ASSOCIATION MINING

Definition

- Search for patterns recurring in the given data set
- Given a set of item sets or transactions, find rules predicting the occurrence of items based on the occurrences of other items in the transactions

Applications

- Netflix movie recommendations

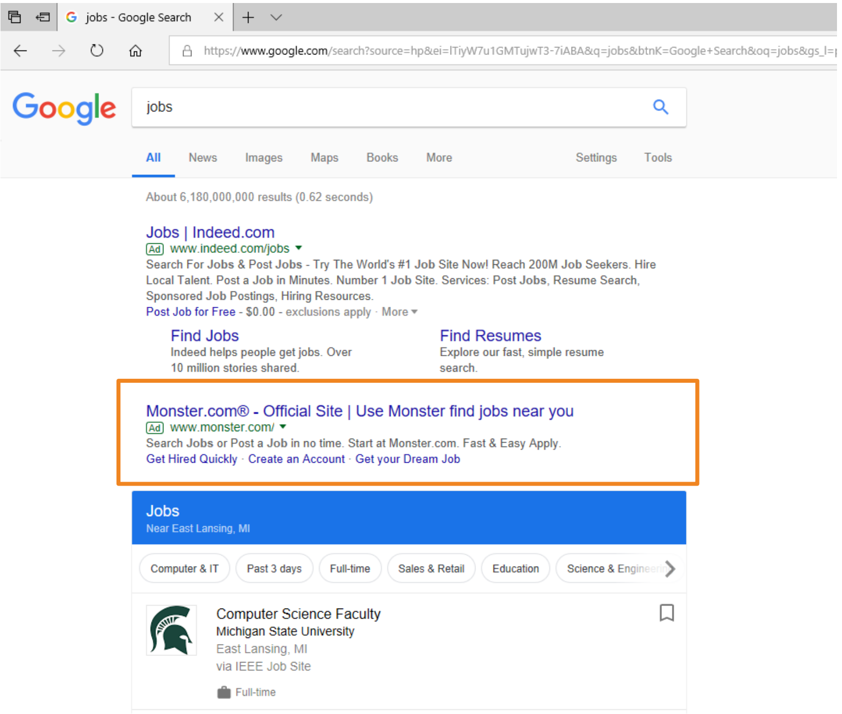
  {Breaking Bad, House of Cards} -> Mad Men



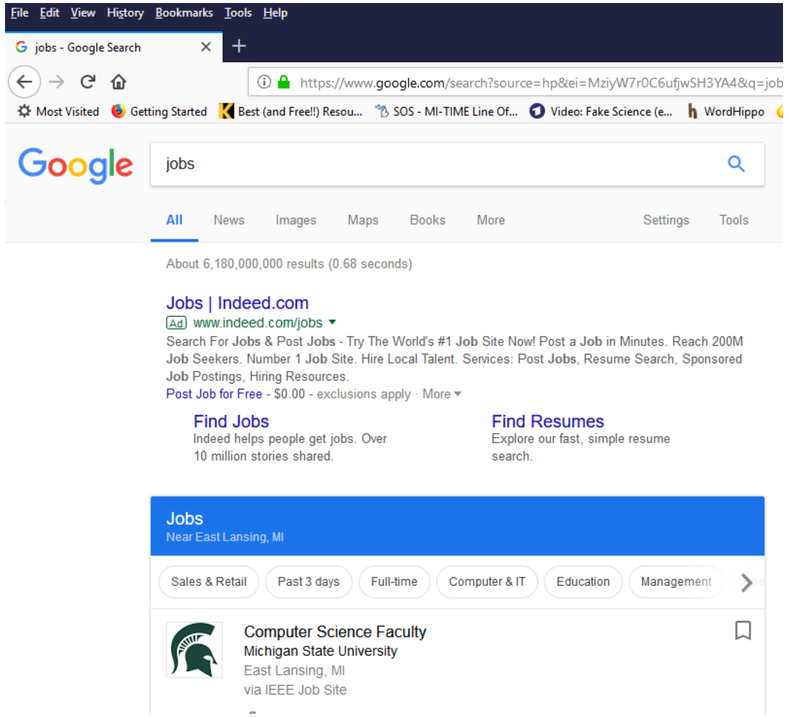
- Google search
  - Suggested autofill
  - Google.com, search “how to”, top 3 suggestions
- Order of webpages / Ads
  - Search history
  - Location

Search history: “jobs”

Microsoft Edge

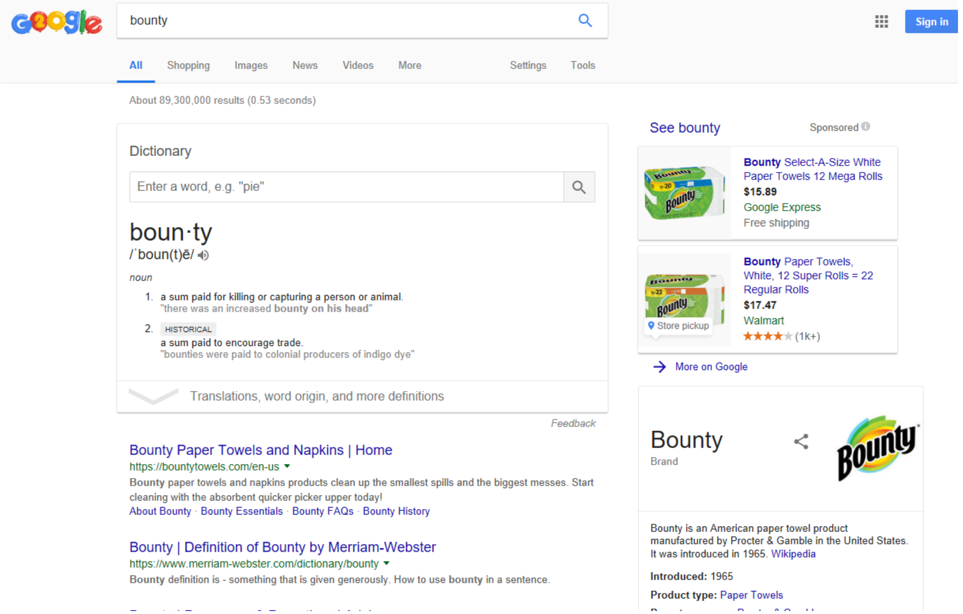


Mozilla Firefox

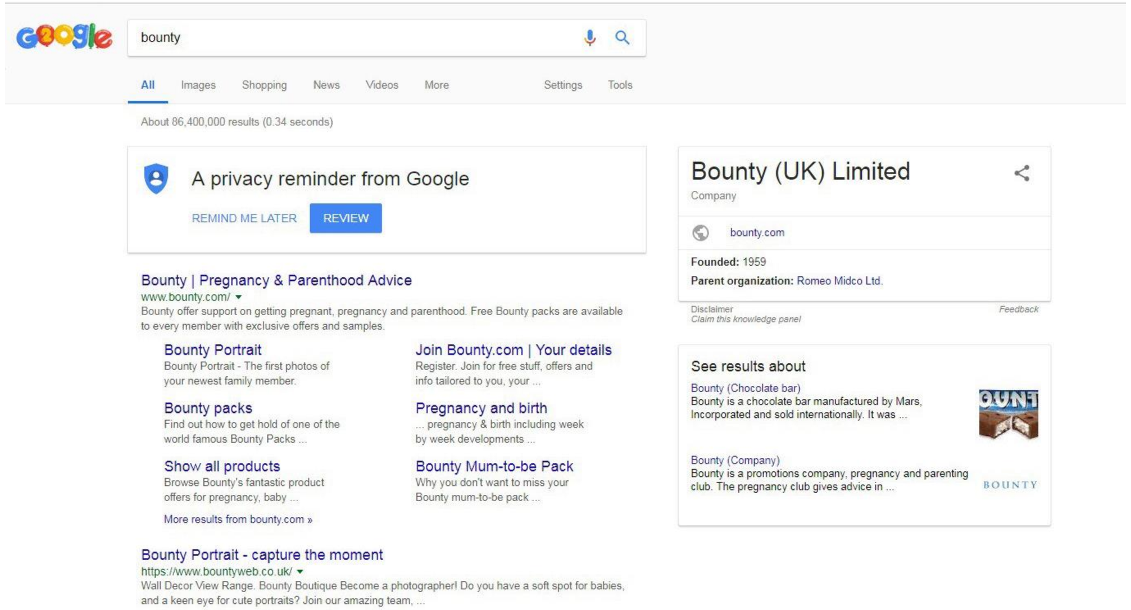


Location: “bounty”

United States



United Kingdom



Applications (cont'd)

- Market basket analysis: what items do customers buy together

  {Bread, Milk} => {Paper Towel}

- Recommender System: A sales manager at an electronic store talking to a customer who recently purchased a computer and a camera, what should he recommend next?

  {Video camera} => {warranty, memory card}

- Customer relationship management: identify preferences of different customer groups {Home, 2 cars} => {Policy A}, {Home, Ann Arbor} => {Policy B}

- Medical Diagnosis: find associations in symptoms and observations to predict diagnosis {Fever, lethargic, vomiting} => {Food Poisoning}

Market Basket Analysis

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**

Each row in this table corresponds to a transaction, which contains a unique identifier and a set of items bought by a given customer. Retailers are interested in analyzing the data to learn about the purchasing behavior of their customers.

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!

Issues

- Discovering patterns from large transaction data: computationally extensive
- Discovery of fake patterns

Definitions

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**

- Itemset: $I = \{i_1, i_2, \ldots, i_d\}$
  - The set of all items in the market basket data

    $I = \{Bread, Milk, Diaper, Beer, Eggs, Coke\}$

  - K-itemset: an itemset containing k items
  - Null/empty itemset: an itemset that does not contain any items
- Transaction set: $T = \{t_1, t_2, \ldots, t_N\}$
- Each transaction t_i contains a subset of I

Definitions

- **Support** of an itemset X: number of transactions containing X
  - σ({Bread,Diapers})

    3
  - σ({Diapers, Milk, Coke})

    2

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

Definitions

- **Association Rule**:
  - Implication of the form X → Y, where X and Y are *disjoint* itemsets
  - {Bread, Diapers} → {Milk}
  - {Bread} → {Milk}
- **Support of a rule:**
  - The fraction of transactions containing both X and Y

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

- **Confidence of a rule:**
  - The fraction transactions containing X that also contains Y

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

Example

Association Rule: $\{Bread, Diaper\} \longrightarrow \{Milk\}$

$$s(\{Bread, Diaper\} \rightarrow \{Milk\}) = \frac{\sigma(\{Bread, Diaper, Milk\})}{N} = \frac{2}{5}$$

$$c(\{Bread, Diaper\} \rightarrow \{Milk\}) = \frac{\sigma(\{Bread, Diaper, Milk\})}{\sigma(\{Bread, Diaper\})} = \frac{2}{3}$$

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

Interpretation

- Support:
  - Low: items may occur together by chance
  - Used to eliminate uninteresting rules
    - Transaction set contains 1000 transaction.
    - A single transaction contains the items {Band aids, TV}
    - No other transactions contain either item
    - What is the support for {TV} => {Band aids}?
    - What is the confidence for {TV} => {Band aids}?

σ({TV, Bandaids}) = *# trans. containing both* = 1

s({TV} => {Bandaids}) = σ({TV, Bandaids}) / N = 1/1000 = 0.001

c({TV} => {Bandaids}) = σ({TV, Bandaids}) / σ({TV}) = 1/1 = 1

Interpretation

- Confidence:
  - Measures reliability of implication
  - The higher the confidence, the more likely Y is present in transactions containing X
    - Transaction set contains 1000 transaction.
    - 200 transactions contain the items {Milk, Paper}
    - 250 transactions contain {Milk}
    - 800 transactions contain {Paper}
    - What is the support for {Milk} => {Paper}?
    - What is the confidence for {Milk} => {Paper}?

Co-occurrence / Not causality relationship

$\sigma(\{\text{Milk, Paper}\}) = \textit{\# trans. containing both} = 200$

$s(\{\text{Milk}\} \Rightarrow \{\text{Paper}\}) = \sigma(\{\text{Milk, Paper}\}) / N = 200/1000 = 0.2$

$c(\{\text{Milk}\} \Rightarrow \{\text{Paper}\}) = \sigma(\{\text{Milk, Paper}\}) / \sigma(\{\text{Milk}\}) = 200/250 = 0.8$

Association Rule Discovery Problem

- Given:
  - a set of transactions T
  - a minimum support *minsup*
  - A minimum confidence *minconf*

- Find all association rules having:
  - support >= *minsup*
  - confidence >= *minconf*

Association Rule Discovery

- Brute Force Approach: find all possible rules then filter.
- 2-Step Approach: find frequent items then generate rules.

Brute Force Approach

- Compute support and confidence of every possible rule
- Select only rules satisfying *minsup* and *minconf* threshold

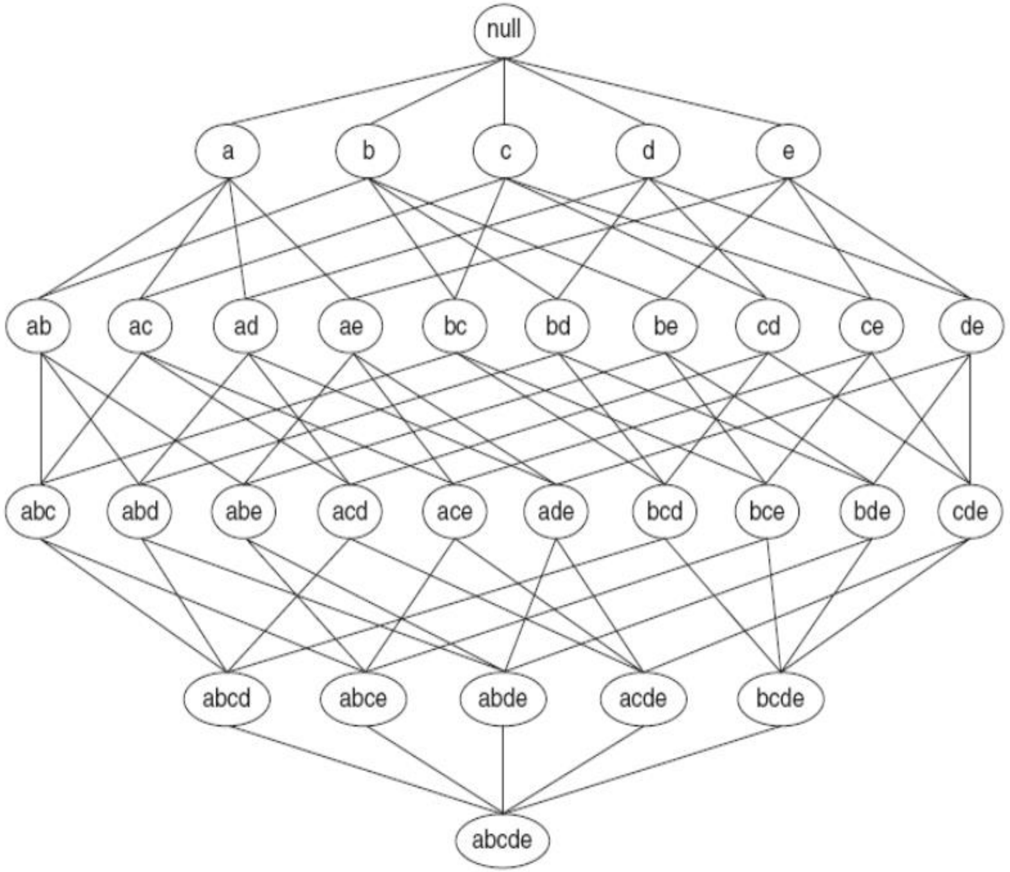
- Possible number of rules:

$R = 3^d - 2^{d+1} + 1$ (d: items set size)

Exponential $O(3^d)$

Prohibitively expensive!

- Example:
  - d = 6 $R = 3^6 - 2^7 + 1 = 602$
  - d = 10 $R = 3^{10} - 2^{11} + 1 = 57{,}002$
  - d = 15 R = 14,283,372



If a store has 1000 different items:

```
> R:= 3^d - 2^(d+1) + 1;
```

$R := 1322070819480806636890455259752144365965422032752148167664920368226828\backslash$
$8597346704899540778313850608061963909777696872582355950954582100618911\backslash$
$53427252579536740276202251983208038565846020852394948554218991363885818\backslash$
$99008515815124222891865584056470656519917837282615127680916298918451454\backslash$
$74664024317511732305069839876918094271408312667904387523472700944332427\backslash$
$97274846309584159379956326025667978550580533115253077585835467099769852\backslash$
$7992693374081496689754702826924329094491519081250$

If 1000 rules can be generated per second

```
> R/1000./3600/24/365;
```

$$.4192259068\ 10^{467}$$

Observation

- Support of a rule $X \longrightarrow Y$ depends on support of itemset $\{X \cup Y\}$

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

- Six possible rules from itemset *{Bread, Diaper, Milk}*:

$$\{Bread, Diaper\} \longrightarrow \{Milk\}, \{Milk\} \longrightarrow \{Bread, Diaper\}$$
$$\{Bread, Milk\} \longrightarrow \{Diaper\}, \{Diaper\} \longrightarrow \{Bread, Milk\}$$
$$\{Diaper, Milk\} \longrightarrow \{Bread\}, \{Bread\} \longrightarrow \{Diaper, Milk\}$$

- If the itemset has low support:
  - All six rules have low support
  - Can be pruned

Better Approach

2-Step Approach: find frequent items then generate rules.

1. **Frequent itemset** generation
   - Generate all itemsets satisfying *minsup*
2. Rule Generation:
   - Extract rules satisfying *minconf* from **frequent itemsets**

1

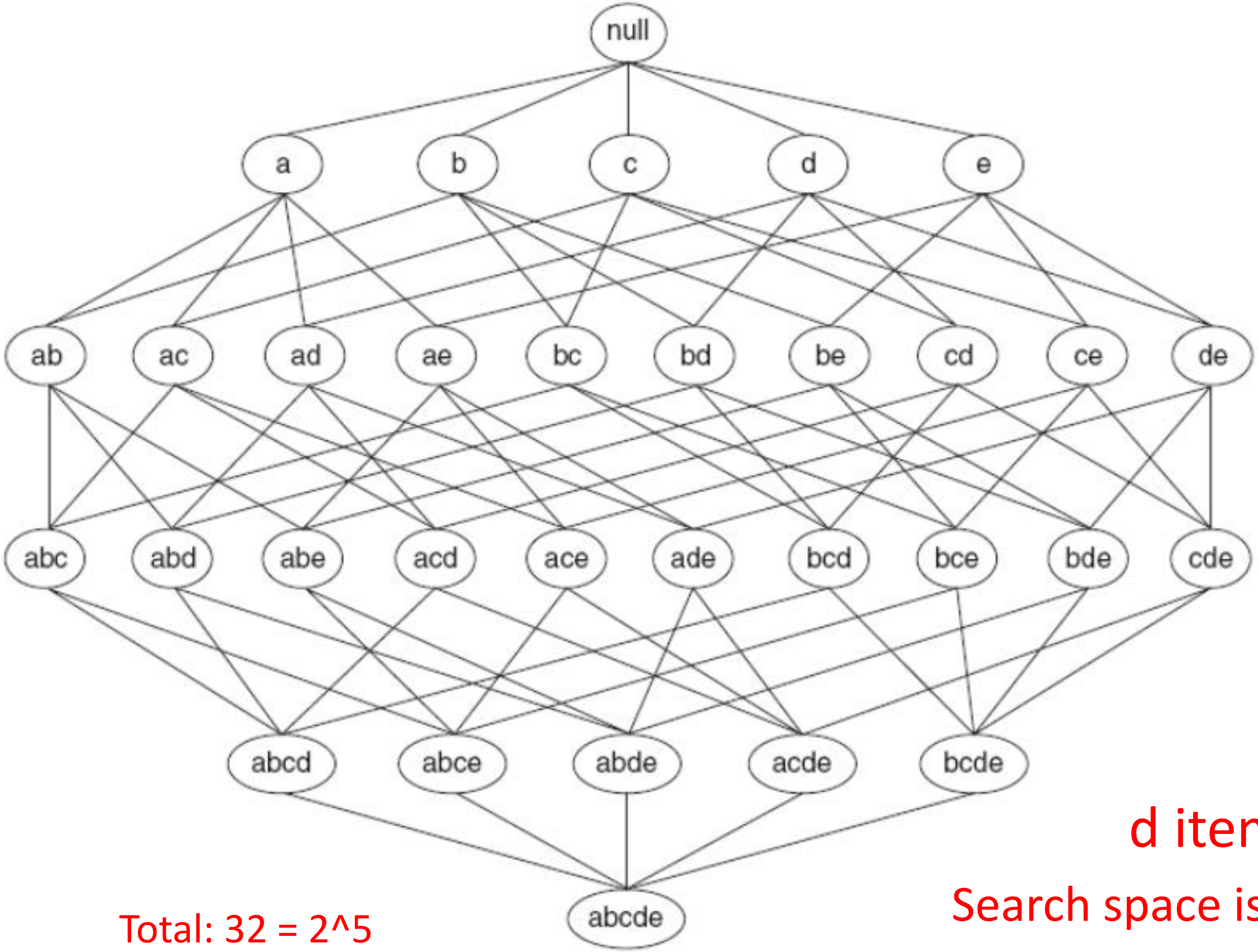
5

5 choose 2: 10

5 choose 3: 10

5 choose 4: 5

5 choose 5: 1



Total: 32 = 2^5

d items: 2^d itemsets

Search space is exponentially large

How to reduce computational complexity

- Reduce the number of candidate itemsets using the *Apriori* principles.
- Reduce the number of comparison: instead of matching each candidate itemset against every transaction, we can reduce the number by using more advanced data structures.
- Reduce the number of transactions

Apriori Principle

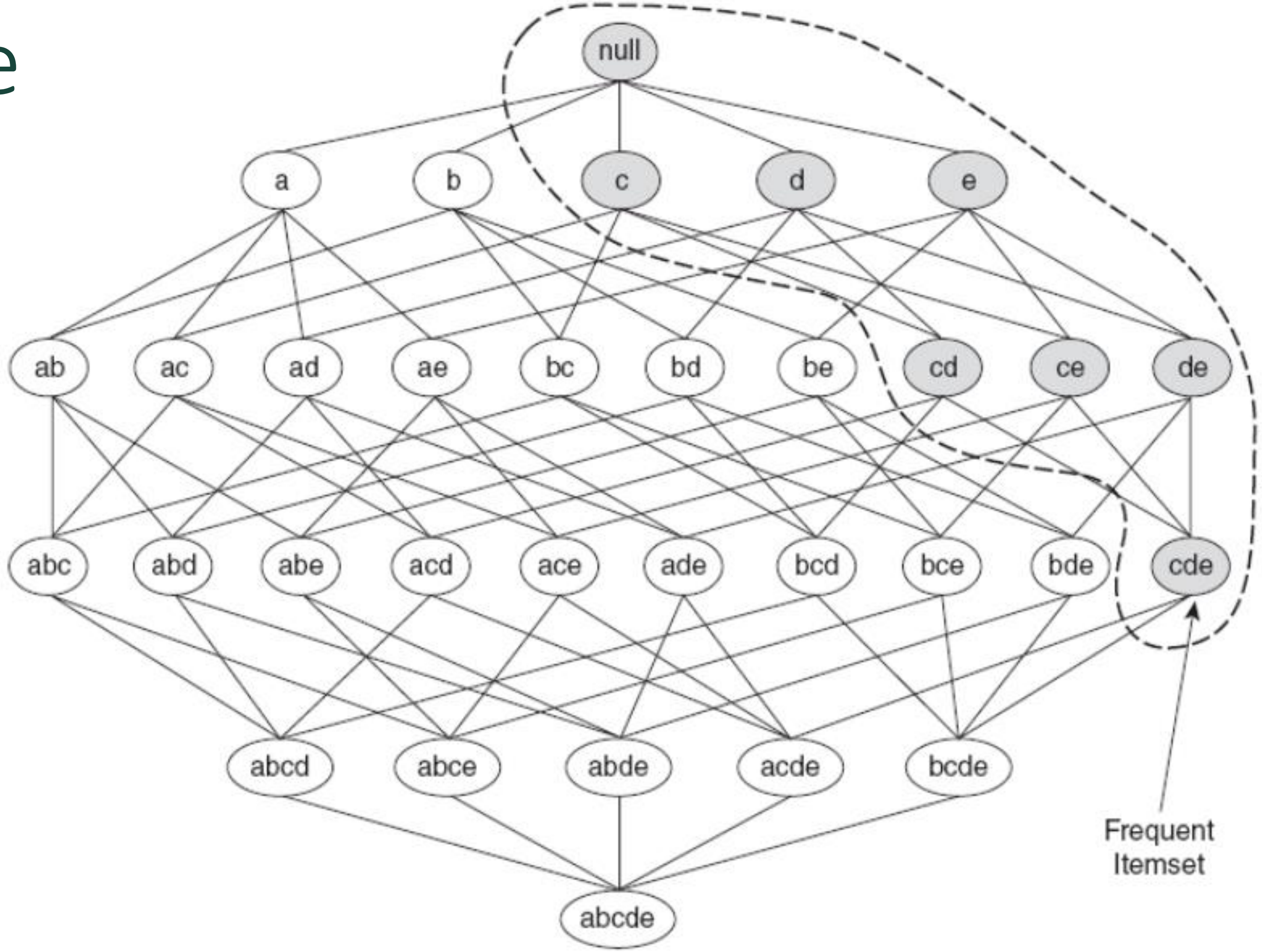
Anti-monotone property:
The support of an itemset never exceeds the support for the subsets

i.e., support of an itemset <= support for its subsets

Apriori Principle

Anti-monotone property: support of an itemset <= support for its subsets

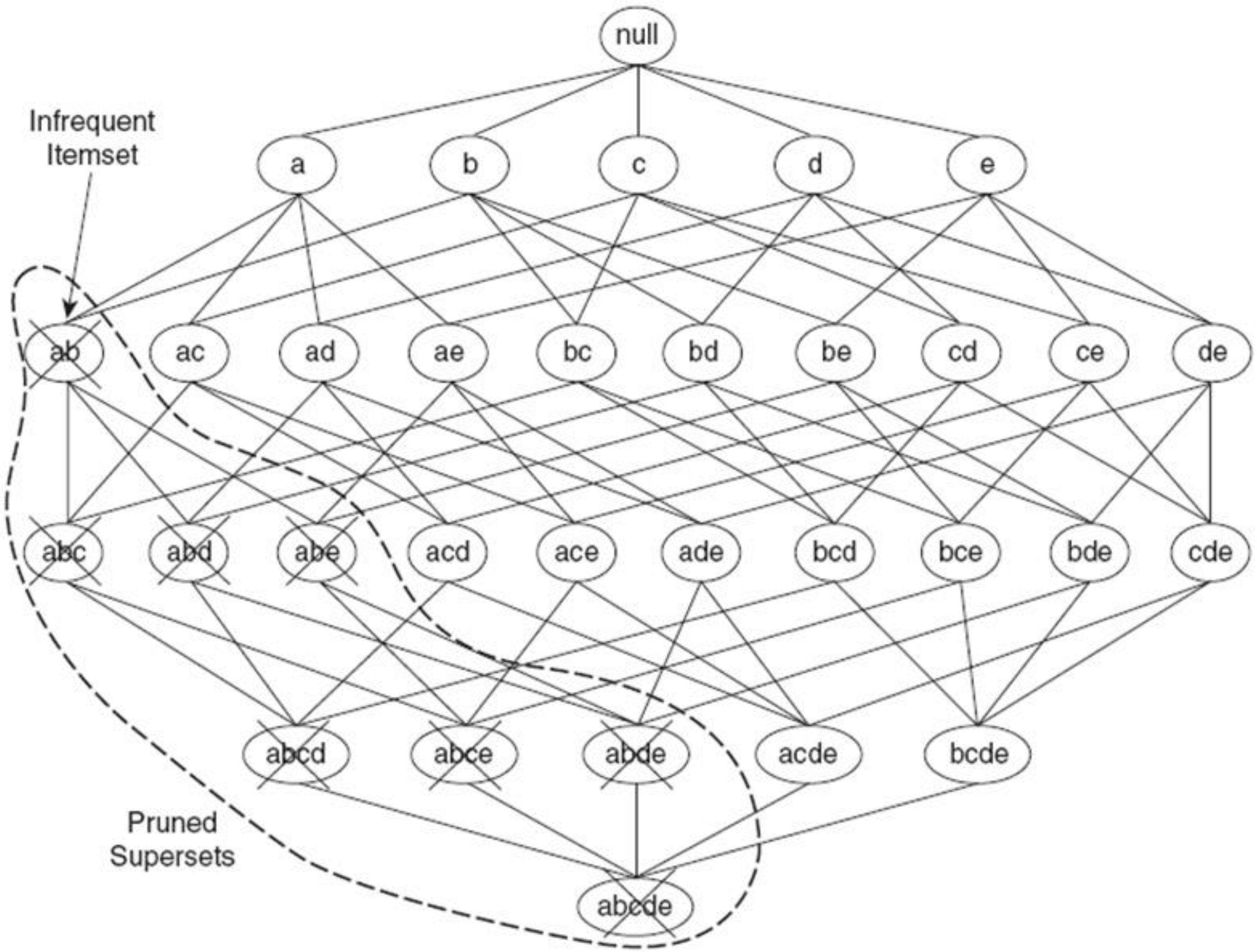
- If an itemset is frequent, then all its subsets are frequent



Apriori Principle

Anti-monotone property: support of an itemset <= support for its subsets

- If an itemset is infrequent, then all its supersets are infrequent



Apriori Algorithm

- C_k: Candidate itemsets of size k (itemsets possibly frequent)
- F_k: Frequent itemsets of size k
- Compute F_1:
  - A single pass over the transactions table to count support of individual items
- Iteratively, use F_{k-1} to compute C_k and then F_k
- Stop when F_k is empty
- A pass over the transactions is needed to count the support of every C_k

Example

Minimum support count = 3

Candidate
1-Itemsets

Item	Count
Beer	3
Bread	4
Cola	2
Diapers	4
Milk	4
Eggs	1

Candidate
2-Itemsets

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

Itemsets removed because of low support

Candidate
3-Itemsets

Itemset	Count
{Bread, Diapers, Milk}	3

Candidate Set Generation

- Avoid generating too many unnecessary candidates
- Ensure the set is complete: no frequent itemset is left out
- Do not generate duplicate itemsets

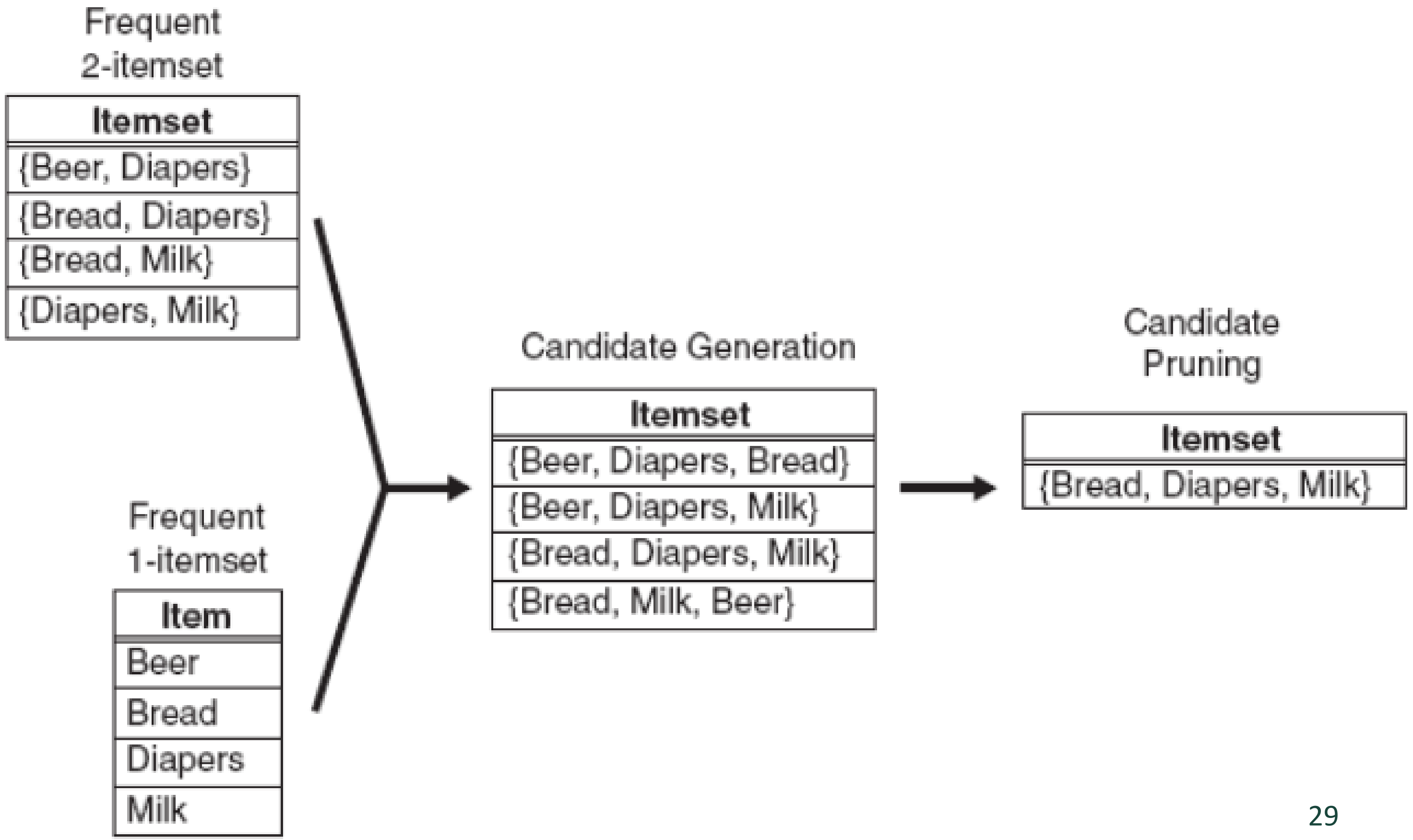
{a, b, c, d} can be generated by merging:

- *{a, b, c} and {d}*
- *{a, c} and {b, d}*
- *{c} and {a, b, d}*

F_{k-1} x F_1 Method

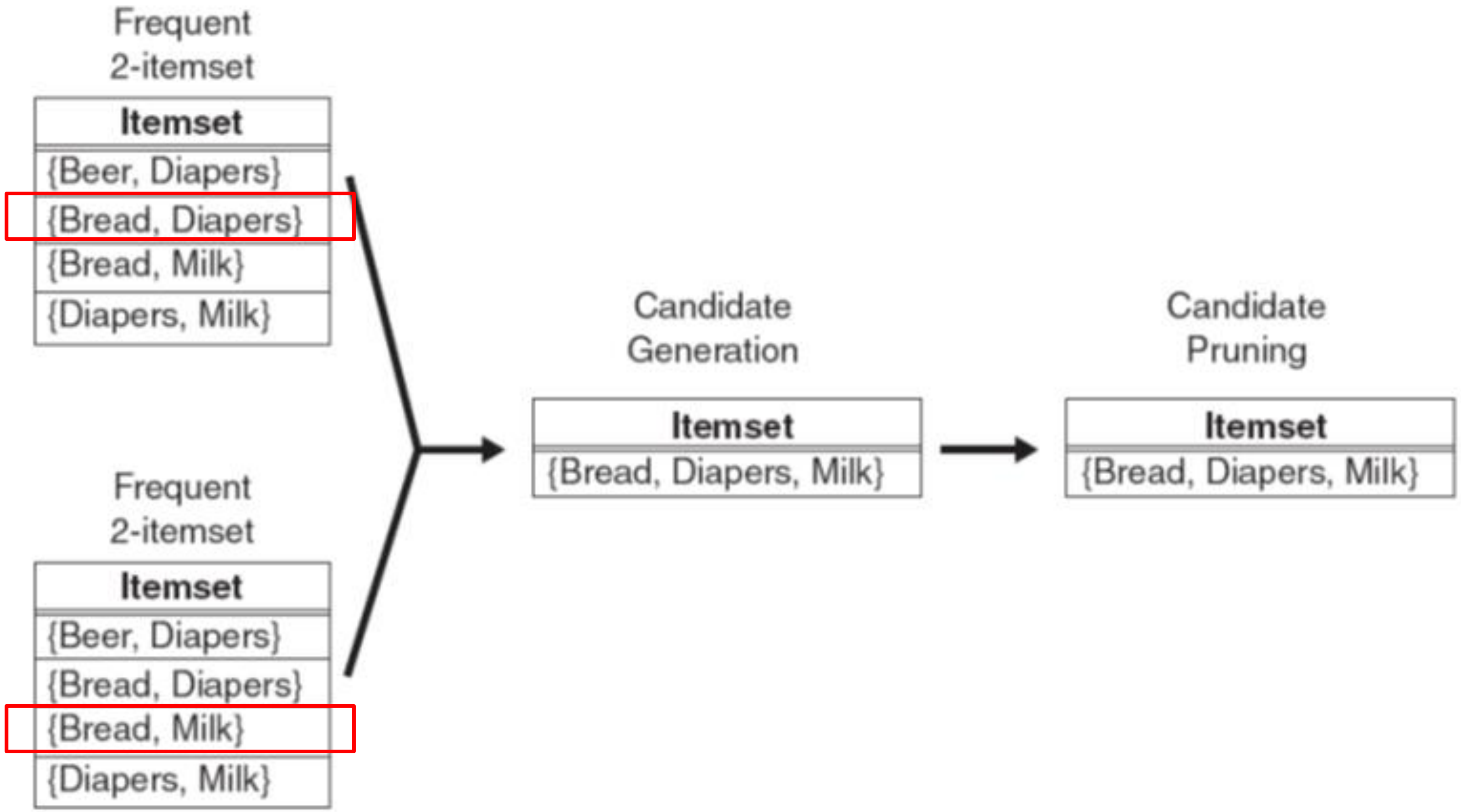
- Extend each itemset in F_{k-1} by a frequent item in F_1

Use F_{k-1} to compute C_k



$F_{k-1} \times F_{k-1}$ Method (Apriori Gen)

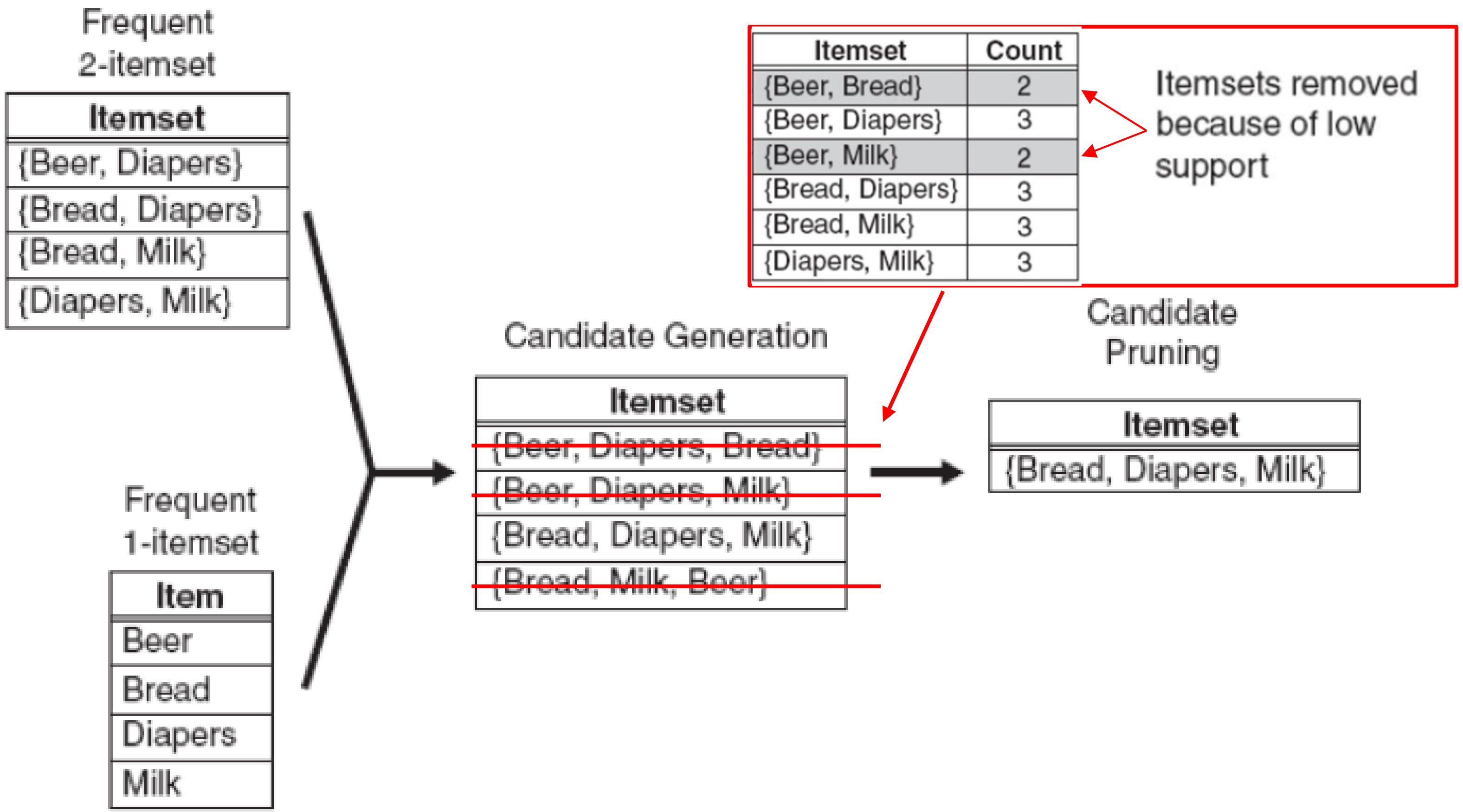
- Merge two itemsets in F_{k-1} if their first k-2 items are identical



Candidate Pruning

- Remove itemsets containing infrequent subsets:

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



Support Counting

- Compare each transaction against every itemset: computationally expensive!

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**

Itemset
{Bread, Diapers, Milk}

Improving Efficiency

- **Transaction Reduction**:
  - a transaction that does not contain any frequent k-itemset cannot contain any frequent (k+1)-itemset

- **Sampling**: pick a random sample and find frequent itemsets on sample. Trading accuracy for efficiency

Computational Complexity

- **Support threshold**: lower support implies:
  - More frequent itemsets, more candidate itemsets
  - Larger frequent itemsets (larger k)
- **Number of items**:
  - More space needed to store support counts
  - Increases the number of candidate itemsets
- **Number of transactions**:
  - Increases the time needed for a pass of the data
- **Transaction Width**:
  - Increases the maximum size of frequent itemsets

Rule Generation

2-Step Approach: find frequent items then generate rules.

✓ **Frequent itemset** generation
- Generate all itemsets satisfying *minsup*

-- Rule Generation:
- Extract rules satisfying *minconf* from **frequent itemsets**

Rule Generation

- Given the minimum confidence *minconf* , generating association rules by going through all possible combinations of **frequent item** sets and pruning the rules according to confidence criterion.

- Given a frequent k-itemset Z:
  - There are 2^k-2 possible association rules
  - Ignoring $\varnothing \rightarrow Z$ and $Z \rightarrow \varnothing$

- When considering rule , both $X \cup Y$ and X are frequent
  - Support is already computed
  - All rules satisfy *minsup*
  - Do not need to traverse transaction table

- Select only those satisfying *minconf*

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

Rule Generation

- Generate all nonempty subsets for each frequent itemset
- For every nonempty subset S of Itemset I , output of the rule:
  - S --> (I - S)
  - **If** support_count (I) / support_count (S) > = minimum confidence threshold **then** rule is a **strong Association Rule.**

TID	Items Bought
1	{ A, C }
2	{ A, B, C, E}
3	{ A, D }
4	{ A, B, C, E }
5	{ A, B, C, D, E }

minimum threshold support = 60%
minimum threshold confidence = 80%

Item set	Support Count	Support
{A, B, C }	3	60%

Rule Generation

minimum threshold confidence = 80%

Item set	Support Count	Support
{A, B, C }	3	60%

Step 1:

- Generate all nonempty subsets for each frequent itemset
  - For Itemset - { A, B, C } , all non empty subsets are {A,B}, {B,C}, {A,C}, {A}, {B}, {C}

Step 2.1:

- For every nonempty subset S of Itemset I , output of the rule:
  - S --> (I - S)
    - {A,B} -> {C}
    - {B,C} -> {A}
    - {A,C} -> {C}
    - {A} -> {B,C}
    - {B} -> {A,C}
    - {C} -> {A,B}

Rule Generation

minimum threshold confidence = 80%

Item set	Support Count	Support
{A, B, C }	3	60%

Step 2.2:

- **If** support_count (I) / support_count (S) > = minimum confidence threshold **then** rule is a **strong Association Rule.**
  - {A,B} -> {C}, Confidence = 3/3 * 100 = 100% - **Yes**, it is a strong association rules
  - {B,C} -> {A}, Confidence = 3/3 * 100 = 100% - **Yes**, it is a strong association rules
  - {A,C} -> {C}, Confidence = 3/4 * 100 = 80% - **Yes**, it is a strong association rules
  - {A} -> {B,C}, Confidence = 3/5 * 100 = 60% - **No**, it is not a strong association rules
  - {B} -> {A,C}, Confidence = 3/3 * 100 = 100% - **Yes**, it is a strong association rules
  - {C} -> {A,B}, Confidence = 3/4 * 100 = 80% - **Yes**, it is a strong association rules

Credit Card Promotion Database

- 10 samples
- Single itemsite can be twice as large than previous example

Magazine Promo	Watch Promo	Life Ins Promo	Credit Card Ins.	Sex
Yes	No	No	No	Male
Yes	Yes	Yes	No	Female
No	No	No	No	Male
Yes	Yes	Yes	Yes	Male
Yes	No	Yes	No	Female
No	No	No	No	Female
Yes	No	Yes	Yes	Male
No	Yes	No	No	Male
Yes	No	No	No	Male
Yes	Yes	Yes	No	Female

Single item sets at a 40% coverage threshold:

single item sets	Number of items
A. Magazine Promo=Yes	7
B. Watch Promo=Yes	4
C. Watch Promo=No	6
D. Life Ins Promo=Yes	5
E. Life Ins Promo=No	5
F. Credit Card Ins=No	8
G. Sex=Male	6
H. Sex=Female	4

Credit Card Promotion Database

Single item sets at a 40% coverage threshold:

single item sets	Number of items
A. Magazine Promo=Yes	7
B. Watch Promo=Yes	4
C. Watch Promo=No	6
D. Life Ins Promo=Yes	5
E. Life Ins Promo=No	5
F. Credit Card Ins=No	8
G. Sex=Male	6
H. Sex=Female	4

Now begin pairing up combinations with the same minimal support threshold (40%)

	A	B	C	D	E	F	G	H
B	3	-						
C	4		-					
D	5			-				
E	2		4		-			
F	5		5		5	-		
G	4		4		4	4	-	
H						4		-

Credit Card Promotion Database

	A	B	C	D	E	F	G	H
B	3	-						
C	4		-					
D	5			-				
E	2		4		-			
F	5		5		5	-		
G	4		4		4	4	-	
H						4		-

Resulting rules from two item sets. Consider rules in both directions:

1. (A → D)
   (MagazinePromo=Yes)→ (LifeInsPromo=Yes) at 5/7 confidence
2. (D → A)
   (LifeInsPromo=Yes) → (MagazinePromo=Yes) at 5/5 confidence
3. twenty more rules from the 10 two-item-sets (A then C, C then A, A then F, F then A, etc.)

Now apply minimum confidence threshold

If confidence threshold would be 80%, then the first rule (A → D) is eliminated.