Frequent Itemset and Association Rule Mining

Shantanu Jain

Market Basket Analysis

Baskets of items

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

Association Rules

{Milk} --> {Coke}
{Diaper, Milk} --> {Beer}

The Market-Basket Model

Input:

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

Output:

Rules Discovered:

{Milk} --> {Coke}

{Diaper, Milk} --> {Beer}

- Items = products/goods; Itemset: any set of items. k-Itemset: a set of k items
- Basket/Transaction = set of items purchased by a customer at a given point in time.
- Brick and Mortar: Track purchasing habits
  - Chain stores have TBs of transaction data
  - Tie-in “tricks”, e.g., sale on diapers + raise price of beer
  - Need the rule to occur frequently, or no $$’s
- Online: Might be able to make profit from infrequent, but strong association rules.

adapted from: J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Frequent Itemsets

- Simplest question: Find sets of items that appear together "frequently" in baskets
- Support $\sigma(X)$ for itemset X: Number of baskets containing all items in X
  - Fractional Support s(X) for itemset **X**: Fraction of baskets containing all items in **X**, σ(X)/N
- Given a *support threshold* σ_{min}, then sets of items X that appear in at least $\sigma(X) \geq \sigma_{min}$ baskets are called *frequent itemsets*

Example: Frequent Itemsets

- Items = {milk, coke, pepsi, beer, juice}
- Baskets

  B_1 = {m, c, b}　　B_2 = {m, p, j}

  B_3 = {m, b}　　B_4 = {c, j}

  B_5 = {m, c, b}　　B_6 = {m, c, b, j}

  B_7 = {c, b, j}　　B_8 = {b, c}

- Itemsets with frequency $\sigma(X) \geq 3$

  {m}:5, {c}:6, {b}:6, {j}:4,
  {m,c}: 3, {m,b}:4, {c,b}:5, {c,j}:3,
  {m,c,b}:3

Association Rules

- If-then rules about the contents of baskets
- $\{a_1, a_2, \ldots, a_k\} \rightarrow \{b\}$ means: "if a basket contains all of $a_1, \ldots, a_k$ then it is *likely* to contain *b*"
- In practice there are many rules, want to find significant/interesting ones!
- Two measures of significance for purchase $B = \{b\}$ given $A = \{a_1, \ldots, a_k\}$

Support (fractional): $s(A \cup B) = \sigma(A \cup B) / N$

Confidence: $s(A \cup B) / s(A) = \sigma(A \cup B) / \sigma(A)$

Interest of Association Rules

- Not all high-confidence rules are interesting
  - The rule $\boldsymbol{A} \rightarrow \boldsymbol{milk}$ may have high confidence because milk is just purchased very often (independent of $\boldsymbol{A}$)

- Lift of a rule $\boldsymbol{A} \rightarrow \boldsymbol{B}$:

Confidence and Interest

$B_1 = \{m, c, b\}$ $B_2 = \{m, p, j\}$

$B_3 = \{m, b\}$ $B_4 = \{c, j\}$

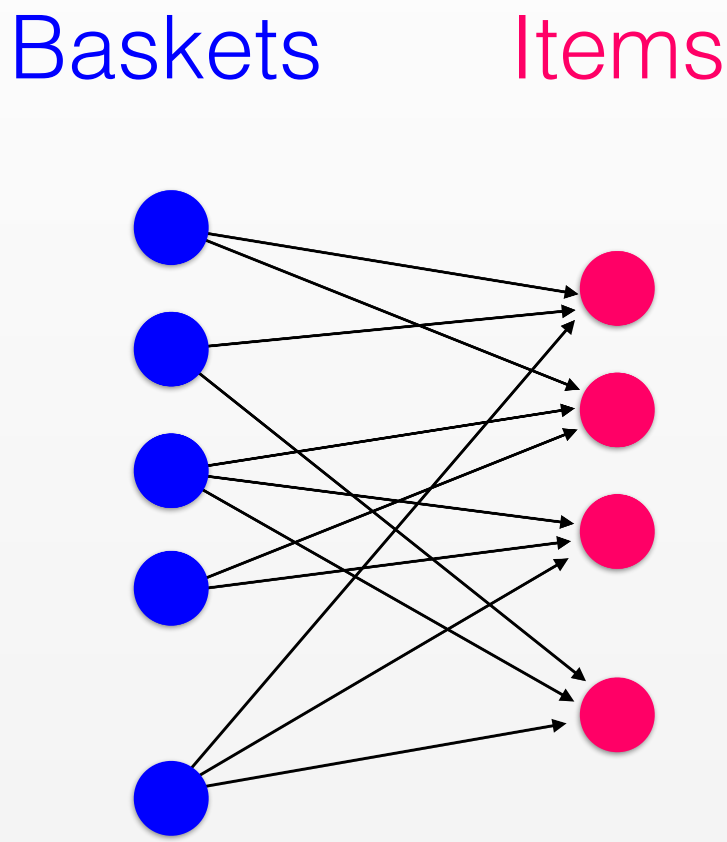
$B_5 = \{m, c, b\}$ $B_6 = \{m, c, b, j\}$

$B_7 = \{c, b, j\}$ $B_8 = \{b, c\}$

- Association rule: $\{m\} \rightarrow \{b\}$
  - Confidence = 4/5
  - Lift = 4/8 / (5/8 * 6/8) = 1.06
    - Item *b* appears in 6/8 of the baskets
    - Rule is not very interesting!

adapted from: J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

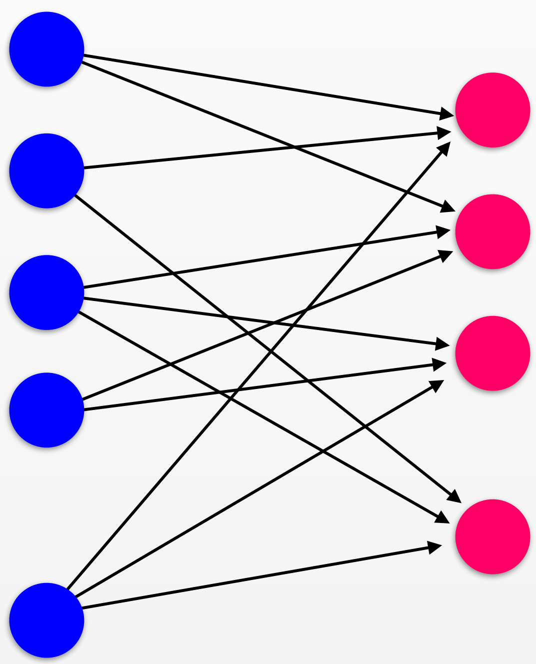
Other Applications



- **General view:** Association rules predict links between “basket” nodes and “item” nodes
- What is a “basket” and what is an “item” can vary from application to application.

Other Applications

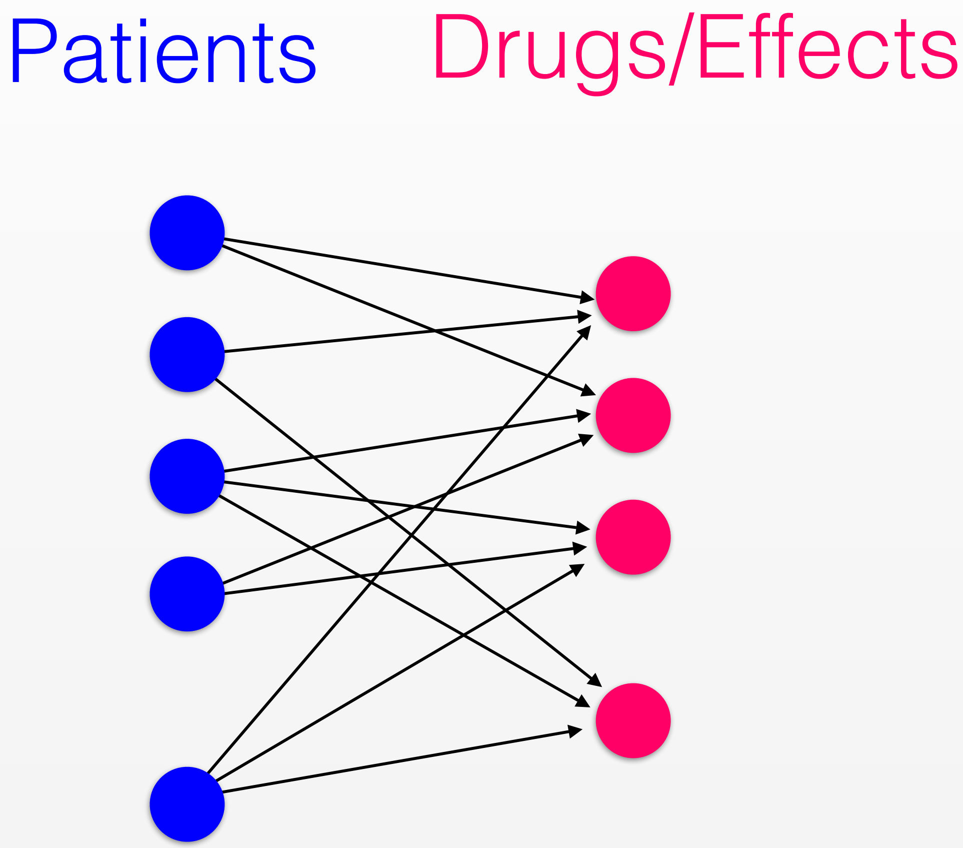
Sentences Documents



Plagiarism Detection

- Baskets = sentences; Items = documents containing those sentences
  - Frequents sets of documents could indicate plagiarism
  - Notice items do not have to be “inside” baskets

Other Applications



Drug Side Effects

- Baskets = patients;
  Items = drugs & side-effects
  - Detect combinations of drugs that result in side-effects
  - *Requires extension:* Needs to store absence as well as presence

Other Applications: Voting Records

Association Rule	Confidence
{budget resolution = no, MX-missile=no, aid to El Salvador = yes } $\longrightarrow$ {Republican}	91.0%
{budget resolution = yes, MX-missile=yes, aid to El Salvador = no } $\longrightarrow$ {Democrat}	97.5%
{crime = yes, right-to-sue = yes, physician fee freeze = yes} $\longrightarrow$ {Republican}	93.5%
{crime = no, right-to-sue = no, physician fee freeze = no} $\longrightarrow$ {Democrat}	100%

- Baskets = politicians; Items = party & votes
  - Can extract set of votes most associated with each party (or or faction within a party)

Up Next: Mining Association Rules

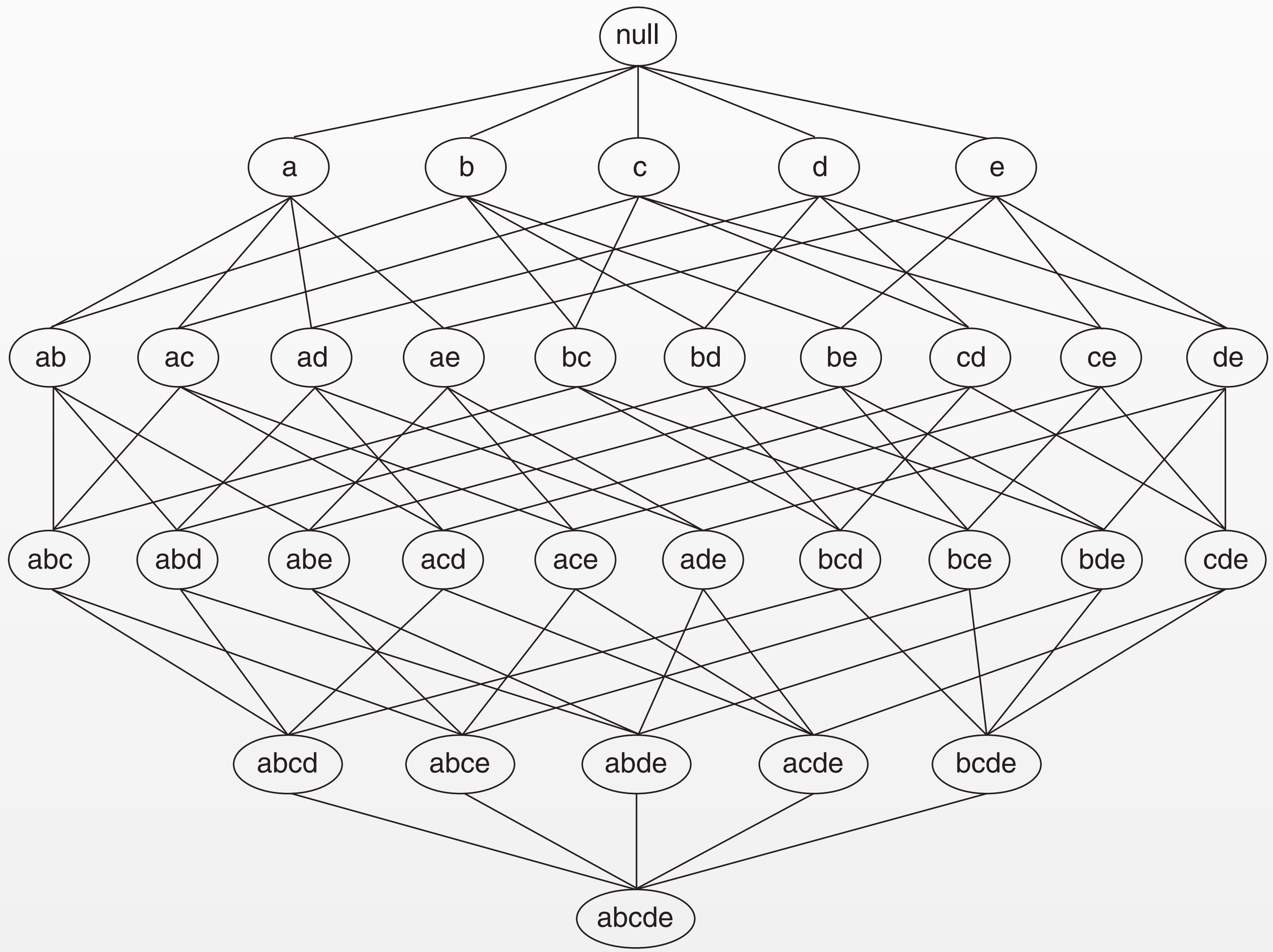
$$\{i_1, i_2, \ldots, i_k\} \rightarrow j$$

- **Problem:** Find **all** association rules with support $\geq s$ and confidence $\geq c$
  - **Note:** Support of an association $A \rightarrow B$ rule is the support of $A \cup B$
  - **Hard part:** Finding **all** frequent itemsets!
  - If $\{i_1, i_2, \ldots, i_k\} \rightarrow j$ has high support and confidence, then $\{i_1, i_2, \ldots, i_k\}$ and $\{i_1, i_2, \ldots, i_k, j\}$ will be frequent

Mining Frequent Itemsets with A-Priori

Finding Frequent Item Sets

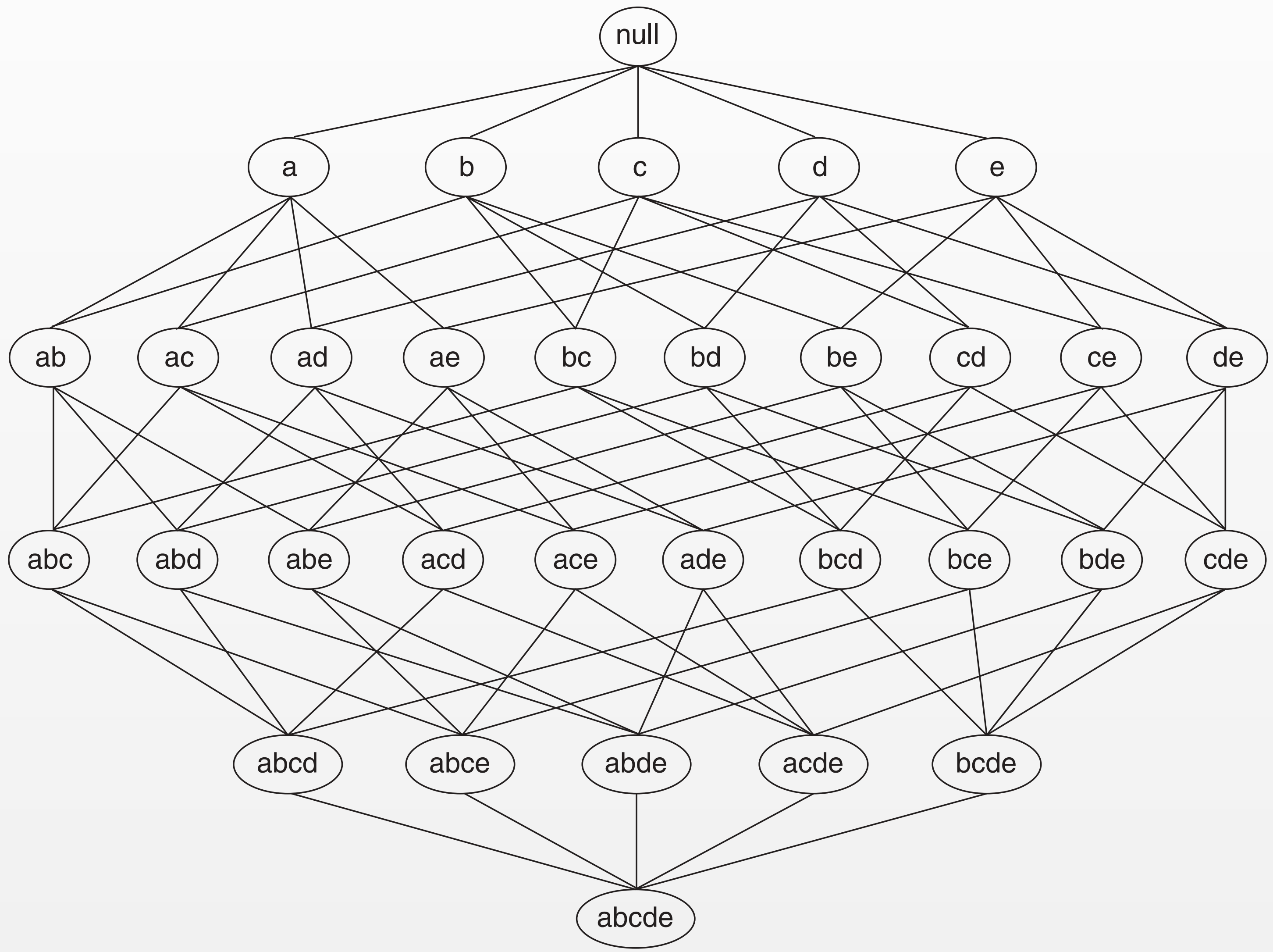
Given ***I*** products, how many possible item sets are there?



adapted from: Tan, Steinbach & Kumar, “Introduction to Data Mining”, http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf

Finding Frequent Item Sets

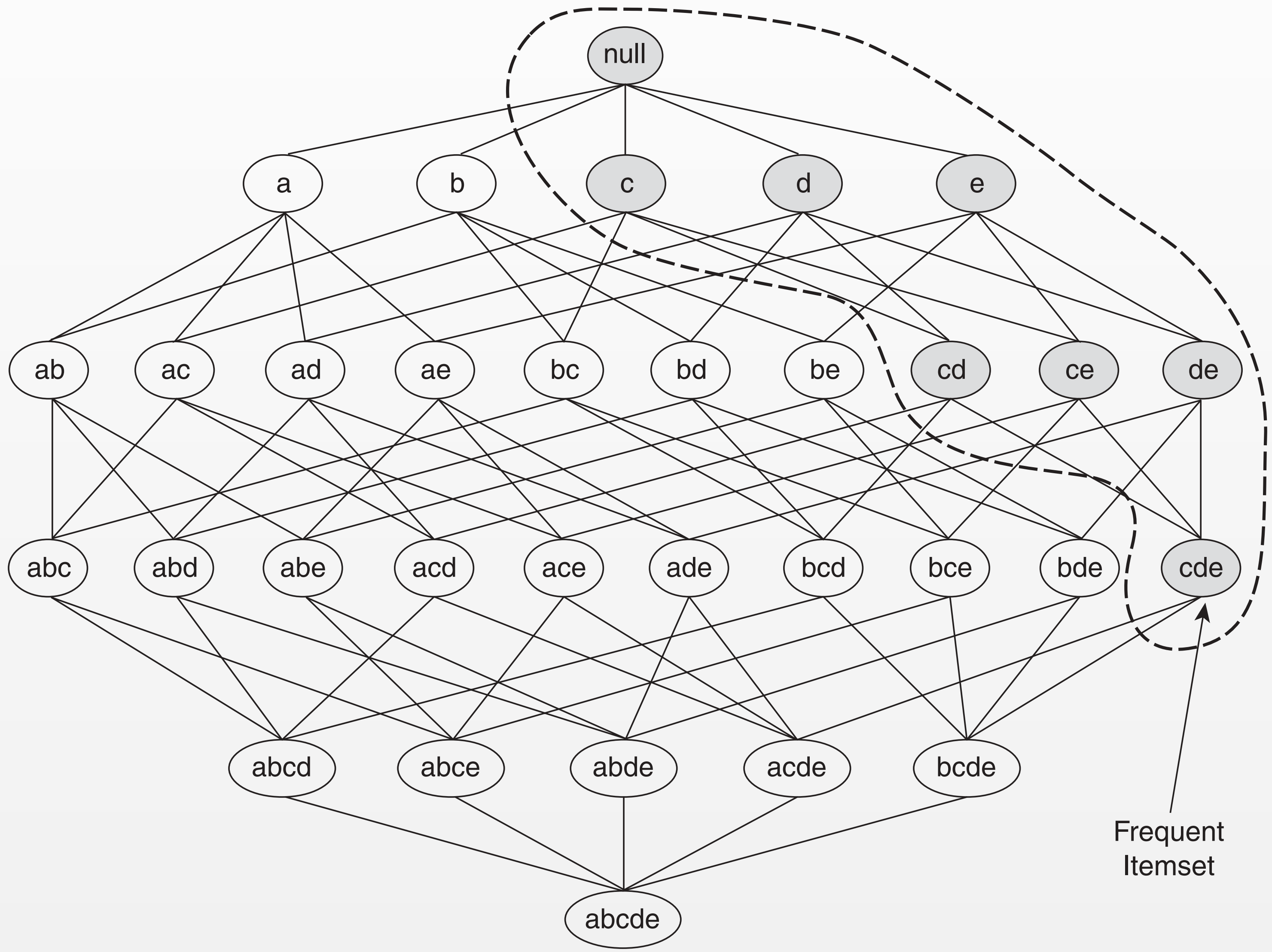
Answer: $2^l - 1$; Cannot enumerate all possible sets



adapted from: Tan, Steinbach & Kumar, "Introduction to Data Mining", http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf

Intuition: A-priori Principle

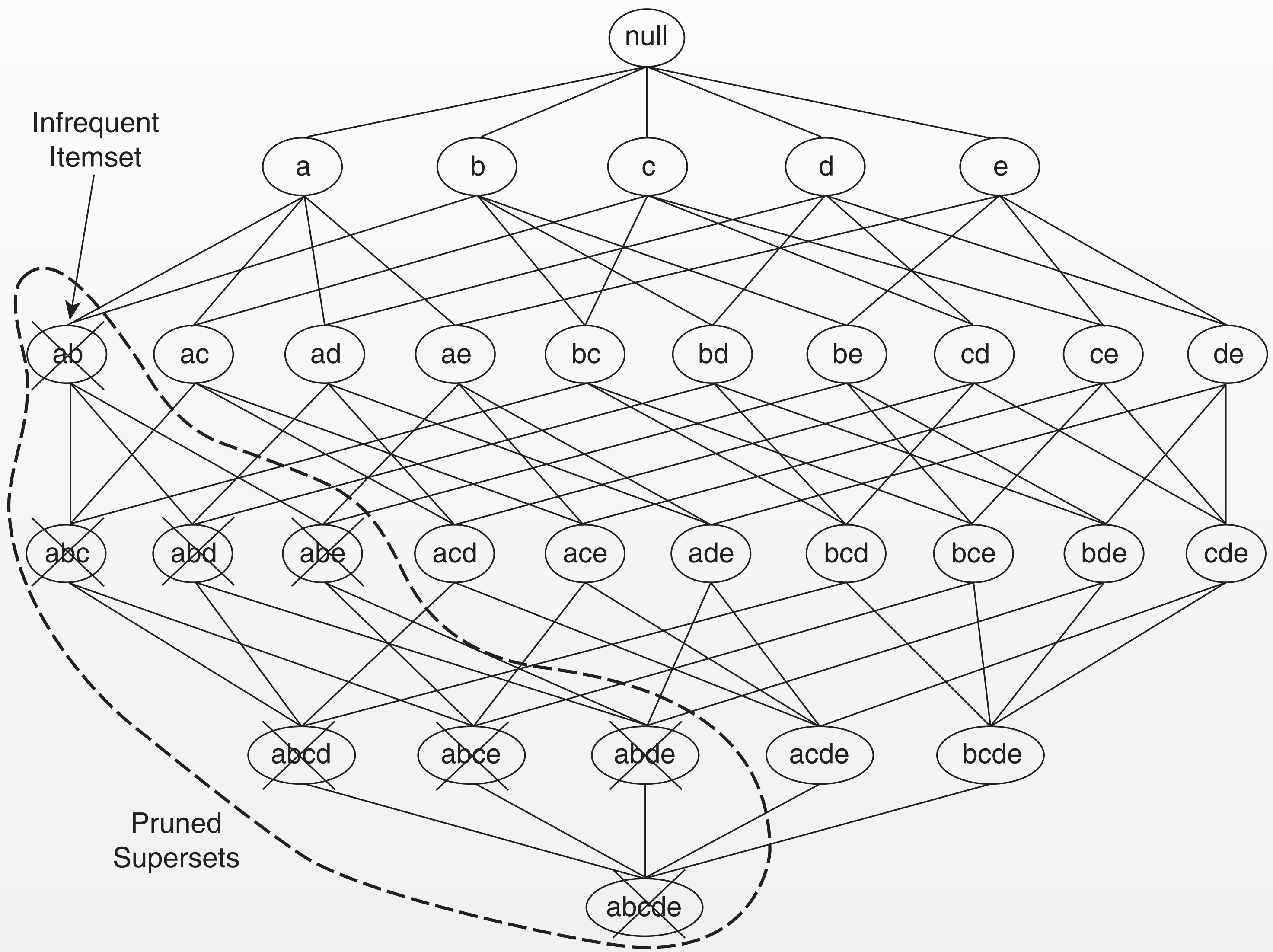
Observation: Subsets of a frequent item set are **also** frequent



adapted from: Tan, Steinbach & Kumar, "Introduction to Data Mining", http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf

Intuition: A-priori Principle

Corollary: If a set is **not** frequent, then its supersets are **also** not frequent



adapted from: Tan, Steinbach & Kumar, "Introduction to Data Mining", http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf

A-priori Algorithm

1. Find all frequent sets of size ***k*** **= 1**
   (*only have to check* ***I*** *possible sets*)
2. For ***k*** **= 2** … ***I***
   - Extend frequent sets of size ***k*** **- 1**
     to create *candidate* sets of size ***k***
   - Find candidate sets that are frequent