CMR
INSTITUTE OF
TECHNOLOGY

USN □□□□□□□□□□

Internal Assesment Test - II

Sub:	Data Warehousing & Data Mining							Code:	13MCA442
Date:	09.05.2017	Duration:	90 mins	Max Marks:	50	Sem:	IV	Branch:	MCA

Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1(a)	What is anti monotone property of support? Explain Apriori Algorithm with example.	[10]	CO4	L2
2(a)	Explain FP Growth algorithm with example.	[10]	CO4	L2
3(a)	What is Rule based classifier? Explain Sequential Covering Algorithm.	[10]	CO3	L2
4(a)	Explain the property of Rule Based classifier.	[5]	CO3	L2
(b)	What is nearest neighbor classifier? Explain.	[5]	CO3	L2
5(a)	Explain Hunt Algorithm for decision tree induction.	[5]	CO4	L2
(b)	What are node impurity measures? Explain with example.	[5]	CO3	L2
6(a)	Explain types of clusters.	[10]	CO3	L2
7(a)	What is k means algorithm for clustering? Explain bisect k means algorithm also.	[6+4]	CO4	L2
8(a)	Explain the alternative method of item sets generation.	[10]	CO3	L2

Course Outcomes		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1:	Describe the designing of Data Warehousing so that it can be able to solve the root problems.	1	-	3	2	-	-	3	3
CO2:	Understanding of the value of data mining in solving real-world problems.	2	2	3	2	-	-	2	3
CO3:	Understanding of foundational concepts underlying data mining.	2	3	3	1	-	-	3	3
CO4:	Understanding of algorithms commonly used in data mining tools.	2	-	3	2	-	-	3	3
CO5:	To develop further interest in research and design of new Data Mining techniques.	1	-	2	-	-	-	3	3

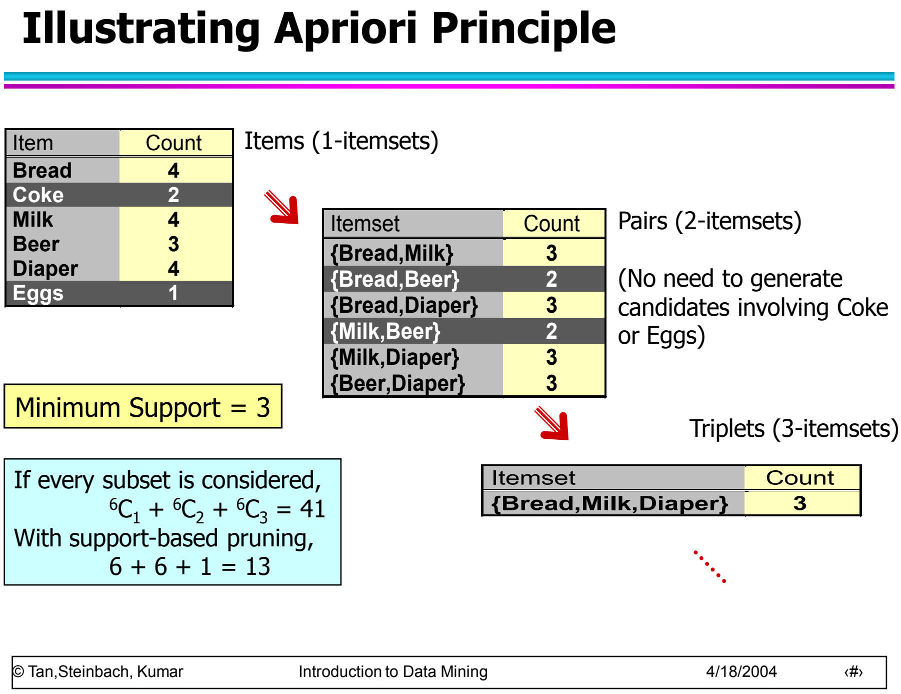
Cognitive level	KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PO1 - Apply *knowledge*; PO2 - *Problem analysis*; PO3 - *Design/development of solutions*; PO4 - team work; PO5 - *Ethics*; PO6 - Communication; PO7- *Business Solution*; PO8 – Life-long learning;

Q1. What is anti monotone property of support. Explain Apriori Algorithm with example.

Ans. Apriori principle:

- If an itemset is frequent, then all of its subsets must also be frequent
    - Apriori principle holds due to the following property of the support measure: $\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$
- Support of an itemset never exceeds the support of its subsets
- This is known as the anti-monotone property of support



Illustrating Apriori Principle

Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
$^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
6 + 6 + 1 = 13

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Apriori Algorithm

Method:

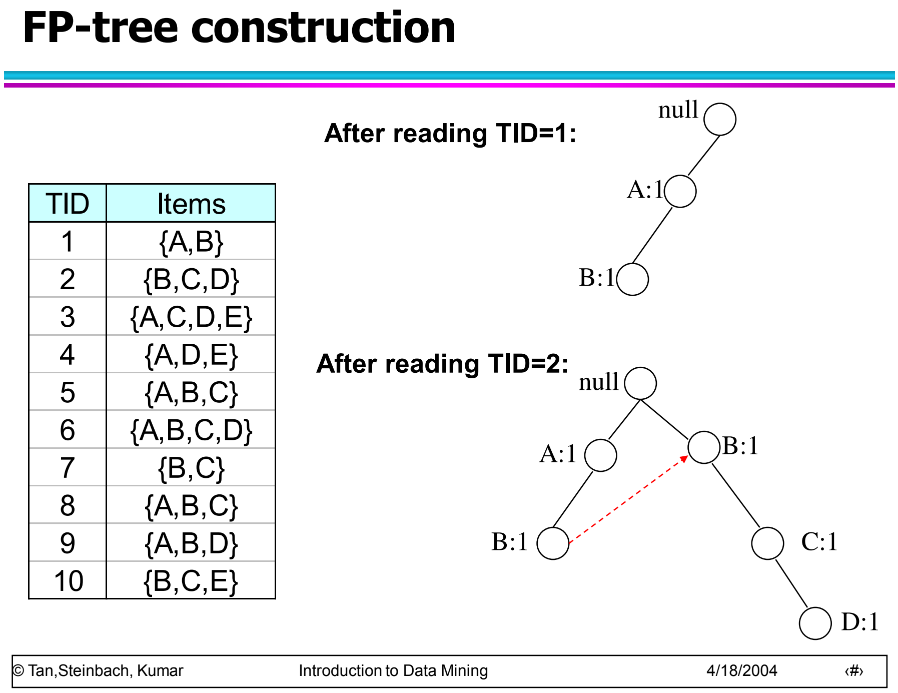
- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Q2. Explain FP Growth algorithm with example.

Ans:

FP-growth Algorithm

- Use a compressed representation of the database using an FP-tree
- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets



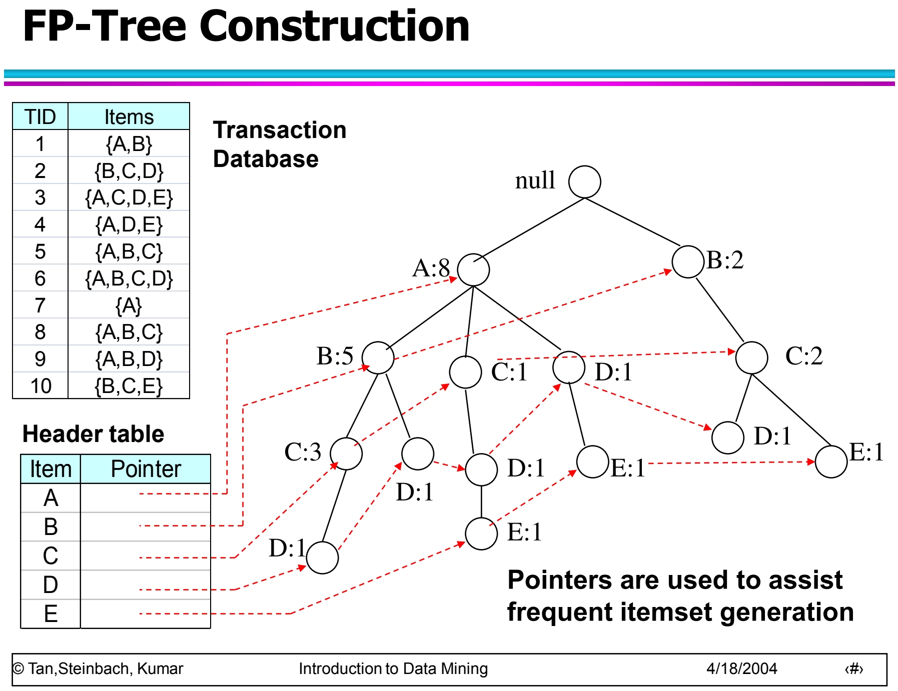
FP-tree construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:

After reading TID=2:

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



FP-Tree Construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

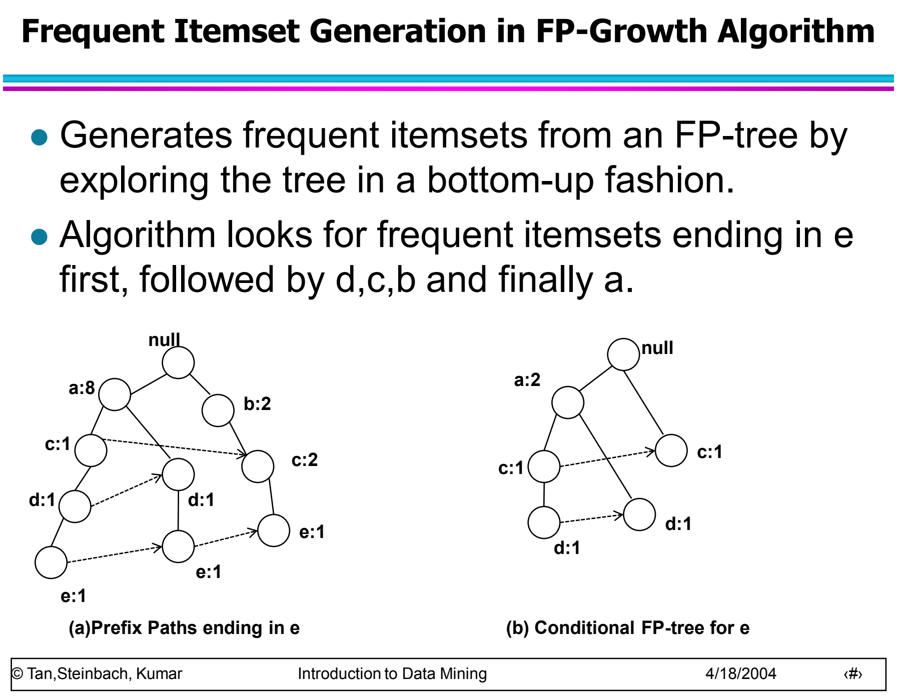
Transaction Database

Header table

Item	Pointer
A	
B	
C	
D	
E	

Pointers are used to assist frequent itemset generation

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



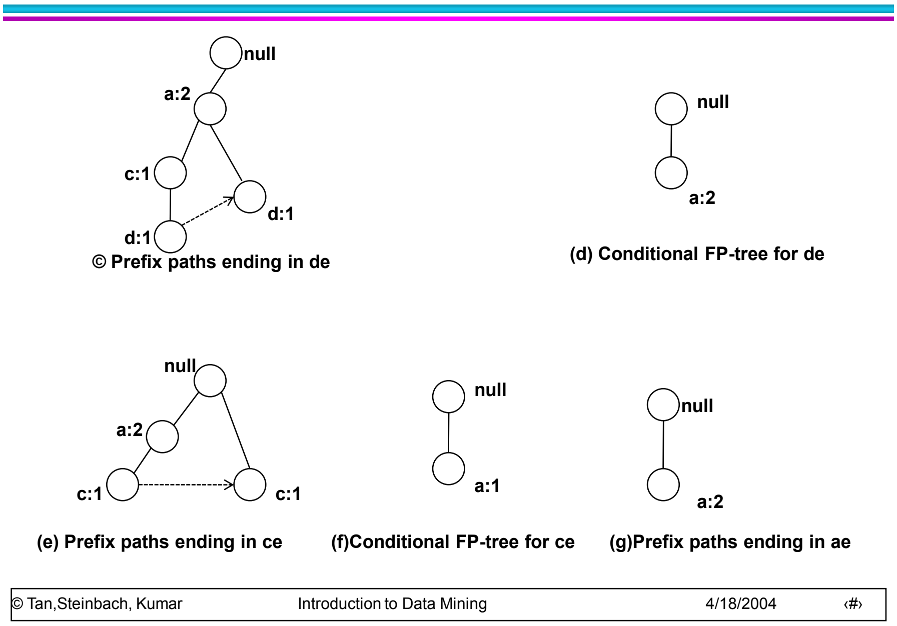
Frequent Itemset Generation in FP-Growth Algorithm

- Generates frequent itemsets from an FP-tree by exploring the tree in a bottom-up fashion.
- Algorithm looks for frequent itemsets ending in e first, followed by d,c,b and finally a.

(a)Prefix Paths ending in e

(b) Conditional FP-tree for e

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



© Prefix paths ending in de

(d) Conditional FP-tree for de

(e) Prefix paths ending in ce

(f)Conditional FP-tree for ce

(g)Prefix paths ending in ae

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

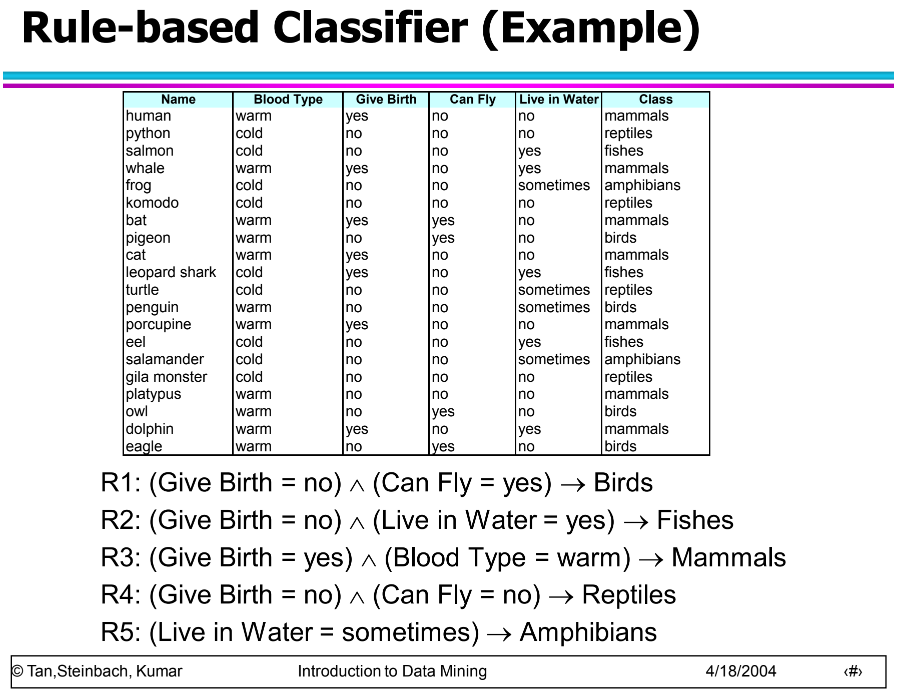
- Used divide and conquer strategy to split the problem into smaller subproblems.
- Gather all the paths(prefix paths) containing node e.
- Count the support for e. Assume that the minimum support count is 2, {e} is declared a frequent itemset because its support count is 3.
- Solve the subproblems of finding frequent itemsets ending in de, ce, be, and ae.
- Convert the prefix paths into conditional FP-tree.
- Support counts along the prefix paths must be updated.
- The prefix paths are truncated by removing the nodes for e.
- Ignore the infrequent item from subsequent analysis.

- FP-growth uses the conditional FP-tree for e to solve the subproblems of finding frequent itemset ending in de,ce, and ae
- To find the frequent itemsets ending in de, gather the prefix from the conditional FP-tree for e.
- Obtain the support count for{d,e} by adding the frequency counts associated with node d .
- Support count is 2 so {d,e} is declared a frequent itemset.
- Construct the conditional FP-tree for de.
- Update the support count and remove the infrequent item c.
- Tree contains only one item a, whose support count is equal to minsup, extracts the frequent itemset {a,d,e}
- Generate frequent itemsets ending in ce.
- Only {c,e} is found to be frequent.
- Then generate next frequent item ending in ae which is {a,e}

Q3. What is Rule based classifier? Explain Sequential Covering Algorithm.

Ans: **Rule-Based Classifier**

- Classify records by using a collection of "if…then…" rules
    - Rule: (*Condition*) → *y*
  - where
      - *Condition* is a conjunctions of attributes
      - *y* is the class label
  - *LHS*: rule antecedent or precondition
  - *RHS*: rule consequent
  - Examples of classification rules:
      - (Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) → Birds
      - (Taxable Income < 50K) $\wedge$ (Refund=Yes) → Evade=No



Rule-based Classifier (Example)

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

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

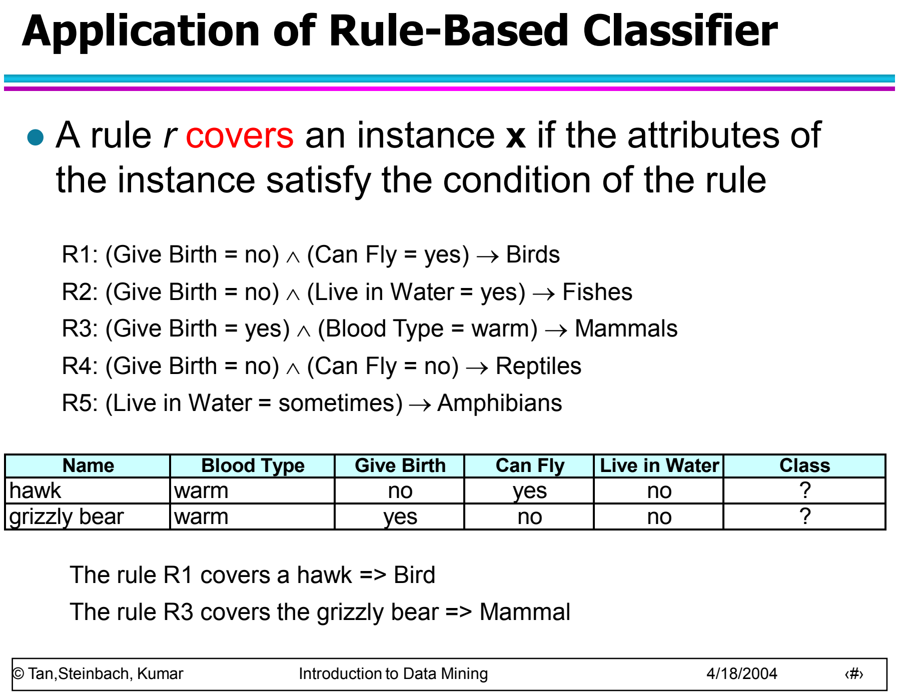
R2: (Give Birth = no) $\wedge$ (Live in Water = yes) → Fishes

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

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

R5: (Live in Water = sometimes) → Amphibians

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



Application of Rule-Based Classifier

- A rule *r* covers an instance **x** if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → Amphibians

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

The rule R1 covers a hawk => Bird
The rule R3 covers the grizzly bear => Mammal

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Direct Method: Sequential Covering

Extracts the rules one class at a time for data sets having more than two classes. Criterion for choosing class depend on the factor such as class prevalence.

1. Start from an empty decision list, R.
2. Grow a rule using the Learn-One-Rule function
3. Add the new rule to the bottom of the decision list
4. Remove training records covered by the rule
5. Repeat Step (2) and (3) until stopping criterion is met

Sequential Covering Algorithm:

```
Let E be the training records and A be the set of attribute-value pairs, {(Aj,Vj)}.
Let Yo be an ordered set of classes {y1,y2,….yk}
Let R={ } be the initial rule list.
for each class y E Yo – {yk} do
    while stopping condition is not met do
        r ← Learn-One-Rule(E,A,y).
        Remove training records from E that are covered by r.
        Add r to the bottom of the rule list: R->RVr.
    end while
end for
Insert the default rule, { }->yk, to the bottom of the rule list R
```

Q4a. Explain the property of Rule Based classifier.

Ans:

Characteristics of Rule-Based Classifier

Mutually exclusive rules

- Classifier contains mutually exclusive rules if the rules are independent of each other

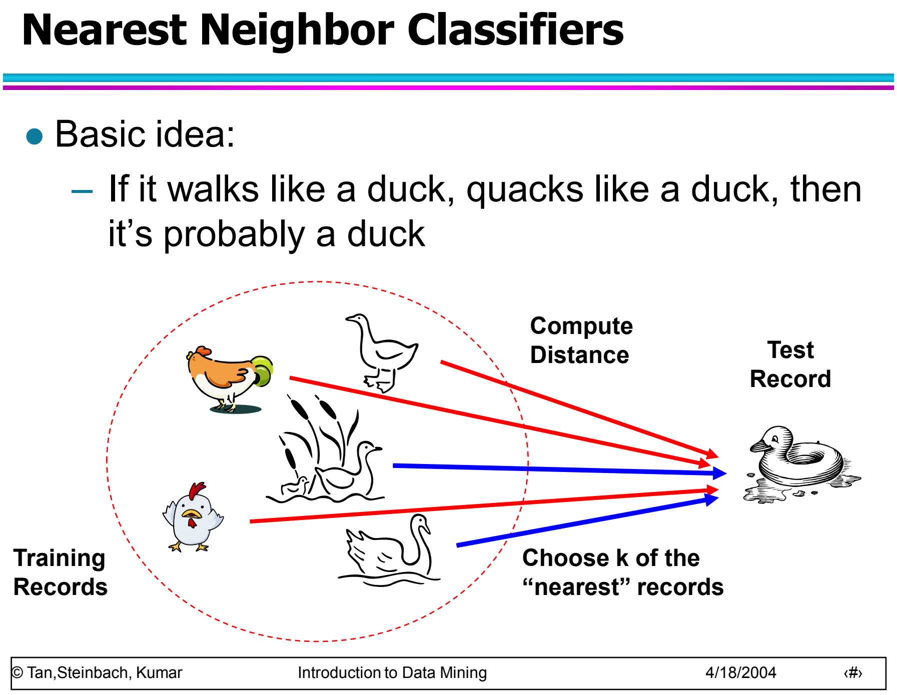
- Every record is covered by at most one rule

Exhaustive rules

- Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
- Each record is covered by at least one rule

Q4b. What is nearest neighbor classifier? Explain.

Nearest Neighbor Classifiers:



Nearest Neighbor Classifiers

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

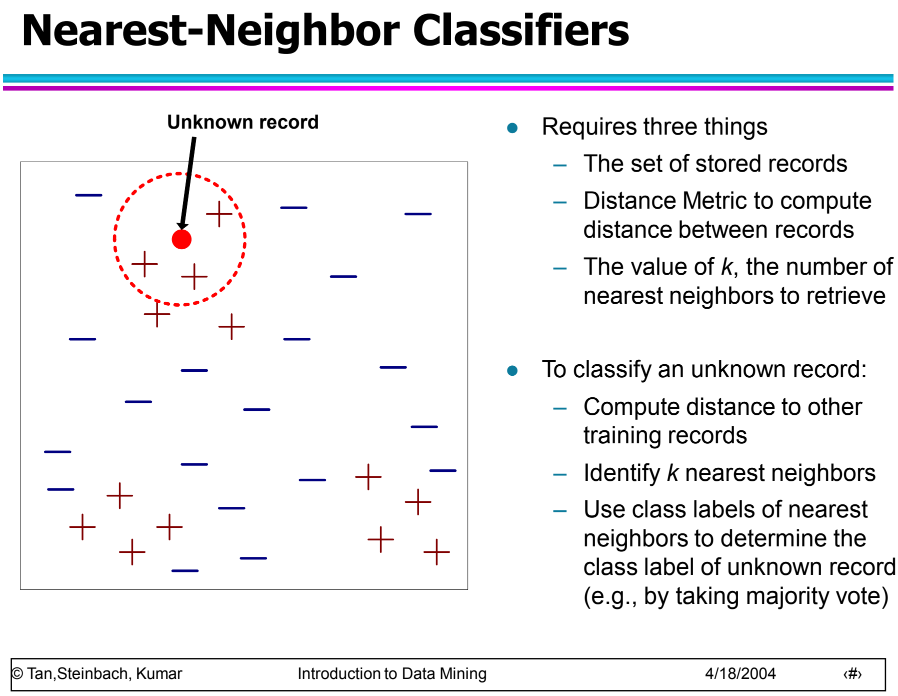
Compute Distance

Test Record

Training Records

Choose k of the "nearest" records

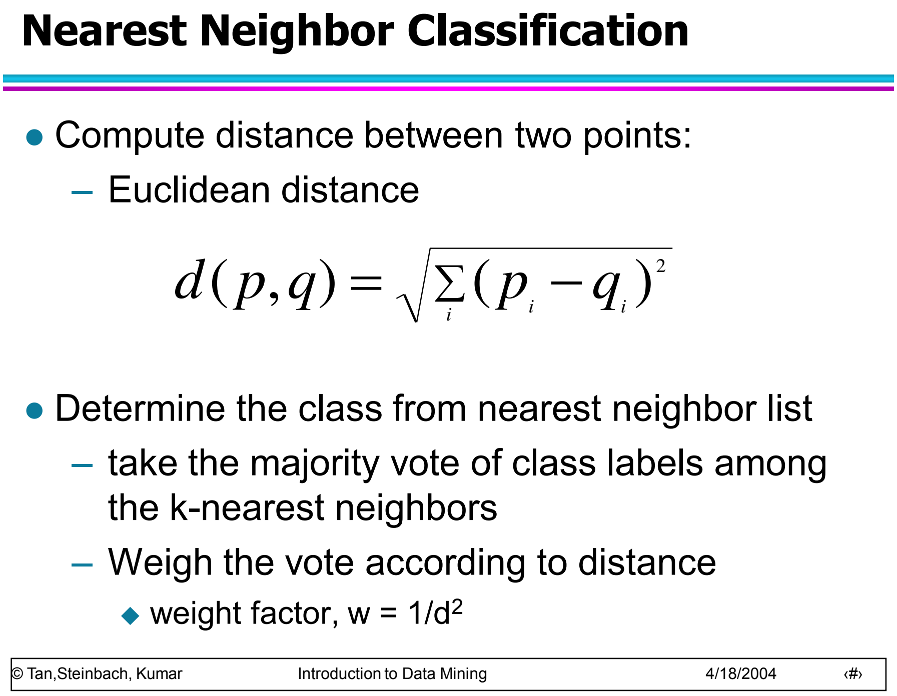
© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



Nearest-Neighbor Classifiers

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

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



Nearest Neighbor Classification

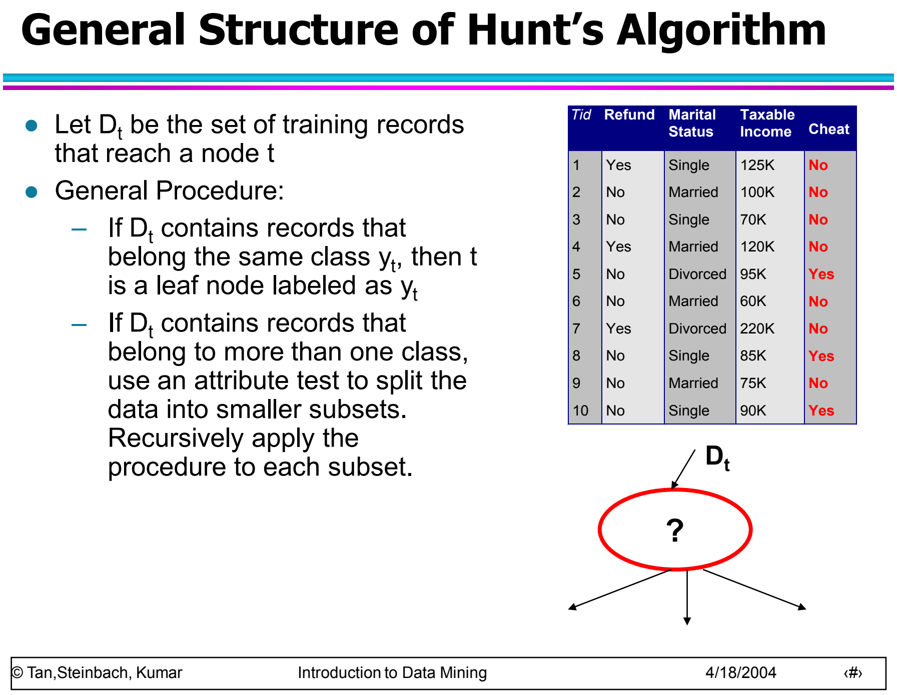
- Compute distance between two points:
  - Euclidean distance

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

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

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Q5a. Explain Hunt Algorithm for decision tree induction.
Ans:

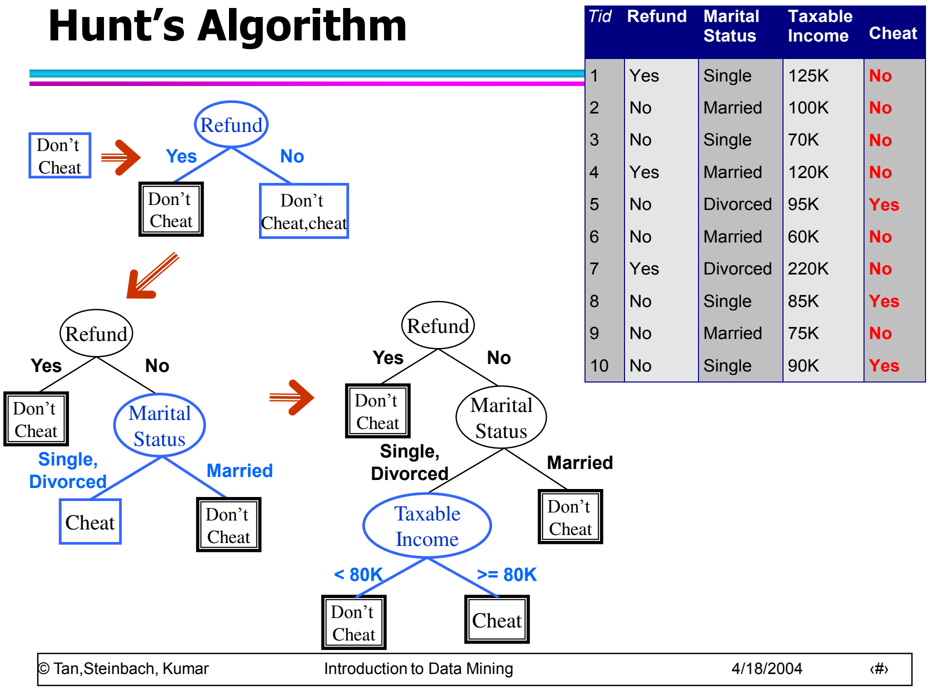


General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

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

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



Hunt's Algorithm

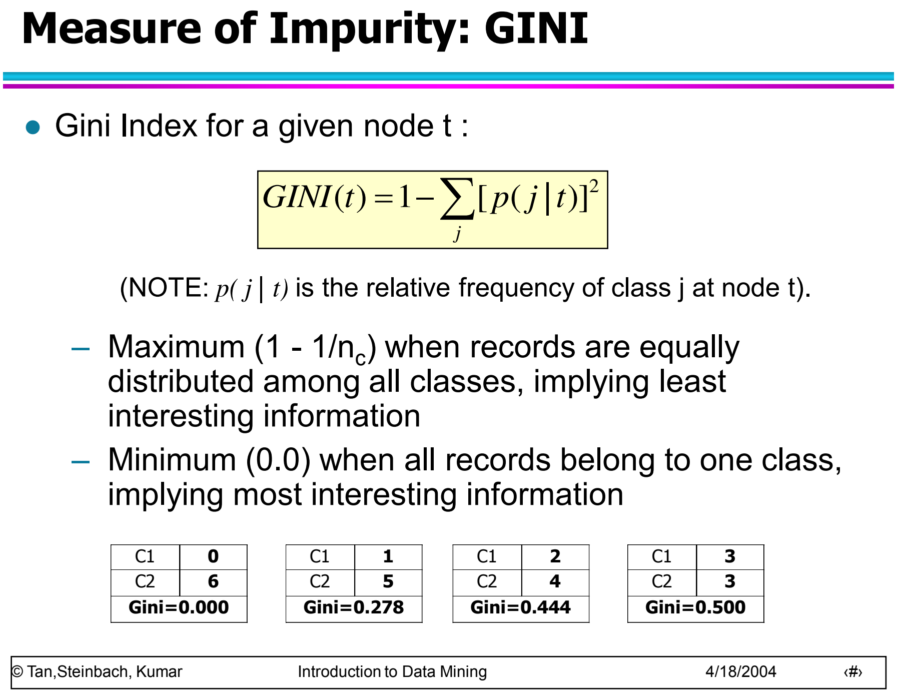
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Q5b. What are node impurity measures? Explain with example.

Ans: **Measures of Node Impurity**

- Gini Index
- Entropy
- Misclassification error



Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

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

– Maximum $(1 - 1/n_c)$ when records are equally distributed among all classes, implying least interesting information

– Minimum (0.0) when all records belong to one class, implying most interesting information

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Examples for computing GINI

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

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

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

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

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

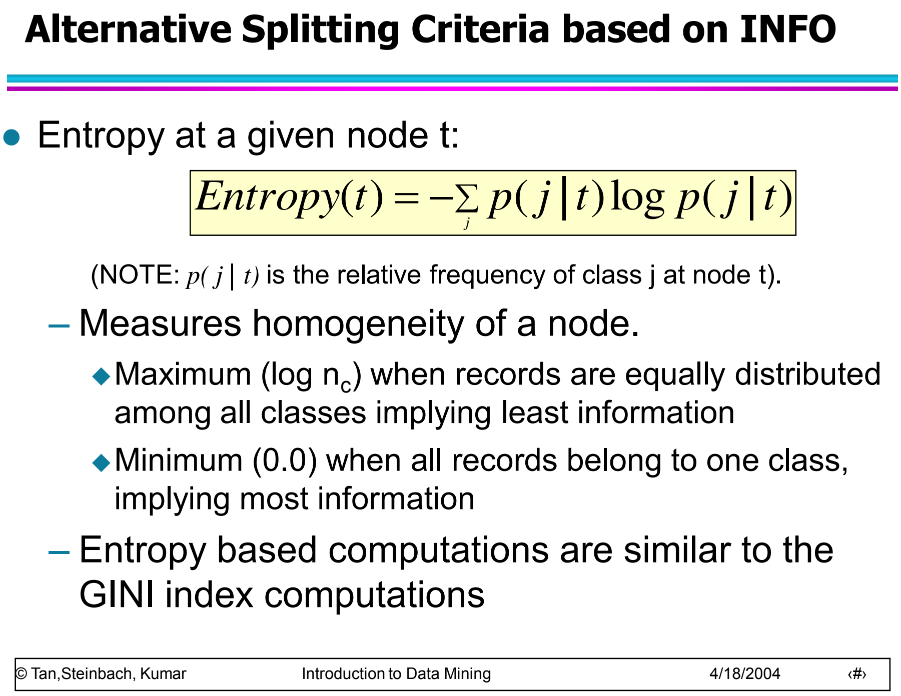
Splitting Based on GINI

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

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

where, n_i = number of records at child i,

n = number of records at node p.



Alternative Splitting Criteria based on INFO

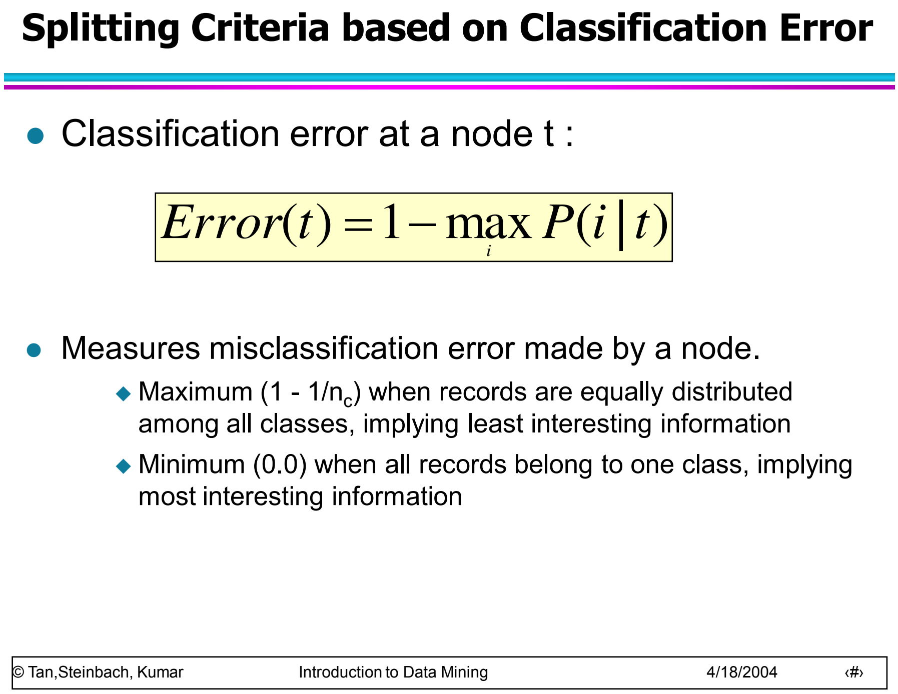
- Entropy at a given node t:

$$Entropy(t) = -\sum_{j} p(j \mid t) \log p(j \mid t)$$

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

- Measures homogeneity of a node.
  - Maximum (log n_c) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›



Splitting Criteria based on Classification Error

- Classification error at a node t :

$$Error(t) = 1 - \max_{i} P(i \mid t)$$

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

© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Q6. Explain types of clusters.

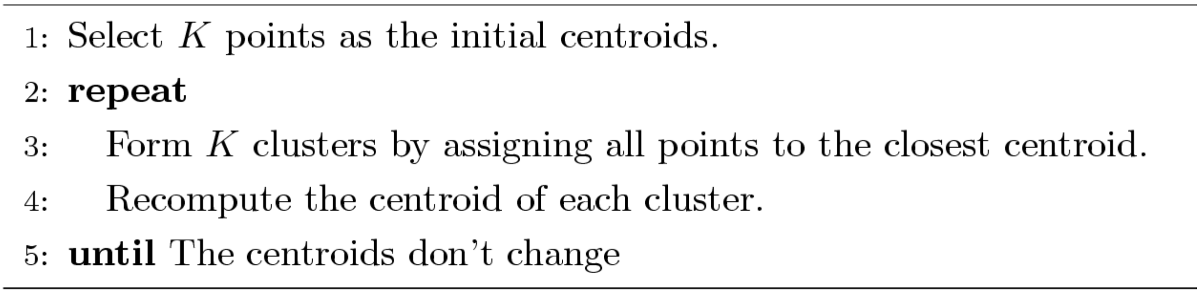
Ans: Types of Clusters:

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

- Well-Separated Clusters:
    - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.
- Center-based
    - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
    - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster
- Contiguous Cluster (Nearest neighbor or Transitive)
    - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.
- Density-based
    - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
    - Used when the clusters are irregular or intertwined, and when noise and outliers are present.
- Shared Property or Conceptual Clusters
    - Finds clusters that share some common property or represent a particular concept.

Q7. What is k means algorithm for clustering? Explain bisect k means algorithm also.
Ans : K means Algorithm:

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple



```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

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

n = number of points,
K=number of clusters,
I = number of iterations,

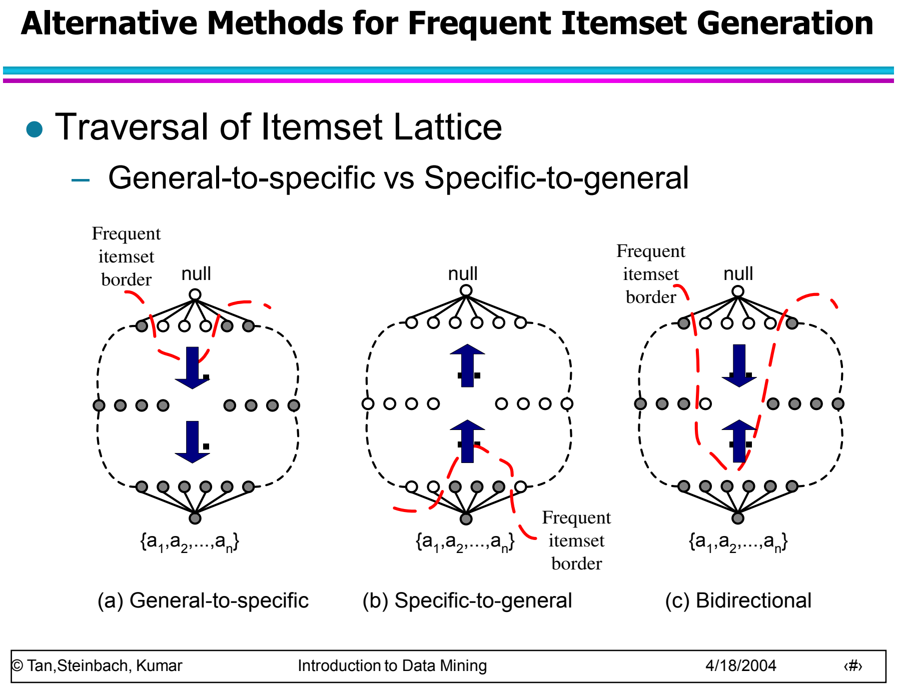
d = number of attributes

Bisecting K-means algorithm

– Variant of K-means that can produce a partitional or a hierarchical clustering

```
Initialize the list of clusters to contain the cluster containing all points.
repeat
    Select a cluster from the list of clusters
    for $i = 1$ to $number\_of\_iterations$ do
        Bisect the selected cluster using basic K-means
    end for
    Add the two clusters from the bisection with the lowest SSE to the list of clusters.
until Until the list of clusters contains $K$ clusters
```

Q8. Explain the alternative method of item sets generation.



Alternative Methods for Frequent Itemset Generation

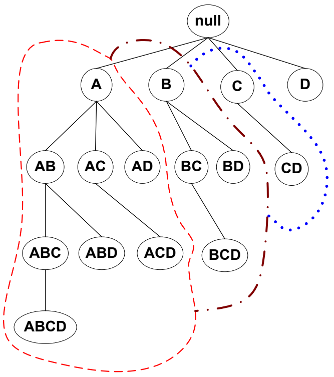
- Traversal of Itemset Lattice
  - General-to-specific vs Specific-to-general

(a) General-to-specific (b) Specific-to-general (c) Bidirectional

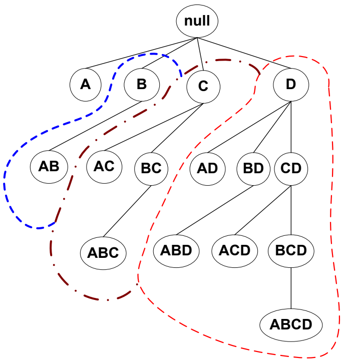
© Tan,Steinbach, Kumar Introduction to Data Mining 4/18/2004 ‹#›

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
  - Equivalent Classes



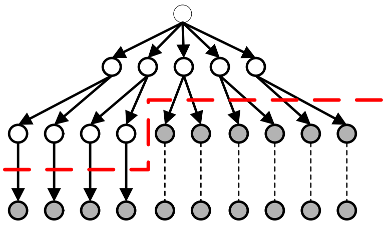
(a) Prefix tree



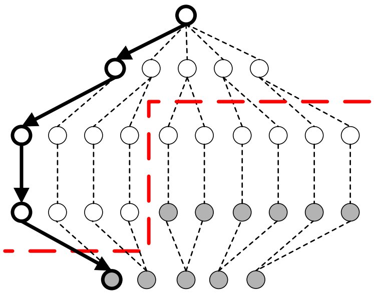
(b) Suffix tree

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
  - Breadth-first vs Depth-first



(a) Breadth first



(b) Depth first

Alternative Methods for Frequent Itemset Generation

- Representation of Database
  - horizontal vs vertical data layout

Horizontal
Data Layout

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

Vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				