Association Rule Mining

Lesson 4

Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection;
  - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

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

Rules Discovered: (Example only)

{Milk} --> {Coke}

{Diaper, Milk} --> {Beer}

Association Rule Discovery: Application 1

- Marketing and Sales Promotion:
  - Let the rule discovered be

    {Coke, … } --> {Potato Chips}

  - Potato Chips as consequent => Can be used to determine what should be done to boost its sales.

  - Coke in the antecedent => can be used to see which products would be affected if the store discontinues selling coke.

  - Coke in antecedent *and* Potato chips in consequent => Can be used to see what products should be sold with coke to promote sale of Potato chips!

Association Rule Discovery: Application 2

- Supermarket shelf management.
  - Goal: To identify items that are bought together by sufficiently many customers.
  - Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
  - A classic rule --
    - If a customer buys diaper and milk, then he is very likely to buy beer:

$$Diapers \rightarrow Beer,\ \textit{support} = 20\%,\ \textit{confidence} = 85\%$$

Association rule discovery basic concepts

- Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of items.
- Let D, the task-relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID.
- Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$

- An association rule is an implication of the form
$$A \Rightarrow B, \text{where } A \subset I, B \subset I, \text{and } A \cap B = \phi.$$
- The rule $A \Rightarrow B$ holds in the transaction set *D* with support *s*, where *s* is the percentage of transactions in *D* that contain *A u B* (i.e., the *union* of sets *A* and *B*, or say, both *A* and *B*). This is taken to be the probability, *P*(*A u B*).
- The rule $A \Rightarrow B$ has confidence *c* in the transaction set *D*, where *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, *P*(*B*|*A*). That is,

$$\begin{aligned} support(A \Rightarrow B) &= P(A \cup B) \\ confidence(A \Rightarrow B) &= P(B|A). \end{aligned}$$

- Rules that satisfy both a minimum support threshold (*min sup*) and a minimum confidence threshold (*min conf*) are called strong. By convention, we write support and confidence values so as to occur between 0% and 100%, rather than 0 to 1.0.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

- In general, association rule mining can be viewed as a two-step process:

- **1.** Find all frequent itemsets:
  - By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min sup*.

- **2.** Generate strong association rules from the frequent itemsets:
  - By definition, these rules must satisfy minimum support and minimum confidence.

Let D be database of transactions

- e.g.:

Transaction ID	Items
1000	A, B, C
2000	A, B
3000	A, D
4000	B, E, F

- Let I be the set of items that appear in the database, e.g., $I=\{A,B,C,D,E,F\}$
  - Each transaction t is a subset of I
- A ***rule*** is an implication among ***itemsets*** X and Y, of the form by $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$
  - e.g.: $\{B,C\} \rightarrow \{A\}$ is a rule

Itemset

- A set of one or more items
  - E.g.: {Milk, Bread, Diaper}
- k-itemset
  - An itemset that contains k items

Support count (σ)

- Frequency of occurrence of an itemset (number of transactions it appears)
- E.g. σ({Milk, Bread,Diaper}) = 2

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

Support

- Fraction of the transactions in which an itemset appears
- E.g. s({Milk, Bread, Diaper}) = 2/5

Frequent Itemset

- An itemset whose support is greater than or equal to a ***minsup*** threshold

Association Rule discovery steps

1. Find the *frequent* itemsets

   (item sets are the sets of items that have minimum support)

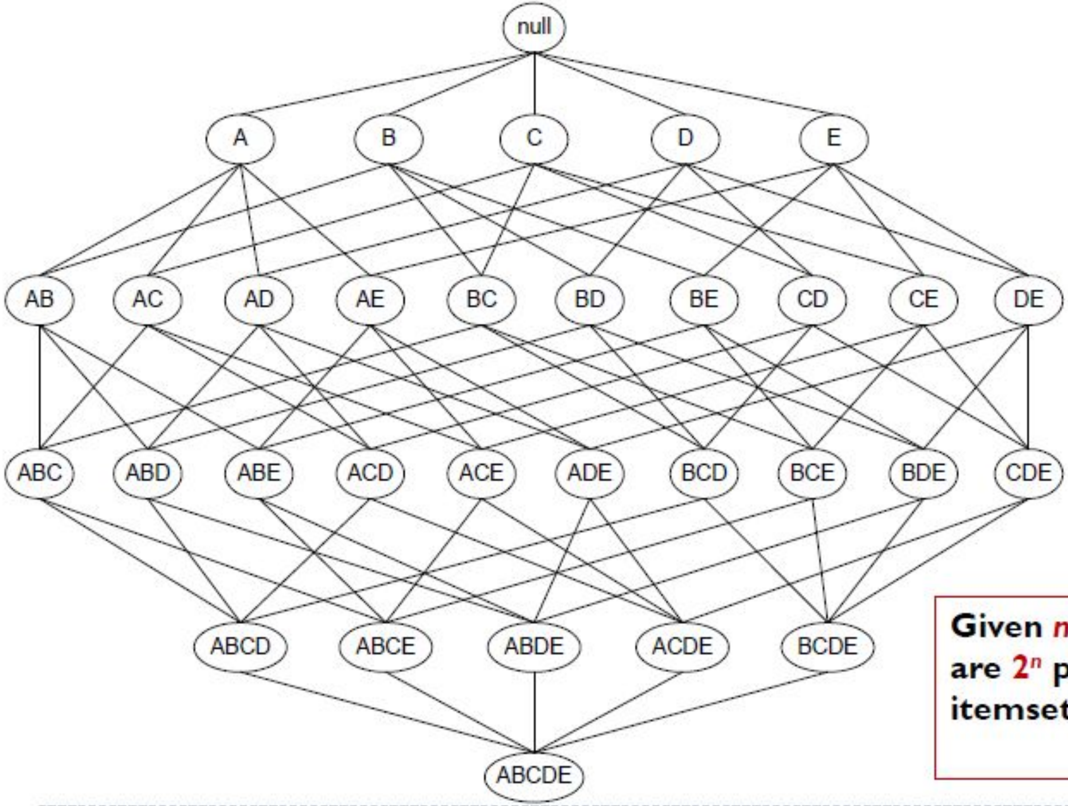
2. Use the frequent itemsets to generate association rules

Brute Force Algorithm:

- **List all possible itemsets and compute their support**
- **Generate all rules from frequent itemset**
- **Prune rules that fail the *minconf* threshold**

Would this work?!

How many itemsets are there?



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

Scalable methods for mining Frequent Patterns

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
    - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
      - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}

- Scalable mining methods: Three major approaches
  - **Apriori** (Agrawal & Srikant@VLDB'94)
  - **Freq. pattern growth** (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - **Vertical data format approach** (Charm—Zaki & Hsiao @SDM'02)

Apriori Algorithm

C_k : Candidate itemset of size k
L_k : Frequent itemset of size k

L_1 = {frequent items};
for ($k = 1$; L_k != $\varnothing$; k++) **do begin**
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database **do**
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$;

Join Step: C_k is generated by joining L_{k-1} with itself

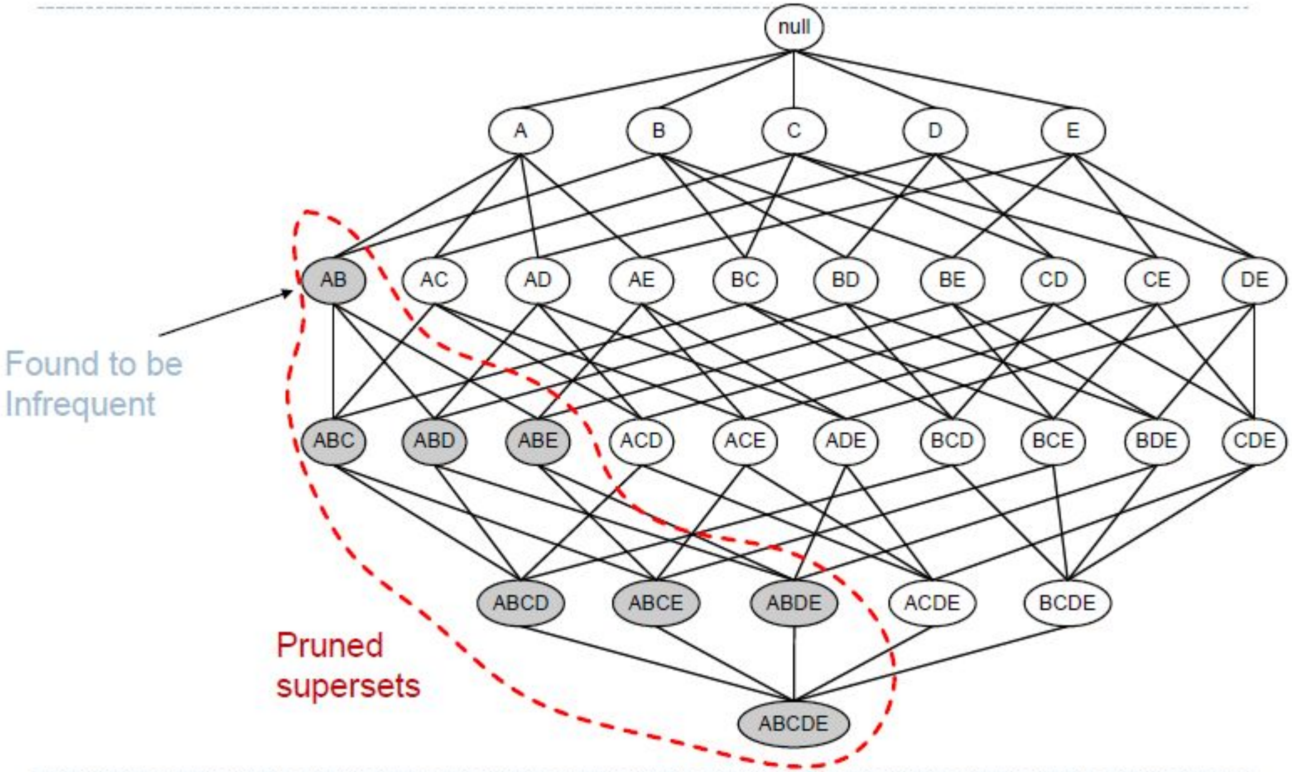
Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Apriori

- Support is "downward closed"
  - If an itemset is frequent (has enough support), then all of its subsets must also be frequent
    - if {AB} is a frequent itemset, both {A} and {B} are frequent itemsets
  - This is due to the ***anti-monotone*** property of support

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

- **Corollary:** if an itemset doesn't satisfy minimum support, none of its supersets will either
  - this is essential for pruning search space)


null
A
B
C
D
E
AB
AC
AD
AE
BC
BD
BE
CD
CE
DE
ABC
ABD
ABE
ACD
ACE
ADE
BCD
BCE
BDE
CDE
ABCD
ABCE
ABDE
ACDE
BCDE
ABCDE
Found to be Infrequent
Pruned supersets

support based pruning

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

Items (1-itemsets)

minsup = 3/5

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)

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

Association Rule

- $X \rightarrow Y$, where X and Y are non-overlapping itemsets
- **{Milk, Diaper} → {Beer}**

Rule Evaluation Metrics

- **Support (s)**
  - Fraction of transactions that contain both X and Y
  - **i.e., support of the itemset $X \cup Y$**
- **Confidence (c)**
  - Measures how often items in Y appear in transactions that contain X

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

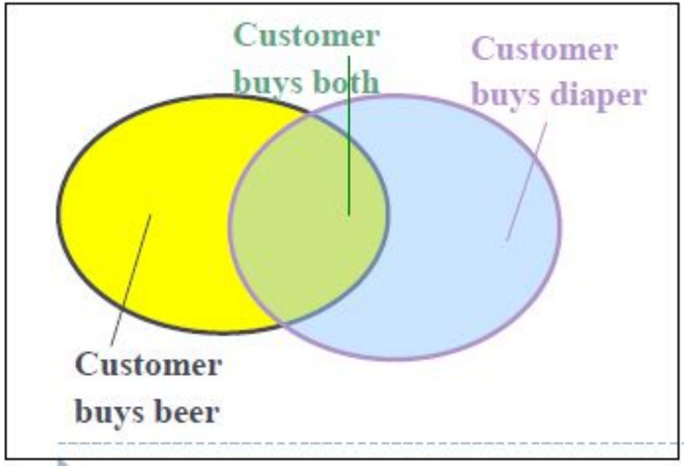
Example:

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

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

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

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset $X = \{x_1, \ldots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let $sup_{min} = 50\%$, $conf_{min} = 50\%$

Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}

Association rules:

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

Transaction ID	Items Bought
1001	A, B, C
1002	A, C
1003	A, D
1004	B, E, F
1005	A, D, F

Itemset {A, C} has a support of 2/5 = 40%

Rule {A} ==> {C} has confidence of 50%

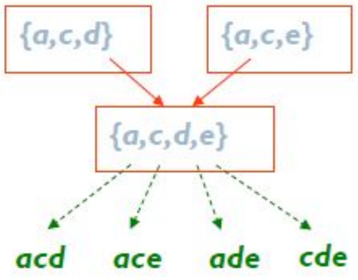
Rule {C} ==> {A} has confidence of 100%

Support for {A, C, E} ?
Support for {A, D, F} ?

Confidence for {A, D} ==> {F} ?
Confidence for {A} ==> {D, F} ?

Example of Generating Candidates

- L_3={*abc*, *abd*, *acd*, *ace*, *bcd*}
- Self-joining: $L_3 * L_3$
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4 = {*abcd*}



Apriori Example: (minsup = 2)

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

L_2

item set	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

L_3

itemset	sup
{2 3 5}	2

The final "frequent" item sets are those remaining in L2 and L3.
However, {2,3}, {2,5}, and {3,5} are all contained in the larger item set {2, 3, 5}. Thus, the final group of item sets reported by Apriori are {1,3} and {2,3,5}. These are the only item sets from which we will generate association rules.

- Item sets: {1,3} and {2,3,5}
- Recall that confidence of a rule LHS → RHS is Support of itemset (i.e. LHS ∪ RHS) divided by support of LHS.

Candidate rules for {1,3}		Candidate rules for {2,3,5}			
Rule	Conf.	Rule	Conf.	Rule	Conf.
{1}→{3}	2/2 = 1.0	{2,3}→{5}	2/2 = 1.00	{2}→{5}	3/3 = 1.00
{3}→{1}	2/3 = 0.67	{2,5}→{3}	2/3 = 0.67	{2}→{3}	2/3 = 0.67
		{3,5}→{2}	2/2 = 1.00	{3}→{2}	2/3 = 0.67
		{2}→{3,5}	2/3 = 0.67	{3}→{5}	2/3 = 0.67
		{3}→{2,5}	2/3 = 0.67	{5}→{2}	3/3 = 1.00
		{5}→{2,3}	2/3 = 0.67	{5}→{3}	2/3 = 0.67

Assuming a min. confidence of 75%, the final set of rules reported by Apriori are: {1}→{3}, {3,5}→{2}, {5}→{2} and {2}→{5}