Applied Analytics and Predictive Modeling

Spring 2020

Lecture-9

Lydia Manikonda

manikl@rpi.edu

Rensselaer

Some of the slides adapted from Intro to Data Mining Tan et al. 2nd edition

Today’s agenda

- 2-slide project presentations
- Class Exercise
- Association Rules

2-slide project Presentations

Association Rules

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, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

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

Implication means co-occurrence, not causality!

Definitions

- **Itemset**
  - A 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**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 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

Association Rule

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

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

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

Example – Association Rule

- {Milk, Diaper} => {Beer}

- Support

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

- Confidence

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

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**

Association Rule Mining Task

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

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

  ⇒ Computationally prohibitive!

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 of Rules:

{Milk,Diaper} $\rightarrow$ {Beer} (s=0.4, c=0.67)
{Milk,Beer} $\rightarrow$ {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} $\rightarrow$ {Milk} (s=0.4, c=0.67)
{Beer} $\rightarrow$ {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} $\rightarrow$ {Milk,Beer} (s=0.4, c=0.5)
{Milk} $\rightarrow$ {Diaper,Beer} (s=0.4, c=0.5)

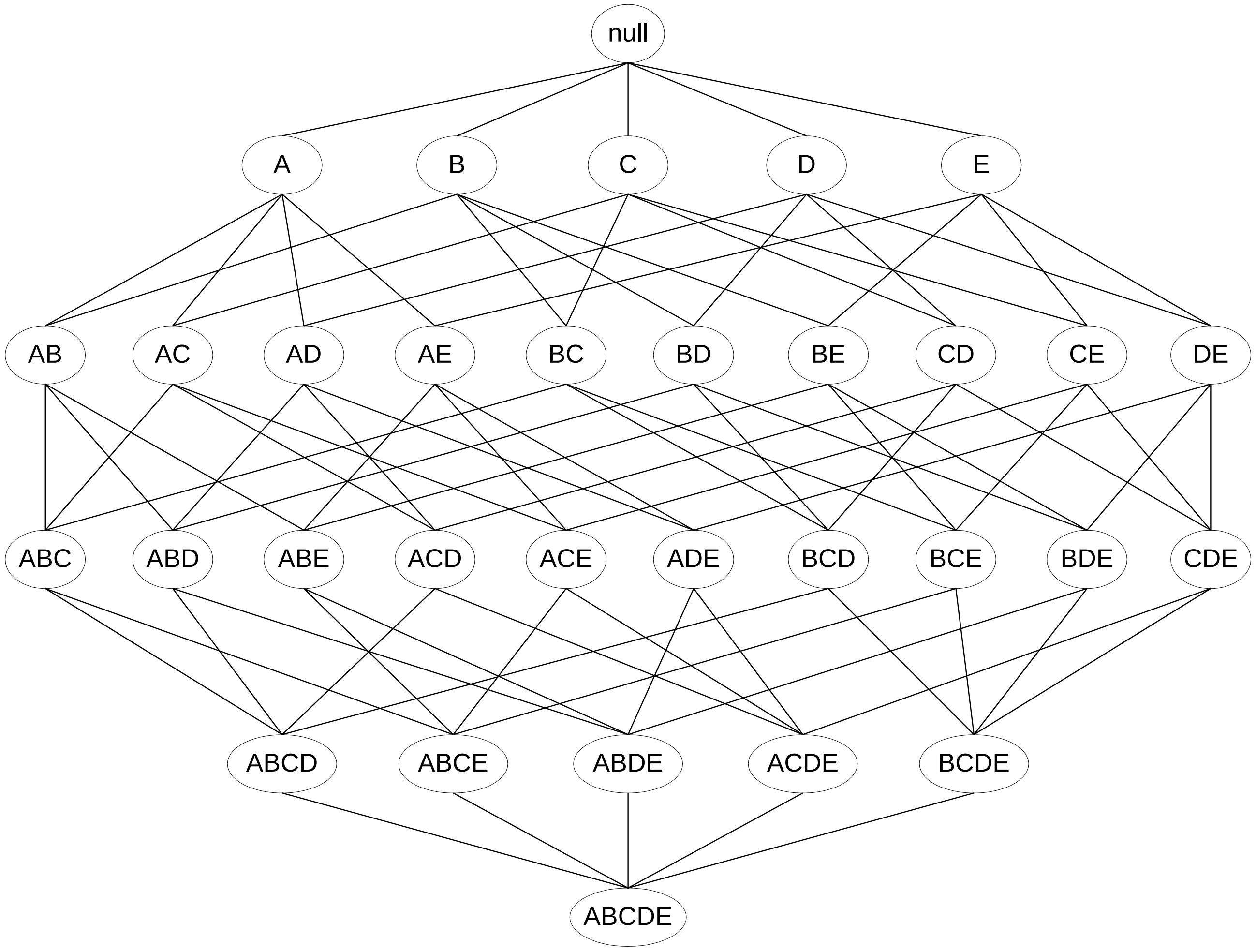
Observations:

- 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

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 binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

Frequent Itemset Generation



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

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

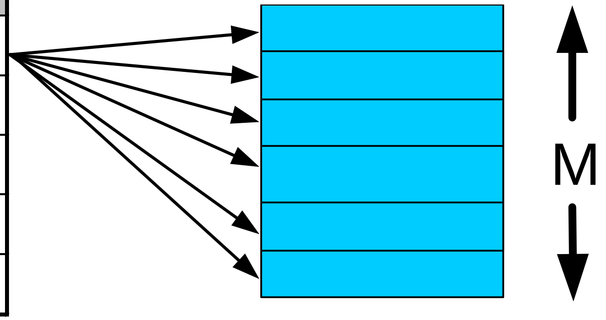
Transactions

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

N

w

List of Candidates



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

Frequent Itemset Generation Strategies

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by DHP and vertical-based mining algorithms
- Reduce the number of comparisons (NM)
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction

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:

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

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

Apriori Algorithm

- F_k: frequent k-itemsets
- L_k: candidate k-itemsets

- Algorithm
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: Generate L_{k+1} from F_k
    - **Candidate Pruning**: Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
    - **Support Counting**: Count the support of each candidate in L_{k+1} by scanning the DB
    - **Candidate Elimination**: Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}

Exercise-1 – Solved in the last class

Transaction 1	**Apple, beer, rice, chicken**
Transaction 2	**Apple, beer, rice**
Transaction 3	**Apple, beer**
Transaction 4	**Milk, beer, rice, chicken**
Transaction 5	**Milk, beer, rice**
Transaction 6	**Milk, beer**

Find all the frequent itemsets where, *min_sup* = 0.2

Exercise-2

- Using Apriori algorithm, identify frequent itemsets where *min_sup* =2

Transaction 1	**a, b, e**
Transaction 2	**b, d**
Transaction 3	**b, c**
Transaction 4	**a, b, d**
Transaction 5	**a, c**
Transaction 6	**b, c**
Transaction 7	**a, c**
Transaction 8	**a, b, c, e**
Transaction 9	**a, b, c**

Association Rules

- Association rules
  - An implication expression of the form X $\rightarrow$ Y, where X and Y are itemsets
- Rule: X => Y (X is antecedent; Y is consequent)
- Support = $\frac{Frequency(X\ and\ Y)}{Total\ \#\ transactions}$
- Confidence = $\frac{Frequency(X\ and\ Y)}{Frequency(X)}$
- Lift = $\frac{Support}{Support(X) * Support(Y)}$

A **lift** value greater than 1 indicates positive dependence between the antecedent and consequent – that the **antecedent and consequent** appear more often together than expected

Association Rules

- Support = $\frac{Frequency(X\ and\ Y)}{Total\ \#\ transactions}$
- Tells us how often the itemset appears in the dataset

- Confidence = $\frac{Frequency(X\ and\ Y)}{Frequency(X)}$
- Tells us how often the rule is true with regard to our dataset

- Lift = $\frac{Support}{Support(X) * Support(Y)}$
- Tells us if X and Y are independent of each other

Association Rules Examples

- Rule-1: A => D

- Support = Freq(A, D)/Total #T
  = 2/5

- Confidence = Freq(A,D)/Freq(A)
  = 2/3

- Lift = Support/Supp(A)*Support(D) = (2/5)/((3/5)*(3/5)) = (2/5)/(9/25) = 10/9 >1 so dependent on each other.

Transaction ID	Items
T1	**A, B, C**
T2	**A, C, D**
T3	**B, C, D**
T4	**A, D, E**
T5	**B, C, E**

Example-2

- Rule: C => A

Transaction ID	Items
T1	A, B, C
T2	A, C, D
T3	B, C, D
T4	A, D, E
T5	B, C, E

Example-3

- Rule A => C

Transaction ID	Items
T1	A, B, C
T2	A, C, D
T3	B, C, D
T4	A, D, E
T5	B, C, E

Example-4

- {B, C} => D

Transaction ID	Items
T1	A, B, C
T2	A, C, D
T3	B, C, D
T4	A, D, E
T5	B, C, E