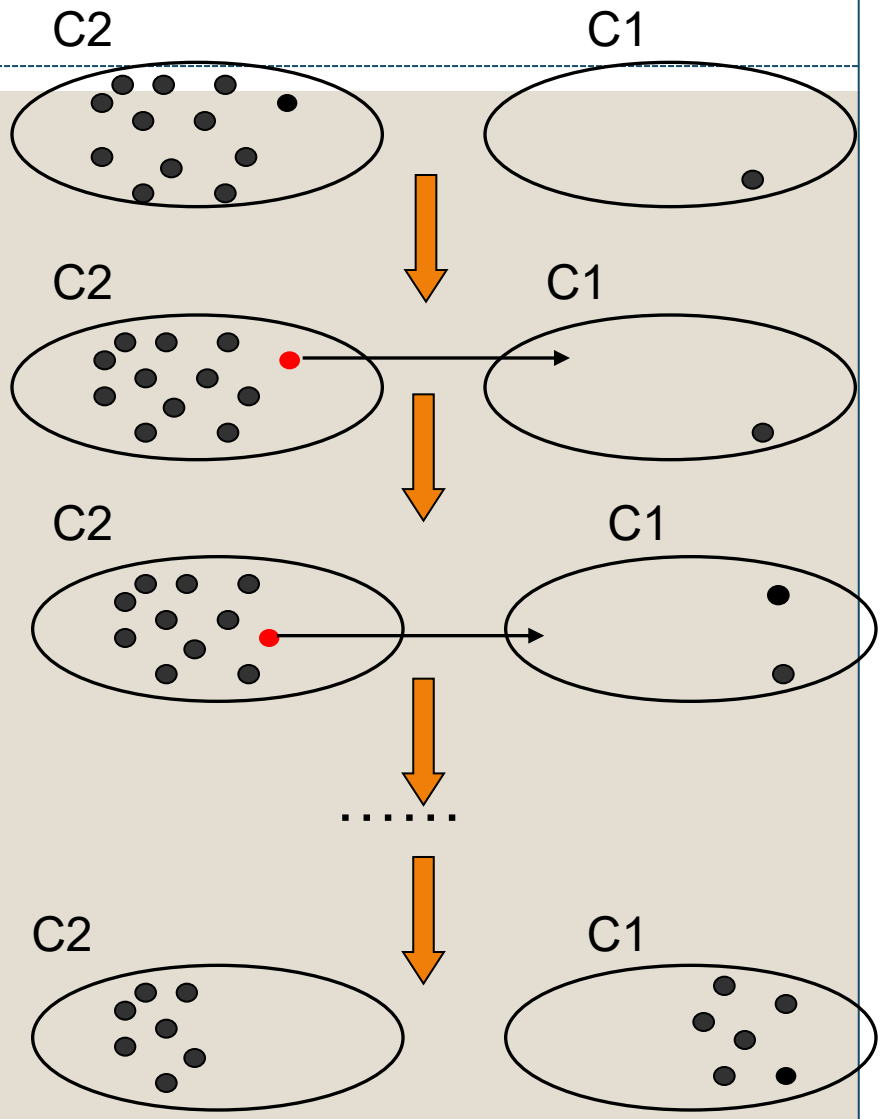
3. For each object Oi in C2, tell whether it is more close to C1 or to other objects in C2

$$D_i = \underset{j \in C_2}{avg}\, d(O_i, O_j) - \underset{j \in C_1}{avg}\, d(O_i, O_j)$$

4. Choose the object O_k with greatest D score.

5. If D_k>0, move Ok from C2 to C1, and repeat 3-5.

6. Otherwise, stop splitting process.



Discussion on Hierarchical Approaches

- **Strengths**
  - Do not need to input k, the number of clusters
- **Weakness**
  - Do not scale well; time complexity of at least $O(n^2)$, where n is total number of objects
  - Can never undo what was done previously

Hierarchical clustering comparison

Agglomerative (bottom up)

1. Start with 1 point (singleton)
2. Recursively add two or more appropriate clusters at each iteration
3. Stop when all the clusters are merged into one big cluster.

Divisive (top down)

1. Start with a big cluster
2. Recursively divide into smaller clusters at each iteration
3. Stop when k number of clusters is achieved.

THANK YOU