COMP6237 Data Mining
Lecture 9: Market Basket Analysis

Zhiwu Huang

Zhiwu.Huang@soton.ac.uk

Lecturer (Assistant Professor) @ VLC of ECS

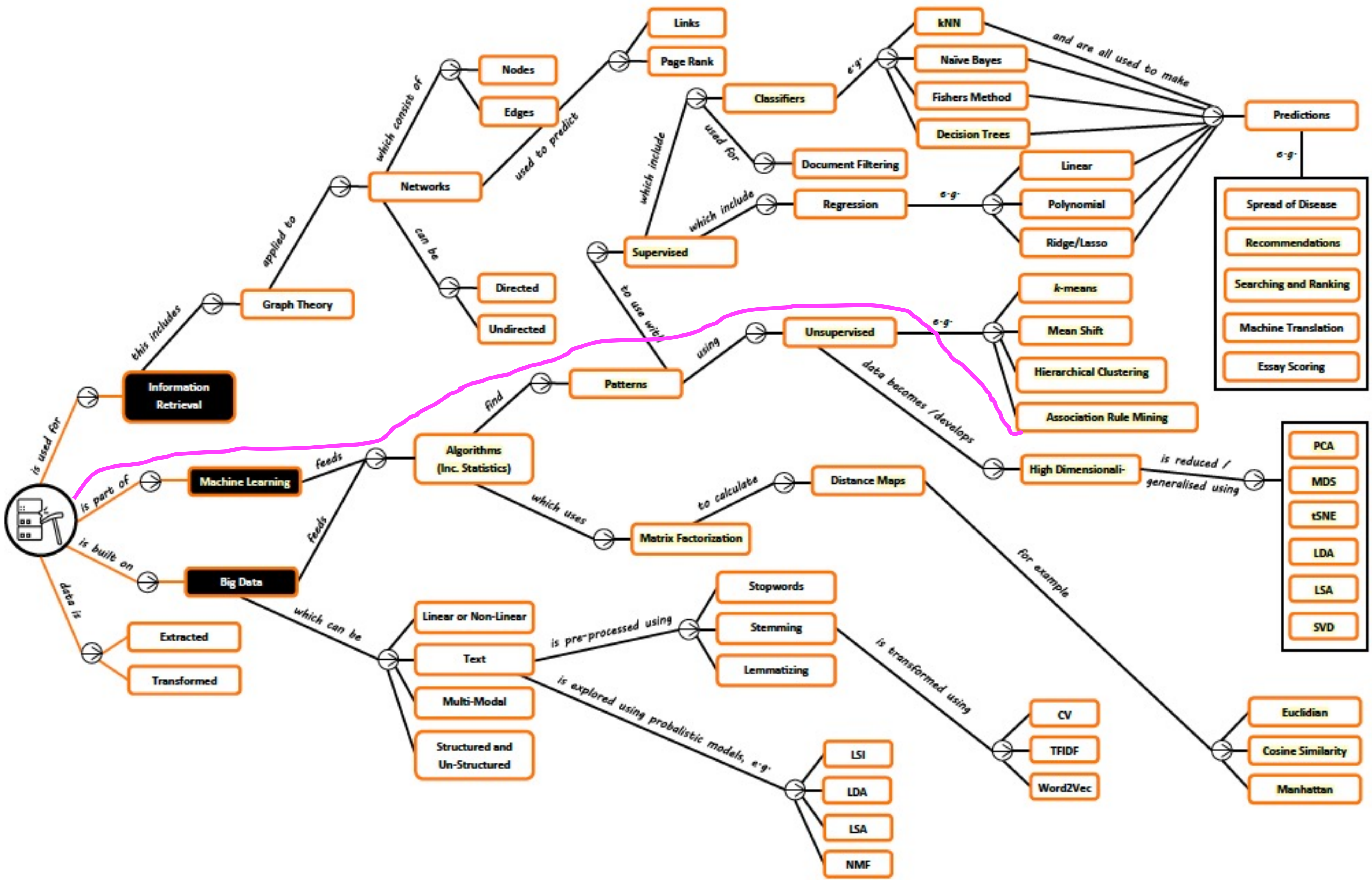
University of Southampton

Lecture slides available here:

http://comp6237.ecs.soton.ac.uk/zh.html

(Thanks to Prof. Jonathon Hare and Dr. Jo Grundy for providing the lecture materials used to develop the slides.)

Market Basket – **Roadmap**



Market Basket – **Textbook**

5 Association Analysis: Basic Concepts and Algorithms

Many business enterprises accumulate large quantities of data from their day-to-day operations. For example, huge amounts of customer purchase data are collected daily at the checkout counters of grocery stores. ***Table 5.1*** *gives an example of such data, commonly known as* ***market basket transactions****. Each row in this table corresponds to a transaction, which contains a unique identifier labeled TID and a set of items bought by a given customer. Retailers are interested in analyzing the data to learn about the purchasing behavior of their customers. Such valuable information can be used to support a variety of business-related applications such as marketing promotions, inventory management, and customer relationship management.*

- Introduction to Data Mining, *P. Tan et al*
  https://www-users.cse.umn.edu/~kumar001/dmbook/index.php

Market Basket – **Overview (1/4)**

Why analyse market baskets?

Get insight:

- do products sell quickly or slowly
- which products are sold together?
- which might need a promotion?

Use that to take action:

- store layout
- promotions
- recommendations

Credit: Jo Grundy

Market Basket – **Overview (2/4)**

Beer and Nappies
Back in 1992 A data consultant was using SQL queries to find things were often bought along side nappies (Diapers in the US), as nappies are high margin, they wanted to sell more of them. They were looking to find things to put on the shelves near each other. She found a correlation between beer sales, and nappy sales, and emailed her colleagues about it.
There was no good statistical basis for this link, but the story has become well known, one of the first to 'go viral'

Credit: Jo Grundy

Market Basket – **Overview (3/4)**

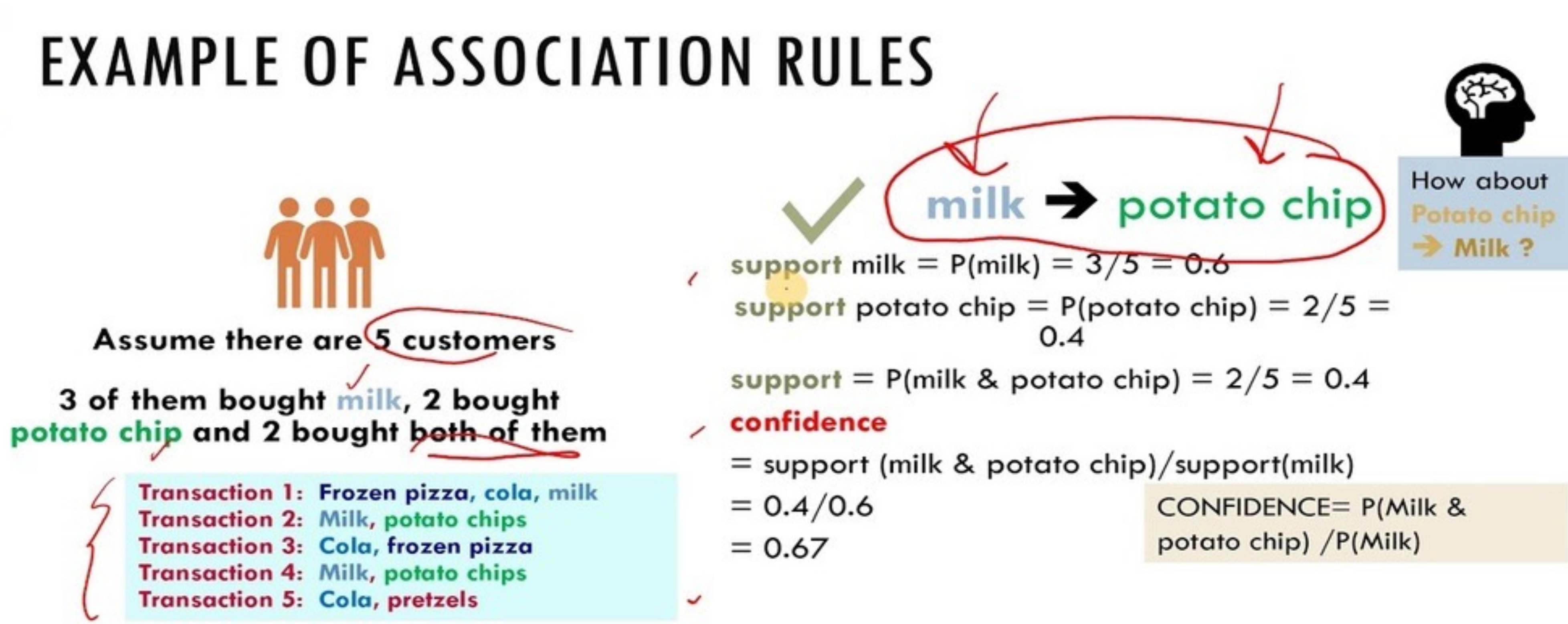
Market Basket analysis:

Given a database of transactions
Find groups of items that are frequently bought together

Each transaction is a set of items, a basket, called here an *itemset*
This allows companies to understand why people make certain purchases

Credit: Jo Grundy

Market Basket – **Overview (4/4)**



https://www.analyticsvidhya.com

Market Basket – **Learning Outcomes**

- **LO1**: Demonstrate an understanding of market basket analysis concepts and techniques, such as: (exam)
  - ❖ Calculating support and confidence for itemsets
  - ❖ Understanding the key steps of the Apriori algorithm for association rule mining
  - ❖ Using the Apriori algorithm to generate association rules from transaction data
- **LO2**: Implement the learned algorithms using associate rule mining algorithms (coursework)

Assessment hints: Multi-choice Questions (single answer: concepts, calculation etc)

- *Textbook Exercises: textbooks (Programming + Mining)*
- *Other Exercises: https://www-users.cse.umn.edu/~kumar001/dmbook/sol.pdf*
- *ChatGPT or other AI-based techs*

Market Basket – Definition

- Market basket transaction data:

  t_1: {bread, cheese, milk}

  t_2: {apple, eggs, salt, yogurt}

  …

  t_n: {biscuit, eggs, milk}

- Concepts:
  - An *item*: an item/article in a basket (i)
  - I: the set of *all items* sold in the store ($\{i_1, i_2, \ldots, i_m\}$)
  - A *transaction*: items purchased in a basket; it may have TID (transaction ID) (t)
  - A *transactional dataset*: A set of transactions ($T = \{t_1, t_2, \ldots, t_n\}$)

Market Basket – Definition

- An association rule is an implication of the form:

  $X \rightarrow Y$, where $X, Y \subset I$, *and* $X \cap Y = \varnothing$

  – $I = \{i_1, i_2, \ldots, i_m\}$: a set of *all items*

  – An itemset X is a set of items, where $X \subset I$.
    - E.g., X = {milk, bread} is an itemset.
    - A *k*-itemset is an itemset with *k* items.
      – E.g., {milk, bread, cereal} is a 3-itemset

  – A transaction t contains an itemset X, if $X \subseteq t$, $X \subset I$

Market Basket – Definition

- **Support:** The rule $X \to Y$ holds with support, ***sup***, in T (the transaction data set) if ***sup*** % of transactions contain $X \cup Y$.
  - ***sup*** = $\Pr(X \cup Y)$.

- **Confidence:** The rule $X \to Y$ holds in T with confidence, ***conf***, if ***conf*** % of transactions that contain X also contain Y.
  - ***conf*** = $\Pr(Y \mid X)$

Market Basket – Definition

- **Support count:** The support count of an itemset *A*, denoted by *A.count*, in a data set *T* is the number of transactions in *T* that contain *A*. Assume *T* has *n* transactions.
- Then, support and confidence for the rule $X \rightarrow Y$

$$support = \frac{(X \cup Y).count}{n}$$

number of transactions that simultaneously contains X and Y

total number of transactions

$$confidence = \frac{(X \cup Y).count}{X.count}$$

number of transactions that only contain itemset X

Market Basket – Definition

- Minimum Support Threshold: min_sup = s% (e.g., 40%)
- Minimum Confidence threshold: $min\_conf = c\%$(e.g., 60%)

Market Basket – Definition

- **Frequent itemset**
  - Suppose *min_sup* is the minimum support threshold
  - An itemset satisfies minimum support if the occurrence frequency of the itemset is greater or equal to *min_sup*
  - If an itemset satisfies minimum support, then it is a frequent itemset

- **Itemset**
  - A set of items is referred to as itemset
  - An itemset containing *k* items is called **k-itemset**

Market Basket – Definition

Rules that satisfy both a *minimum support* threshold and a *minimum confidence* threshold are called **strong rules**

Market Basket – Example

- **Itemset**
- **Support count (σ)**
- **Support (s)**
- **Frequent Itemset**

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

Market Basket – Example

University of Southampton

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - *k*-itemset
    - An itemset that contains k items
  - *3*-itemset: {Milk, Diaper, Beer}
- **Support count (σ)**
- **Support (s)**
- **Frequent Itemset**

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

Market Basket – Example

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - *k*-itemset
    - An itemset that contains k items
  - *3*-itemset: {Milk, Diaper, Beer}
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Diaper, Beer}) = 2
- **Support (s)**
- **Frequent Itemset**

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

Market Basket – Example

University of Southampton

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - *k*-itemset
    - An itemset that contains k items
  - *3*-itemset: {Milk, Diaper, Beer}
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Diaper, Beer}) = 2
- **Support (s)**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Diaper, Beer}) = 2/5
- **Frequent Itemset**

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

Market Basket – Example

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

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

Market Basket – Association Rule Mining

- An association rule *r* is ***strong*** if
  - *Support(r) ≥ min_sup*
  - *Confidence(r) ≥ min_conf*

Market Basket – Association Rule Mining

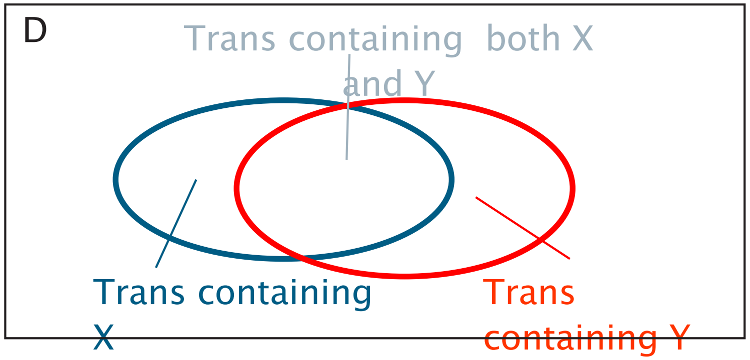
- An association rule *r* is ***strong*** if
  - *Support(r)* ≥ *min_sup*
  - *Confidence(r)* ≥ *min_conf*

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *min_sup* threshold
  - confidence ≥ *min_conf* threshold

Market Basket – Example

- Association Rule
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets, and $X \cap Y = \emptyset$
  - Example: $\{Milk, Diaper\} \rightarrow \{Beer\}$

Antecedent → Consequent



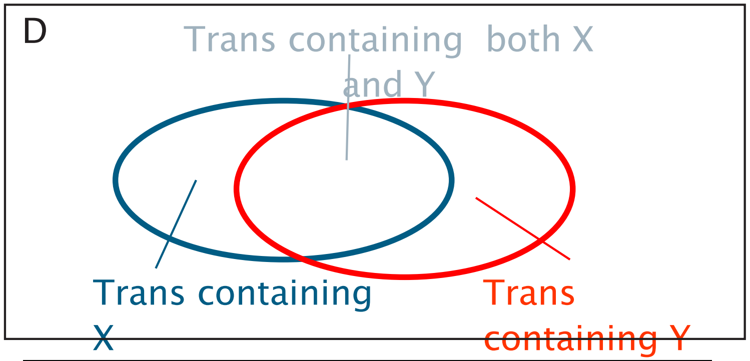
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Rule: $\{Milk, Diaper\} \Rightarrow Beer$

Market Basket – Example

- Association Rule
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets, and $X \cap Y = \emptyset$
  - Example: {Milk, Diaper} → {Beer}
- Rule Evaluation Metrics
  - Support (s)
    - ◆ Fraction of transactions that contain both X and Y

$$P(X \cup Y) = \frac{\#transcontaining(X \cup Y)}{\#transinD}$$



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

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

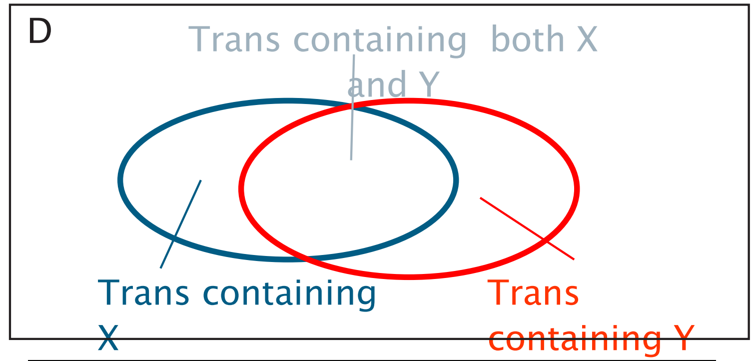
Market Basket – Example

- Association Rule
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets, and $X \cap Y = \emptyset$
  - Example: {Milk, Diaper} → {Beer}
- Rule Evaluation Metrics
  - Support (s)
    - ◆ Fraction of transactions that contain both X and Y

$$P(X \cup Y) = \frac{\#transcontaining(X \cup Y)}{\#transinD}$$

  - Confidence (c)
    - ◆ Measures how often items in Y appear in transactions that contain X

$$P(Y|X) = \frac{\#transcontaining(X \cup Y)}{\#transcontainingX}$$



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

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

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

Market Basket – Association Rule Mining

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *min_sup* and *min_conf* thresholds

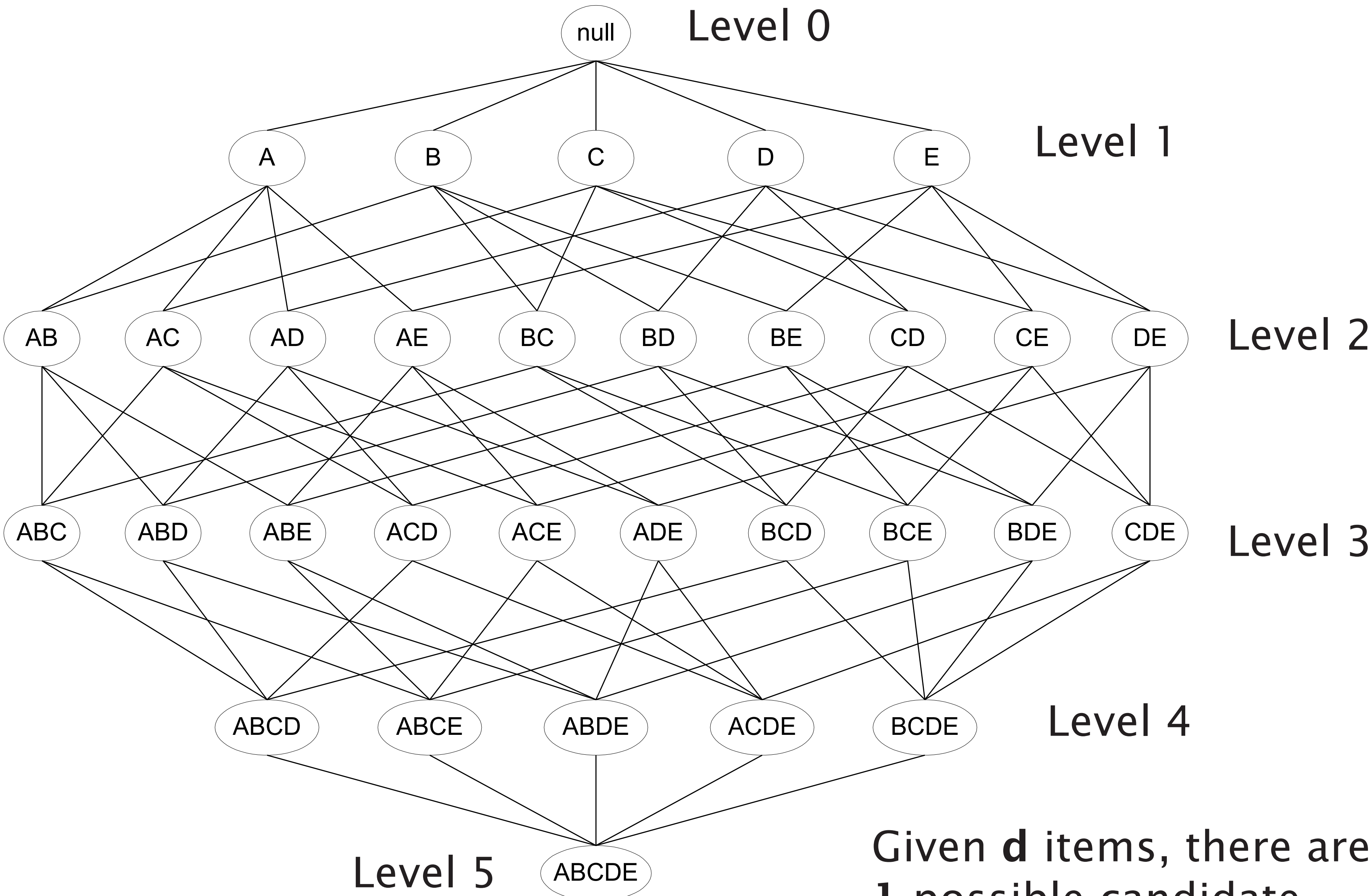
$\Rightarrow$ Computationally prohibitive!

Market Basket – Association Rule Mining

- Two-step approach:
  - Frequent Itemset Generation
    - Generate all itemsets whose support $\geq$ minsup
  - 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!

Market Basket – Association Rule Mining



Given **d** items, there are $\mathbf{2^d - 1}$ possible candidate itemsets

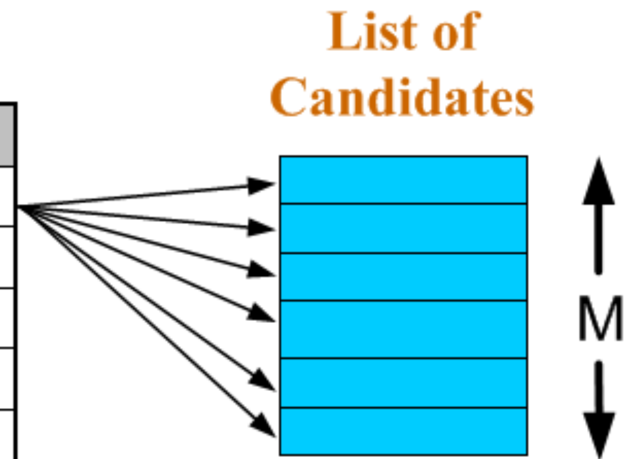
Market Basket – Association Rule Mining

Frequent Itemset Generation

- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

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



  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => **Expensive** since M = **2^d-1**!!!*

Market Basket – Association Rule Mining

- There are a large number of them!!
    - Multi-level or generalized rule mining
    - Constrained rule mining
    - Incremental rule mining
    - ….

- They use different strategies and data structures.
    - these algorithms generally find the same set of rules although their computational efficiencies and memory requirements may be different.

- Their resulting sets of rules
    - Given a transaction data set *T*, a minimum support and a minimum confidence, the set of association rules existing in *T* is **uniquely determined**.

- We study only one: **Apriori Algorithm**

Market Basket – Association Rule Mining
Apriori Algorithm

- One of the most well-known algorithms
- **Two steps** or two phases:
  - Find all itemsets that have minimum support (*frequent itemsets*, also called large itemsets)-- discover ***frequent itemsets*** from a given dataset
  - **Generate rules** from these frequent itemsets.
- ., a frequent itemset

{Chicken, Clothes, Milk}   [sup = 3/7]

and one rule from the frequent itemset

Clothes $\rightarrow$ Milk, Chicken   [sup = 3/7, conf = 3/3]

Market Basket – Association Rule Mining
Apriori Algorithm

Step 1: Mining or Finding all Frequent Itemsets

- **Key idea:** reduce the number of candidate itemsets with a strong prior knowledge that any subsets of a frequent itemset are also frequent itemsets
  - A frequent *itemset* is an itemset whose support is ≥ minsup.
  - Example: If {milk diaper} is a frequent itemset, then we can derive that its subsets {milk} and {diaper} are both frequent itemsets.

Market Basket – Association Rule Mining
Apriori Algorithm

University of Southampton

Step 2: Generating Rules from Frequent Itemsets

- **Frequent itemsets ≠ association rules**
- One more step is needed to generate association rules
- For each frequent itemset *X*,

  For each proper nonempty subset *A* of *X*,

  – Let *B* = X - *A*

  – A $\longrightarrow$ B is an association rule if

    - Confidence($A \rightarrow B$) ≥ minconf,

      support($A \rightarrow B$) = support($A \cup B$) = support(X)

      confidence($A \rightarrow B$) = support($A \cup B$) / support(A)

Antecedent → Consequent

Market Basket – Apriori Algorithm
Reducing 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:

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

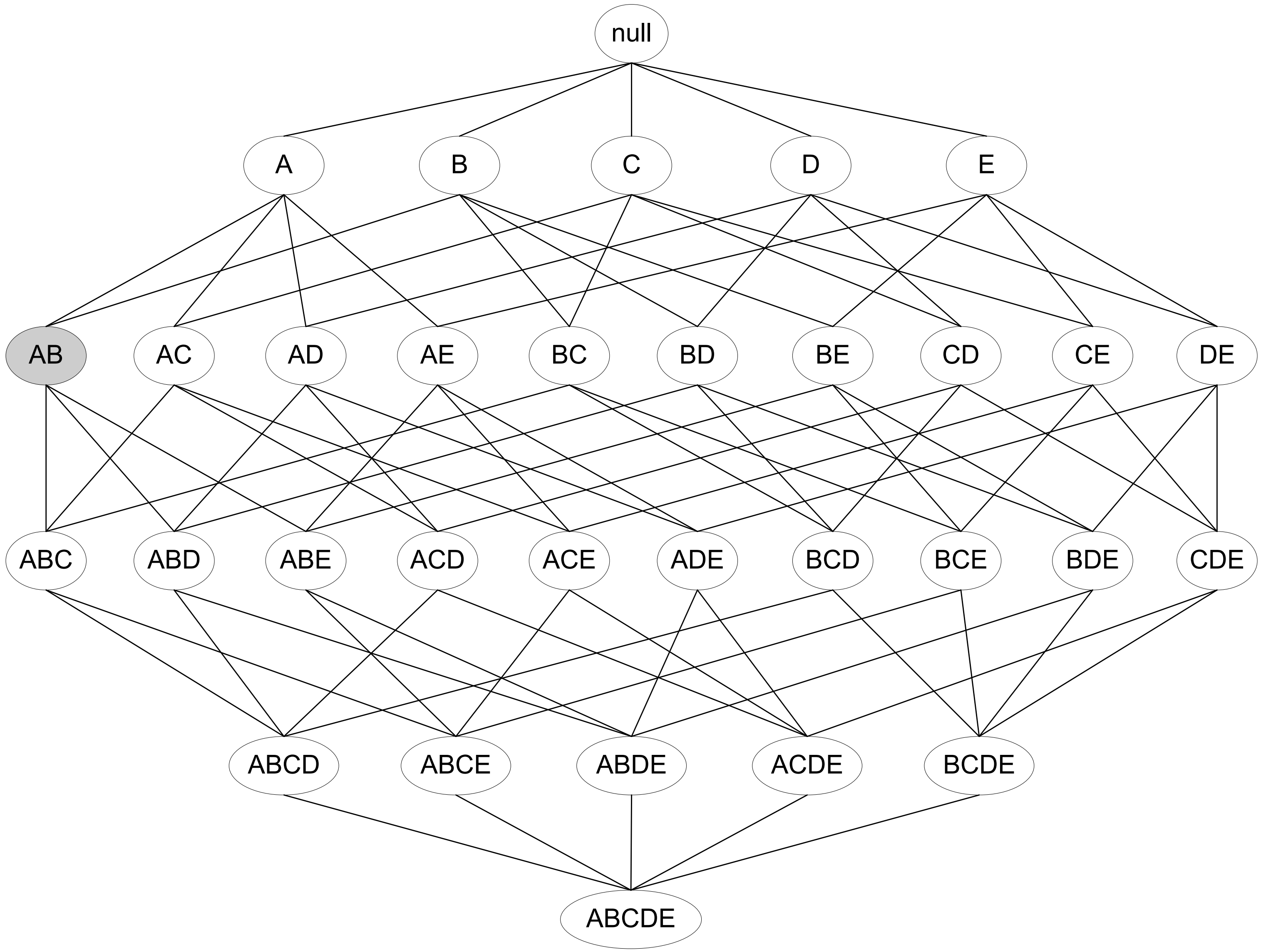
  - Support of an itemset never exceeds the support of its subsets
  - This is known as the **anti-monotone** property of support

Market Basket – Apriori Algorithm
Anti-Monotone Property

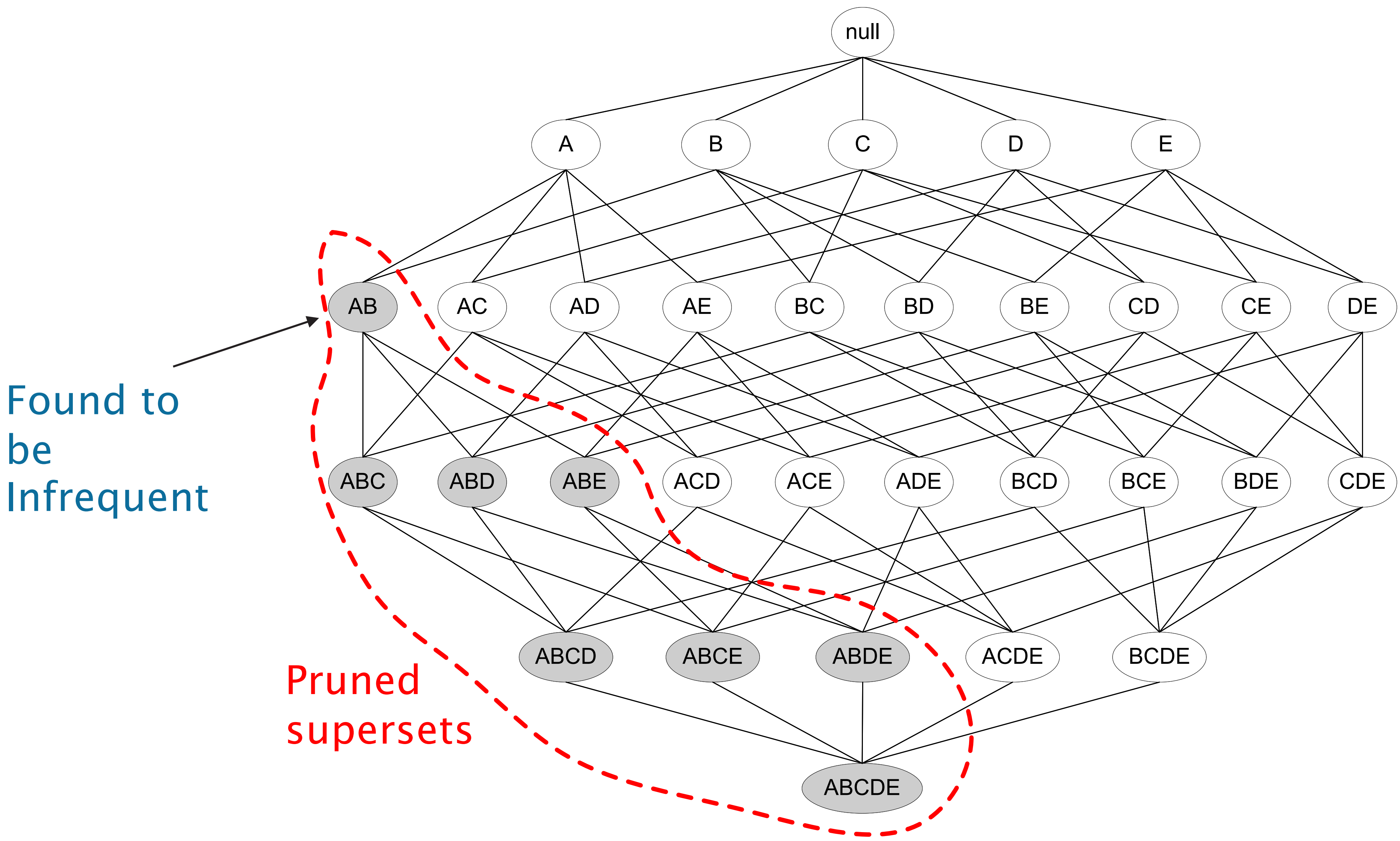
- Any subset of a frequent itemset must be also frequent — An anti-monotone property
  - Any transaction containing {beer, diaper, milk} also contains {beer, diaper}
  - {beer, diaper, milk} is frequent → {beer, diaper} must also be frequent
- In other words, any superset of an infrequent itemset must also be infrequent
  - No superset of any infrequent itemset should be generated or tested
- **Many item combinations can be pruned!**

Market Basket – Apriori Algorithm

Found to be Infrequent



Market Basket – Apriori Algorithm



Market Basket – Apriori Algorithm

- Method:
  - Let length of itemset be k=1
  - Generate frequent itemsets of length 1 (i.e., 1-itemset)
  - Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Market Basket – Apriori Algorithm
Example

- Five transactions from a supermarket

TID	List of Items
1	Beer,Diaper,Baby Powder,Bread,Umbrella
2	Diaper,Baby Powder
3	Beer,Diaper,Milk
4	Diaper,Beer,Detergent
5	Beer,Milk,Coca-Cola

Market Basket – Apriori Algorithm
Example

- Min_sup 40% (2/5)

C1

Item	Support
Beer	"4/5"
Diaper	"4/5"
Baby Powder	"2/5"
Bread	"1/5"
Umbrella	"1/5"
Milk	"2/5"
Detergent	"1/5"
Coca-Cola	"1/5"

L1

Item	Support
Beer	"4/5"
Diaper	"4/5"
Baby Powder	"2/5"
Milk	"2/5"

Market Basket – Apriori Algorithm
Example

C2

Item	Support
Beer, Diaper	"3/5"
Beer, Baby Powder	"1/5"
Beer, Milk	"2/5"
Diaper,Baby Powder	"2/5"
Diaper,Milk	"1/5"
Baby Powder,Milk	"0"

L2

Item	Support
Beer, Diaper	"3/5"
Beer, Milk	"2/5"
Diaper,Baby Powder	"2/5"

Market Basket – Apriori Algorithm
Example

C3

Item	Support
Beer, Diaper,Baby Powder	"1/5"
Beer, Diaper,Milk	"1/5"
Beer, Milk,Baby Powder	"0"
Diaper,Baby Powder,Milk	"0"

Empty

Market Basket – Apriori Algorithm

Example

Discovery: Support >*min_sup=40%*

- *min_sup=40%* *min_conf=70%*

Item	Support
Beer, Diaper	"3/5"
Beer, Milk	"2/5"
Diaper,Baby Powder	"2/5"

Generate Rules based on the searched frequent 2-itemsets

{Beer} → {Diaper}, {Beer} → {Milk}, {Diaper} → {Baby Powder}
{Diaper} → {Beer}, {Milk} → {Beer}, {Baby Powder} → {Diaper}

Item	Support(A,B)	Support A	Confidence
Beer, Diaper	60%	80%	75%
Beer, Milk	40%	80%	50%
Diaper,Baby Powder	40%	80%	50%
Diaper,Beer	60%	80%	75%
Milk,Beer	40%	40%	100%
Baby Powder, Diaper	40%	40%	100%

Market Basket – Apriori Algorithm **Example**

Results: Association Rules

$$Beer \Rightarrow Diaper$$

- support 60%, confidence 75%

$$Diaper \Rightarrow Beer$$

- support 60%, confidence 75%

$$Milk \Rightarrow Beer$$

- support 40%, confidence 100%

$$Baby\_Powder \Rightarrow Diaper$$

- support 40%, confidence 100%

Market Basket – Summary

Terms were defined:

- **Association rules**: if X then Y, $X \implies Y$
- **Items** I, set of all possible items i
- **Transaction**: set of items t_i such that $t_i \subset I$
- **Database** D containing all transactions $\{t_i\}_1^d$
- **Itemset**: subset of I, with k items is a $k - itemset$

Measures were defined:

- **Support** of itemset X is % transactions in D that contain X
- Support of Association rule $X \implies Y$ is $\frac{|t \in D; X \cup Y \subset t|}{|t \in D; X \subset t|}$
- **Confidence** is $\frac{Sup(X \cup Y)}{Sup(X)}$

A Priori Algorithm described

Credit: Jo Grundy