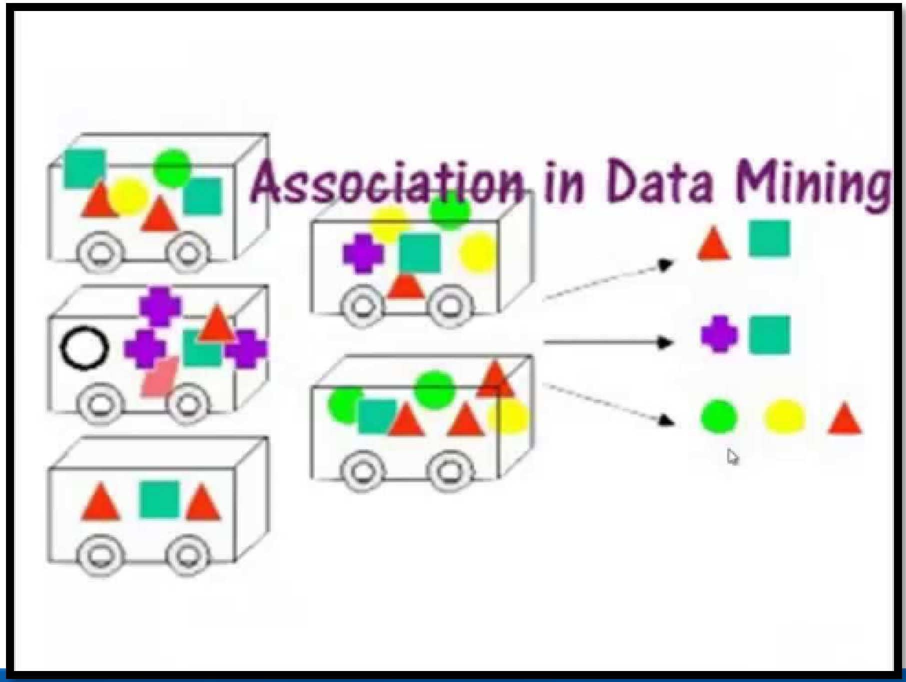
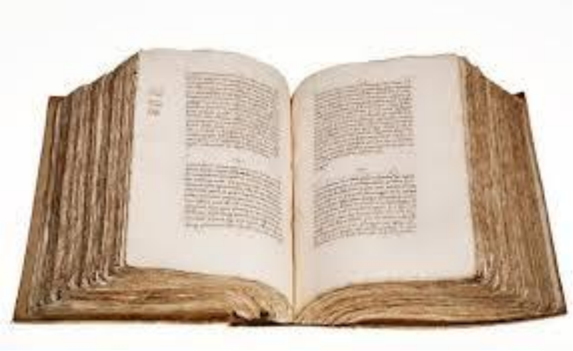


Association Rule Mining

Zareen Alamgir

Book Chapters to Read

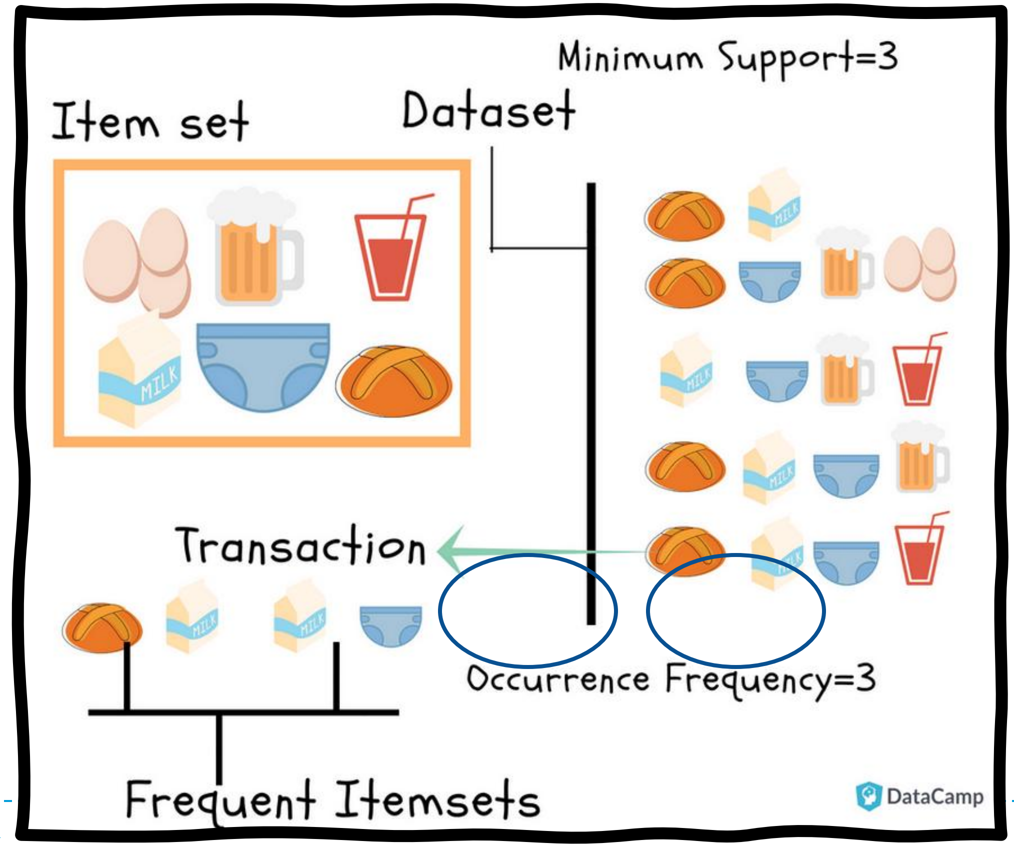
- Book Mining of Massive Dataset
  - Chapter 6: Frequent itemsets



- Book Introduction to Data Mining
  - Chapter 6: Basic Association rule Mining

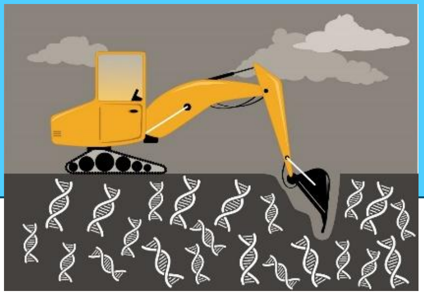
Frequent Pattern Analysis

- **Frequent pattern**
  - a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set



What products were often purchased together?

Frequent Pattern Analysis

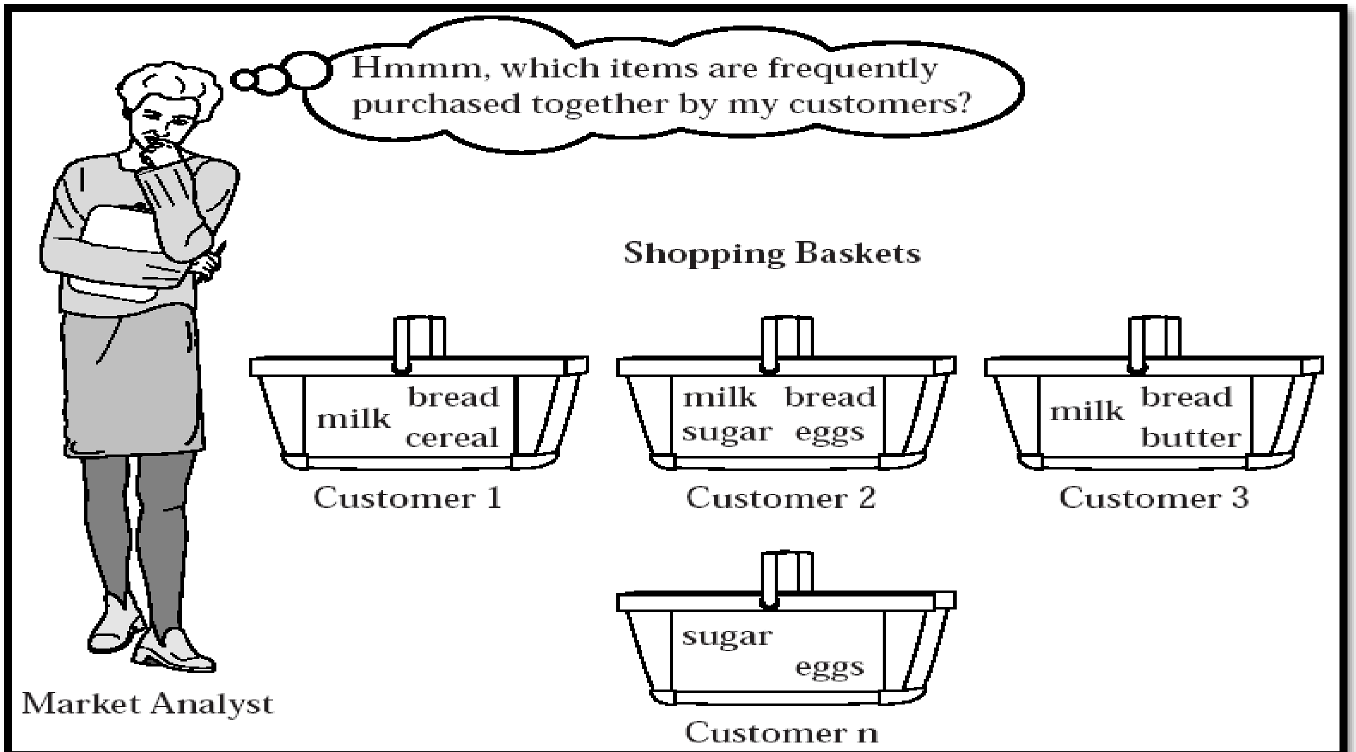


Motivation: Finding inherent regularities in data

- **Recommender Systems**
  - discover patterns in user behavior and preferences to make personalized recommendations.
- **Fraud Detection**
  - identify abnormal patterns of behavior that may indicate fraudulent activity.
- **Network Intrusion Detection**
  - detect patterns of network activity that may indicate a security threat.
- **Medical Analysis:**
  - identify patterns in data that may indicate a particular disease or condition.
  - find what kinds of DNA are sensitive to the new drug?

Association Mining

- Association rule mining
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items in data



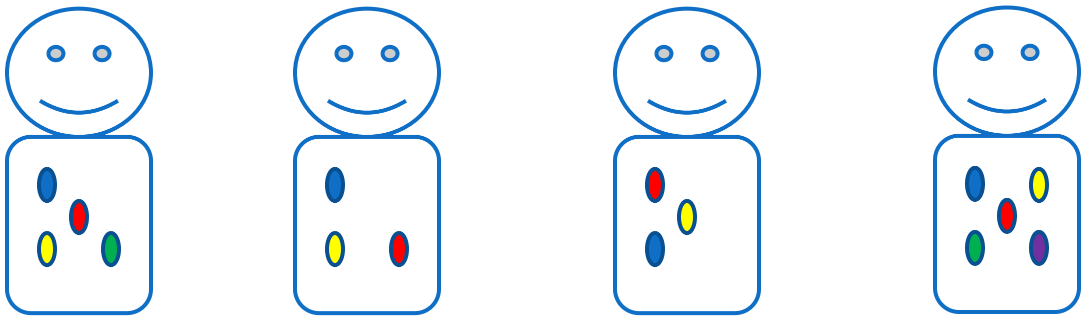
Application

- Items = products
- Baskets = sets of products someone bought in one trip to the store

- Suppose many people buy cereal and diapers together
  - Run a sale on diapers; raise price of cereal
  - Only useful if many buy cereal & diapers

Application

- Baskets = patients
- Items = drugs and side effects



- If patients
  - are taking the same combination of drugs and
  - facing same side effects
- then its probably caused by that combination.

Application

- Baskets = sentences
- Items = documents containing those sentences

S1:Data Mining helps.	S2: Data Mining…	S3:Data Mining ….
Doc1	Doc1	Doc1
Doc2	Doc2	Doc2
Doc3	Doc4	Doc5

- Items that appear together too often could represent plagiarism
- ***Notice items do not have to be "in" baskets***

Application

- Baskets = Web pages
- Items = words.

Webpage 1	Webpage 2	Webpage 3
Data Mining	Data Mining	Data Mining
Data Analysis	Visualization	Warehousing
Map Reduce	Map Reduce	Map Reduce

- Co-occurrence of relatively rare words, e.g., “Nawaz Sharif” and “Imran Khan,” may indicate an interesting relationship

Association Rule Mining

Given a set of transactions, ***find rules*** that will *predict the occurrence of an item* based on the *occurrences of other items* in the transaction

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Cereal, Eggs
3	Milk, Diaper, Cereal, Coke
4	Bread, Milk, Diaper, Cereal
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} $\rightarrow$ {Cereal},

{Milk, Bread} $\rightarrow$ {Eggs, Coke},

{Cereal, Bread} $\rightarrow$ {Milk},

Definition: Frequent Itemset

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Cereal, Eggs
3	Milk, Diaper, Cereal, Coke
4	Bread, Milk, Diaper, Cereal
5	Bread, Milk, Diaper, Coke

- **Itemset**
  - **A collection of one or more items**
    - E.g. {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - *An itemset with support greater than or equal to a min support threshold*

Definitions

- **Association Rule**
  - An implication X → Y, where X and Y are itemsets
  - Example: {Bread, Milk} → {Eggs}
- **Rule Evaluation Metrics**
  - **Support, s :**
    - *Fraction of transactions that contain both X and Y*
    - probability that a transaction contains X ∪ Y

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Cereal, Eggs**
3	**Milk, Diaper, Cereal, Coke**
4	**Bread, Milk, Diaper, Cereal**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow Cereal$$

$$\boxed{s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Cereal})}{|\text{T}|} = \frac{2}{5} = 0.4}$$

Interesting Association Rules

- Not all high-confidence rules are interesting
  - The rule **X** $\rightarrow$ **milk** may have high confidence for many itemsets **X**, because milk is just purchased very often (independent of **X**)
- Interesting rules are those with high positive or negative *interest values*

Association Rule Mining

- Goal: Find rules with high support/confidence
- How to compute?
  - Support: Find sets of items that occur frequently
  - *Confidence: Find frequency of subsets of supported itemsets*

- *If we have all frequently occurring sets of items (frequent itemsets), we can compute support and confidence!*

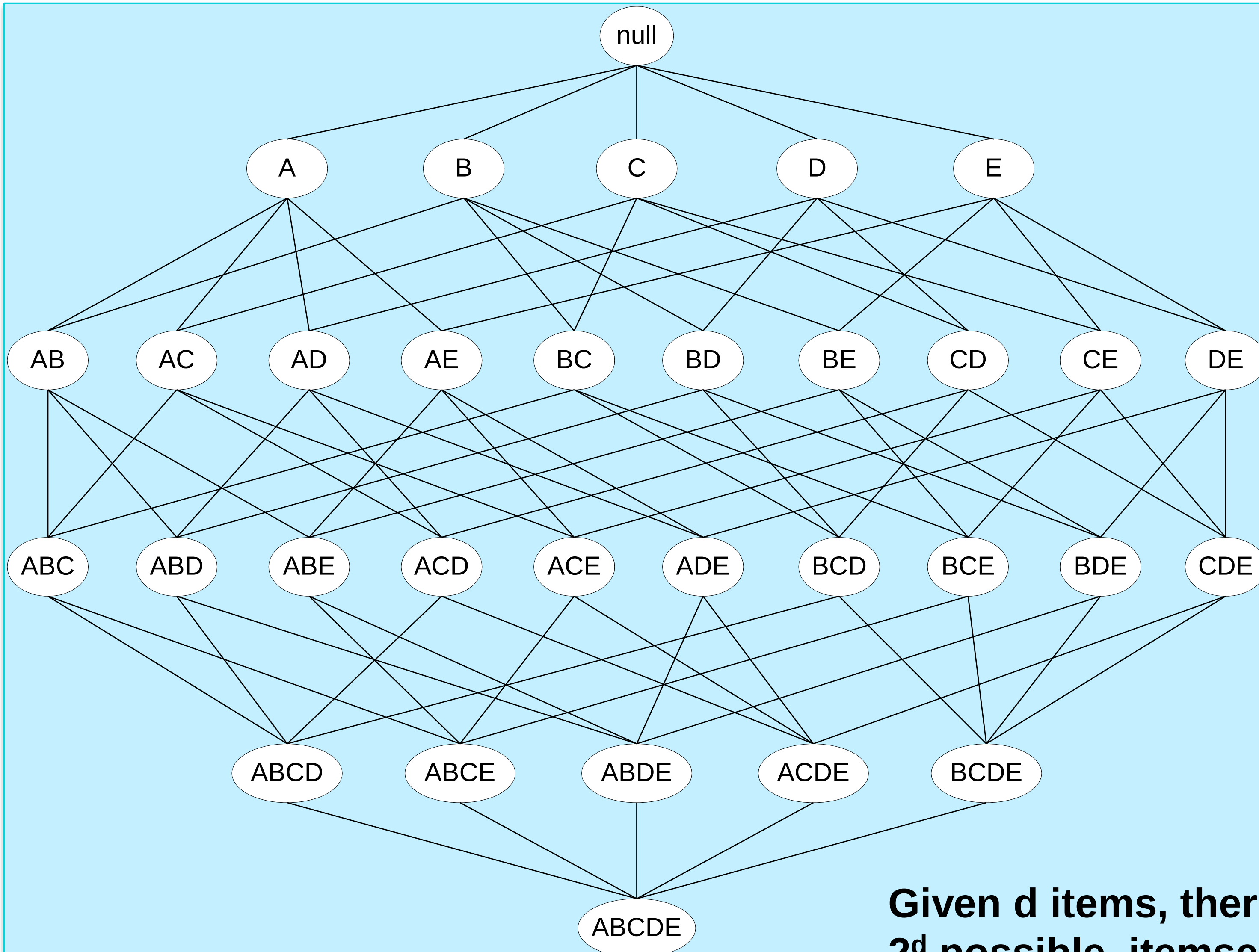
Mining Frequent Itemsets task

- **Input:** A set of transactions T, over a set of items I
- **Output:** All itemsets with items in I having
  - support ≥ *minsup* threshold

- Problem parameters:
  - N = |T|: number of transactions
  - d = |I|: number of (distinct) items
  - w: maximum width of a transaction
  - Number of possible itemsets?
    - $M = 2^d$

- Scale of the problem:
  - WalMart sells 100,000 items and can store billions of baskets.
  - Web has billions of words and many billions of pages.

The itemset lattice



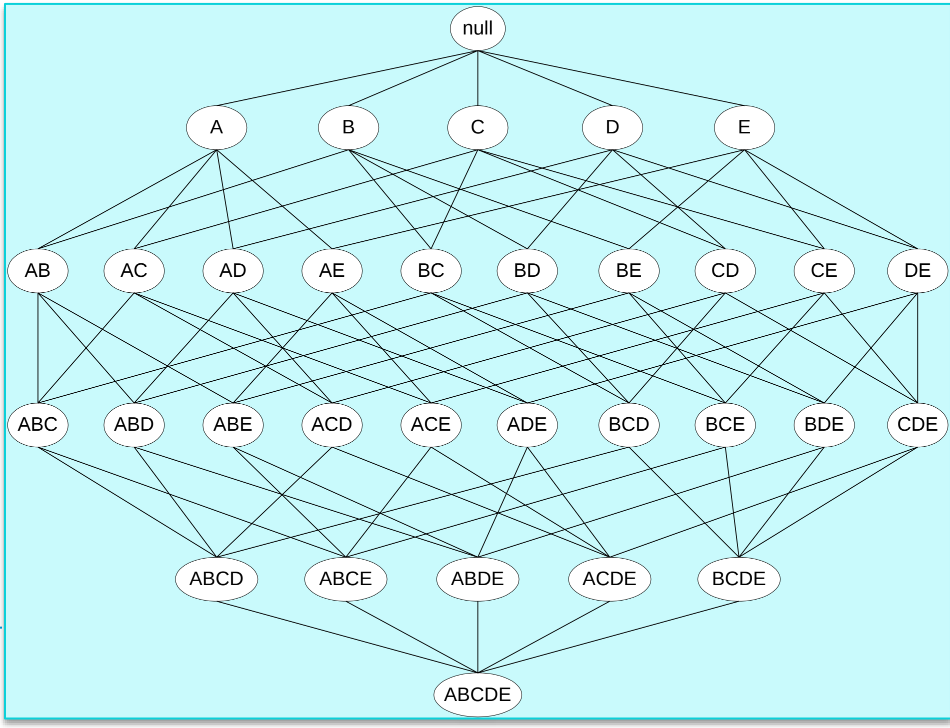
Given d items, there are 2^d possible itemsets

Naïve Algorithm 1

- Brute-force approach
  - Each itemset is a candidate
  - Count the support of each candidate by scanning the data
- Time Complexity ~ $O(N\ 2^d\ w)$,
- Space Complexity ~ $O(2^d)$

where
N = no of transactions
d = no of (distinct) items
w = max width of a transaction

Expensive 2^d !!!



A Naïve Algorithm 2

- Scan the data and for each transaction
  - generate all possible itemsets.
  - Keep a count for each itemset in the data.
- Time Complexity ~ $O(N2^w)$
- Space Complexity ~ $O(2^d)$

Expensive 2^d !!!

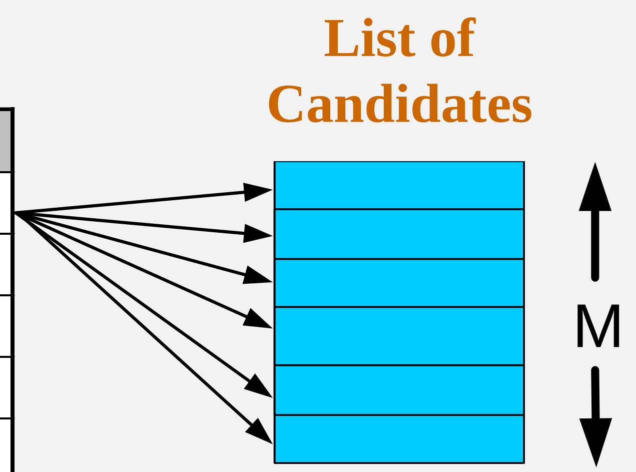
where
- N = no of transactions
- d = no of (distinct) items
- w = max width of a transaction

Transactions

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

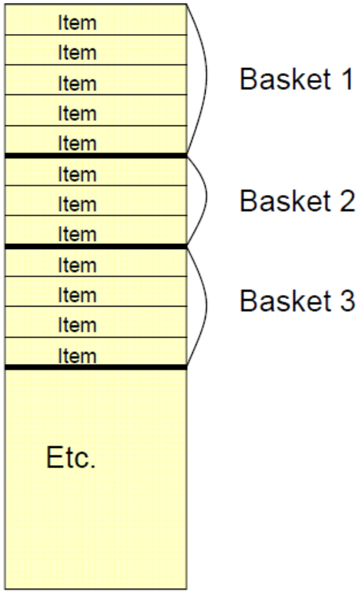
N

w



Computation Model

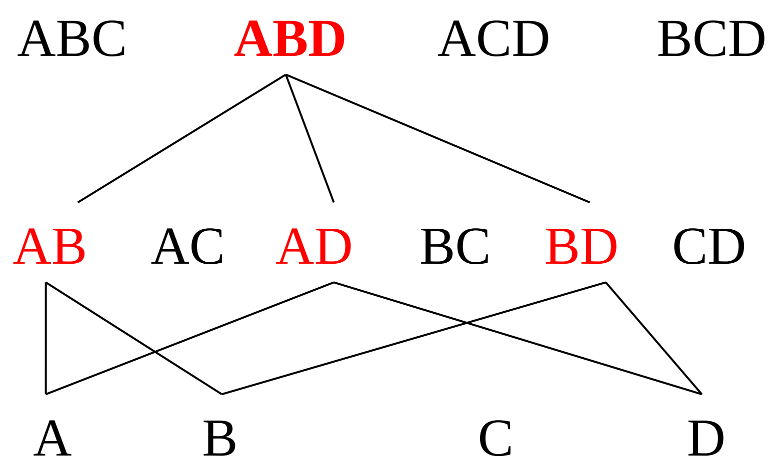
- Typically, data is kept in flat files and stored on disk
- The true cost of mining disk-resident data is usually the number of disk I/O's.
- In practice, association-rule algorithms read the data in passes – all itemsets read in turn.
- Thus, we measure the cost by the number of passes an algorithm takes.



Example: items are positive integers, and boundaries between baskets are –1.

Apriori Algorithm

- **Apriori property**: any subset of a frequent itemset must be frequent
  - if **{cereal, diaper, nuts}** is frequent, so is **{cereal, diaper}**
  - Every transaction having {cereal, diaper, nuts} also contains {cereal, diaper}



Apriori Algorithm

- **Apriori pruning principle:**
  - If there is any itemset which is infrequent, its superset should not be generated/tested!
- Method:
  - Generate length (k+1) candidate itemsets from length k frequent itemsets, and
  - Test the candidates against DataBase
- Performance studies show its efficiency and scalability

Illustration of the Apriori principle

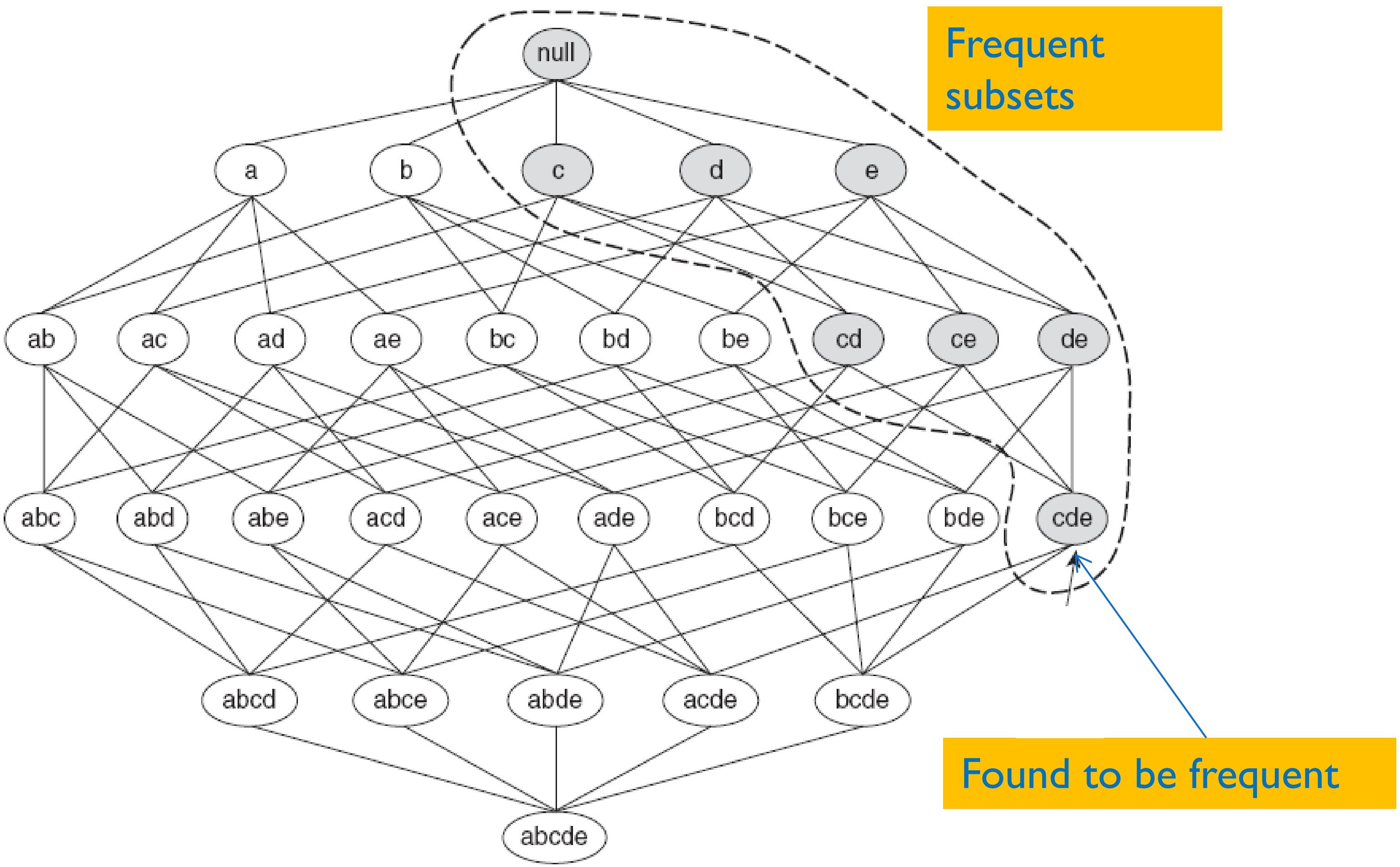
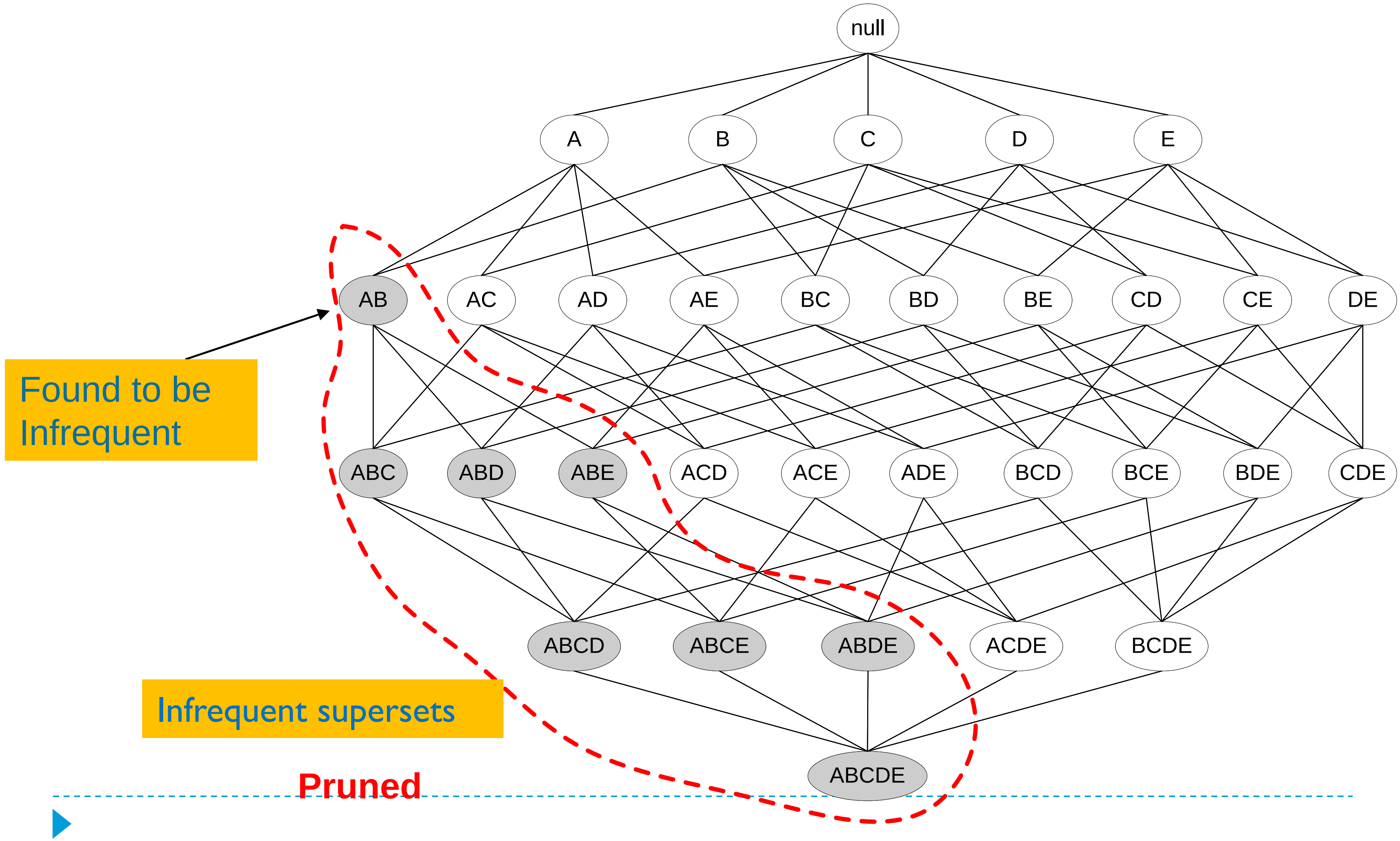


Figure 6.3. An illustration of the *Apriori* principle. If $\{c, d, e\}$ is frequent, then all subsets of this itemset are frequent.

Illustration of the Apriori principle



The Apriori algorithm

Level-wise approach

C_k = candidate itemsets of size k
L_k = frequent itemsets of size k

Frequent itemset generation

Candidate generation

1. $k = 1$, C_1 = all items
2. While C_k not empty
3. Scan the database to find which itemsets in C_k are frequent and put them into L_k
4. Use L_k to generate a collection of candidate itemsets C_{k+1} of size $k+1$
5. $k = k+1$

R. Agrawal, R. Srikant: "Fast Algorithms for Mining Association Rules", *Proc. of the 20th Int'l Conference on Very Large Databases*, 1994.

Apriori principle

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

minsup = 3

Items (1-itemsets)

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Cereal	**3**
Diaper	**4**
Eggs	**1**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Cereal}	**2**
{Bread,Diaper}	**3**
{Milk,Cereal}	**2**
{Milk,Diaper}	**3**
{Cereal,Diaper}	**3**

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**2**

If every subset is considered,

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 6 + 15 + 20 = 41$$

With support-based pruning,

$$\binom{6}{1} + \binom{4}{2} + 1 = 6 + 6 + 1 = 13$$

Only this triplet has all subsets to be frequent

But it is below the minsup threshold

Candidate Generation

- Basic principle (Apriori):
  - An itemset of size k+1 is candidate to be frequent only if all of its subsets of size k are known to be frequent
- Main idea:
  - Construct a candidate of size k+1 by combining frequent itemsets of size k
    - If **k = 1**, take the all pairs of frequent items
    - If **k > 1**, join pairs of itemsets that *differ by just one item*
    - For each generated candidate itemset ensure that all subsets of size k are frequent.

The Apriori Algorithm—An Example

Min-support =2

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1^{st} scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2^{nd} scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3^{rd} scan

L_3

Itemset	sup
{B, C, E}	2

The Apriori Algorithm—An Example

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

L_3

Itemset	sup
{B, C, E}	2

Some Rules

A → C , C → A

B → C, C → B

B → E, E → B

BC → E

CE → B

BE → C

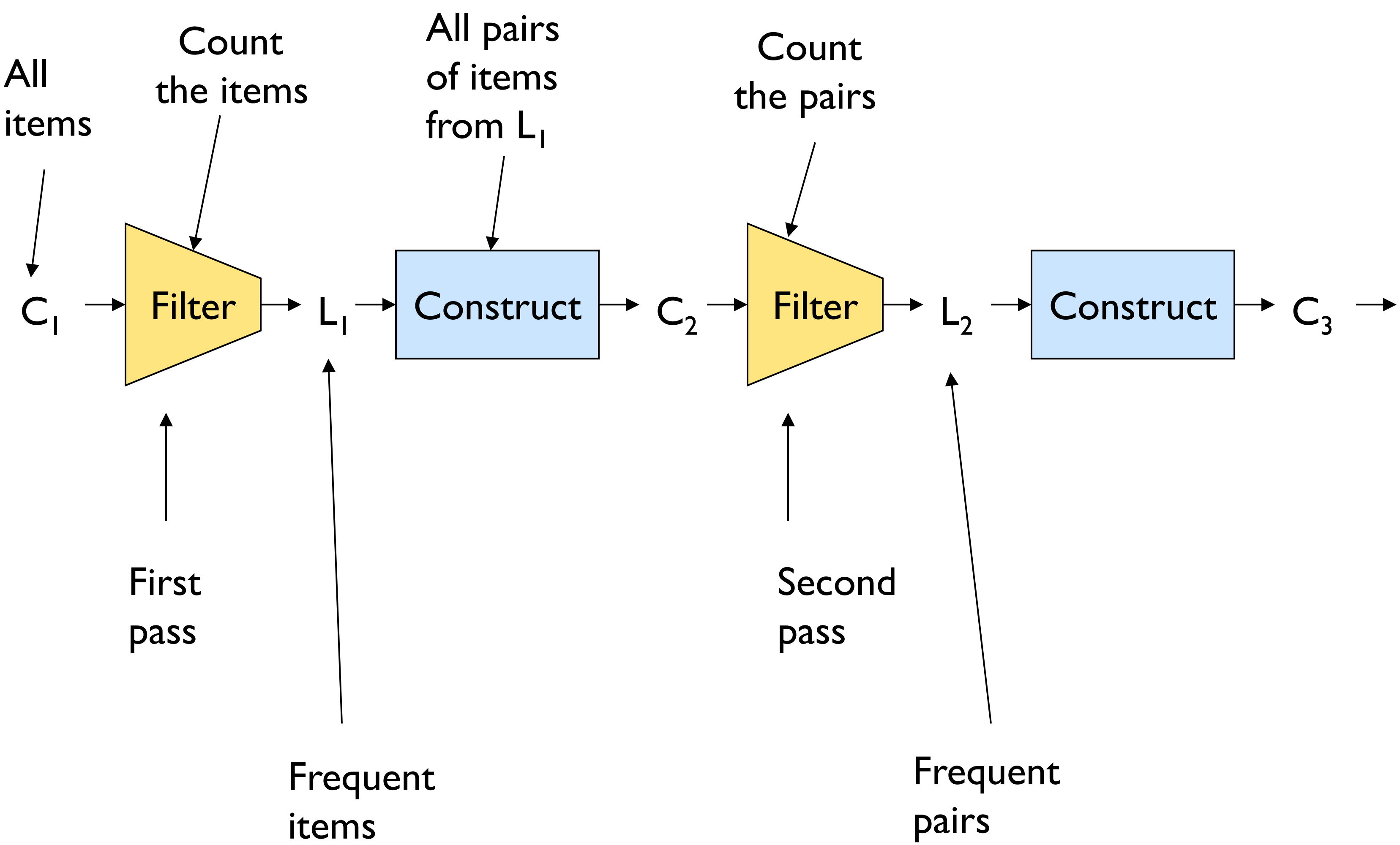
Frequency ≥ 50%, Confidence 100%:

A → C

B → E, E → B

BC → E

CE → B


All
items
Count
the items
All pairs
of items
from L_1
Count
the pairs
C_1
Filter
L_1
Construct
C_2
Filter
L_2
Construct
C_3
First
pass
Second
pass
Frequent
items
Frequent
pairs

Generate Candidates C_{k+1}

- Assumption: The items in an itemset are ordered
  - if integers ordered in increasing order,
  - if strings ordered in lexicographic order
  - The order ensures that if item y > x appears before x, then x is not in the itemset
- The items in L_k are also listed in an order

Create a candidate itemset of size k+1, by joining two itemsets of size k, that share the first k-1 items

Item 1	Item 2	Item 3
a	b	c
a	b	e
a	d	e

Generate Candidates C_{k+1}

- Assumption: The items in an itemset are ordered
  - E.g., if integers ordered in increasing order, if strings ordered in lexicographic order
    - The order ensures that if item y > x appears before x, then x is not in the itemset
- The items in L_k are also listed in an order

Create a candidate itemset of size k+1, by joining two itemsets of size k, that share the first k-1 items

Item 1	Item 2	Item 3
a	b	c
a	b	e
a	d	e

a	b	c	e

Generate Candidates C_{k+1}

- Assumption: The items in an itemset are ordered
  - E.g., if integers ordered in increasing order, if strings ordered in lexicographic order
    - The order ensures that if item y > x appears before x, then x is not in the itemset
- The items in L_k are also listed in an order

Create a candidate itemset of size k+1, by joining two itemsets of size k, that share the first k-1 items

Item 1	Item 2	Item 3
a	b	c
a	b	e
a	d	e

Are we missing something? What about this candidate?

a	b	d	e

Generate Candidates C_{k+1}

- Are we done? Are all the candidates valid?

Item 1	Item 2	Item 3
a	b	c
a	b	e
a	d	e

a	b	c	e

Is this a valid candidate?

No. Subsets (a, c, e) and (b, c, e) should also be frequent

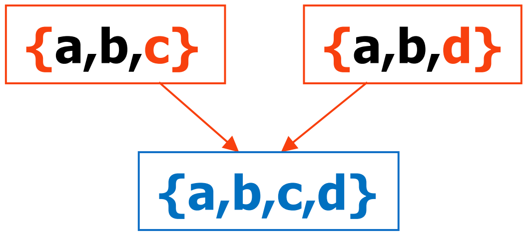
- Pruning step:
  - For each candidate (k+1)-itemset create all subset k-itemsets
  - Remove a candidate if it contains a subset k-itemset that is not frequent

Apriori principle

Example 2

- L_3={abc, abd, acd, ace, bcd}
- **Generate Candidates**
  - abcd from abc and abd
  - acde from acd and ace

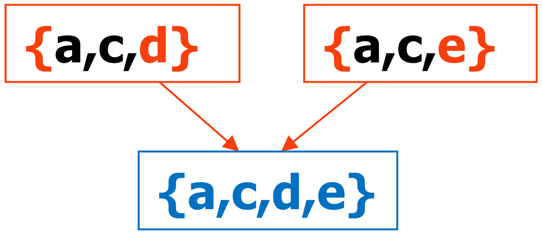
item1	item2	item3
a	b	c
a	b	d
a	c	d
a	c	e
b	c	d



Example 2

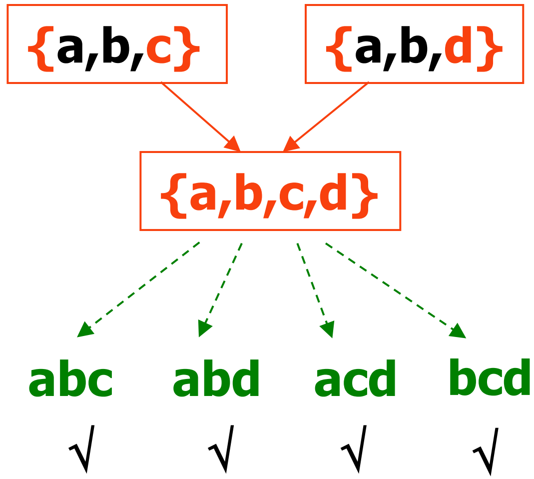
- L_3={abc, abd, acd, ace, bcd}
- **Generate Candidates**
  - abcd from abc and abd
  - acde from acd and ace

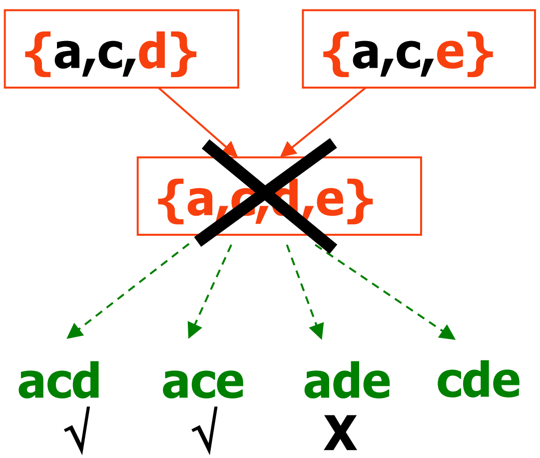
item1	item2	item3
a	b	c
a	b	d
a	c	d
a	c	e
b	c	d



Example 2

- L_3={*abc*, *abd*, *acd*, *ace*, *bcd*}
- **Generate Candidates**
  - ***abcd*** from ***abc*** and ***abd***
  - ***acde*** from ***acd*** and ***ace***
- ***Pruning:***
  - ***abcd*** is kept since all itscsubset itemsets are in L_3
  - ***acde*** is removed because ***ade*** is not in L_3
- C_4={*abcd*}





Generate Candidates C_{k+1}

- We have all frequent k-itemsets L_k
- **Step 1: Generate Candidates L_k**
  - Create set C_{k+1} by joining frequent k-itemsets that share the first k-1 items
- **Step 2: prune**
  - Remove from C_{k+1} the itemsets that contain a subset k-itemset that is not frequent

Apriori Example

Table 6.1: Transactional data for an *AllElectronics* branch.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Minimum support count is 2.

Scan *D* for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Apriori Example

Table 6.1: Transactional data for an *AllElectronics* branch.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Minimum support count is 2.

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Apriori Example

Table 6.1: Transactional data for an *AllElectronics* branch.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Minimum support count is 2.

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Apriori Example

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Association Rule Mining Task

- Input: A set of transactions T, over a set of items I
- Output: All rules with items in I having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

– **Confidence, c :**

  – Measures how often items in Y appear in transactions that contain X

TID	*Items*
1	**Bread, Milk**
2	**Bread, Diaper, Cereal, Eggs**
3	**Milk, Diaper, Cereal, Coke**
4	**Bread, Milk, Diaper, Cereal**
5	**Bread, Milk, Diaper, Coke**

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \textit{Cereal}$$

$$\boxed{s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Cereal})}{|\text{T}|} = \frac{2}{5} = 0.4}$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Cereal})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

- **From Book 'Introduction to Data Mining'**

Mining Association Rules

- Two-step approach:
  1. Frequent Itemset Generation
     - Generate all itemsets whose support $\geq$ minsup
  2. Rule Generation
     - Generate high confidence rules from each frequent itemset, where each rule is a partitioning of a frequent itemset into Left-Hand-Side (LHS) and Right-Hand-Side (RHS)

Frequent itemset: {A,B,C,D}
Rule: AB→CD

Rule Generation

- We have all frequent itemsets, how do we get the rules?
  - For every frequent itemset S, we find rules of the form $L \rightarrow S - L$, where $L \subset S$, that satisfy the minimum confidence requirement
  - Example: $L = \{A,B,C,D\}$
  - Candidate rules:

    $A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$
    $AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BD \rightarrow AC$, $CD \rightarrow AB$,
    $ABC \rightarrow D$, $BCD \rightarrow A$, $BC \rightarrow AD$,
- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

Rule Generation

- How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property
    c(ABC $\rightarrow$D) can be larger or smaller than c(AB $\rightarrow$D)
  - But confidence of rules generated from the same itemset has an anti-monotone property
  - e.g., L = {A,B,C,D}:

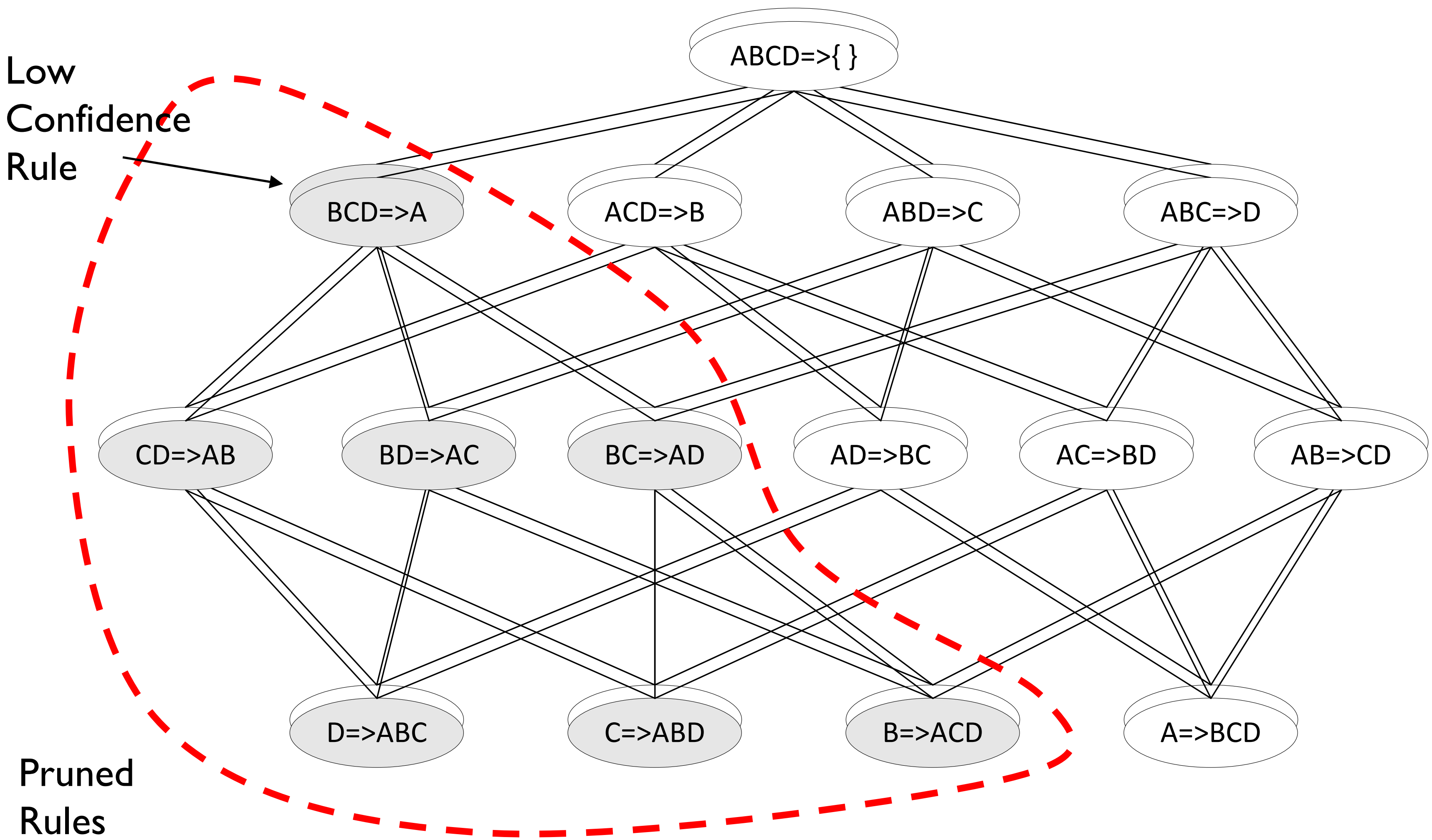
A → B

$$c = \frac{\sigma(A, B)}{\sigma(A)}$$

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

- σ(ABCD)/ σ(ABC) >= σ(ABCD)/ σ(AB) >= σ(ABCD)/ σ(A)
- Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

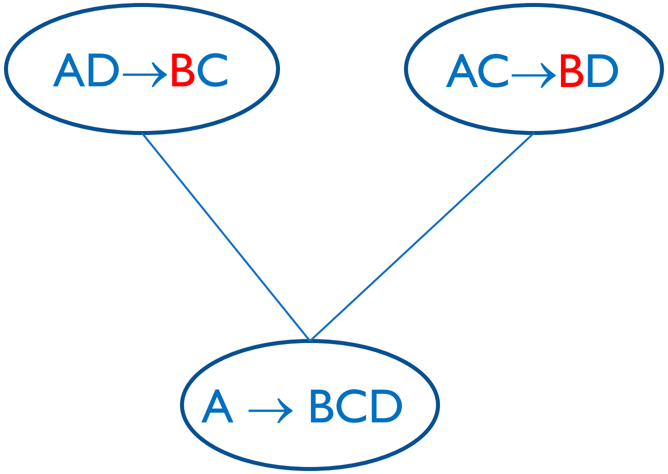
Rule Generation for Apriori Algorithm



Lattice of rules created by the RHS

Rule Generation for Apriori Algorithm

- Candidate rule is generated by merging two rules that share the same prefix in the RHS
- join(AD→BC,AC→BD) would produce the candidate rule A → BCD
- Prune rule A → BCD if its subset AB→CD does not have high confidence
- Essentially we are doing **Apriori** on the RHS



Interestingness Measurements

- Objective measures

  Two popular measurements:
  - *support*
  - *confidence*

- Subjective measures

  A rule (pattern) is interesting if
  - it is *unexpected* (surprising to the user); and/or
  - *actionable* (the user can do something with it)

Computing Interestingness Measure

- Given a rule $X \rightarrow Y$, information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11}: support of X and Y
f_{10}: support of X and $\overline{Y}$
f_{01}: support of $\overline{X}$ and Y
f_{00}: support of $\overline{X}$ and $\overline{Y}$

Used to define various measures

- support, confidence, lift, Gini, Jaccard, etc.

Drawback of Confidence

	Coffee	Not Coffee	
Tea	15	5	20
Not Tea	75	5	80
	90	10	100

The pitfall of confidence can be traced to the fact that measure ignores the support of the itemset in the rule consequent.

Association Rule: Tea $\rightarrow$ Coffee

Confidence X->Y = Support (X,Y)/ Support X

Confidence= P(Coffee|Tea) = 15/20=0.75

but P(Coffee) = 0.9

$\Rightarrow$ Although confidence is high, rule is misleading

$\Rightarrow$ P(Coffee|NotTea) = 75/80 = 0.9375

Example: Lift/Interest

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.9

Lift = $\dfrac{c(X\text{->}Y)}{s(Y)} = \dfrac{s(X, Y)}{s(X)\,s(Y)}$

$\Rightarrow$ Lift = 0.75/0.9= 0.8333 (< 1, therefore is negatively associated)

For binary variables lift is equivalent to another objective measure **Interest Factor**

Lift or Interest Factor

$$\text{Lift} = \frac{c(X \to Y)}{s(Y)} = \frac{s(X, Y)}{s(X)\, s(Y)}$$

- If Lift = 1 then X and Y are independent
- If Lift > 1 then X and Y are positively correlated
- If Lift < 1 then X and Y are negatively correlated

Drawback Lift & Interest

	Y	$\bar{Y}$	
X	10	0	10
$\bar{X}$	0	90	90
	10	90	100

	Y	$\bar{Y}$	
X	90	0	90
$\bar{X}$	0	10	10
	90	10	100

$$Lift = \frac{0.1}{(0.1)(0.1)} = 10$$

$$Lift = \frac{0.9}{(0.9)(0.9)} = 1.11$$

$$\text{Lift} = \frac{c(X\text{->}Y)}{s(Y)} = \frac{s(X, Y)}{s(X)\,s(Y)}$$

Statistical independence: If $P(X,Y)=P(X)P(Y)$ => Lift = 1

X and Y are independent. However lift is positive > 1 …

Other Measures

- Correlation
- Conviction
- IS measure (asymmetric)
  - equivalent to Cosine measure for binary variables
- Jaccard
- Interest
- Gini Index

There are lots of measures proposed in the literature

Some measures are good for certain applications, but not for others

What criteria should we use to determine whether a measure is good or bad?

What about Apriori-style support based pruning? How does it affect these measures?

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\Big(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\Big)$
13	Conviction (V)	$\max\Big(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\Big)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\Big(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\Big)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B),P(A\|B)-P(A))$

Continuous and Categorical Attributes

Continuous and Categorical Attributes

How to apply association analysis formulation to **non-asymmetric binary variables?**

Session Id	Country	Session Length (sec)	Number of Web Pages viewed	Gender	Browser Type	Buy
1	USA	982	8	Male	IE	No
2	China	811	10	Female	Netscape	No
3	USA	2125	45	Female	Mozilla	Yes
4	Germany	596	4	Male	IE	Yes
5	Australia	123	9	Male	Mozilla	No
…	…	…	…	…	…	…

Example of Association Rule:

$$\{\text{Number of Pages} \in [5,10) \wedge (\text{Browser=Mozilla})\} \rightarrow \{\text{Buy = No}\}$$

Handling Categorical Attributes

- Categorical Attributes:
  - finite number of possible values,
  - no ordering among value
- Transform categorical attribute into asymmetric binary variables
- Introduce a new "item" for each distinct attribute-value pair
  - Example: replace Browser Type attribute with
    - Browser Type = Internet Explorer
    - Browser Type = Mozilla
    - Browser Type = Chrome

Handling Categorical Attributes

- Potential Issues
  - What if attribute has many possible values
    - Example: attribute country has more than 200 possible values
    - Many of the attribute values may have very low support
    - ***Potential solution:*** Aggregate the low-support attribute values

Replace less frequent attribute values into category called others

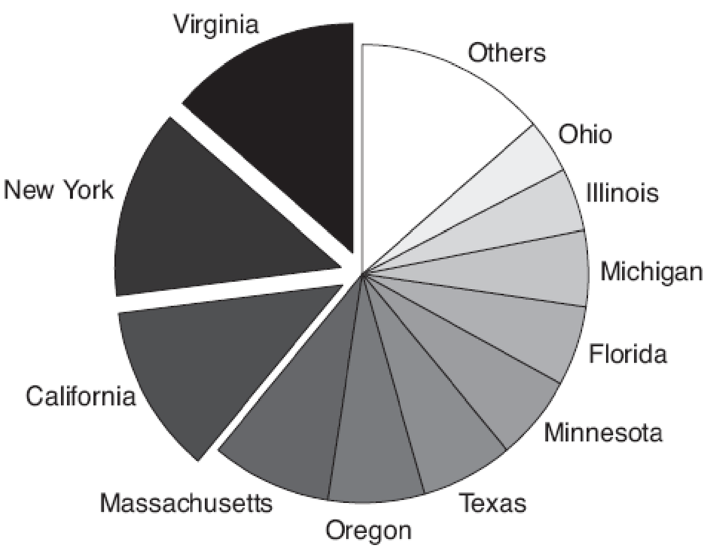


Figure 7.1. A pie chart with a merged category called `Others`.

Handling Categorical Attributes

- Potential Issue: What if distribution of attribute values is highly skewed

  - Example: In an online survey, we collected information regarding attributes gender, education, state, computer at home, chat online, shop online and privacy concern.
    - 85 % of the participant have computer at home
      - {Computer at home =yes, shop Online =yes} ->{ privacy concerns = yes}
      - Better: {shop Online =yes} ->{ privacy concerns = yes}

  - ***Potential solution:*** drop the highly frequent items

Handling Continuous Attributes

- Different kinds of rules:
  - $Age \in [21,35) \wedge Salary \in [70k,120k) \rightarrow Buy$
  - $Salary \in [70k,120k) \wedge Buy \rightarrow Age: \mu=28, \sigma=4$

- Different methods:
  - Discretization-based
    - Equal-width binning
    - Equal-depth binning
    - Clustering
  - Statistics-based

Discretization Issues

- Size of the discretized intervals affect support & confidence

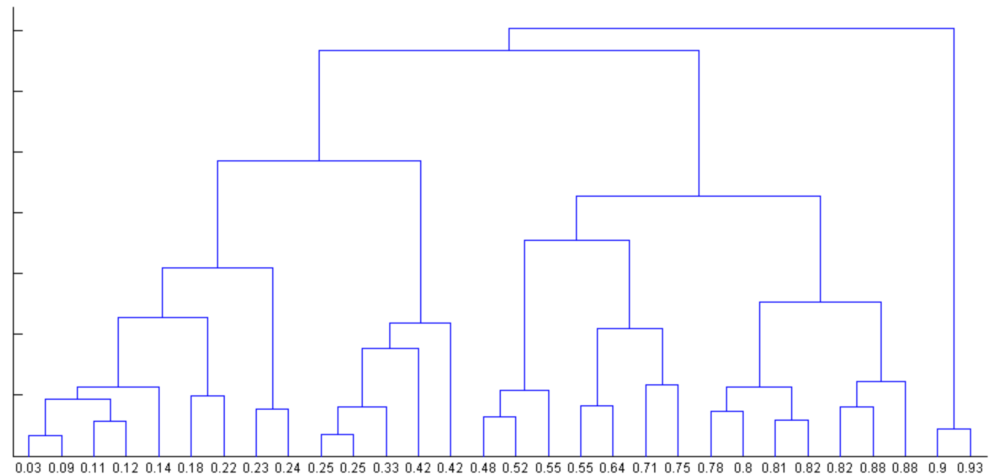
  Age $\in$ [16,24) -> chat online = yes (s=8.8%, c=81.5%)

  Age $\in$ [44,60) -> chat online = no (s=16.8%, c=70%)

  - If intervals too small: may not have enough support
    - Age $\in$ [16,24) -> chat online = yes (s=4.4%, c=84.7%)
    - Age $\in$ [20,24) -> chat online = no (s=4.3%, c=78.3%)
  - If intervals too large: may not have enough confidence
    - Age $\in$ [12,36) -> chat online = yes (s=30%, c=57.7%)
    - Age $\in$ [44,60) -> chat online = no (s=28%, c=58.3%)
- Potential solution: use all possible intervals

Discretization Issues

- Execution time
  - If intervals contain n values, there are on average $O(n^2)$ possible ranges



- Too many rules (redundant rules)
  - $\{Age \in [16,20) \wedge$ gender =male $\} \rightarrow$ chat online = yes
  - $\{Age \in [16,24) \wedge$ gender = male$\} \rightarrow$ chat online = yes