Spring 2019

CSEN1083: Data Mining

Association Analysis

Seif Eldawlatly

Association Analysis

- Point-of-sale data collection (bar code scanners, radio frequency identification (RFID), and smart card technology) have allowed retailers to collect up-to-the-minute data

Association Analysis

- Discover patterns that describe strongly associated features in the data

- 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

- Example:

Transaction ID	Items
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Eggs}
9	{Bread, Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

Rules Discovered:
{Diapers} --> {Milk}
{Butter} --> {Bread}

Association Analysis

- Itemset: A collection of zero or more items. If an itemset contains k items, it is called a k-itemset

  Example: {Milk, Bread, Diaper} is a 3-itemset

- Transaction Width: number of items present in a transaction
- Example:

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

- Transaction 2 is of width 4 where the itemset {Bread, Diaper} is subset of it

Association Analysis

- Support Count (σ): the number of transactions that contain a particular itemset X

$$\sigma(X) = \left|\left\{t_i \,\middle|\, X \subseteq t_i, t_i \in T\right\}\right|$$

where | . | represents the number of transactions t_i for which the itemset X is subset and T is the set of all transactions in the dataset

- Example:
  - σ({Bread, Milk}) = 3
  - σ({Bread, Diaper, Wipes}) = 2

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

Association Analysis

- Association Rule: An association rule is an implication expression of the form $X \rightarrow Y$, where X and Y are disjoint itemsets

- The strength of an association rule can be measured in terms of its support and confidence

- Support: Determines how often a rule is applicable to a given dataset

- Measured by

$$s\left(X \rightarrow Y\right) = \frac{\sigma\left(X \cup Y\right)}{Total\ \ Number\ \ of\ \ Transactions}$$

Association Analysis

- Support is an important measure because a rule that has very low support may occur simply by chance

- Confidence: Determines how frequently items in *Y* appear in transactions that contain *X*

$$c\left(X \rightarrow Y\right) = \frac{\sigma\left(X \cup Y\right)}{\sigma\left(X\right)}$$

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

- Example:
- s({Milk, Diaper} → {Wipes}) = 2/5
- c({Milk, Diaper} → {Wipes}) = 2/3

- Confidence measures the reliability of the inference made by a rule

Association Rule Mining

- Given a set of transactions *T*, find all the rules having support ≥ *minsup* and confidence ≥ *minconf*, where *minsup* and *minconf* are the corresponding support and confidence thresholds

- Frequent Itemset: An itemset whose support is greater than or equal to a *minsup* 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

- This approach is computationally very expensive

Association Rule Mining

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

Example of Rules:

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

- Observations:
  - All the above rules are binary partitions of the same itemset: {Milk, Diaper, Wipes}
  - Rules originating from the same itemset have identical support but can have different confidence
  - Thus, we may decouple the support and confidence requirements

Association Rule Mining

- A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:

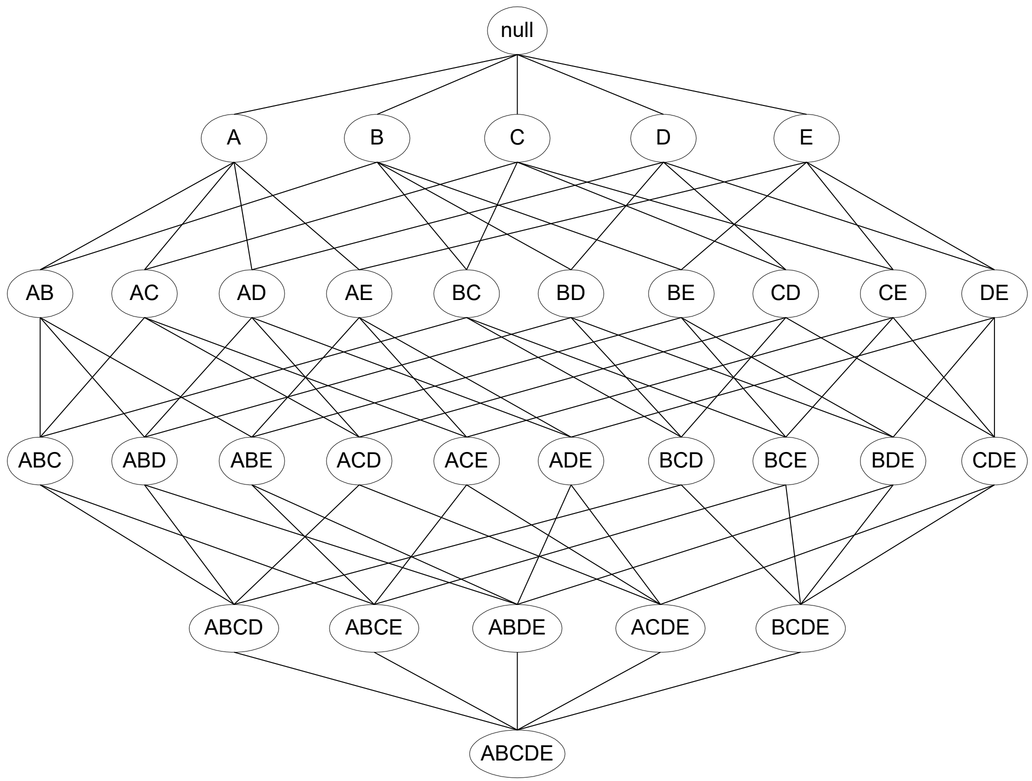
  1. Frequent Itemset Generation
     – Generate all itemsets whose support $\geq minsup$

  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

- A lattice structure can be used to enumerate the list of all possible itemsets

- Example: Itemset lattice for I: {A, B, C, D, E}
- Given d items, there are 2^d possible candidate itemsets



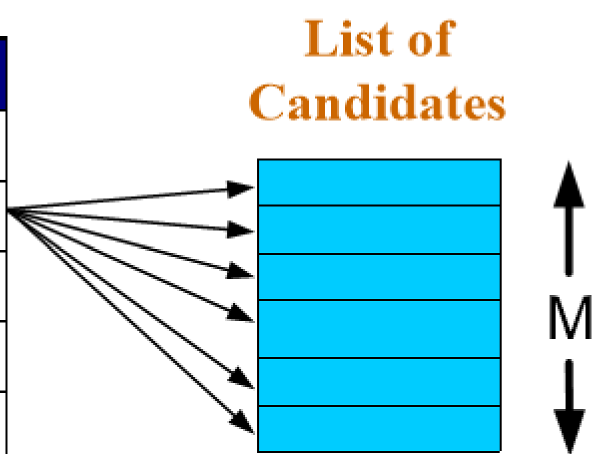
Frequent Itemset Generation

- Because itemsets can be very large in many practical applications, the search space of itemsets that need to be explored is exponentially large

- Brute-force approach: Determine the support count for every candidate itemset in the lattice structure

- Example: Match each transaction against every candidate

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



Frequent Itemset Generation Strategies

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M

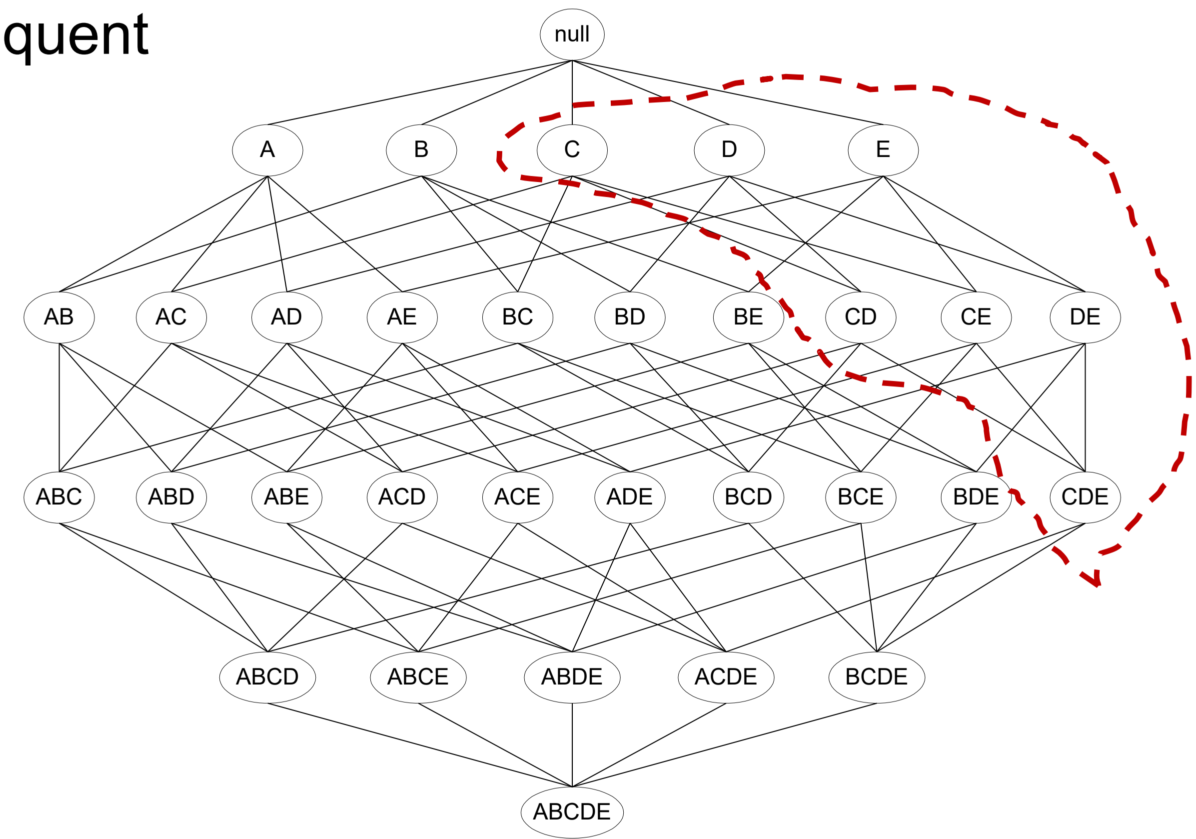
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by vertical-based mining algorithms

- 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

Apriori Algorithm

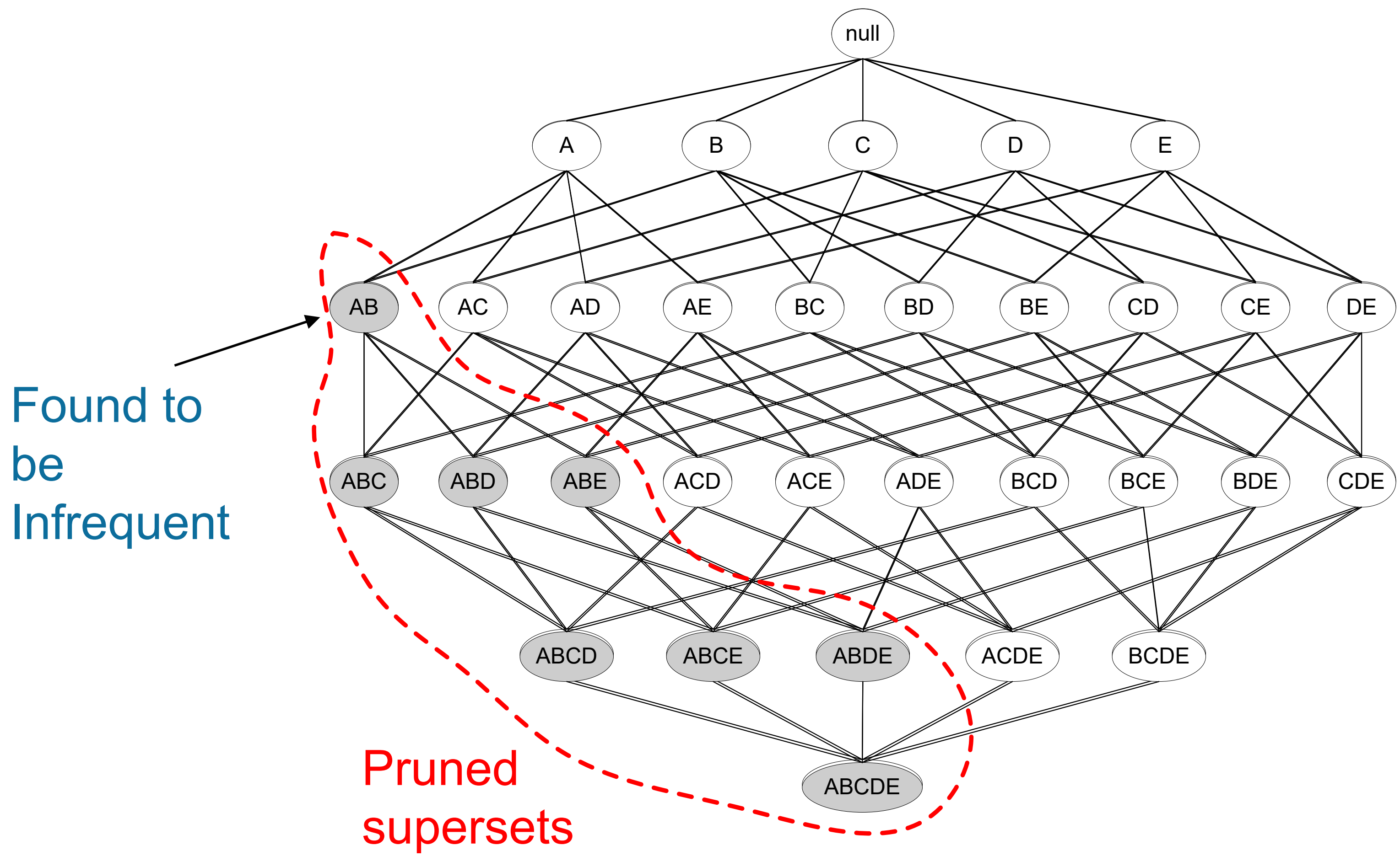
- Apriori Principle: If an itemset is frequent, then all of its subsets must also be frequent

- Example: Suppose that CDE is a frequent itemset, then any transaction that includes CDE will include its subsets. They will all be frequent



Apriori Algorithm

- Conversely, if an itemset such as {a, b} is infrequent, then all of its supersets must be infrequent too



Apriori Algorithm Frequent Itemset Generation

- Example: For the dataset below, assume that the support threshold is 60%, which is equivalent to a minimum support count equal to 3

- Generating frequent 1-itemsets

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

Items (1-itemsets)

Item	Count
Bread	**4**
~~**Coke**~~	~~**2**~~
Milk	**4**
Wipes	**3**
Diaper	**4**
~~**Eggs**~~	~~**1**~~

Apriori Algorithm Frequent Itemset Generation

- Generating frequent 2-itemsets

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

Items (1-itemsets)

Item	Count
Bread	**4**
~~**Coke**~~	~~**2**~~
Milk	**4**
Wipes	**3**
Diaper	**4**
~~**Eggs**~~	~~**1**~~

Items (2-itemsets)

Itemset	Count
{Bread,Milk}	**3**
~~**{ Bread, Wipes}**~~	~~**2**~~
{Bread,Diaper}	**3**
~~**{ Milk , Wipes}**~~	~~**2**~~
{ Milk, Diaper }	**3**
{Wipes,Diaper}	**3**

(No need to generate candidates involving Coke or Eggs)

Apriori Algorithm Frequent Itemset Generation

- Generating frequent 3-itemsets

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

Items (1-itemsets)

Item	Count
Bread	**4**
~~**Coke**~~	~~**2**~~
Milk	**4**
Wipes	**3**
Diaper	**4**
~~**Eggs**~~	~~**1**~~

Items (2-itemsets)

Itemset	Count
{Bread,Milk}	**3**
~~**{ Bread, Wipes}**~~	~~**2**~~
{Bread,Diaper}	**3**
~~**{ Milk , Wipes}**~~	~~**2**~~
{ Milk, Diaper }	**3**
{Wipes,Diaper}	**3**

Items (3-itemsets)

Itemset	Count
{ Wipes, Diaper, Milk}	**2**
{ Wipes,Bread, Diaper}	**2**
{Bread, Diaper, Milk}	**2**
{Wipes, Bread, Milk}	**1**

(All not satisfying the *minsup*)

Apriori Algorithm Frequent Itemset Generation

- Brute-force Method: Considers every k-itemset as a potential candidate and then applies the candidate pruning step to remove any unnecessary candidate

Items

Item
Beer
Bread
Cola
Diapers
Milk
Eggs

Candidate Generation

Itemset
{Beer, Bread, Cola}
{Beer, Bread, Diapers}
{Beer, Bread, Milk}
{Beer, Bread, Eggs}
{Beer, Cola, Diapers}
{Beer, Cola, Milk}
{Beer, Cola, Eggs}
{Beer, Diapers, Milk}
{Beer, Diapers, Eggs}
{Beer, Milk, Eggs}
{Bread, Cola, Diapers}
{Bread, Cola, Milk}
{Bread, Cola, Eggs}
{Bread, Diapers, Milk}
{Bread, Diapers, Eggs}
{Bread, Milk, Eggs}
{Cola, Diapers, Milk}
{Cola, Diapers, Eggs}
{Cola, Milk, Eggs}
{Diapers, Milk, Eggs}

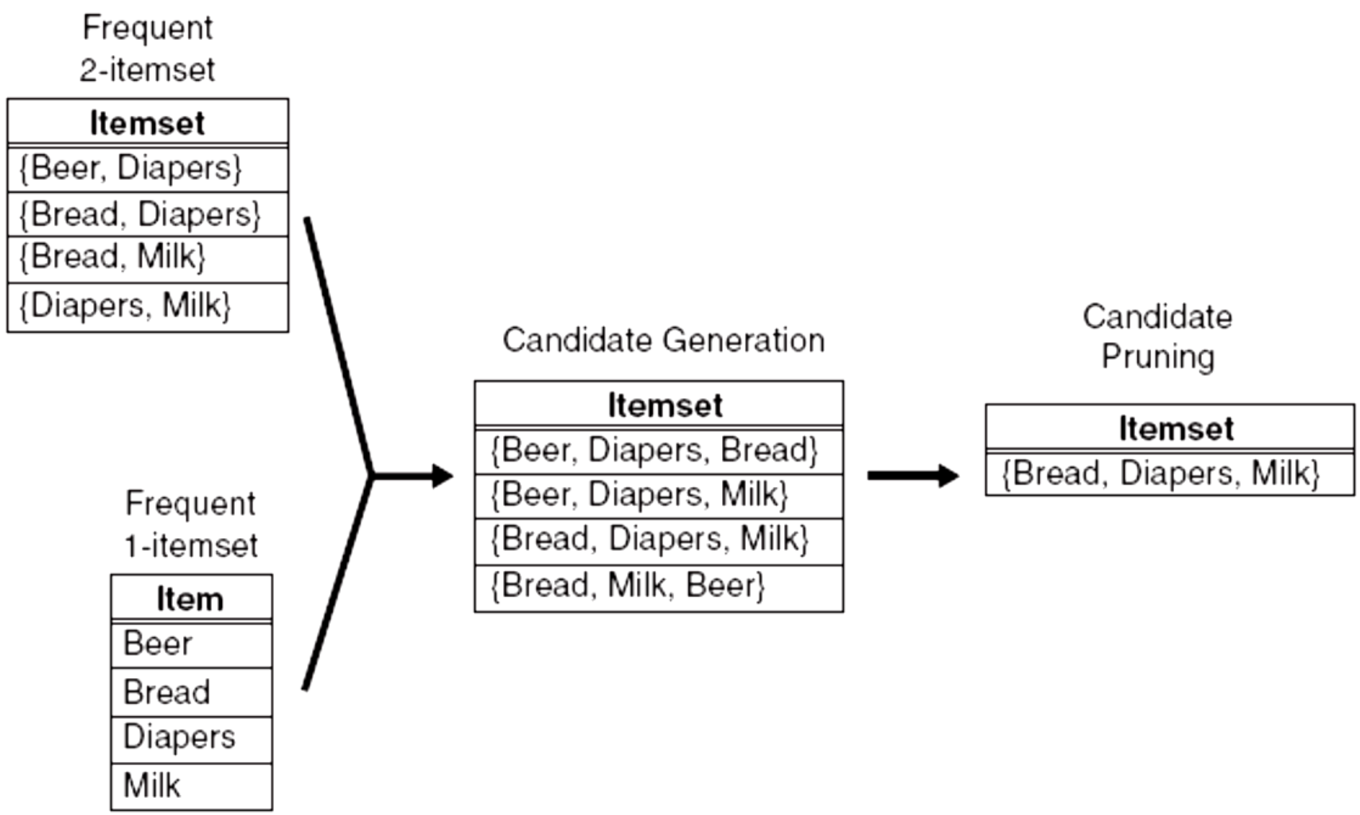
Candidate Pruning

Itemset
{Bread, Diapers, Milk}

Apriori Algorithm Frequent Itemset Generation

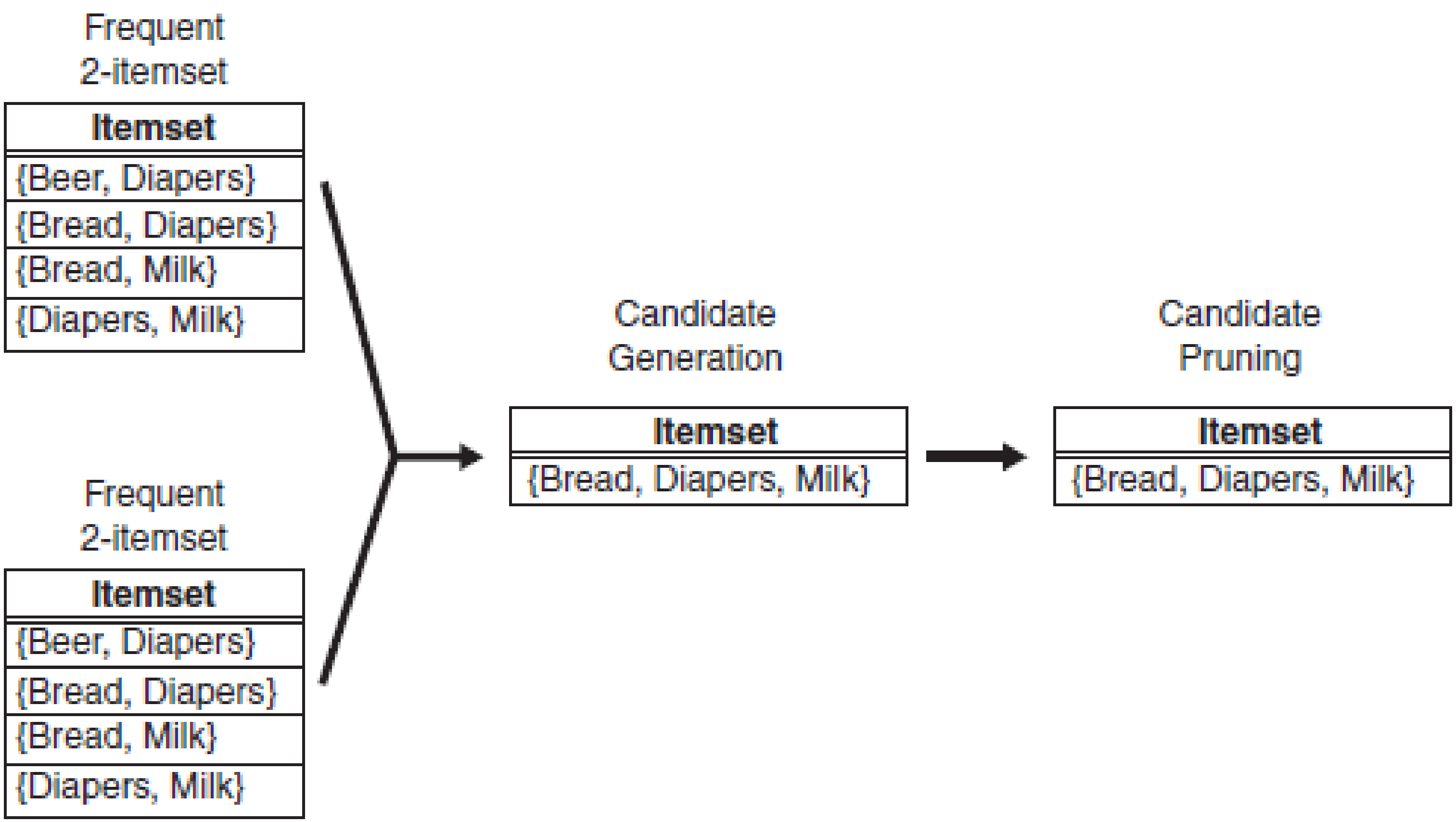
- $F_{k-1} \times F_1$ Method: Extend each frequent (k - 1)-itemset with other frequent items

- The procedure is complete because every frequent k-itemset is composed of a frequent (k - 1)-itemset and a frequent 1-itemset



Apriori Algorithm Frequent Itemset Generation

- $F_{k-1} \times F_{k-1}$ Method: Merges a pair of frequent (k-1)-itemsets if their first k-2 items are identical



Apriori Algorithm Rule Generation

- If {A,B,C,D} is a frequent itemset, candidate rules:

ABC $\rightarrow$D,	ABD $\rightarrow$C,	ACD $\rightarrow$B,	BCD $\rightarrow$A,
A $\rightarrow$BCD,	B $\rightarrow$ACD,	C $\rightarrow$ABD,	D $\rightarrow$ABC
AB $\rightarrow$CD,	AC $\rightarrow$ BD,	AD $\rightarrow$ BC,	BC $\rightarrow$AD,
BD $\rightarrow$AC,	CD $\rightarrow$AB		

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

- All such rules must have already met the support threshold because they are generated from a frequent itemset

Apriori Algorithm Rule Generation

- Confidence of rules generated from the same itemset has an anti-monotone property

- Example: Suppose {A,B,C,D} is a frequent 4-itemset:

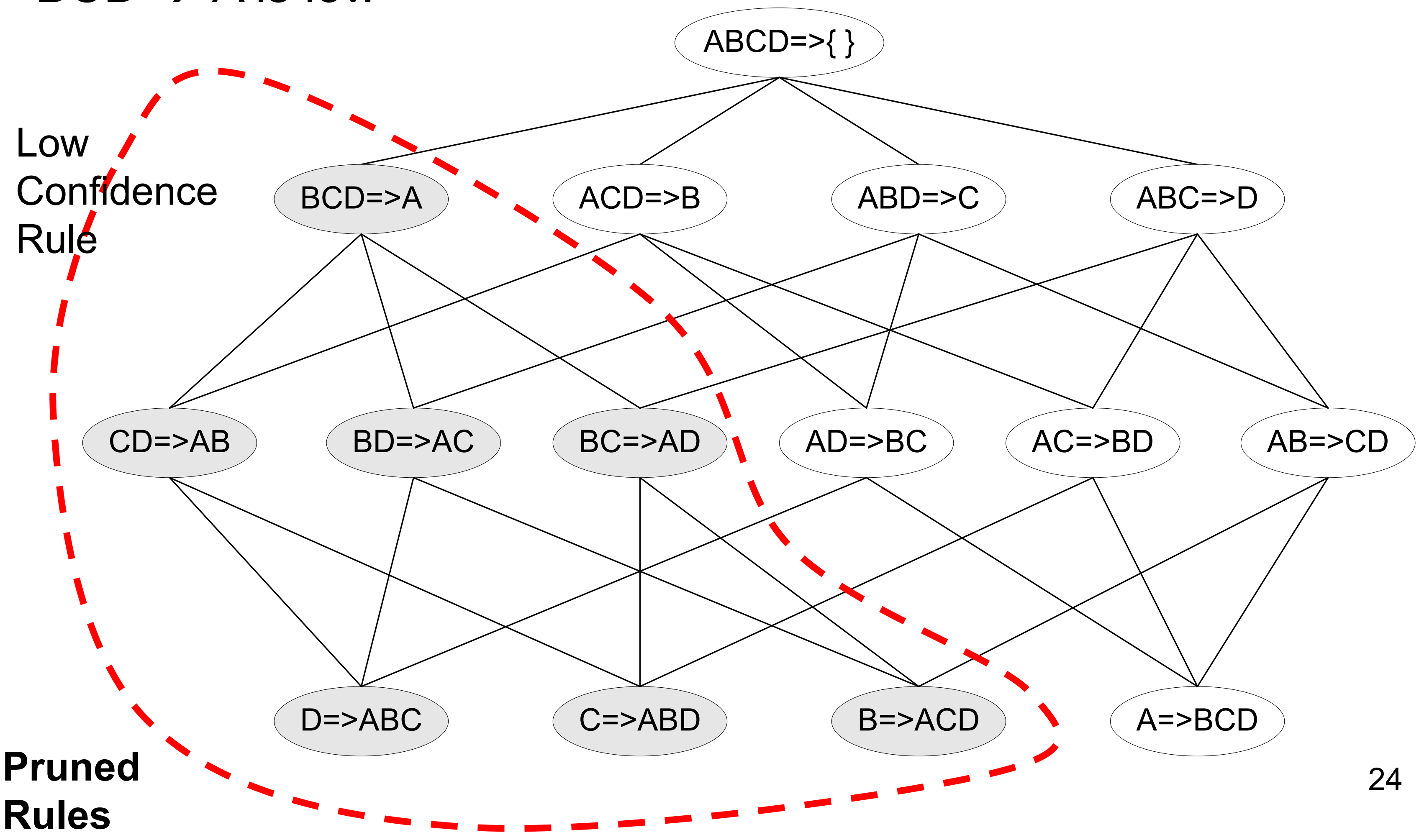
$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

- Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

- Therefore, if $c(ABC \rightarrow D)$ is less than *minconf*, then all $ABC \rightarrow D$, $AB \rightarrow CD$ and $A \rightarrow BCD$ rules could be pruned

Apriori Algorithm Rule Generation

- Example: Let the itemset {A, B, C, D} be a frequent itemset. The following lattice of rules can be generated. If the confidence of BCD → A is low



Apriori Algorithm Rule Generation

- Back to the Example:

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

Items (1-itemsets)

Item	Count
Bread	**4**
~~Coke~~	~~2~~
Milk	**4**
Wipes	**3**
Diaper	**4**
~~Eggs~~	~~1~~

Items (2-itemsets)

Itemset	Count
{Bread,Milk}	**3**
~~{ Bread, Wipes}~~	~~2~~
{Bread,Diaper}	**3**
~~{ Milk , Wipes}~~	~~2~~
{ Milk, Diaper }	**3**
{Wipes,Diaper}	**3**

Items (3-itemsets)

Itemset	Count
{ Wipes, Diaper, Milk}	**2**
{ Wipes,Bread, Diaper}	**2**
{Bread, Diaper, Milk}	**2**
{Wipes, Bread, Milk}	**1**

(All not satisfying the *minsup*)

- The frequent itemsets are then:
  - {Bread, Milk}, {Bread, Diaper}, {Milk, Diaper}, {Wipes, Diaper}

Apriori Algorithm Rule Generation

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

- From each frequent itemset, we generate rules and compute their confidence. Let *minconf* be 90%
  - {Bread, Milk}:
    - c(Bread → Milk) = ¾
    - c(Milk → Bread) = ¾
  - {Bread, Diaper}:
    - c(Bread → Diaper) = ¾
    - c(Diaper → Bread) = ¾
  - {Milk, Diaper}:
    - c(Milk → Diaper) = ¾
    - c(Diaper → Milk) = ¾
  - {Wipes, Diaper}:
    - c(Wipes → Diaper) = 1 (Only rule satisfying the *minconf*)
    - c(Diaper → Wipes) = ¾
- Final Rule Inferred: Wipes → Diaper