


# Big Data @ Walmart

## Walmart Big Data Facts and Figures

 **245 million** customers visiting 10,900 stores and 10 active websites across the globe—Walmart is a name to reckon in the retail sector.

Walmart sees close to **300,000** social mentions every week.



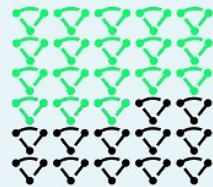
It has **2 million** associates and approximately half a million associates hired every year.

Walmart's employee numbers are more than some of the retailer's customer numbers.



Walmart takes in approximately **\$36 million** from across 4300 US stores every day.

Walmart collects **2.5 petabytes** of unstructured data from 1 million customers every hour.



Walmart made a move from the experiential 10 node Hadoop cluster to a **250 node** Hadoop cluster in 2012.

Walmart has exhaustive customer data of close to 145 million Americans of which 60% of the data is of U.S adults.



## How Walmart uses Big Data?



The analytics systems at Walmart analyse close to 100 million keywords on daily basis to optimize the bidding of each keyword.

The analysis covers millions of products and 100's of millions customers from different sources.



Walmart observed a significant 10% to 15% increase in online sales for \$1 billion in incremental revenue.

Walmart Labs analyses every clickable action on Walmart.com—

- 1) What consumers buy in-store and online?
- 2) What is trending on Twitter?
- 3) Local events such as San Francisco giants winning the World Series?
- 4) How local weather deviations affect the buying patterns?

*“The most important thing about Wal-Mart is the scale of Wal-Mart. Its scale in terms of customers, its scale in terms of products and its scale in terms of technology.”*

*—Anand Rajaram, head of WalmartLabs*

*“We want to know what every product in the world is. We want to know who every person in the world is. And we want to have the ability to connect them together in a transaction.”*

*—said Walmart's CEO of global e-commerce in 2013*

# Association Analysis and the Apriori Algorithm



# Association Analysis

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

Set of all items in the data:  $I = \{i_1, i_2, \dots, i_n\}$

Set of all transactions in the data:  $T = \{t_1, t_2, \dots, t_n\}$

# Terminology

<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

- **Transaction width**: the number of items in a transaction
- **Itemset**: any collection of 0 or more items
  - Ex: {Beer, Diapers, Eggs} is an itemset
- **k-itemset**: if an itemset contains k items, it is called a k-itemset
  - Ex: {Beer, Diapers, Eggs} is a 3-itemset
- A transaction **contains** itemset X if X is a subset of the transaction
  - Ex:  $t_2$  contains {Bread, Diapers} but not {Bread, Milk}
- **Support count  $[\sigma(X)]$** : the number of transactions that contain this itemset
  - Ex: The support count of {Beer, Diapers, Milk} is 2
- **Frequent itemset**: an itemset whose support is equal to or greater than some *minsup* threshold

# Association Rules

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

- Support: Fraction of transactions that contain both  $X$  and  $Y$

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{|T|}$$

- Confidence: Measures how often items in  $Y$  appear in transactions that contain  $X$

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

<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:** {Milk, Diaper}  $\Rightarrow$  Beer

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

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



# Association Rule Mining

- 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

<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 Rules:

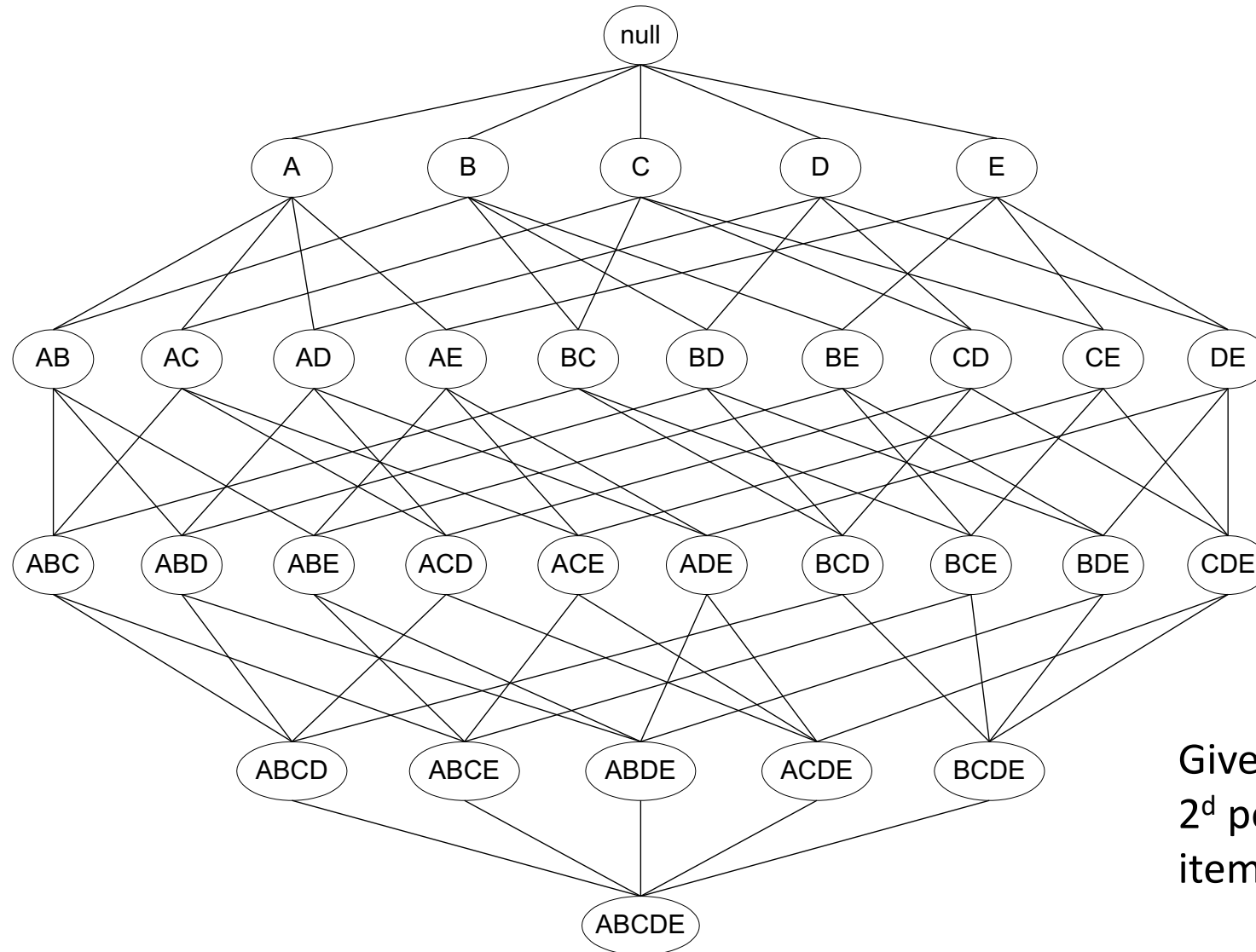
$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$  ( $s=0.4, c=0.67$ )  
 $\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\}$  ( $s=0.4, c=1.0$ )  
 $\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\}$  ( $s=0.4, c=0.67$ )  
 $\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\}$  ( $s=0.4, c=0.67$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

- All the example rules are subsets of the same itemset:  $\{\text{Milk, Diaper, Beer}\}$
- Rules originating from the same itemset have identical support but can have different confidence

# Association Rule Mining Steps

1. **Frequent Itemset** generation: find all the itemsets that satisfy *minsup*
2. **Strong Rule** generation: find all the rules in the frequent itemsets that satisfy *minconf*

# Candidate Itemsets



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



# Apriori Principle

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

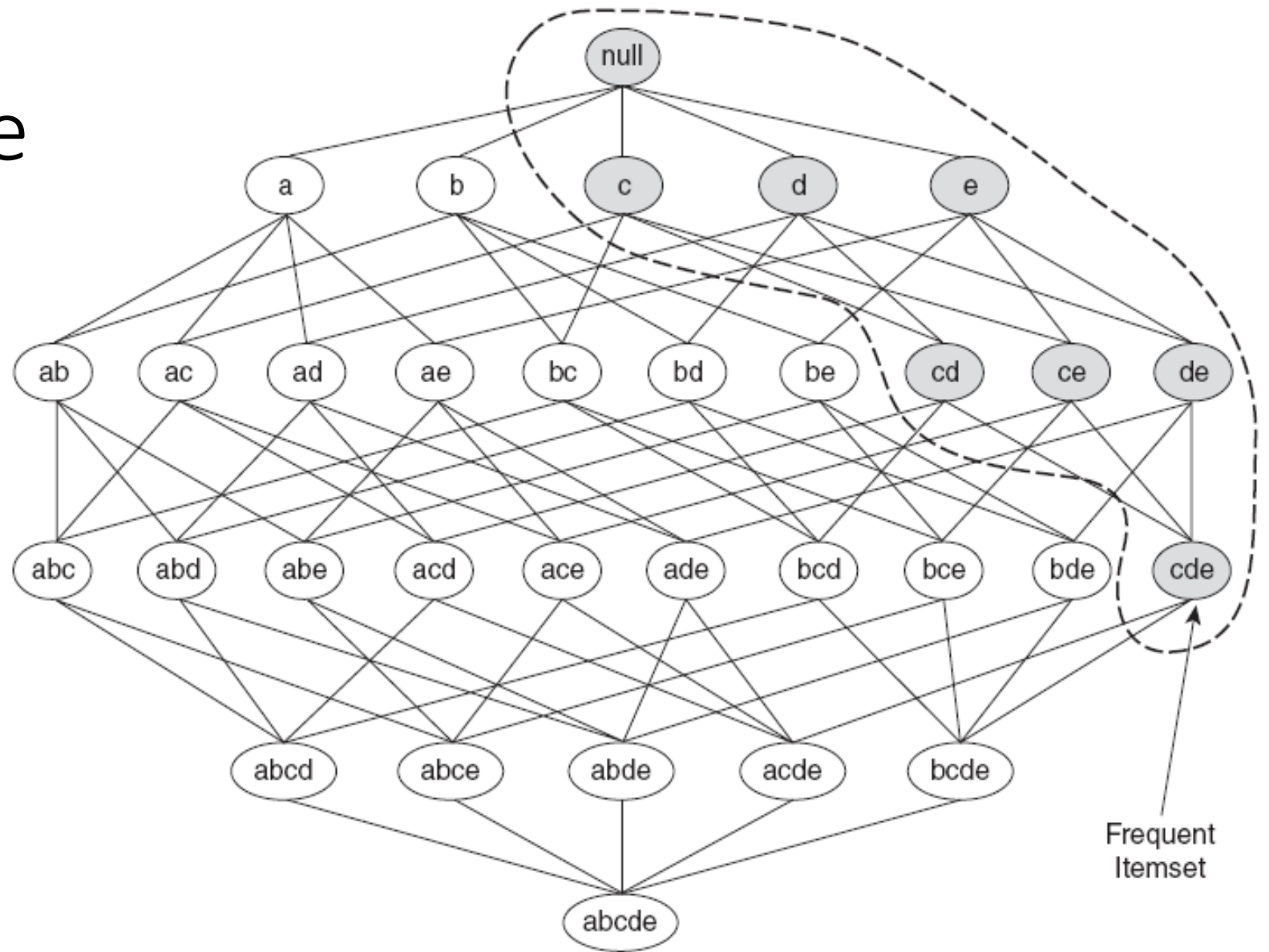
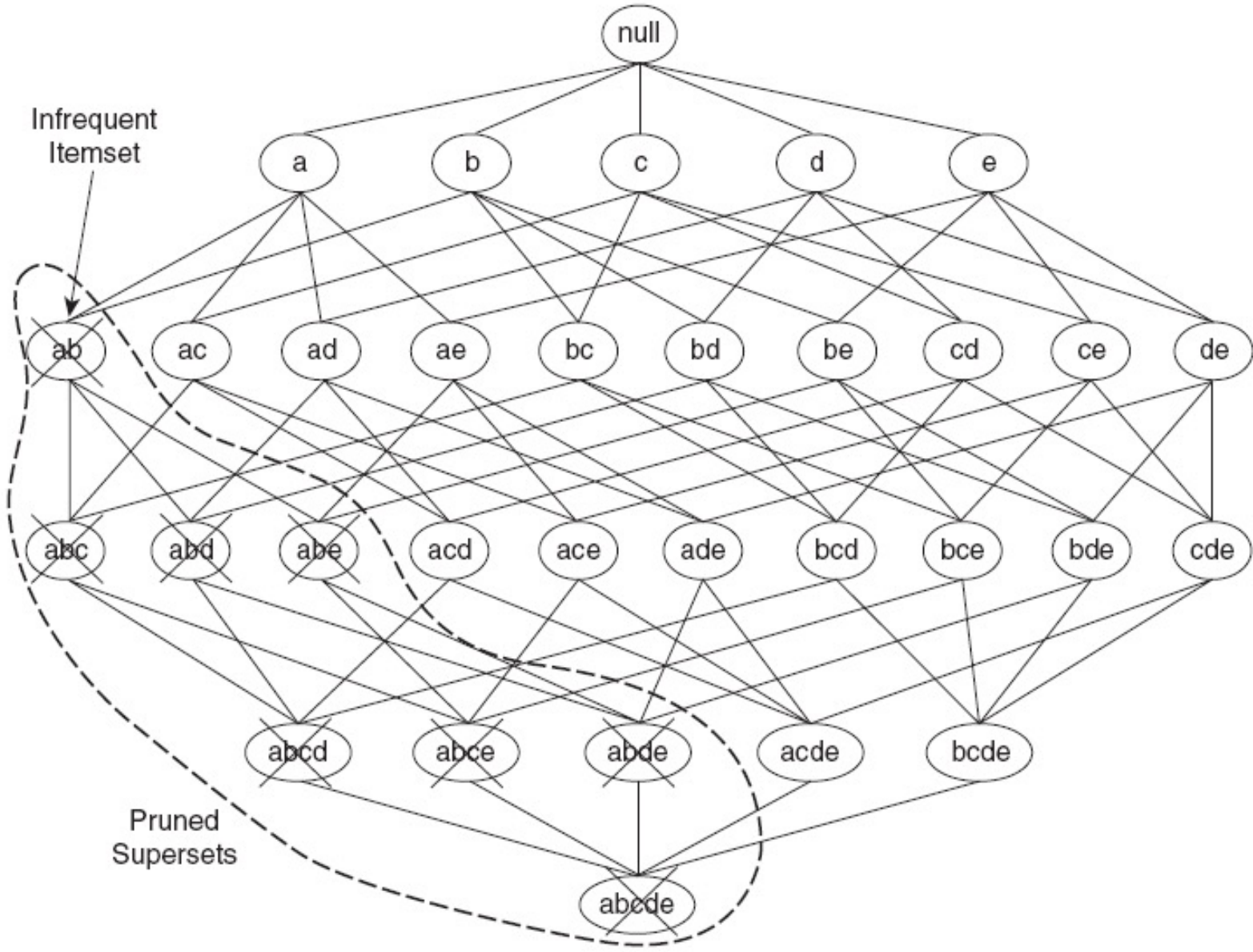


Figure 6.3. An illustration of the *Apriori* principle. If  $\{c, d, e\}$  is frequent, then all subsets of this itemset are frequent.

# Apriori Principle

- If an itemset is infrequent, then all of its supersets must also be infrequent

The anti-monotone property of support: Support of an itemset never exceeds the support of its subsets



**Figure 6.4.** An illustration of support-based pruning. If  $\{a, b\}$  is infrequent, then all supersets of  $\{a, b\}$  are infrequent.

# Apriori Algorithm, $k = 1$

- Initially, every individual item is considered as a candidate 1-itemset (let  $k=1$ )
- Their supports are counted; anything below *minsup* is discarded

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
6	Bread, Diaper, Milk, Eggs

Minimum Support = 50% = 3/6

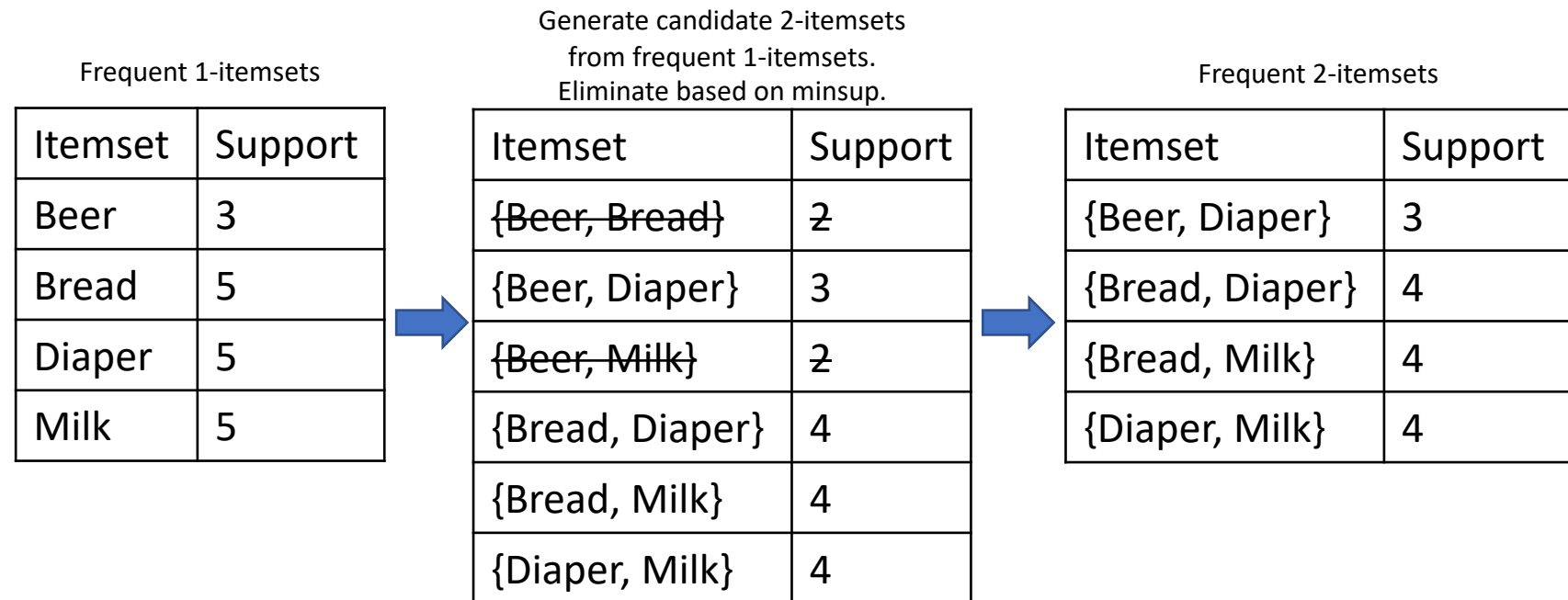
Candidate 1-itemsets		Eliminate based on minsup		Frequent 1-itemsets	
Itemset	Support	Itemset	Support	Itemset	Support
Beer	3	Beer	3	Beer	3
Bread	5	Bread	5	Bread	5
Coke	2	<del>Coke</del>	<del>2</del>	Diaper	5
Diaper	5	Diaper	5	Milk	5
Eggs	2	<del>Eggs</del>	<del>2</del>		
Milk	5	Milk	5		

# Apriori Algorithm, $k = 2$

- Candidate  $(k+1)$ -itemsets are generated from the frequent  $k$ -itemsets
- Their supports are counted; anything below *minsup* is discarded

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
6	Bread, Diaper, Milk, Eggs

Minimum Support = 50% = 3/6



# Apriori Algorithm, $k > 2$

- Candidate  $(k+1)$ -itemsets are generated from the frequent  $k$ -itemsets
- Itemsets are pruned ***apriori*** if they contain an infrequent subset
- Remaining itemset supports are counted; anything below *minsup* is discarded
- Repeat for each  $k > 2$  until no additional frequent itemsets are found

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
6	Bread, Diaper, Milk, Eggs

Minimum Support = 50% = 3/6

Frequent 2-itemsets

Itemset	Support
{Beer, Diaper}	3
{Bread, Diaper}	4
{Bread, Milk}	4
{Diaper, Milk}	4

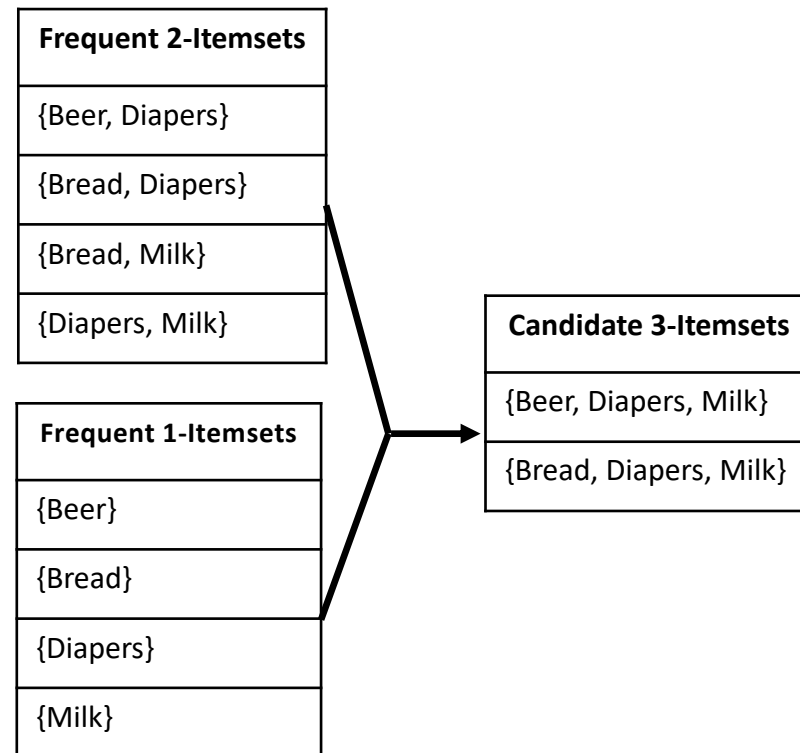


Generate candidate 3-itemsets  
from frequent 2-itemsets.

There are two alternative  
methods for generating  
the candidate  
 $(k+1)$ -itemsets,  
once  $k > 2$   
(see next two slides...)

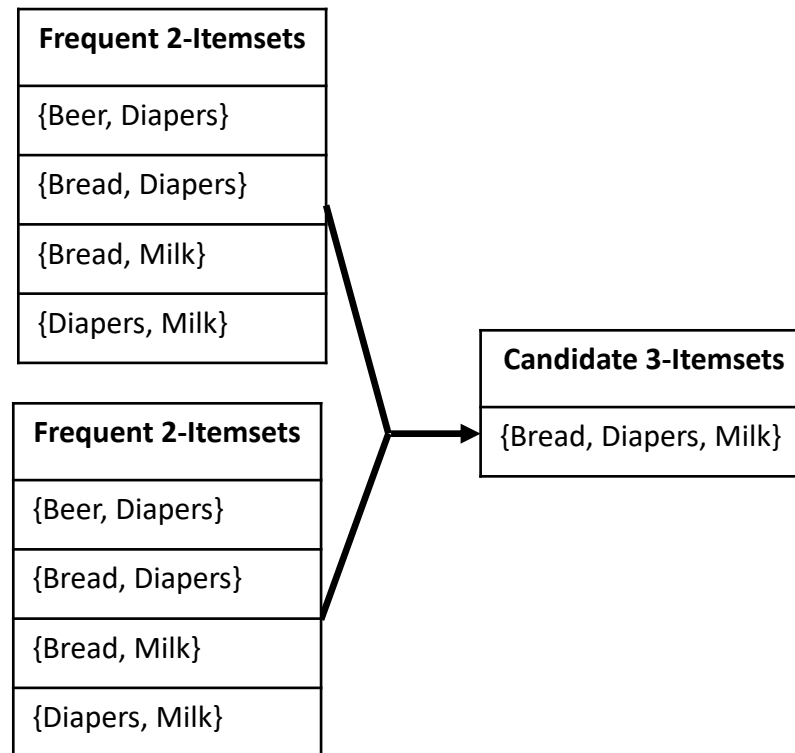
# Candidate Generation: $F_{k-1} \times F_1$ Method

- Items in each frequent itemset must be sorted (i.e. in alphabetical order)
- Extend each frequent (k-1)-itemset with a frequent 1-itemset that is alphabetically larger than the items already in the (k-1)-itemset



# Candidate Generation: $F_{k-1} \times F_{k-1}$ Method (Apriori-gen)

- Items in each frequent itemset must be sorted (i.e. in alphabetical order)
- Merge pairs of (k-1)-itemsets if all but their last item are the same





# Apriori Algorithm, $k > 2$

- Candidate  $(k+1)$ -itemsets are generated from the frequent  $k$ -itemsets
- Itemsets are pruned ***apriori*** if they contain an infrequent subset
- Remaining itemset supports are counted; anything below *minsup* is discarded
- Repeat for each  $k > 2$  until no additional frequent itemsets are found

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
6	Bread, Diaper, Milk, Eggs

Minimum Support = 50% = 3/6

Frequent 2-itemsets

Itemset	Sup
{Beer, Diaper}	3
{Bread, Diaper}	4
{Bread, Milk}	4
{Diaper, Milk}	4



Generate candidate 3-itemsets using method of choice.

First, eliminate candidates based on ***apriori***.

Then, eliminate based on minsup.

Itemset	Support
<del>{Beer, Diapers, Milk}</del>	Eliminated apriori because {Beer, Milk} is not frequent
{Bread, Diapers, Milk}	3

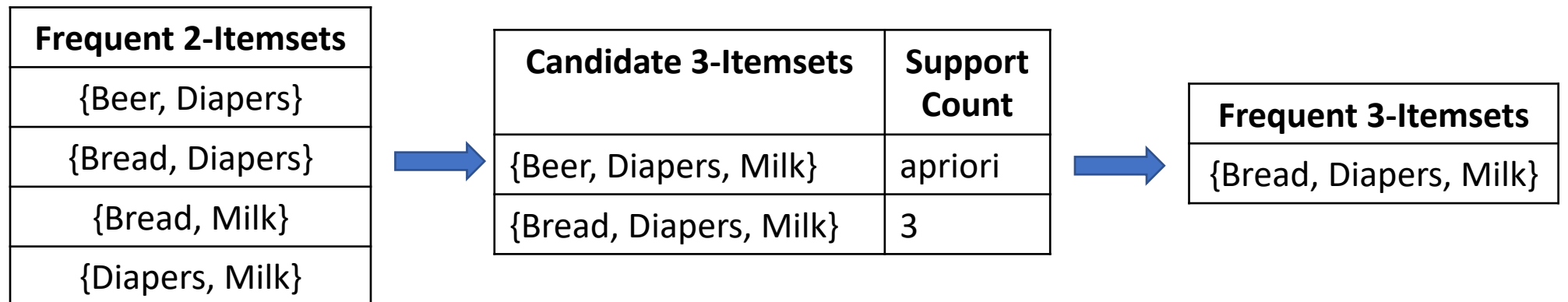


Frequent 3-itemsets

Itemset	Sup
{Bread, Diapers, Milk}	3

# Methods for Apriori Pruning

- If any  $(k-1)$ -subset of the candidate is not a frequent  $(k-1)$ -itemset, the candidate is pruned
  - Ex: {Bread, Diapers, Milk} is only a frequent 3-itemset if {Bread, Diapers}, {Bread, Milk}, and {Diapers, Milk} are all frequent 2-itemsets.
- Or, for any frequent  $k$ -itemset, every item in the set must be contained in at least  $k-1$  of the frequent  $(k-1)$ -itemsets
  - Ex: {Beer, Diapers, Milk} is only a frequent 3-itemset if Beer, Diapers, and Milk all show up in 2 of the frequent 2-itemsets. Since Beer is only in 1 of the frequent 2-itemsets, Beer will not be in a frequent 3-itemset.



# Frequent Itemsets

- The final list of frequent itemsets is all of the frequent itemsets you found at every k.

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
6	Bread, Diaper, Milk, Eggs

Minimum Support = 3/6 (50%)

Frequent Itemsets	Support Count
Bread	5
Milk	5
Beer	3
Diaper	5
{Bread, Milk}	4
{Bread, Diaper}	4
{Milk, Diaper}	4
{Beer, Diaper}	3
{Bread, Diapers, Milk}	3

# Apriori Practice

TID	Items
1	A,B,E
2	B,D
3	B,C
4	A,B,D
5	A,C
6	B,C
7	A,C
8	A,B,C,E
9	A,B,C

MinSup = 2

Clearly Show:

- Which candidate generation method you are using ( $F_{k-1} \times 1$  or  $F_{k-1} \times F_{k-1}$ )
- The **candidate** k-itemsets that are generated at each k
- Which of the candidate itemsets get pruned **before** counting (apriori)
- Which of the candidate itemsets get **counted** for support
- The **frequent** k-itemsets at each k