Data Mining & Data Warehouse

Dr. Raed Ibraheem Hamed

University of Human Development,
College of Science and Technology
Department of Information Technology

2016 – 2017

Road map

- Association rule mining
- Market-Basket Data
- Frequent Itemsets
- Association rule Applications
- Association Rules Definition
- Measure 1: Support
- Measure 2: Confidence
- Transaction data: supermarket data
- Rule strength measures

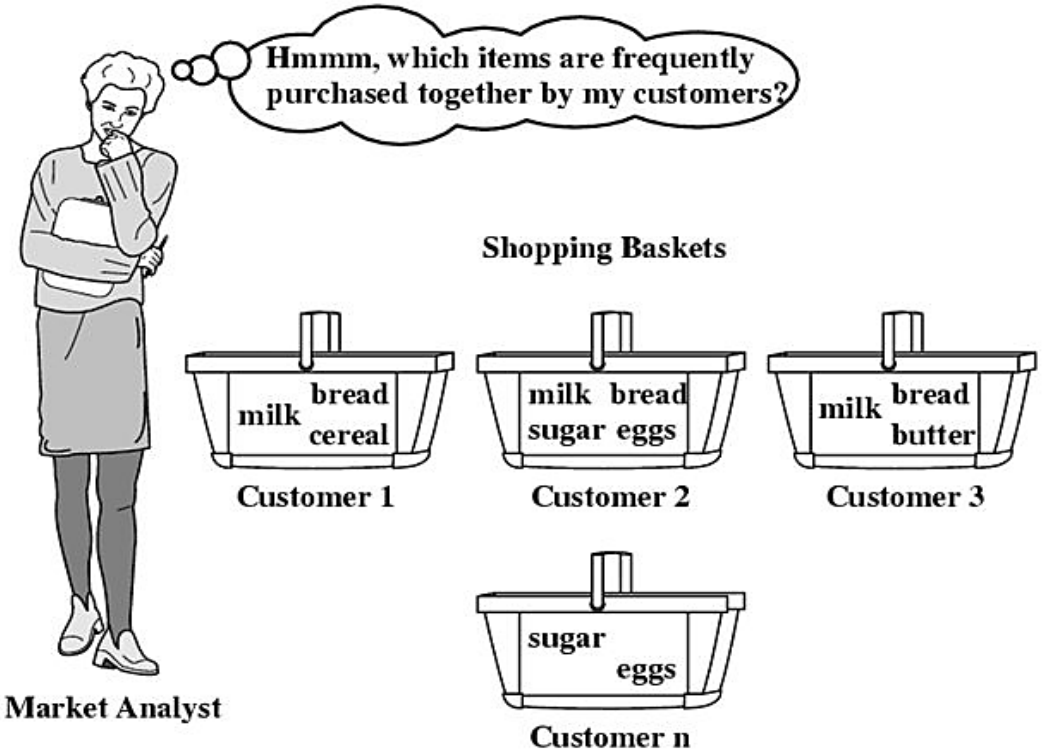
Association rule mining

- Proposed by **Agrawal et al in 1993**.
- It is an important data mining model studied extensively by the database and data mining community.
- Initially used for **Market Basket Analysis** to find how items purchased by customers are related.

Market-Basket Data

- A large set of items, e.g., things sold in a supermarket.
- A large set of baskets, each of which is a small set of the items, e.g., the things one customer buys on one day.

Market Basket Analysis



Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.

Frequent Itemsets

- Given a set of transactions, find combinations of items (itemsets) that occur frequently

Support $s(I)$: number of transactions that contain itemset I

Market-Basket transactions

Items: {Bread, Milk, Diaper, Beer, Eggs, Coke}

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 frequent itemsets $s(I) \geq 3$

{Bread}: 4
{Milk} : 4
{Diaper} : 4
{Beer}: 3
{Diaper, Beer} : 3
{Milk, Bread} : 3

Association rule Applications

- Items = products; baskets = sets of products someone bought in one trip to the store.

- Example application: given that many people buy tea and sugar together:
  - Run a sale on sugar ; raise price of tea.
  - Only useful if many buy sugar & tea.

Association Rules Definition

Association rules are if/then statements that help uncover relationships between seemingly unrelated **data** in a relational database or other information repository. An example of an **association rule** would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

There are two common ways to measure association.

Measure 1: Support.

Measure 1: Support. This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears. In Table 1 below, the support of {apple} is 4 out of 8, or 50%. Itemsets can also contain multiple items.

For instance, the support of {apple, beer, rice} is 2 out of 8, or 25%.

$$\text{Support } \{🍎\} = \frac{4}{8}$$

Measure 1: Support.

Transaction 1	
Transaction 2	
Transaction 3	
Transaction 4	
Transaction 5	
Transaction 6	
Transaction 7	
Transaction 8	

Table 1. Example Transactions

If you discover that sales of items beyond a certain proportion tend to have a significant impact on your profits, you might consider using that proportion as your support **threshold**.

You may then identify itemsets with support values above this threshold as significant itemsets.

Measure 2: Confidence.

Measure 2: Confidence. This says how likely item Y is purchased when item X is purchased, expressed as {X ⟶ Y}.

This is measured by the proportion of transactions with item X, in which item Y also appears. In Table 1, the confidence of {apple ⟶ beer} is 3 out of 4, or 75%.

$$\text{Confidence}\{🍎 \rightarrow 🍺\} = \frac{\text{Support}\{🍎, 🍺\}}{\text{Support}\{🍎\}}$$

3 / 8 = 0.375 4 / 8 = 0.5

Confidence = 0.375 / 0.5 = **0.75**

Support and Confidence Example

Transaction ID	Items Bought
1	Shoes, Shirt, Jacket
2	Shoes,Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

If the **support** is 50%, then {Shoes, Jacket} is the only 2- itemset that satisfies the support.

Frequent Itemset	Support
{Shoes}	75%
{Shirt}	50%
{Jacket}	50%
{Shoes, Jacket}	50%

If the **confidence** is 50%, then the only two rules generated from this 2-itemset, that have confidence are:

Shoes $\Rightarrow$ Jacket Support=50%, Confidence=66%
Jacket $\Rightarrow$ Shoes Support=50%, Confidence=100%

Support and Confidence Example

- Given a database of transactions:

Transaction	Items
t_1	Bread,Jelly,PeanutButter
t_2	Bread,PeanutButter
t_3	Bread,Milk,PeanutButter
t_4	Beer,Bread
t_5	Beer,Milk

- Find all the association rules:

$X \Rightarrow Y$	s	α
Bread $\Rightarrow$ PeanutButter	60%	75%
PeanutButter $\Rightarrow$ Bread	60%	100%
Beer $\Rightarrow$ Bread	20%	50%
PeanutButter $\Rightarrow$ Jelly	20%	33.3%
Jelly $\Rightarrow$ PeanutButter	20%	100%
Jelly $\Rightarrow$ Milk	0%	0%

The model: data

- $I = \{i_1, i_2, \ldots, i_m\}$: a set of *items*.
- Transaction t :
  - t a set of items, and $t \subseteq I$.
- Transaction Database T: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$.

Transaction data: supermarket data

- Market basket transactions:

  t1: {bread, cheese, milk}

  t2: {apple, eggs, salt, yogurt}

  … …

  tn: {biscuit, eggs, milk}

- Concepts:
  - An *item*: an item/article in a basket
  - *I*: the set of all items sold in the store
  - A *transaction*: items purchased in a basket; it may have TID (transaction ID)
  - A *transactional dataset*: A set of transactions

Transaction data: a set of documents

- **A text document data set. Each document is treated as a "bag" of keywords**

doc1: Student, Teach, School

doc2: Student, School

doc3: Teach, School, City, Game

doc4: Baseball, Basketball

doc5: Basketball, Player, Spectator

doc6: Baseball, Coach, Game, Team

doc7: Basketball, Team, City, Game

The model: rules

- A transaction t contains X, a set of items (itemset) in I, if $X \subseteq t$.
- An association rule is an implication of the form:

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

- An itemset is a set of items.
  - E.g., X = {milk, bread, cereal} is an itemset.
- A k-itemset is an itemset with k items.
  - E.g., {milk, bread, cereal} is a 3-itemset

Rule strength measures

- Support: The rule 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 holds in T with confidence *conf* if *conf*% of tranactions that contain X also contain Y.
  - $conf = \Pr(Y \mid X)$
- An association rule is a pattern that states when X occurs, Y occurs with certain probability.

Support and Confidence

- Support count: The support count of an itemset X, denoted by $X.count$, in a data set T is the number of transactions in T that contain X. Assume T has n transactions.
- Then,

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

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

Goal and key features

- **Goal:** Find all rules that satisfy the user-specified minimum support (minsup) and minimum confidence (minconf).
- **Key Features**
  - Completeness: find all rules.
  - No target item(s) on the right-hand-side
  - Mining with data on hard disk (not in memory)

An example

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

- Transaction data
- Assume:

  minsup = 30%

  minconf = 80%
- An example frequent *itemset*:

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

- Association rules from the itemset:

Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]

… …

Clothes, Chicken → Milk, [sup = 3/7, conf = 3/3]

Transaction data representation

- A simplistic view of shopping baskets,
- Some important information not considered. E.g,
  - the quantity of each item purchased and
  - the price paid.

Many mining algorithms

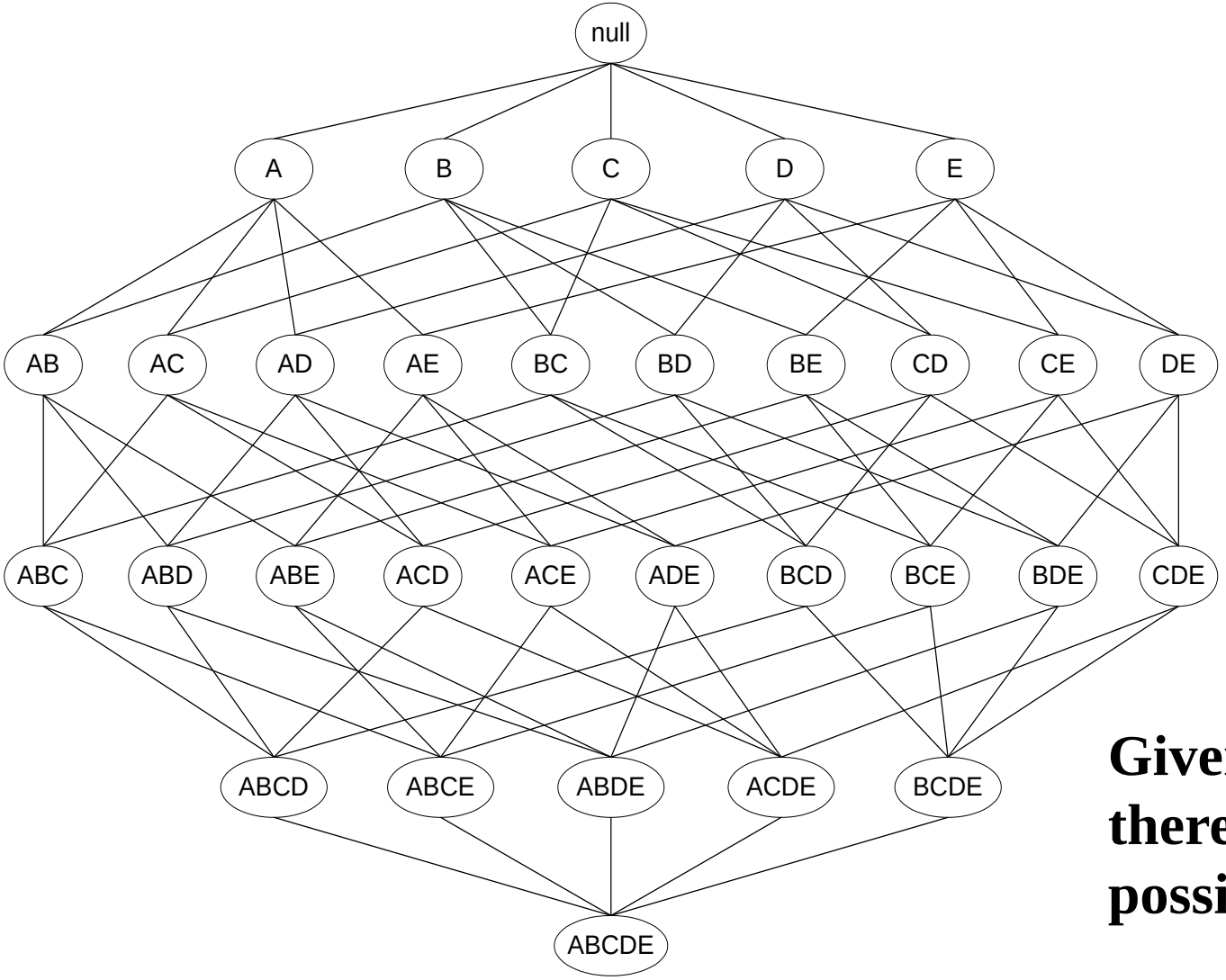
- There are a large number of them!!
- They use different strategies and data structures.
- Their resulting sets of rules are all the same.
  - Given a transaction data set *T*, and a minimum support and a minimum confident, the set of association rules existing in *T* is uniquely determined.
- Any algorithm should find the same set of rules although their computational efficiencies and memory requirements may be different.
- We study only one: the Apriori Algorithm

Mining Frequent Itemsets task

- Input: A set of transactions T, over a set of items I
- Output: All possible itemsets

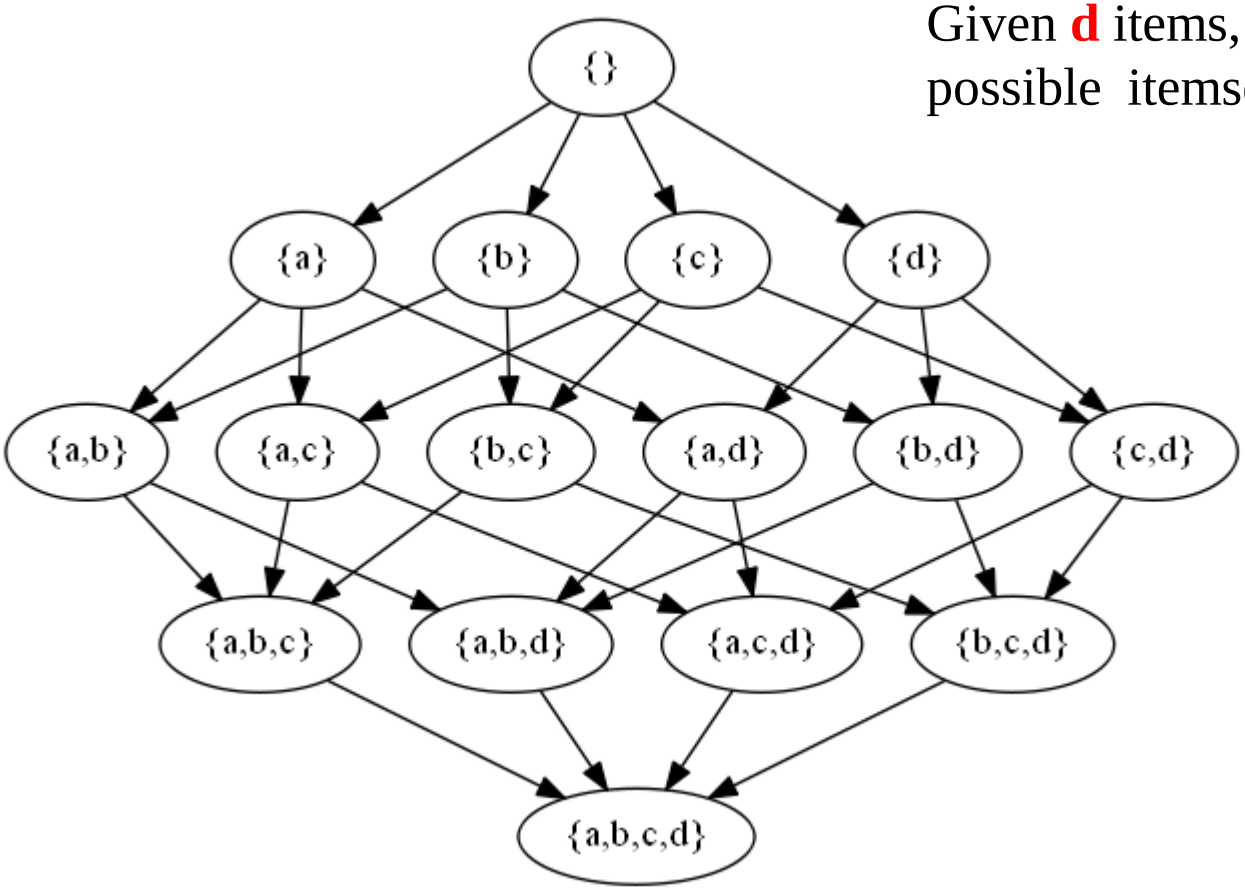
- Problem parameters:
  - N = |T|: number of transactions
  - d = |I|: number of (distinct) items
  - w: max width of a transaction
  - Number of possible itemsets $M = 2^d$?

Frequent Itemset Generation Network



Given d items, there are 2^d possible itemsets

Frequent Itemset Generation Network



Given **d** items, there are 2^d possible itemsets

A Binary Data Matrix of a Transactions Database

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

	Beer	Bread	Milk	Diaper	Eggs	Coke
T_1	0	1	1	0	0	0
T_2	1	1	0	1	1	0
T_3	1	0	1	1	0	1
T_4	1	1	1	1	0	0
T_5	0	1	1	1	0	1

Thank you