

Applied Analytics and Predictive Modeling

Spring 2020

Lecture-8

Lydia Manikonda
manikl@rpi.edu



Rensselaer

Today's agenda

- Case Study-3 presentations
- Association Rules
- Class Exercises

Case Study-3 Presentations

Association Rules (Focus on Frequent Itemsets)

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

<i>TID</i>	<i>Items</i>
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

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

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. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$

- **Support**

- Fraction of transactions that contain an itemset
- E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$

- **Frequent Itemset**

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

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example – Itemset metrics

- Itemset (I1): {Bread, Milk, Diaper}
- Support
 - #occurrences (support count) = 2
 - Fraction of occurrences (support) = 2/5
- Lets say *minsup* = 0.1
- Is I1 a frequent itemset?
 - Yes
 - Support of I1 = 0.4 (> *minsup*)

<i>TID</i>	<i>Items</i>
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

- 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

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example – Association Rule

- {Milk, Diaper} => {Beer}

- Support

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

- Confidence

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

<i>TID</i>	<i>Items</i>
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 $\geq \textit{minsup}$ threshold
 - confidence $\geq \textit{minconf}$ 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

<i>TID</i>	<i>Items</i>
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:

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

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

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

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

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

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

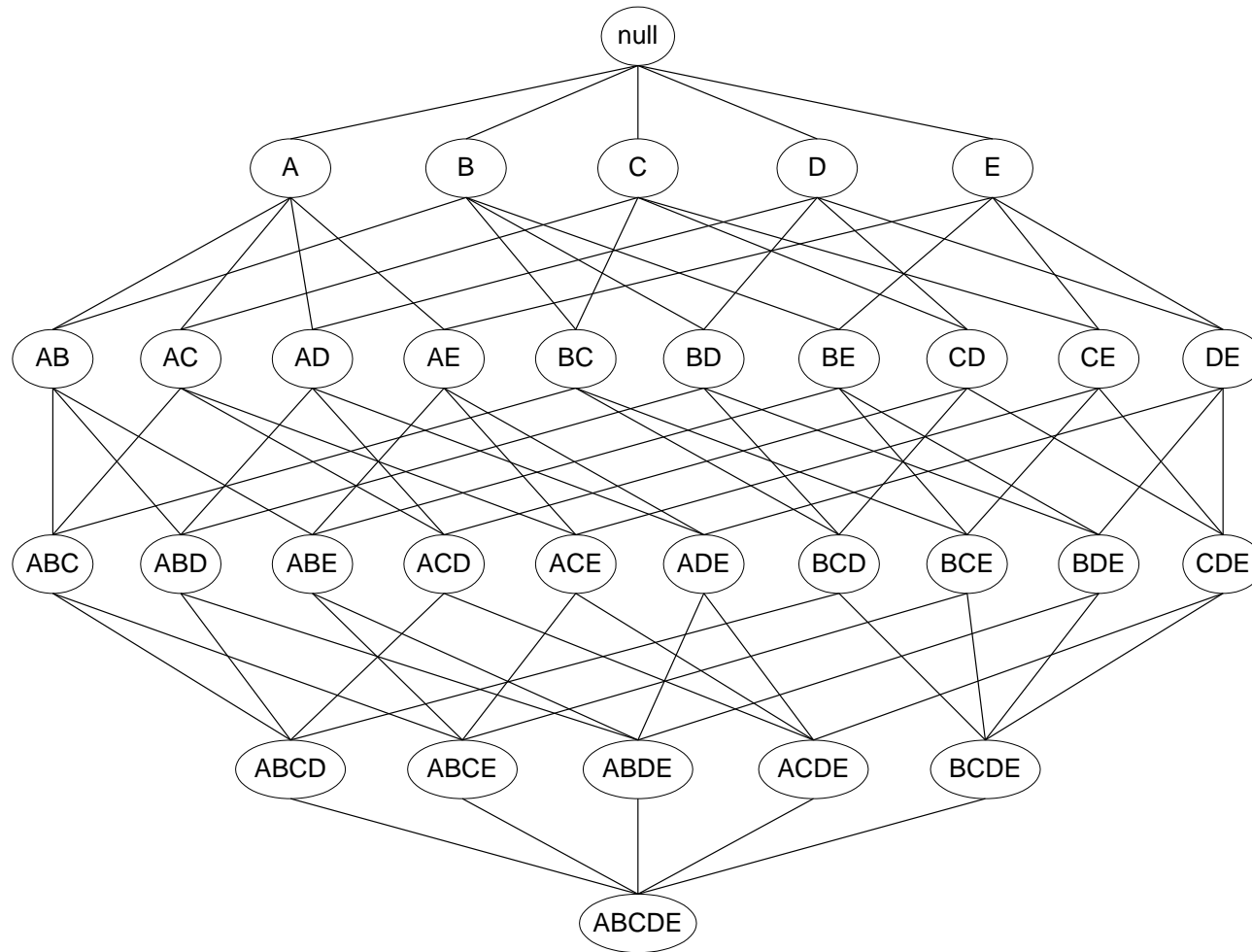
Observations:

- All the above rules are binary partitions of the same itemset:
 $\{\text{Milk, Diaper, Beer}\}$
- 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

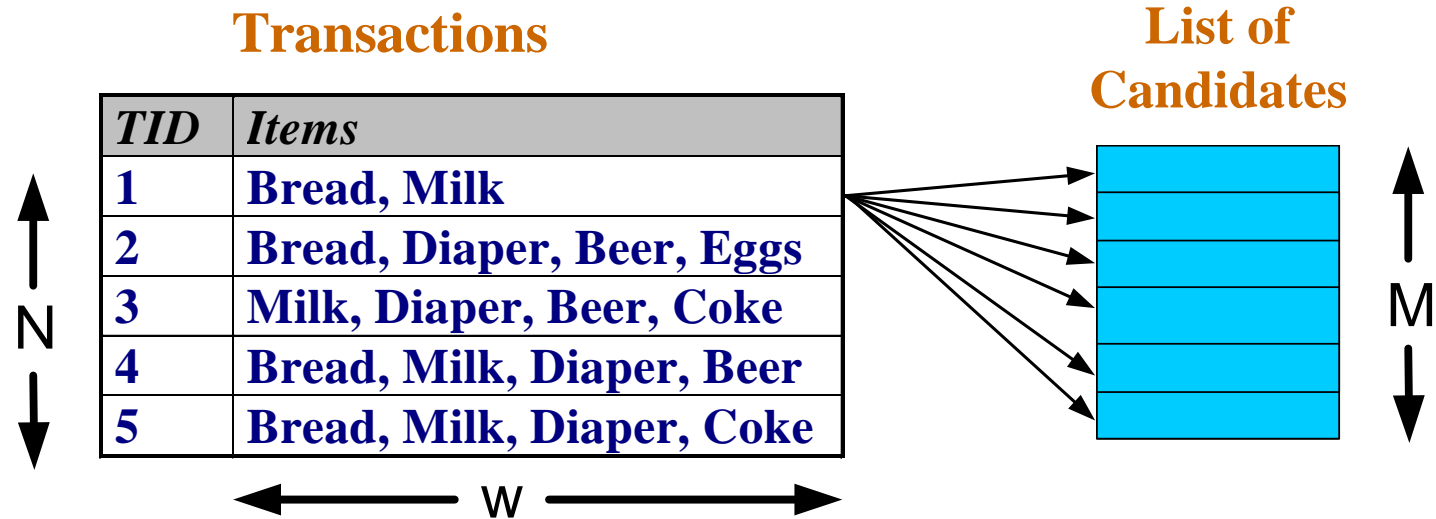
Frequent Itemset Generation



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 $\sim O(NMw) \Rightarrow$ **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

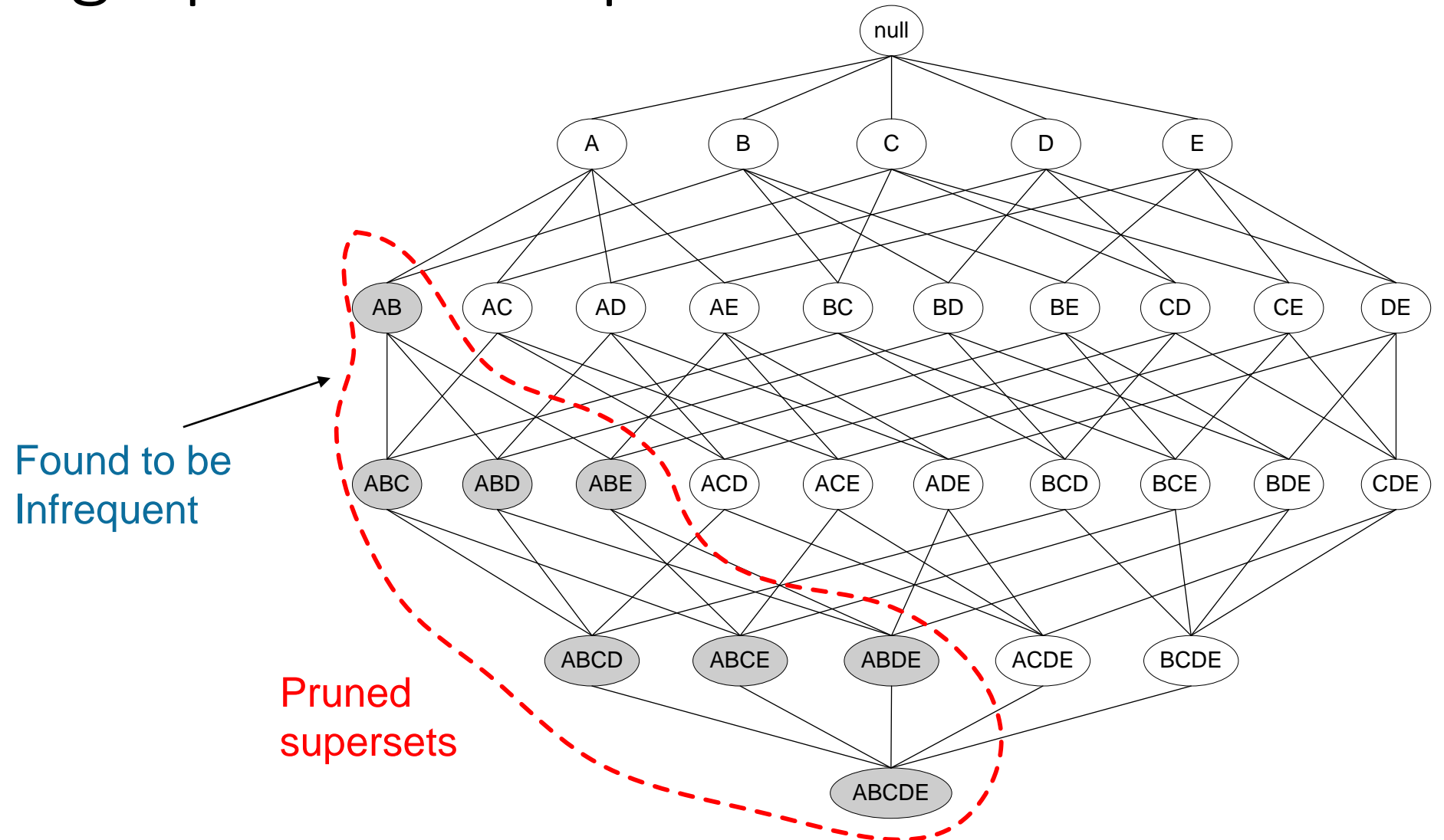
Reducing Number of Candidates

- **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

Illustrating Apriori Principle



Illustrating Apriori Principle

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Coke, Diaper, Milk



Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

Illustrating Apriori Principle

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Coke, Diaper, Milk



Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

Illustrating Apriori Principle

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

Items (1-itemsets)



Itemset
{Bread,Milk}
{Bread, Beer }
{Bread,Diaper}
{Beer, Milk}
{Diaper, Milk}
{Beer,Diaper}

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 \\ 6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

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

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 \\ 6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

Triplets (3-itemsets)

Itemset
{ Beer, Diaper, Milk }
{ Beer,Bread,Diaper }
{Bread, Diaper, Milk }
{ Beer, Bread, Milk }

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
{ Beer, Diaper, Milk}	2
{ Beer,Bread, Diaper}	2
{Bread, Diaper, Milk}	2
{Beer, Bread, Milk}	1

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

$$6 + 6 + 1 = 13$$

Apriori Algorithm

- F_k : frequent k-itemsets
- L_k : candidate k-itemsets
- Algorithm
 - Let $k=1$
 - Generate $F_1 = \{\text{frequent 1-itemsets}\}$
 - 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 $\Rightarrow F_{k+1}$

Candidate Generation: Brute-Force Method

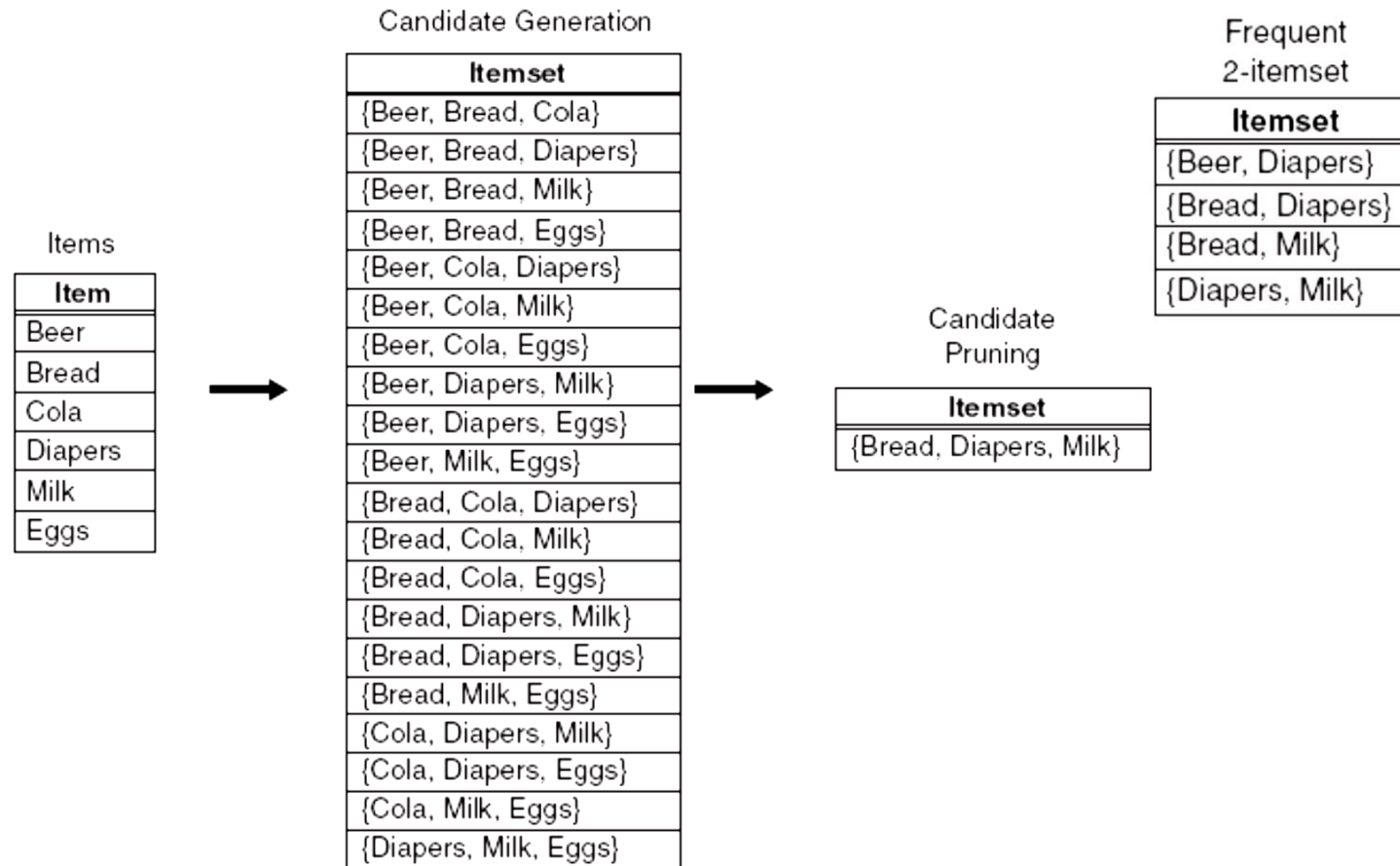


Figure 6.6. A brute-force method for generating candidate 3-itemsets.

Candidate Generation: Merge F_{k-1} and F_1 itemsets

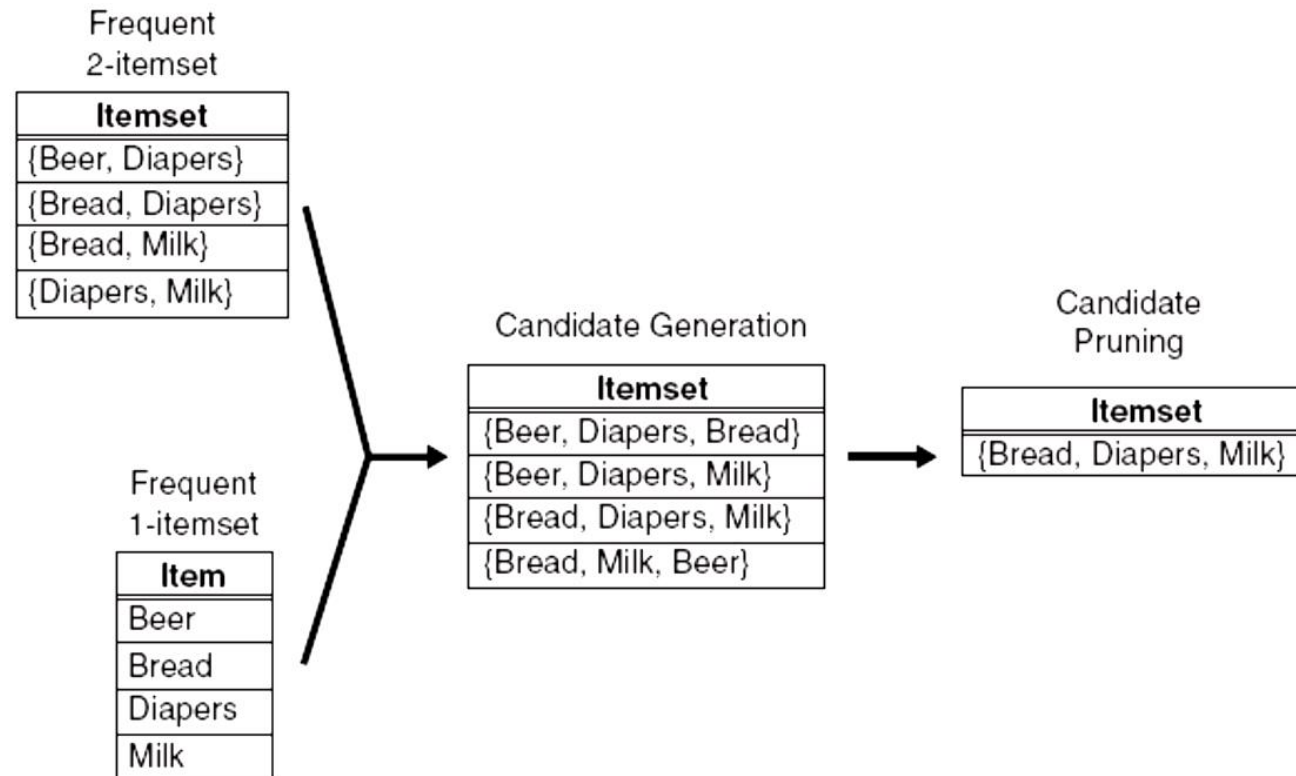


Figure 6.7. Generating and pruning candidate k -itemsets by merging a frequent $(k-1)$ -itemset with a frequent item. Note that some of the candidates are unnecessary because their subsets are infrequent.

Candidate Generation: Merge F_{k-1} and F_{k-1} itemsets

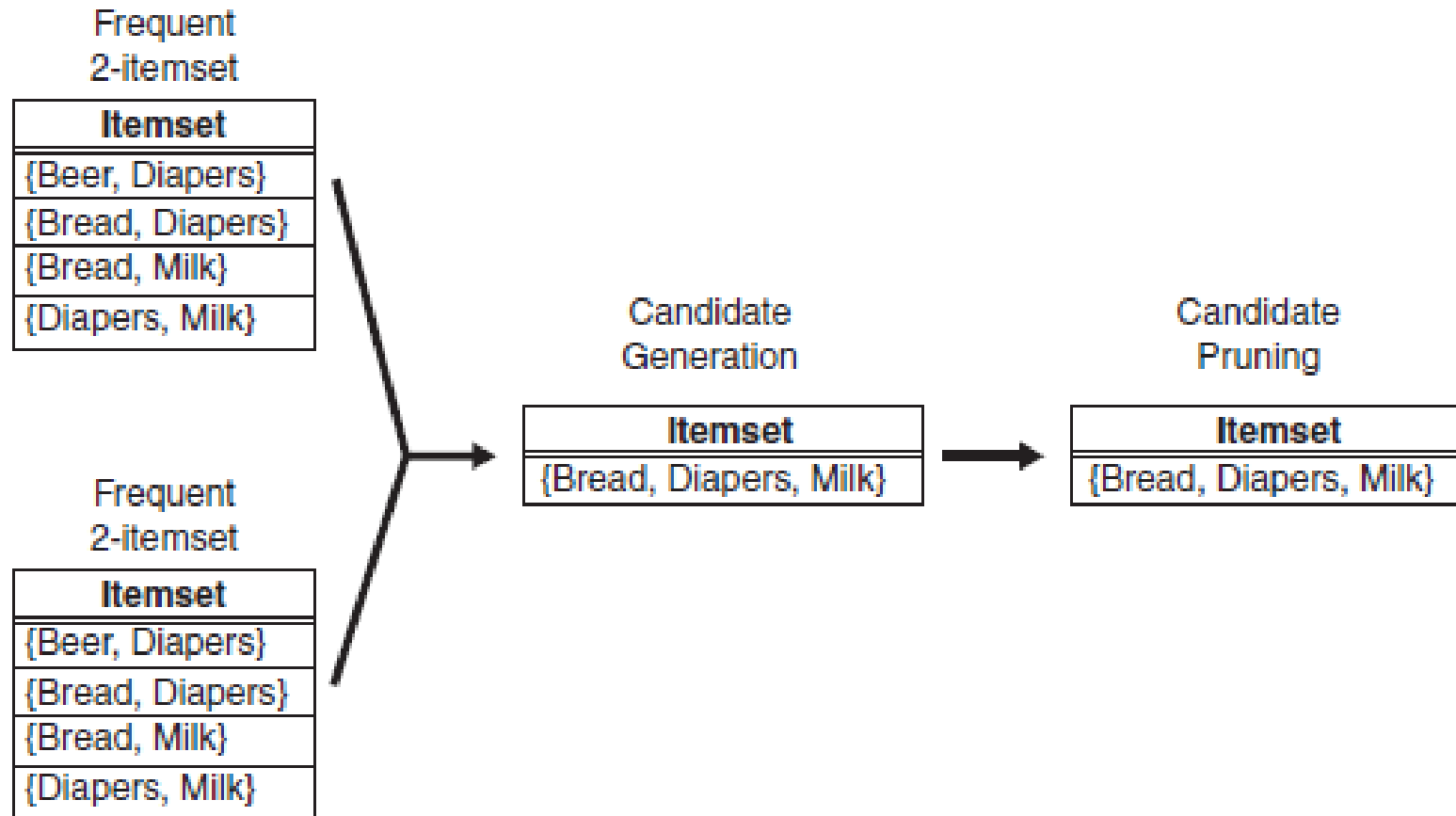


Figure 6.8. Generating and pruning candidate k -itemsets by merging pairs of frequent $(k-1)$ -itemsets.

Candidate Generation: Merge F_{k-1} and F_{k-1} itemsets

- Merge two frequent $(k-1)$ -itemsets if their first $(k-2)$ items are identical
- $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$
 - Merge(ABC, ABD) = ABCD
 - Merge(ABC, ABE) = ABCE
 - Merge(ABD, ABE) = ABDE
- Do not merge(ABD, ACD) because they share only prefix of length 1 instead of length 2

Candidate Pruning

- Let $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$ be the set of frequent 3-itemsets
- $L_4 = \{ABCD, ABCE, ABDE\}$ is the set of candidate 4-itemsets generated (from previous slide)
- Candidate pruning
 - Prune ABCE because ACE and BCE are infrequent
 - Prune ABDE because ADE is infrequent
- After candidate pruning: $L_4 = \{ABCD\}$

Alternate $F_{k-1} \times F_{k-1}$ Method

- Merge two frequent $(k-1)$ -itemsets if the last $(k-2)$ items of the first one is identical to the first $(k-2)$ items of the second.
- $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$
 - Merge(ABC, BCD) = ABCD
 - Merge(ABD, BDE) = ABDE
 - Merge(ACD, CDE) = ACDE
 - Merge(BCD, CDE) = BCDE

Candidate Pruning for Alternate $F_{k-1} \times F_{k-1}$ Method

- Let $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$ be the set of frequent 3-itemsets
- $L_4 = \{ABCD, ABDE, ACDE, BCDE\}$ is the set of candidate 4-itemsets generated (from previous slide)
- Candidate pruning
 - Prune ABDE because ADE is infrequent
 - Prune ACDE because ACE and ADE are infrequent
 - Prune BCDE because BCE
- After candidate pruning: $L_4 = \{ABCD\}$

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)



Triplets (3-itemsets)

Itemset	Count
{Bread, Diaper, Milk}	2

Minimum Support = 3

If every subset is considered,
 ${}^6C_1 + {}^6C_2 + {}^6C_3$
 $6 + 15 + 20 = 41$
With support-based pruning,
 $6 + 6 + 1 = 13$

Use of $F_{k-1} \times F_{k-1}$ method for candidate generation results in only one 3-itemset. This is eliminated after the support counting step.

Exercise-1

Transaction 1	Apple, beer, rice, chicken
Transaction 2	Apple, beer, rice
Transaction 3	Apple, beer
Transaction 4	Apple, pear
Transaction 5	Milk, beer, rice, chicken
Transaction 6	Milk, beer, rice
Transaction 7	Milk, beer
Transaction 8	Milk, pear

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