UNIVERSITY of LOUISIANA LAFAYETTE

CSCE566-DATA MINING
WEEK 2

Frequent Itemset Mining

Min Shi
min.shi@louisiana.edu

Sep 4, 2024

Outline: Frequent Itemset Mining

1. Introduction
2. Association Analysis Basic Concepts
3. Frequent Itemset Generation
4. Association Rules Generation
5. Interestingness Measures

Introduction

- We are often interested in co-occurrence relationships
- Marketing
  1. identify items that are bought together by sufficiently many customers
  2. use this information for marketing or supermarket shelf management purposes

- Inventory Management
  1. identify parts that are often needed together for repairs
  2. use this information to equip your repair vehicles with the right parts

- Usage Mining
  1. identify words that frequently appear together in search queries
  2. use this information to offer auto-completion features to the user

What are patterns?

- Originated from transaction databases
- Example: Supermarket transactions
  - Tales of “Beers” and “Diapers”
- **Pattern:** A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - Unsupervised
  - Captures intrinsic and important properties of the data

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

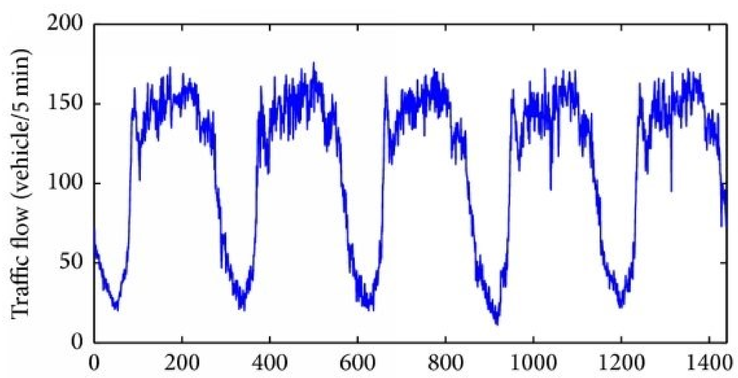
What are pattern analyses?

- **Pattern analysis:** Uncovering patterns from massive data sets
- Application scenarios:
  - What products are typically bought together?
  - What will a customer purchase after buying an iPad? Maybe apple pencil?
  - What code segments look like copy-paste bugs?
  - What word sequences look like high-quality phrases in a large corpus?
  - What gene sequences are popular in certain diseases?
  - …

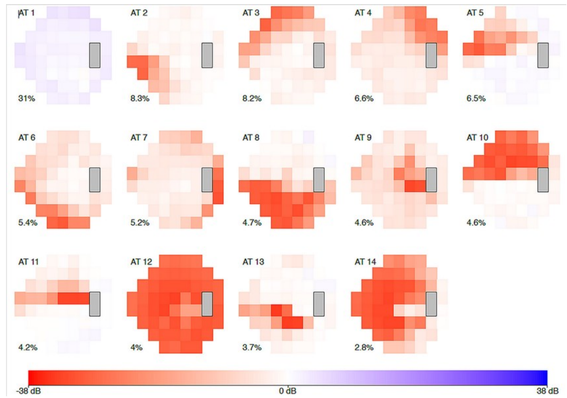
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Why pattern analysis is important?

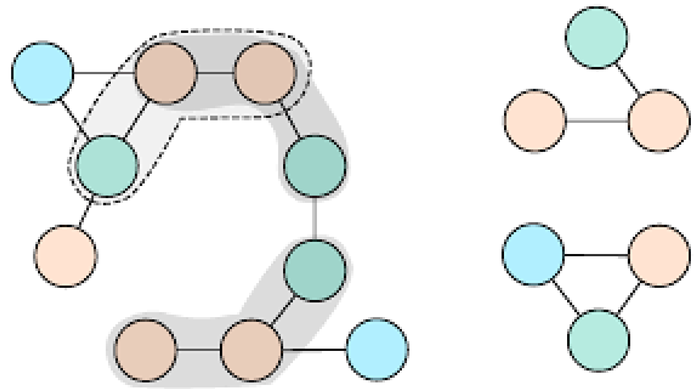
- **Foundation** for many essential data mining tasks
- Relevant tasks:
  - Association, correlation, and causality analysis
  - Mining sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: Discriminative pattern-based analysis
  - Cluster analysis: Pattern-based subspace clustering
  - …



Sequential patterns



Visual loss patterns



Structure patterns

Outline: Frequent Itemset Mining

1. Introduction
2. Association Analysis Basic Concepts
3. Frequent Itemset Generation
4. Association Rules Generation
5. Interestingness Measures

Association analysis

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.

Shopping 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

Examples of Association Rules

{Beer, Bread} → {Milk}
{Milk, Bread} → {Eggs, Coke}
{Diaper} → {Beer}

Implication means co-occurrence, not causality!

Definition: support and frequent itemset

- Itemset
  - collection of one or more items
    - example: {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 (s)
  - fraction of transactions that contain an itemset
  - e.g. s({Milk, Bread, Diaper}) = 2/5 = 0.4
- Frequent Itemset
  - an itemset whose support is greater than or equal to a minimal support (*minsup)* threshold specified by the user

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

Definition: association rule

- Association Rule
  - an implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
  - an association rule states that when X occurs, Y occurs with certain probability.
  - Example:
    {Milk, Diaper} $\rightarrow$ {Beer}
    Condition Consequent

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

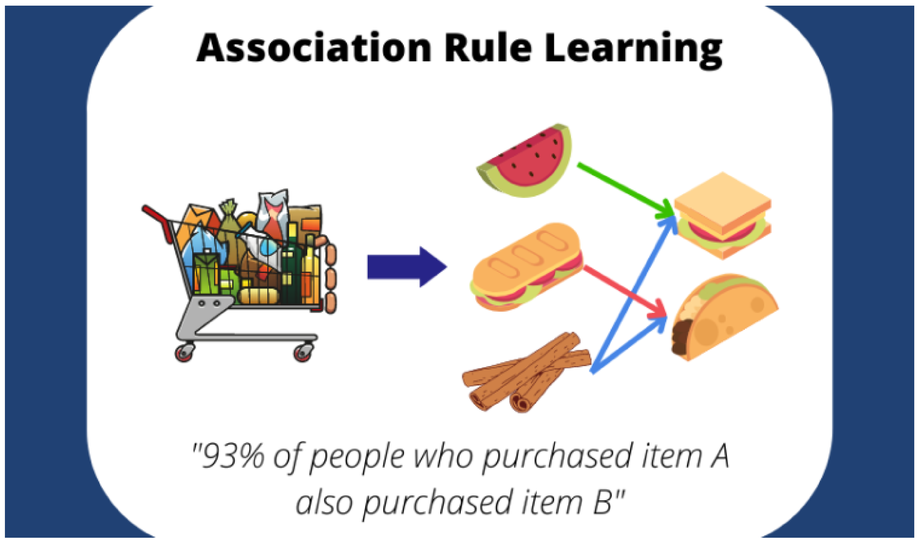
- Rule Evaluation Metrics
  - Support (s)
    fraction of transactions that contain both X and Y
  - Confidence (c)
    measures how often items in X appear in transactions that contain Y

$$s(X \rightarrow Y) = \frac{|X \cup Y|}{|T|} \qquad s = \frac{(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \qquad c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Main challenges in association analysis

1. Mining associations from large amounts of data can be computationally expensive
   - algorithms need to apply smart pruning strategies
2. Algorithms often discover a large number of associations
   - many of them are uninteresting or redundant
   - the user needs to select the subset of the associations that is relevant given the task at hand



The frequent itemset mining task

- Given a set of transactions T, the goal of frequent itemset mining is to frequent itemsets having
  - support ≥ *minsup* threshold
- *minsup* is provided by the user.

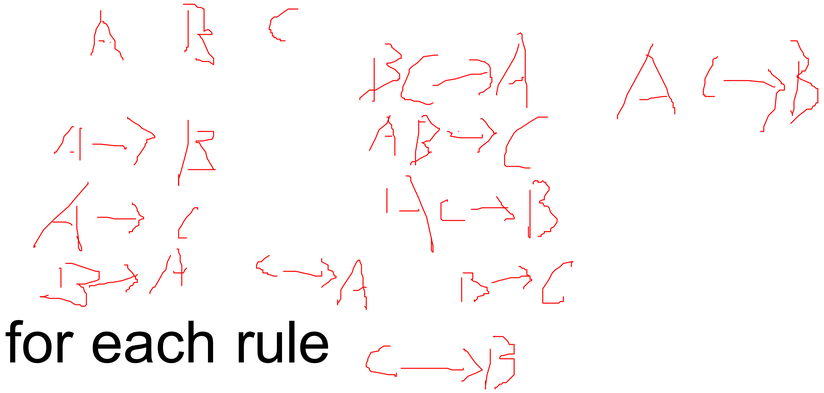
Items bought
Beer, Nuts, Diaper
Beer, Coffee, Diaper
Beer, Diaper, Eggs
Nuts, Eggs, Milk
Nuts, Coffee, Diaper, Eggs, Milk

Example:

- Let minsup = 50%
- Frequent itemset
  - Beer: 3 (60%); Nuts: 3 (60%)
  - Diaper: 4 (80%); Eggs: 3 (60%)
  - {Beer, Diaper}: 3 (60%)

The association rule mining task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  1. support ≥ *minsup* threshold
  2. confidence ≥ *minconf* threshold
- *minsup* and *minconf* are provided by the user.
- Brute force approach:
  1. list all possible association rules
  2. compute the support and confidence for each rule
  3. remove rules that fail the *minsup* and *minconf* thresholds



⇒ Computationally prohibitive due to large number of candidates!

Mining association rules

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

Example rules:

{Milk, Diaper} → {Beer} (s=0.4, c=0.67)
{Milk, Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper, Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk, Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk, Beer} (s=0.4, c=0.5)
{Milk} → {Diaper, Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence.
- Thus, we may decouple the support and confidence requirements.

Frequent itemset vs. association rule mining

Aspect	Frequent Itemset Mining	Association Rule Mining
Definition	Identifies sets of items that appear frequently together in a dataset, measured by support.	Finds rules predicting the occurrence of an item based on others, evaluated by support, confidence, and lift.
Objective	To discover patterns of item combinations frequently appearing in the data.	To extract meaningful correlations and associations from frequent itemsets.
Output	List of all itemsets that meet a minimum support threshold.	Rules expressing dependencies between items, e.g., {milk, bread} → {butter}.
Method	Uses algorithms like Apriori, FP-Growth, and Eclat to generate and evaluate itemsets.	Relies on frequent itemset mining results to generate rules.
Focus	Focuses on finding groups of items that frequently occur together.	Establishes rules based on relationships discovered in itemsets.
Dependence	Standalone process, used as a precursor to association rule mining.	Depends on the results of frequent itemset mining.

Outline: Frequent Itemset Mining

1. Introduction

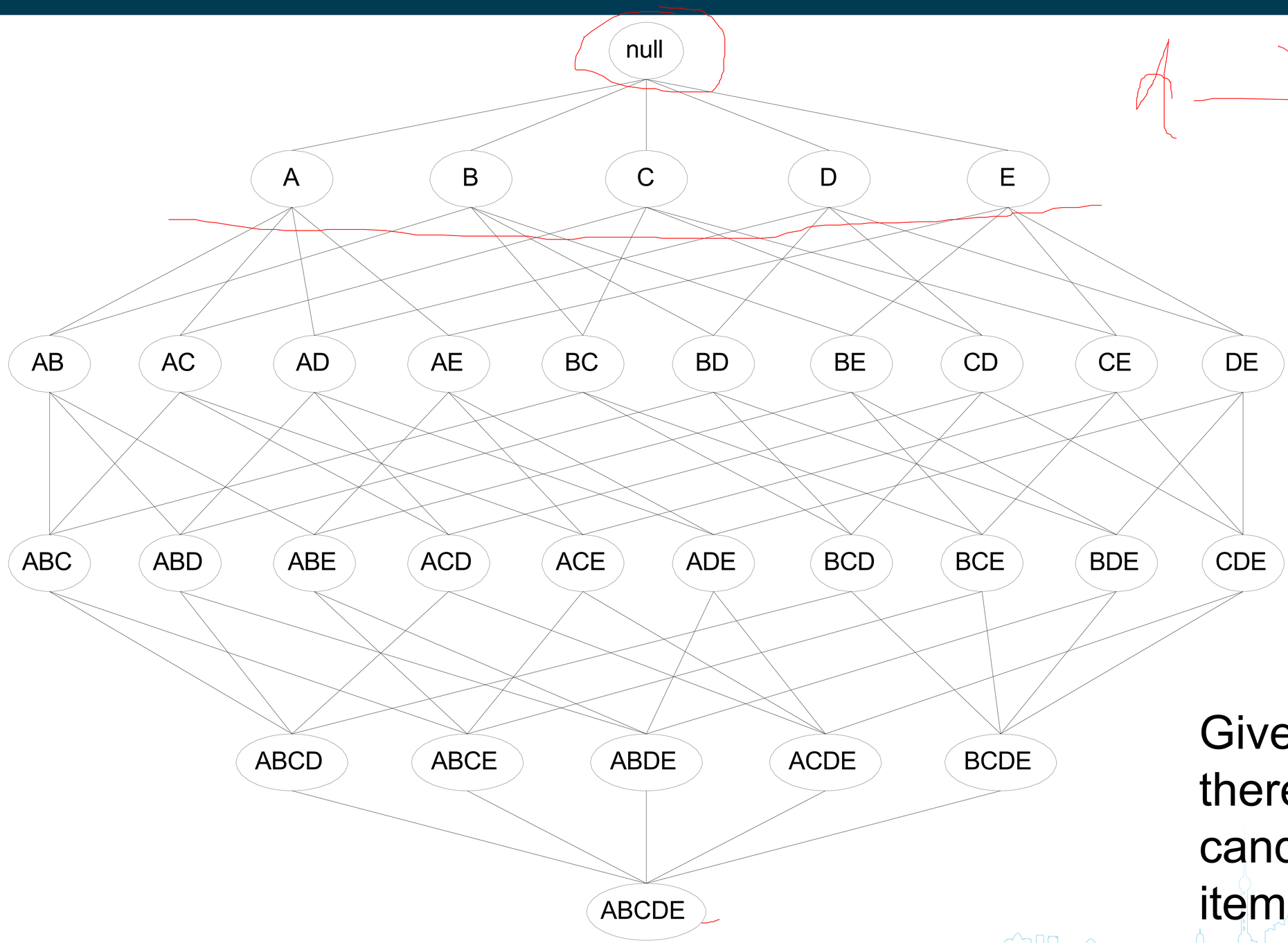
2. Association Analysis Basic Concepts

3. Frequent Itemset Generation

4. Association Rules Generation

5. Interestingness Measures

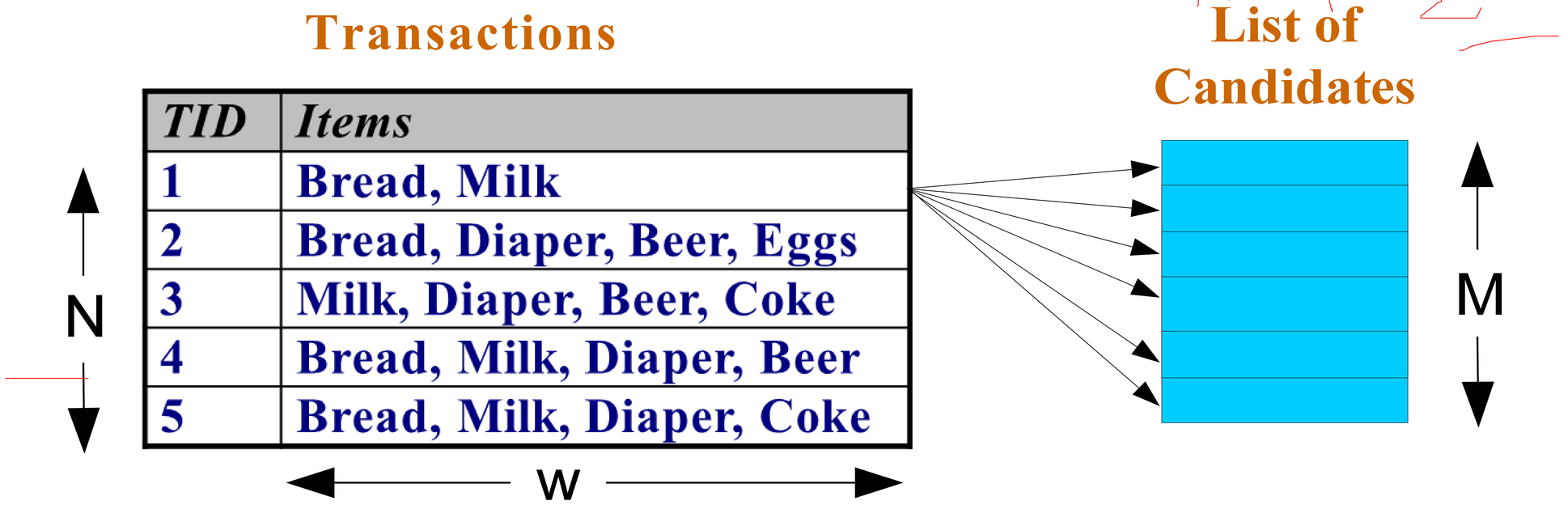
Frequent itemset generation



Given d items, there are 2^d candidate itemsets!

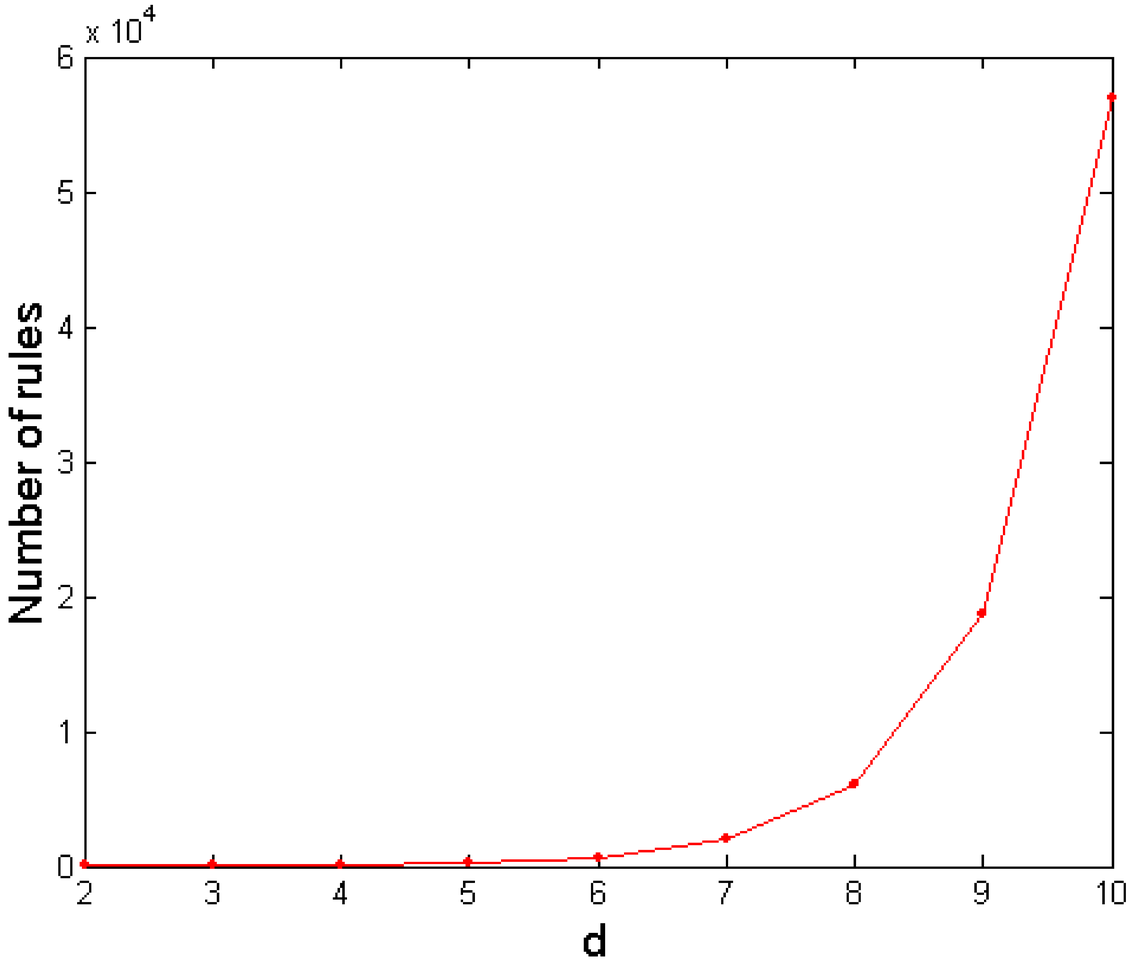
Brute force approach

- Treat every itemset as a candidate frequent itemset
- Count the support of each candidate by scanning the database
- Match each transaction against every candidate



Brute force approach: complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

A smarter algorithm is required!

Example: brute force approach

- Example:
  - Amazon has 10 million books (i.e., Amazon Germany, as of 2011)
- That is $2^{10.000.000}$ possible itemsets
- As a number:
  - $9.04981... \times 10^{3.010.299}$
  - that is: a number with 3 million digits!

- However:
  - most itemsets will not be important at all, e.g., books on Chinese calligraphy, Inuit cooking, and data mining bought together
  - thus, smarter algorithms should be possible
  - intuition for the algorithm: All itemsets containing Inuit cooking are likely infrequent

Reducing the number of candidates

– Apriori Principle

If an itemset is frequent, then all of its subsets must also be frequent.

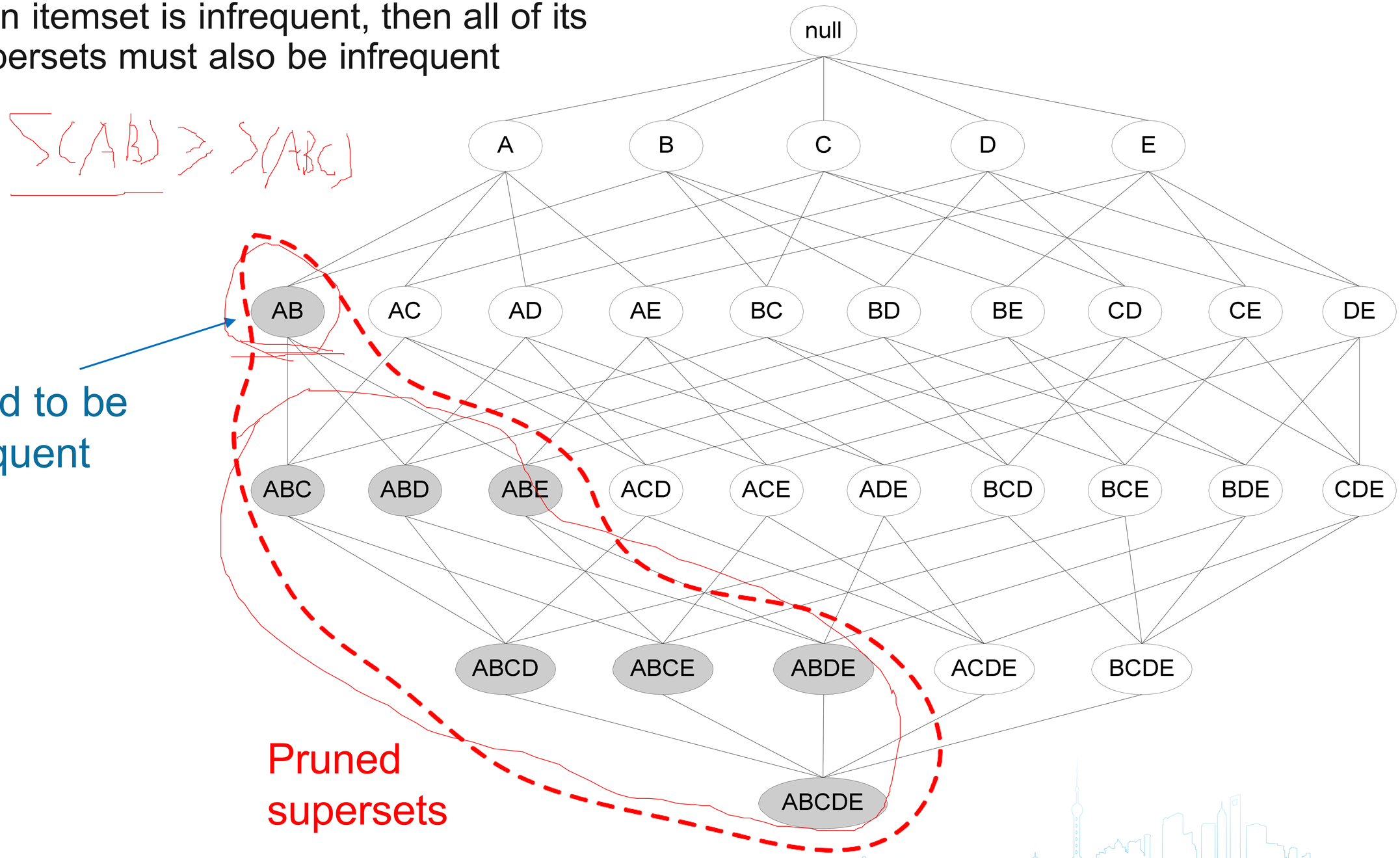
– Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- support of an itemset never exceeds the support of its subsets
- this is known as the anti-monotone property of support

Using the apriori principle for pruning

If an itemset is infrequent, then all of its supersets must also be infrequent



Example: the apriori principle for pruning

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

Items (1-itemsets)

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

Pairs (2-itemsets)

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

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3$

6 + 15 + 20 = 41

With support-based pruning,

6 + 6 + 1 = 13

Triplets (3-itemsets)

Itemset	Count
{Bread, Diaper, Milk}	2

(No need to generate candidate {Milk, Diaper, Beer} as count{Milk, Beer} = 2)

The apriori algorithm

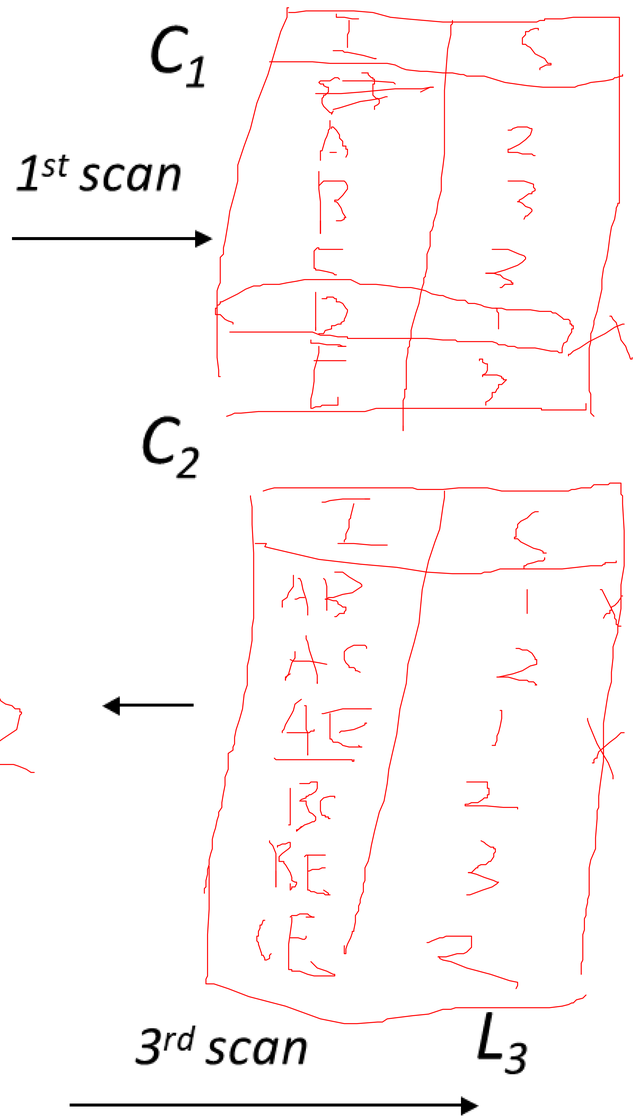
1. Let k = 1
2. Generate frequent itemsets of length 1
3. Repeat until no new frequent itemsets are identified
    1. Generate length (k+1) candidate itemsets from length k frequent itemsets
    2. Prune candidate itemsets that can not be frequent because they contain subsets of length k that are infrequent (Apriori Principle)
    3. Count the support of each candidate by scanning the dataset
    4. Eliminate candidates that are infrequent, leaving only those that are frequent

Example: apriori algorithm

$Sup_{min} = 2$

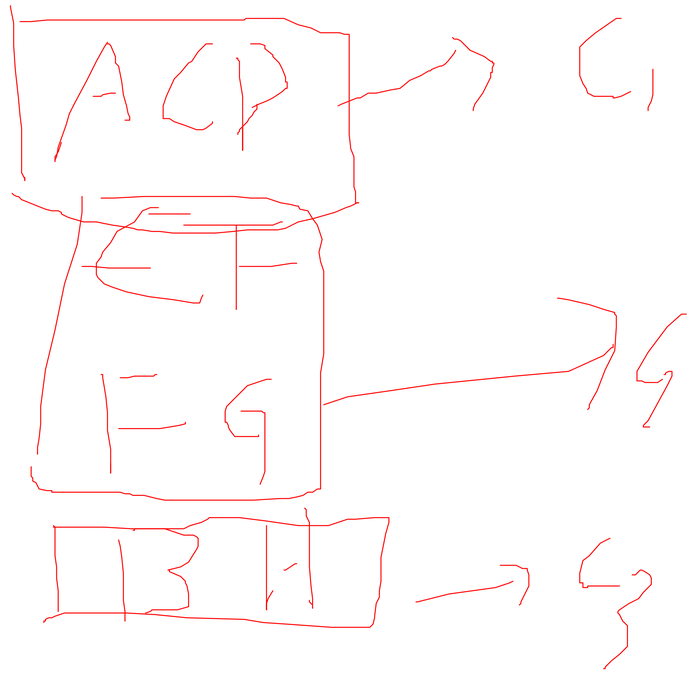
Database TDB

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



Apriori algorithm: prons and cons

- Advantages:
  - Apriori property
  - Easy implementation (in parallel also)



- Disadvantages:
  - requires up to k + 1 database scans

    - k length of the largest frequent itemset found
  - It assumes that the itemsets are in memory

Outline: Frequent Itemset Mining

1. Introduction

2. Association Analysis Basic Concepts

3. Frequent Itemset Generation

4. Association Rules Generation

5. Interestingness Measures

Mining associate rules

- Two-step approach:
  1. Frequent Itemset Generation
     - generate all itemsets whose support ≥ minsup
  2. Rule Generation
     - generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

Mining associate rules: pseudocode

Input:

D //Database of transactions
I //Items
L //Large itemsets
s //Support
α //Confidence

Output:

R //Association Rules satisfying s *and* α

ARGen Algorithm:

$R = \emptyset$;
for each $l \in L$ **do**
  for each $x \subset l$ **such that** $x \neq \emptyset$ **and** $x \neq l$ **do**
    if $\frac{support(l)}{support(x)} \geq \alpha$ **then**
      $R = R \cup \{x \Rightarrow (l - x)\}$;

Association rule generation

- Given a frequent itemset L, find all non-empty subsets f $\subset$ L such that f $\rightarrow$ L – f satisfies the minimum confidence requirement.

Example Frequent Itemset:

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

Example Rule:

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

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

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

Example: association rule generation

tid	X_T
1	{Bier, Chips, Wein}
2	{Bier, Chips}
3	{Pizza, Wein}
4	{Chips, Pizza}

Transaction database

$$I = \{\text{Bier, Chips, Pizza, Wein}\}$$

Itemset	Cover	Sup.	Freq.
{}	{1,2,3,4}	4	100 %
{Bier}	{1,2}	2	50 %
{Chips}	{1,2,4}	3	75 %
{Pizza}	{3,4}	2	50 %
{Wein}	{1,3}	2	50 %
{Bier, Chips}	{1,2}	2	50 %
{Bier, Wein}	{1}	1	25 %
{Chips, Pizza}	{4}	1	25 %
{Chips, Wein}	{1}	1	25 %
{Pizza, Wein}	{3}	1	25 %
{Bier, Chips, Wein}	{1}	1	25 %

Rule	Sup.	Freq.	Conf.
{Bier} ⇒ {Chips}	2	50 %	100 %
{Bier} ⇒ {Wein}	1	25 %	50 %
{Chips} ⇒ {Bier}	2	50 %	66 %
{Pizza} ⇒ {Chips}	1	25 %	50 %
{Pizza} ⇒ {Wein}	1	25 %	50 %
{Wein} ⇒ {Bier}	1	25 %	50 %
{Wein} ⇒ {Chips}	1	25 %	50 %
{Wein} ⇒ {Pizza}	1	25 %	50 %
{Bier, Chips} ⇒ {Wein}	1	25 %	50 %
{Bier, Wein} ⇒ {Chips}	1	25 %	100 %
{Chips, Wein} ⇒ {Bier}	1	25 %	100 %
{Bier} ⇒ {Chips, Wein}	1	25 %	50 %
{Wein} ⇒ {Bier, Chips}	1	25 %	50 %

Challenge: large number of candidate rules

- If {A,B,C,D} is a frequent itemset, then the candidate rules are:

  ABC →D, ABD →C, ACD →B, BCD →A,
  A →BCD, B →ACD, C →ABD, D →ABC
  AB →CD, AC → BD, AD → BC, BC →AD,
  BD →AC, CD →AB

- If |L| = k, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Association 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}:

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

  - Confidence is anti-monotone with respect to the number of items on the right-hand side of the rule

Explanation

Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

- i.e., “moving elements from left to right” cannot increase confidence

Reason:

$$c(AB \rightarrow C) := \frac{s(ABC)}{s(AB)} \qquad c(A \rightarrow BC) := \frac{s(ABC)}{s(A)}$$

- Due to anti-monotone property of support, we know

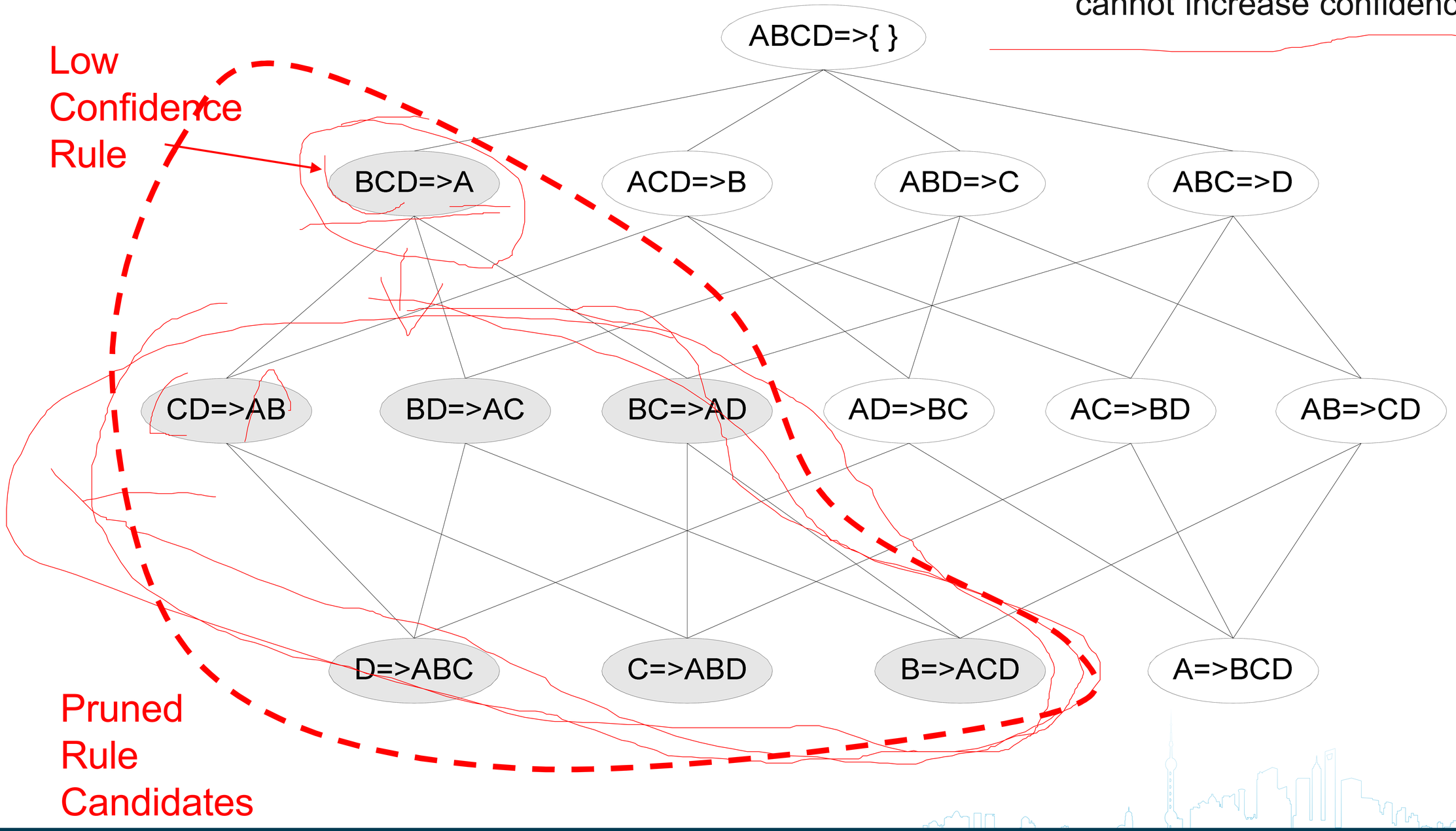
  $s(AB) \leq s(A)$

- Hence

  $c(AB \rightarrow C) \geq C(A \rightarrow BC)$

Candidate rule pruning

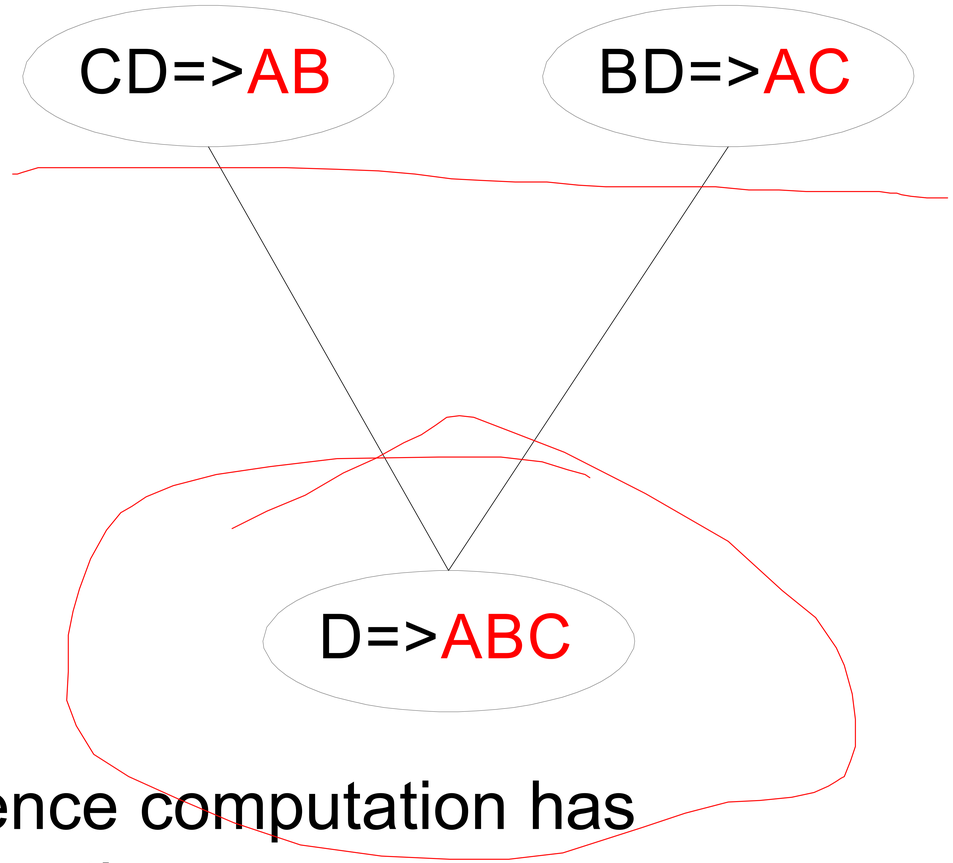
Moving elements from left to right cannot increase confidence



Candidate rule generation with apriori

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent (right hand side of rule)

1. join(CD $\rightarrow$ AB, BD $\rightarrow$ AC) would produce the candidate rule D $\rightarrow$ ABC



2. Prune rule D $\rightarrow$ ABC if one of its parent rules does not have high confidence (e.g. AD $\rightarrow$ BC)

- All the required information for confidence computation has already been recorded in itemset generation.
- Thus, there is no need to scan the transaction data *T* any more

Handling Continuous and Categorical Attributes

- How to apply association analysis to attributes that are not asymmetric binary variables?

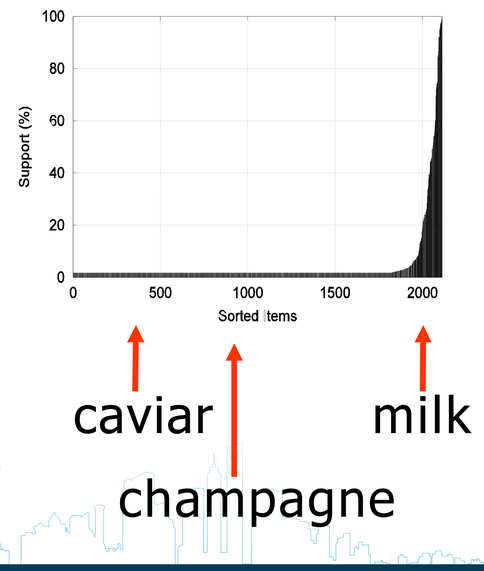
Session Id	Country	Session Length (sec)	Number of Web Pages viewed	Gender	Browser Type	Buy
1	USA	982	8	Male	Chrome	No
2	China	811	10	Female	Chrome	No
3	USA	2125	45	Female	Firefox	Yes
4	Germany	596	4	Male	IE	Yes
5	Australia	123	9	Male	Firefox	No
…	…	…	…	…	…	…

- Example Rule:

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

Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables
- Introduce a new "item" for each distinct attribute-value pair
  - e.g. replace "Browser Type" attribute with
    - attribute: "Browser Type = Chrome"
    - attribute: "Browser Type = Firefox"
    - …..
- Issues
  1. What if attribute has many possible values?
     - many of the attribute values may have very low support
     - potential solution: aggregate low-support attribute values
  2. What if distribution of attribute values is highly skewed?
     - example: 95% of the visitors have Buy = No
     - most of the items will be associated with (Buy=No) item
     - potential solution: drop the highly frequent item

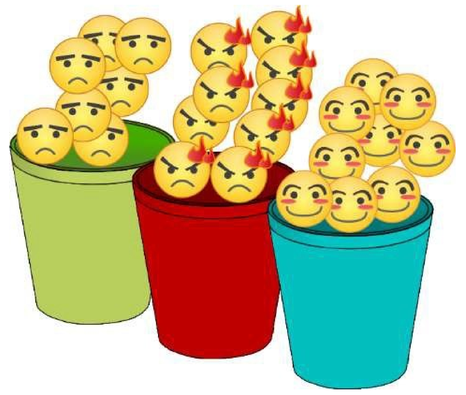


Handling Continuous Attributes

– Transform continuous attribute into binary variables using discretization

  • equal-width binning

  • equal-frequency binning



– Issue: Size of the discretization intervals affects support & confidence

{Refund = No, (Income = $51,251)} → {Cheat = No}

{Refund = No, (60K ≤ Income ≤ 80K)} → {Cheat = No}

{Refund = No, (0K ≤ Income ≤ 1B)} → {Cheat = No}

- If intervals are too small
  - itemsets may not have enough support
- If intervals too large
  - rules may not have enough confidence
  - e.g. combination of different age groups compared to a specific age group

Outline: Frequent Itemset Mining

1. Introduction

2. Association Analysis Basic Concepts

3. Frequent Itemset Generation

4. Association Rules Generation

5. Interestingness Measures

Interestingness Measures

- Association rule algorithms tend to produce too many rules
  - many of them are uninteresting or redundant
  - redundant if {A,B,C} → {D} and {A,B} → {D} have same support & confidence
- Interestingness of patterns depends on application
  - one man's rubbish may be another's treasure
- Interestingness measures can be used to prune or rank the derived rules
- In the original formulation of association rules, support & confidence were the only interestingness measures used
- Later, various other measures have been proposed
  - We will have a look at one: Lift

Drawback of Confidence

Contingency table

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

Association Rule: Tea $\rightarrow$ Coffee

- confidence($\overline{\text{Tea}} \rightarrow$ Coffee) = 0.75
- but support(Coffee) = 0.9
- although confidence is high, rule is misleading as the fraction of coffee drinkers is higher than the confidence of the rule
- we want confidence(X $\rightarrow$ Y) > support(Y)
- otherwise rule is misleading as X reduces probability of Y

Lift

– The lift of an association rule $X \rightarrow Y$ is defined as:

$$Lift = \frac{c(X \rightarrow Y)}{s(Y)}$$

– Confidence normalized by support of consequent

– Interpretation
  - if lift > 1, then X and Y are positively correlated
  - if lift = 1, then X and Y are independent
  - if lift < 1, then X and Y are negatively correlated

Lift

Contingency table

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

$$Lift = \frac{c(X \rightarrow Y)}{s(Y)}$$

Association Rule: $\overline{\text{Tea}} \rightarrow$ Coffee

- confidence($\overline{\text{Tea}} \rightarrow$ Coffee) = 0.75
- but support(Coffee) = 0.9

Lift($\overline{\text{Tea}} \rightarrow$ Coffee) = 0.75/0.9= 0.8333

- lift < 1, therefore is negatively correlated

Frequent itemset mining library: MLXtend

- **Aprior**:
  - Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215. 1994.

Email: min.shi@louisiana.edu