ASSOCIATION RULE MINING

LECTURE 23

Dr. Vani V

What Is Pattern Discovery?

What are patterns?

- *Patterns: A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set*
- *Patterns represent intrinsic and important properties of datasets*

Pattern discovery: Uncovering patterns from massive data sets

Motivation examples:

- *What products were often purchased together?*
- *What are the subsequent purchases after buying an iPad?*
- *What code segments likely contain copy-and-paste bugs?*
- *What word sequences likely form phrases in this corpus?*

Pattern Discovery: Why Is It Important?

Finding **inherent regularities** in a data set

Foundation for many essential data mining 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*

Broad applications

- ***Market basket analysis, cross-marketing, catalog design, sale campaign analysis, Web log analysis, biological sequence analysis***

Association Rule Mining (ARM)

- Association rules mining (or market basket analysis) searches for interesting customer habits by looking at associations.
- The classical example is the one where a store in USA was reported to have discovered that people buying nappies tend also to buy beer. Not sure if this is actually true.
- Applications in marketing, store layout, customer segmentation, medicine, finance, and many more.

Question: Suppose you are able to find that two products x and y (say bread and milk or crackers and cheese) are frequently bought together. How can you use that information?

Transaction ID	Items
10	Bread, Cheese, Newspaper
20	Bread, Cheese, Juice
30	Bread, Milk
40	Cheese, Juice. Milk, Coffee
50	Sugar, Tea, Coffee, Biscuits, Newspaper
60	Sugar, Tea, Coffee, Biscuits, Milk, Juice, Newspaper
70	Bread, Cheese
80	Bread, Cheese, Juice, Coffee
90	Bread, Milk
100	Sugar, Tea, Coffee, Bread, Milk, Juice, Newspaper

Terminology…

- Let the number of different *items* sold be n
- Let the number of *transactions* be N
- Let the set of items be $\{i_1, i_2, \ldots, i_n\}$. The number of items may be large, perhaps several thousands.
- Let the set of transactions be $\{t_1, t_2, \ldots, t_N\}$. Each transaction t_i contains a subset of items from the *itemset* $\{i_1, i_2, \ldots, i_n\}$. These are the things a customer buys when they visit the supermarket. N is assumed to be large, perhaps in millions.
- Not considering the quantities of items bought.

Terminology…

- Want to find a group of items that tend to occur together frequently.
- The association rules are often written as X→Y meaning that whenever X appears Y also tends to appear. X and Y may be single items or sets of items, but the same item does not appear in both.

Terminology…

- Suppose X and Y appear together in only 1% of the transactions but whenever X appears there is 80% chance that Y also appears.
- The 1% presence of X and Y *together* is called the ***support (or prevalence)*** of the rule and 80% is called the ***confidence (or predictability)*** of the rule.
- These are measures of interestingness of the rule.

Terminology…

- **Confidence** denotes the **strength of the association between X and Y**. **Support indicates the frequency of the pattern**. A minimum support is necessary if an association is going to be of some business value.
- Let the chance of finding an item X in the N transactions is x% then we can say probability of X is P(X) = x/100 since probability values are always between 0.0 and 1.0.

Question: Why is minimum support necessary? Why is minimum confidence necessary?

Terminology…

- Suppose we know P(X U Y) then what is the probability of Y appearing if we know that X already exists. It is written as $P(Y \mid X)$.
- The **support for X→Y is the probability of both X and Y appearing** together, that is **P(X U Y).**
- The **confidence of X→Y is the conditional probability** of Y appearing given that X exists. It is written as **$P(Y \mid X)$** and read as P of Y given X.

Terminology

- Sometime the term *lift* is also used. Lift is defined as Support(X U Y)/P(X)P(Y). P(X)P(Y) is the probability of X and Y appearing together if both X and Y appear randomly.
- As an example, if support of X and Y is 1%, and X appears 4% in the transactions while Y appears in 2%, then lift = 0.01/0.04 x 0.02 = 1.25. What does it tell us about X and Y? What if lift was 1?

The task

Want to find all associations which have at least p% support with at least q% confidence such that

- *all rules satisfying any user constraints are found*
- *the rules are found efficiently from large databases*
- *the rules found are actionable*

Applications

Although we are only considering basket analysis, the technique has many other applications as noted earlier.

For example we may have many patients coming to a hospital with various diseases and from various backgrounds. We therefore have a number of attributes (items) and we may be interested in knowing which attributes appear together and may be which attributes are frequently associated to some particular disease.

Question: Can you think of some applications of ARM in an educational institution?

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

Implication means co-occurrence, not causality!

Definition: Frequent Itemset

- **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

Definition: Association Rule

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

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 Y appear in transactions that contain X

Example:

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

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

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

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} → {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)

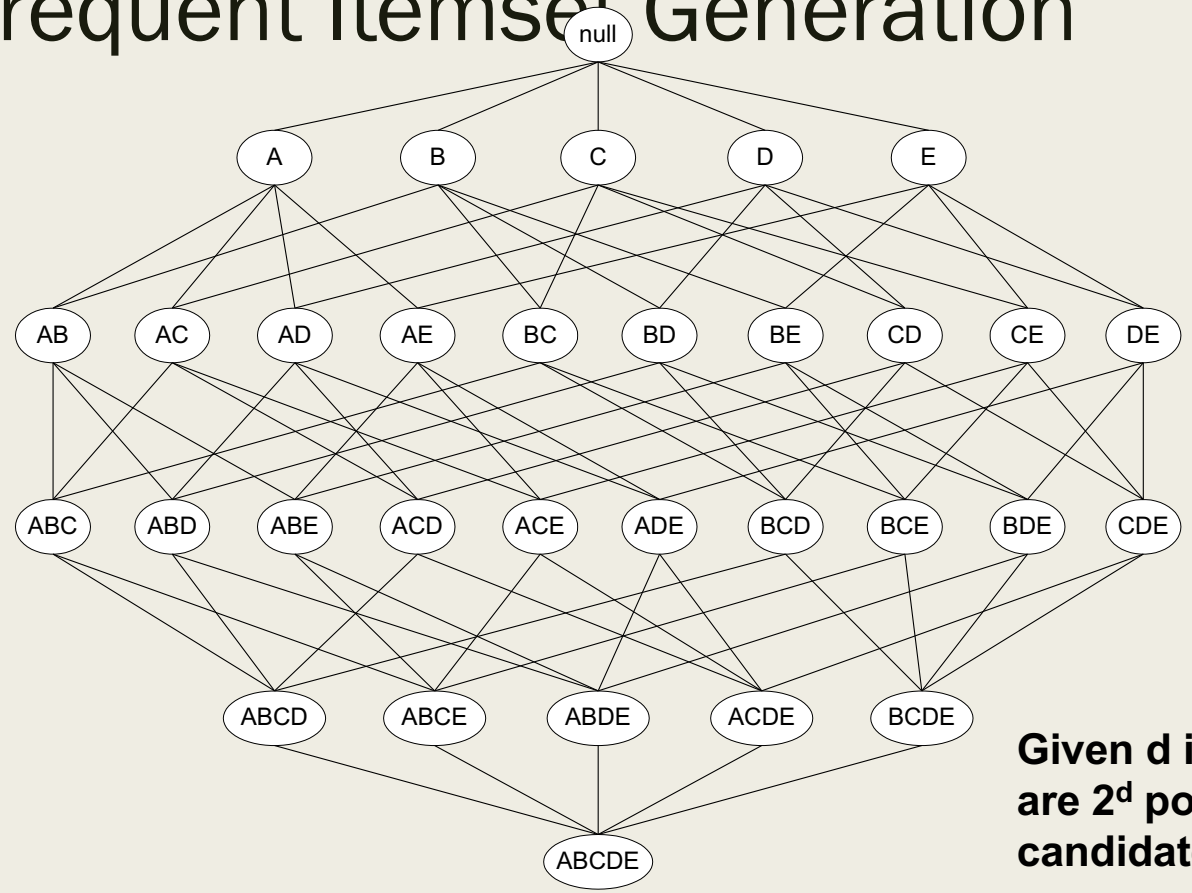
Observations:

- 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

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

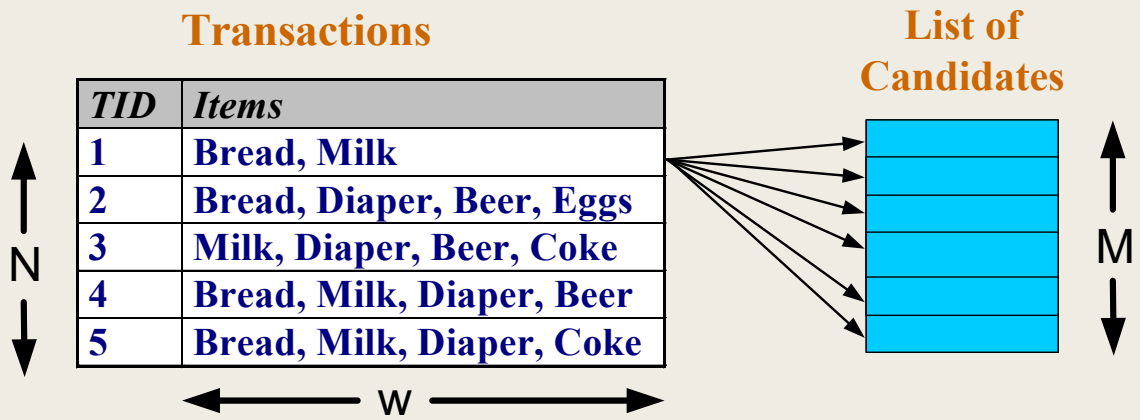
Frequent Itemset Generation



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

Frequent Itemset Generation

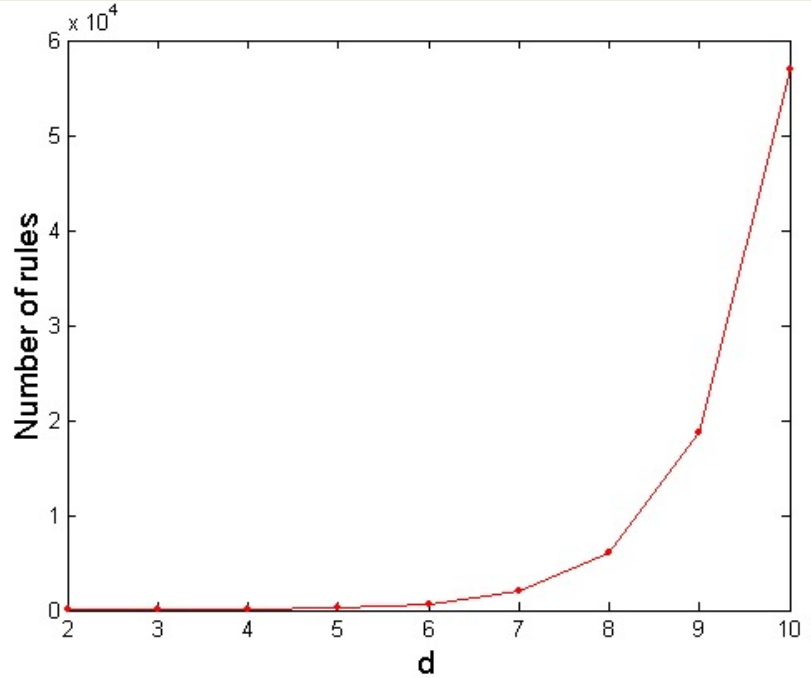
- Brute-force approach:
  - *Each itemset in the lattice is a candidate frequent itemset*
  - *Count the support of each candidate by scanning the database*



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

Computational 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

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*

LECTURE 24

Dr. Vani V

Example

We have 9 items and 10 transactions. Find support and confidence for each pair. How many pairs are there?

Transaction ID	Items
10	Bread, Cheese, Newspaper
20	Bread, Cheese, Juice
30	Bread, Milk
40	Cheese, Juice. Milk, Coffee
50	Sugar, Tea, Coffee, Biscuits, Newspaper
60	Sugar, Tea, Coffee, Biscuits, Milk, Juice, Newspaper
70	Bread, Cheese
80	Bread, Cheese, Juice, Coffee
90	Bread, Milk
100	Sugar, Tea, Coffee, Bread, Milk, Juice, Newspaper

Example

There are 9 x 8 or 72 pairs, half of them duplicates. 36 is too many to analyse in a class, so we use an even **simpler example of n = 4 (Bread, Cheese, Juice, Milk) and N = 4. We want to find rules with minimum support of 50% and minimum confidence of 75%.**

Transaction ID	Items
100	Bread, Cheese
200	Bread, Cheese, Juice
300	Bread, Milk
400	Cheese, Juice. Milk

Example

N = 4 results in the following frequencies. All items have the 50% support we require. Only two pairs have the 50% support and no 3-itemset has the support.

We will look at the two pairs that have the minimum support to find if they have the required confidence.

Itemsets	Frequency
Bread	3
Cheese	3
Juice	2
Milk	2
(Bread, Cheese)	2
(Bread, Juice)	1
(Bread, Milk)	1
(Cheese, Juice)	2
(Cheese, Milk)	1
(Juice, Milk)	1
(Bread, Cheese, Juice)	1
(Bread, Cheese, Milk)	0
(Bread, Juice, Milk)	0
(Cheese, Juice, Milk)	1
(Bread, Cheese, Juice, Milk)	0

Terminology

Items or itemsets that have the minimum support are called *frequent*. In our example, all the four items and two pairs are frequent.

We will now determine if the two pairs {Bread, Cheese} and {Cheese, Juice} lead to association rules with 75% confidence.

Every pair {A, B} can lead to two rules A ⟶ B and B ⟶ A if both satisfy the minimum confidence. Confidence of A ⟶ B is given by the support for A and B together divided by the support of A.

Rules

We have four possible rules and their confidence is given as follows:

Bread $\rightarrow$ *Cheese with confidence of 2/3 = 67%*

Cheese $\rightarrow$ *Bread with confidence of 2/3 = 67%*

Cheese $\rightarrow$ *Juice with confidence of 2/3 = 67%*

Juice $\rightarrow$ *Cheese with confidence of 100%*

Therefore only the last rule Juice $\rightarrow$ Cheese has confidence above the minimum 75% and qualifies. Rules that have more than user-specified minimum confidence are called *confident*.

Problems with Brute Force

This simple algorithm works well with four items since there were only a total of 16 combinations that we needed to look at but if the number of items is say 100, the number of combinations is much larger, in billions.

The number of combinations becomes about a million with 20 items since the number of combinations is 2^n with n items (why?). The naïve algorithm can be improved to deal more effectively with larger data sets.

Problems with Brute Force

Question: Can you think of an improvement over the brute force method in which we looked at every itemset combination possible?

Naïve algorithms even with improvements don’t work efficiently enough to deal with large number of items and transactions. We now define a better algorithm.

Improved Brute Force

Rather than counting all possible item combinations we can look at each transaction and count only the combinations that occur as shown below (that is, we don't count itemsets with zero frequency).

Transaction ID	Items	Combinations
100	Bread, Cheese	{Bread, Cheese}
200	Bread, Cheese, Juice	{Bread, Cheese}, {Bread, Juice}, {Cheese, Juice}, {Bread, Cheese, Juice}
300	Bread, Milk	{Bread, Milk}
400	Cheese, Juice, Milk	{Cheese, Juice}, {Cheese, Milk}, {Juice, Milk}, {Cheese, Juice, Milk}

Improved Brute Force

Removing zero frequencies leads to the smaller table below. We then proceed as before, but the improvement is not large and we need better techniques.

Itemsets	Frequency
Bread	3
Cheese	3
Juice	2
Milk	2
(Bread, Cheese)	2
(Bread, Juice)	1
(Bread, Milk)	1
(Cheese, Juice)	2
(Juice, Milk)	1
(Bread, Cheese, Juice)	1
(Cheese, Juice, Milk)	1

The Apriori Algorithm

To find associations, this classical Apriori algorithm may be simply described by a two-step approach:

Step 1 — discover all frequent (single) items that have support above the minimum support required

Step 2 – use the set of frequent items to generate the association rules that have high enough confidence level

Apriori Algorithm Terminology

- A k-itemset is a set of k items.
- The set C_k is a set of candidate k-itemsets that are potentially frequent.
- The set L_k is a subset of C_k and is the set of k-itemsets that are frequent.

Step-by-step Apriori Algorithm

- Step 1 – Computing L_1

  Scan all transactions. Find all frequent items that have support above the required p%. Let these frequent items be labeled L_1.

- Step 2 – Apriori-gen Function

  Use the frequent items L_1 to build all possible item pairs like {Bread, Cheese} if Bread and Cheese are in L_1. The set of these item pairs is called C_2, the *candidate set*.

Question: Do all members of C_2 have the support? Why?

The Apriori Algorithm

- Step 3 – Pruning

  Scan all transactions and find all pairs in the candidate pair set C_2 that are frequent. Let these frequent pairs be L_2.

- Step 4 – General rule

  A generalization of Step 2. Build candidate set of k items C_k by combining frequent itemsets in the set L_{k-1}.

The Apriori Algorithm

- Step 5 – Pruning

  Step 5, generalization of Step 3. Scan all transactions and find all item sets in C_k that are frequent. Let these frequent itemsets be L_k.

- Step 6 – Continue

  Continue with Step 4 unless L_k is empty.

- Step 7 – Stop

  Stop when L_k is empty.

Reducing Number of Candidates

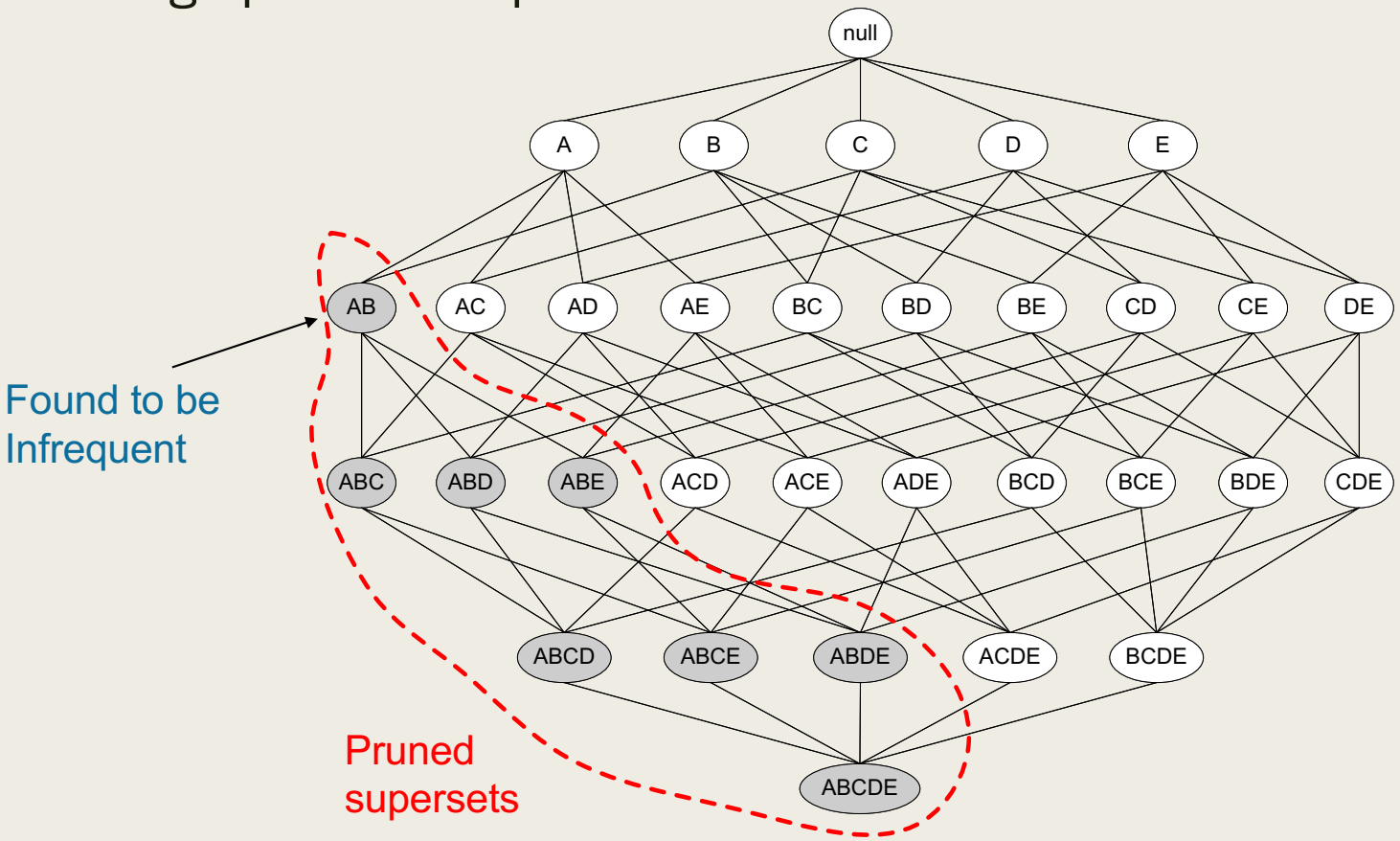
- Apriori principle:
  - *If an itemset is frequent, then all of its subsets must also be frequent*

- 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*

Illustrating Apriori Principle



Illustrating Apriori Principle

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

Items (1-itemsets)

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
$^{6}C_{1} + {}^{6}C_{2} + {}^{6}C_{3} = 41$
With support-based pruning,
$6 + 6 + 1 = 13$

Apriori Algorithm

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

An Example

This example is like the last example, but we have added two more items and another transaction making five transactions and six items. We want to find association rules with 50% support and 75% confidence.

Transaction ID	Items
100	Bread, Cheese, Eggs, Juice
200	Bread, Cheese, Juice
300	Bread, Milk, Yogurt
400	Bread, Juice, Milk
500	Cheese, Juice, Milk

Example

First find L_1. 50% support requires that each frequent item appear in at least three transactions. Therefore L_1 is given by:

Item	Frequnecy
Bread	4
Cheese	3
Juice	4
Milk	3

Example

The candidate 2-itemsets or C_2 therefore has six pairs (why?). These pairs and their frequencies are:

Item Pairs	Frequency
(Bread, Cheese)	2
(Bread, Juice)	3
(Bread, Milk)	2
(Cheese, Juice)	3
(Cheese, Milk)	1
(Juice, Milk)	2

Question: Why did we not select (Egg, Milk)?

Deriving Rules

L_2 has only two frequent item pairs {Bread, Juice} and {Cheese, Juice}. After these two frequent pairs, there are no candidate 3-itemsets (why?) since we do not have two 2-itemsets that have the same first item (why?).

The two frequent pairs lead to the following possible rules:

Bread → *Juice*

Juice → *Bread*

Cheese → *Juice*

Juice → *Cheese*

Deriving Rules

The confidence of these rules is obtained by dividing the support for both items in the rule by the support of the item on the left hand side of the rule.

The confidence of the four rules therefore are

3/4 = 75% (Bread → Juice)

3/4 = 75% (Juice → Bread)

3/3 = 100% (Cheese → Juice)

3/4 = 75% (Juice → Cheese)

Since all of them have a minimum 75% confidence, they all qualify. We are done since there are no 3–itemsets.

References

- TEXTBOOKS :

1. Pang-Ning Tan, Vipin Kumar, Michael Steinbach: **Chapter 6, Introduction to Data Mining**, Pearson, 2012.
2. Jiawei Han and Micheline Kamber: **Chapter 6, Data Mining - Concepts and Techniques**, 3rd Edition, MorganKaufmann Publisher, 2014