Lecture 21: Association Rule Mining

COMP90049
Knowledge Technology

Sarah Erfani and Vinh Nguyen, CIS

Semester 2, 2017

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

Supermarket 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

Example of Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Eggs,Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!

Real-world applications example 1

Marketing and Sales Promotion:

Let the rule discovered be

$$\{\text{Bagels}, \ldots\} \rightarrow \{\text{Potato Chips}\}$$

- Potato Chips as consequent →

  Can be used to determine what should be done to boost its sales.
- Bagels in antecedent and Potato chips in consequent →

  Can be used to co-locate Bagels and Potato Chips to further boost the sales of both products.
- Bagels in antecedent and Potato chips in consequent →

  Can be used to see what products should be sold with Bagels to promote sale of Potato chips!
- Bagels in the antecedent →

  Can be used to see which products would be affected if the store discontinues selling bagels.
- Bagels in antecedent and Potato chips in consequent →

  The store may reduce the price of Bagels to actually increase the profit!

Real-world applications example 2

Consumer appliance repair company:

Goal:

- Anticipate the nature of repairs on its consumer products,
- Keep the service vehicles equipped with right parts to reduce the number of visits required by consumer households, and
- Offer good customers service .

Approach:

- Process the data on tools and parts required in previous repairs at different consumer locations, and
- Discover the co-occurrence patterns.

Other real-world applications

- What products are often purchased together?
- What kinds of DNA are sensitive to this new drug?
- If a user clicks on a particular link, what other links are they likely to click on?

Issues

- What are interesting association rules?
- How do we get at association rules in large/high-dimensional datasets? (scalability)

Definition: Frequent Itemset

- **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. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

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

Definition: Association Rule

Association Rule

- An implication expression of the form $A \rightarrow B$, where A and B are itemsets

  A: antecedent

  B: consequent

- Example:
  {Milk, Diaper} $\rightarrow$ {Beer}

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**

Definition: Rule Evaluation Metrics

- **Support (s)**

  Fraction of transactions that contain both A and B

- **Confidence (c)**

  Measures how often items in A appear in transactions that contain B

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

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

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

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**

Definition: Rule Evaluation Metrics

- The *support count* $\sigma(X)$ of an itemset X is defined as the number of transactions that contain X, i.e.,

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

- We conventionally evaluate the "interestingness" of a given association rule via:
  - **support** $(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(*)} \left(\sim P(A, B)\right)$

    the proportion of transactions in the data set which contain the itemsets A and B
  - **confidence** $(A \rightarrow B) = \frac{\sigma(A \cup B)}{\sigma(A)} \left(\sim P(B|A)\right)$

    the proportion of the transactions for which items in B also appear in transactions containing A

- A **Frequent Itemset** has a support greater than a given minsup support threshold.

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

Brute-force approach

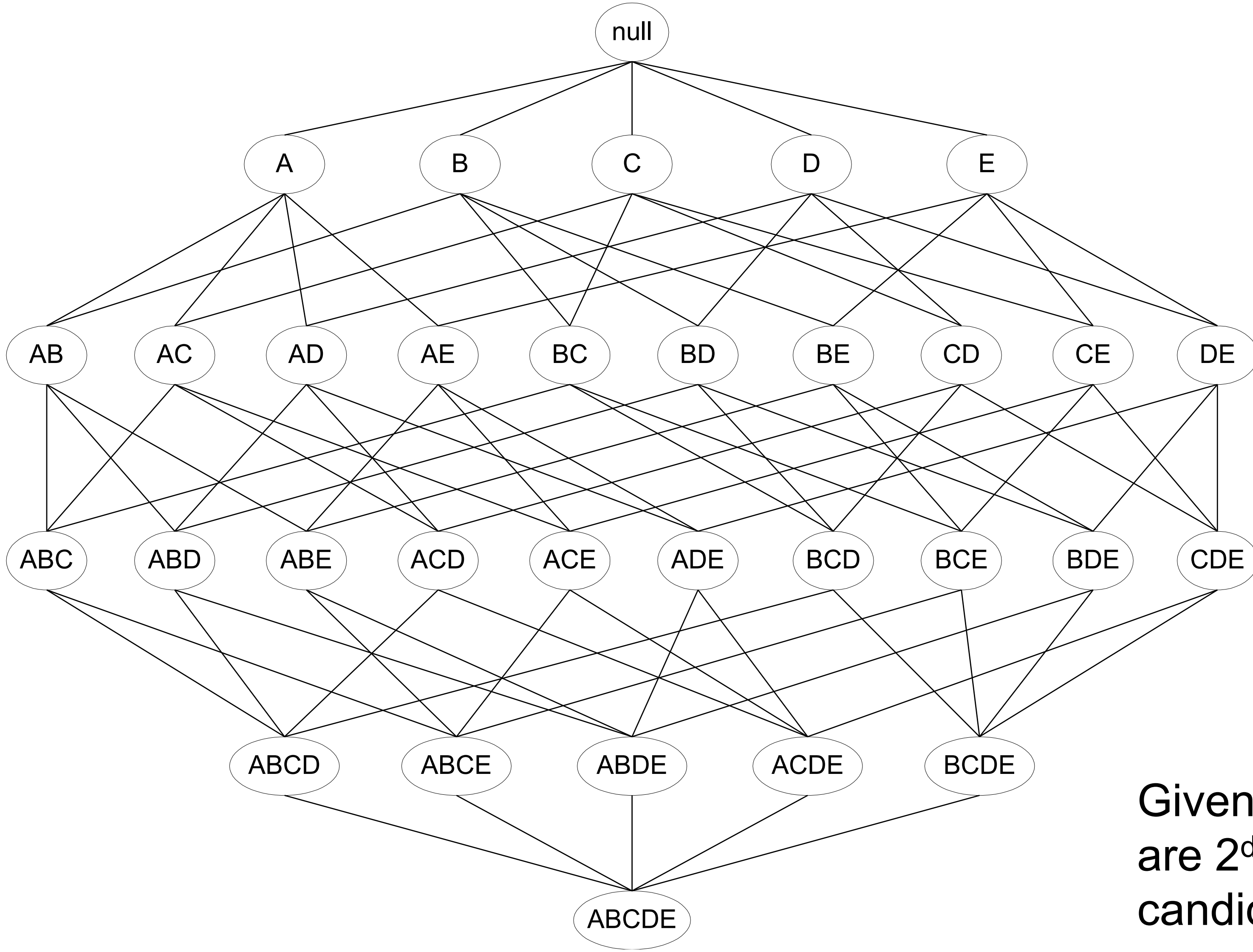
- List all possible association rules
- Compute the support and confidence for each rule
- Prune rules that fail the *minsup* and *minconf* thresholds

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

$\Rightarrow$ Computationally prohibitive!

Itemset Lattice



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

Exponential complexity

- d=100 items
- Total candidates: 2^{100}
- Can process 2 billions candidates per sec
  (Current CPU runs at 2-4 Ghz)
- Time required?

Exponential complexity

- d=100 items
- Total candidates: 2^{100}
- Can process 2 billions candidates per sec

  (Current CPU runs at 2-4 Ghz)
- Time required?

  7.85 Billion Billion years
- Age of the universe?

Mining Association Rules

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

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset:
  {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:

Step 1: Frequent Itemset Generation

Generate all itemsets whose support $\geq$ minsup

Step 2: Rule Generation

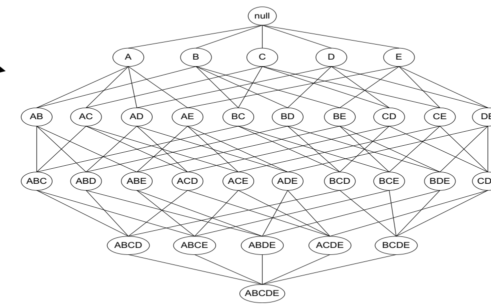
Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent itemset generation is still computationally expensive

Step 1: Frequent Itemset Generation

Brute-force approach:

Lattice



- Each itemset in the lattice is a candidate frequent itemset
- Count the support of each candidate by scanning the database

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

N

w

List of Candidates

M

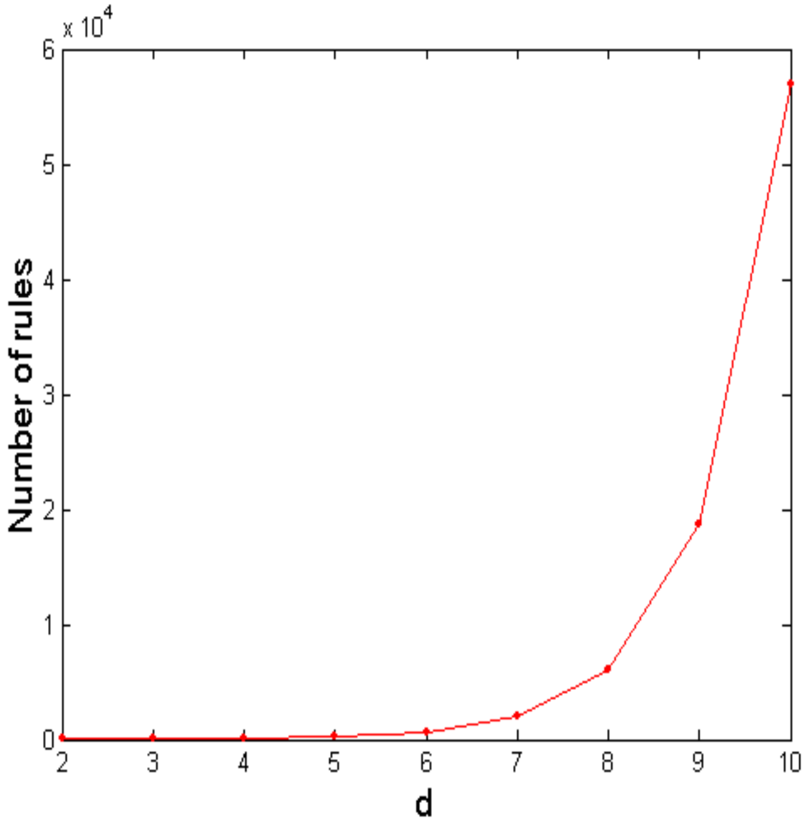
{Bread (Br)}
{Milk (M)}
…
({Br, M}
{Br,D}
…
{Br, M, D}
…
{Br, M, D, Be}
…
{Br, M, D, Be, C}
…
{Br, M, D, Be, C, E}

- Match each transaction against every candidate
- Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Computational Complexity

Given d unique items:

- Total number of itemsets = 2^d
- Total number of possible association rules:



#ways left side items can be chosen out of d items

#ways right side items can be chosen using the remaining d-k items

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

An example d= 3 and item set = {abc}

{a}	{b}	{a}	{c}	{a}	{bc}
{b}	{a}	{b}	{c}	{b}	{ac}
{c}	{a}	{c}	{b}	{c}	{ab}
{ab}	{c}	{ac}	{b}	{bc}	{a}

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 (Direct Hashing and Pruning) 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

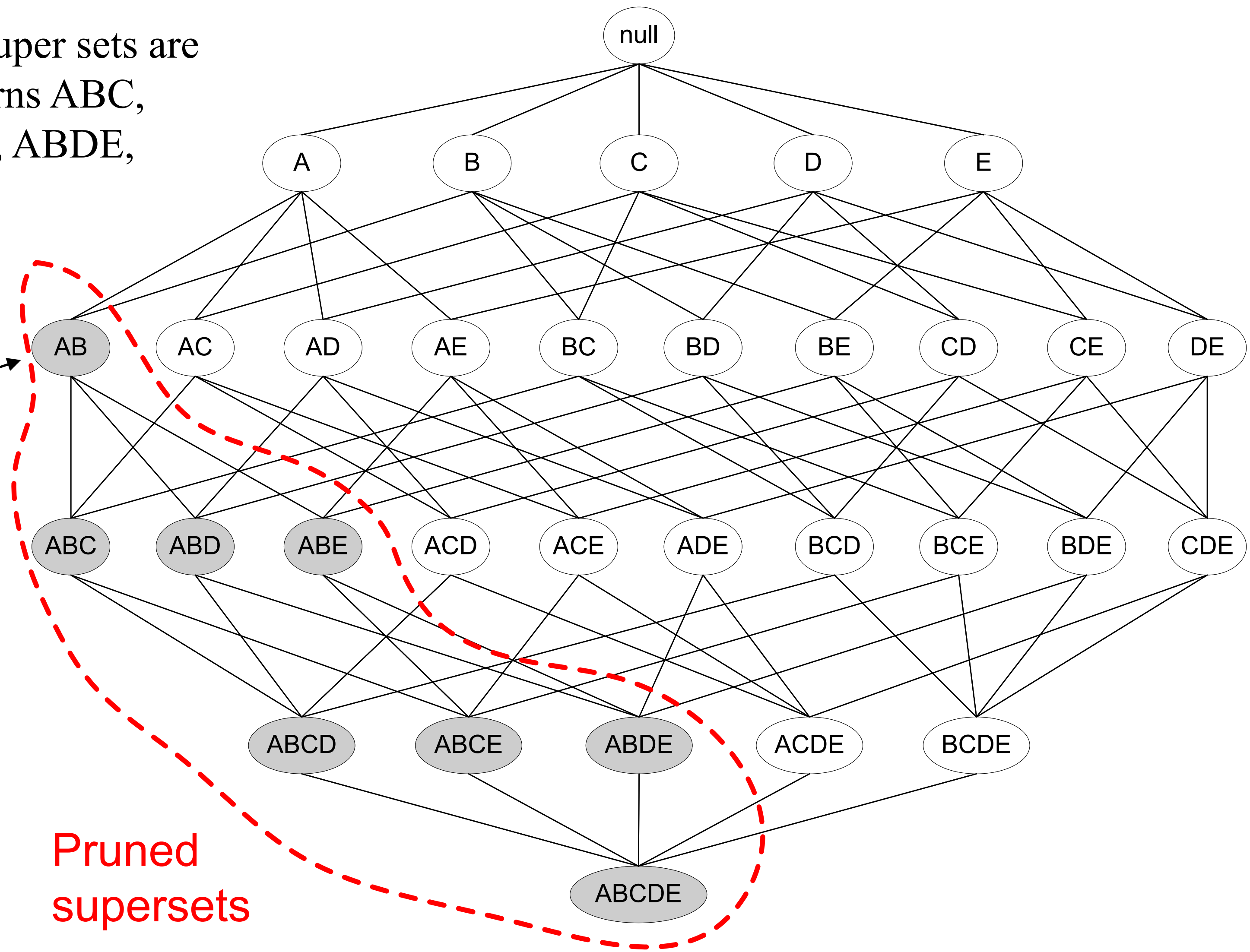
Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- Support of an itemset never exceeds the support of its subsets
- This is known as the anti-monotone property of support

Illustrating Apriori Principle

If AB is infrequent all its super sets are in frequent and hence patterns ABC, ABD, ABE, ABCD, ABCE, ABDE, ABCDE are all infrequent.



Apriori Algorithm

Method:

- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Prune candidate itemsets containing subsets of length k that are infrequent
  - Count the support of each candidate by scanning the database
  - Eliminate candidates that are infrequent, leaving only those that are frequent
  - Generate length (k+1) candidate itemsets from length k frequent itemsets

Apriori Algorithm

$R \leftarrow \phi$
for all $f_k \in \bigcup_{k=2}^{|I|} F_k$ **do** ▷ For each frequent itemset
    $m \leftarrow 1$ ▷ size of the consequent
    $H_m \leftarrow \{i | i \in f_k\}$ ▷ consequent set initially single items
    repeat
        $H_m^* \leftarrow H_m$ ▷ Candidate rules
        for all $h_m \in H_m$ **do**
            $c \leftarrow \sigma(f_k)/\sigma(f_k - h_m)$ ▷ Calculate the confidence of h_m
            if $c \geq N$ **then**
                $R \leftarrow R \cup \{(f_k - h_m) \longrightarrow h_m\}$ ▷ Add to final rule set
            else
                $H_m^* \leftarrow H_m^* - \{h_m\}$ ▷ Prune rule
            end if
        end for
        $H_{m+1} \leftarrow \text{apriori-gen}(H_m^*)$ ▷ Generate $m+1$-consequent rules
        $m \leftarrow m + 1$
    until $H_m = \phi$ **or** $m \geq |f_k| - 1$
end for
return R

Illustrating Apriori Principle

Items (1-itemsets)

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

Minimum Support = 3

Pairs (2-itemsets)
(No need to generate candidates involving Coke or Eggs as min support = 3)

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

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

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	2

If every subset up to 3 itemsets are considered,
Number of subsets = ${}^6C_1$(itemset size of 1)+ ${}^6C_2$ (itemset size of 2)+ ${}^6C_3$ (itemset size of 3)= 41

With support-based pruning (see tables above),
6 + 6 + 1 = 13

Reducing Number of Comparisons

Candidate counting:

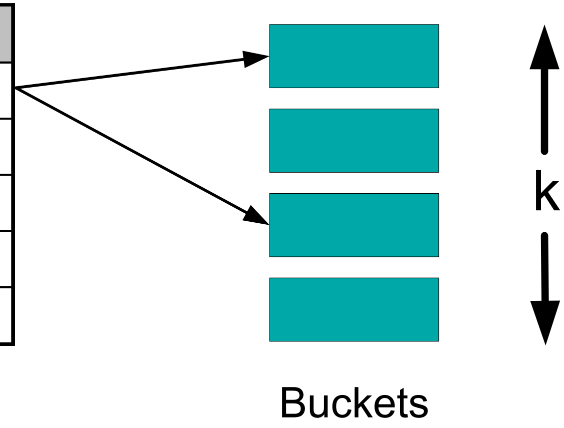
- Scan the database of transactions to determine the support of each candidate itemset
- To reduce the number of comparisons, store the candidates in a hash structure
  - *Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets*

Transactions

N

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

Hash Structure



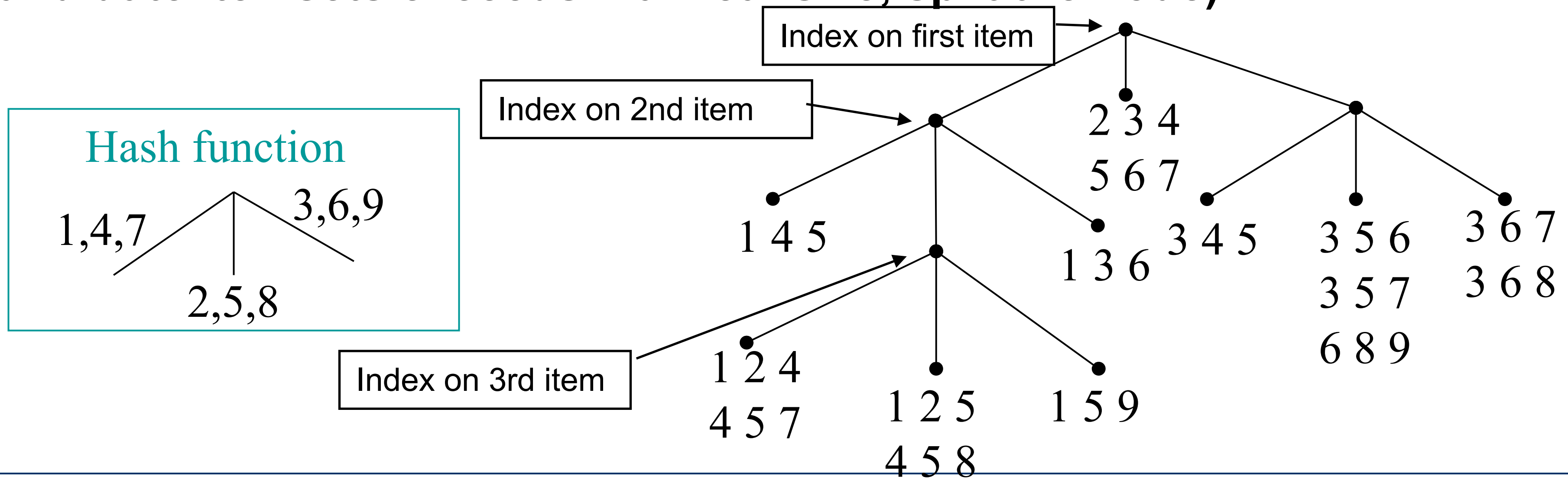
Generate Hash Tree

Suppose you have 15 candidate itemsets of length 3:

{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

We need:

- **Hash function**
- **Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)**

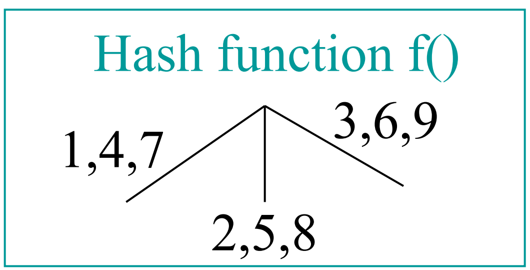


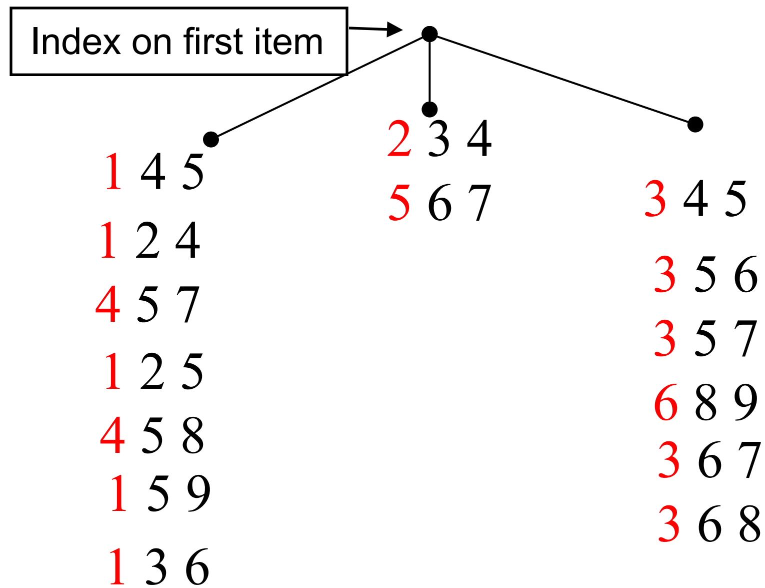
Generate Hash Tree

Suppose you have 15 candidate itemsets of length 3:

{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

f(1 or 4 or 7) = left branch
f(2 or 5 or 8) = middle branch
f(3 or 6 or 9) = right branch

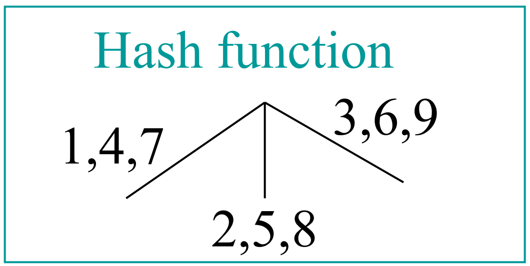


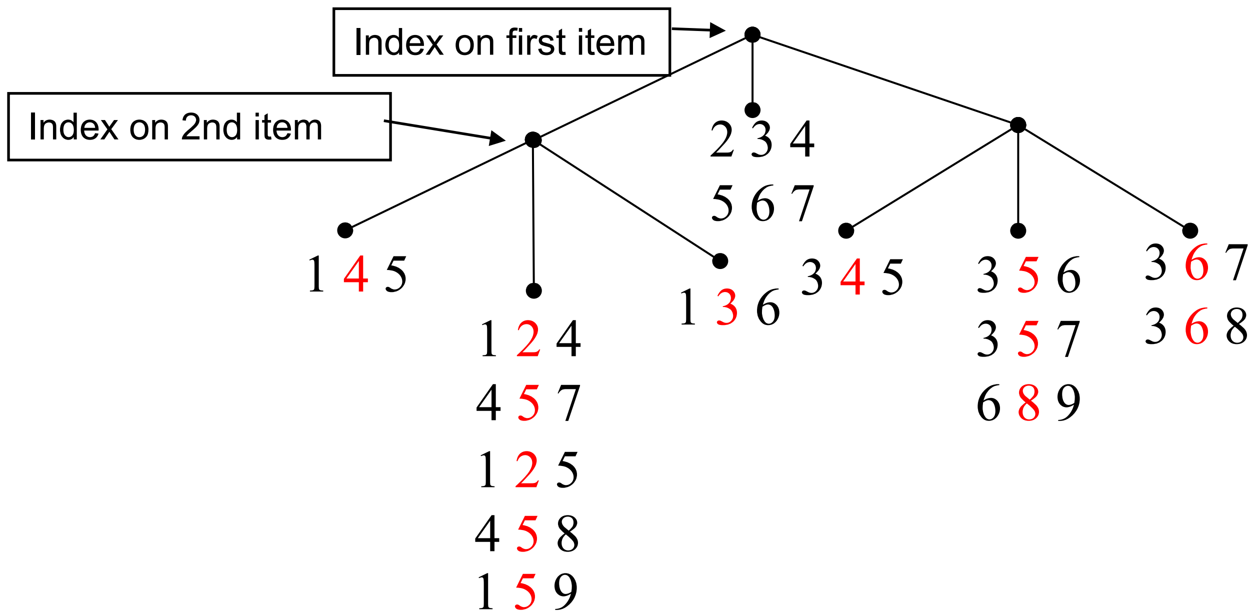


Generate Hash Tree

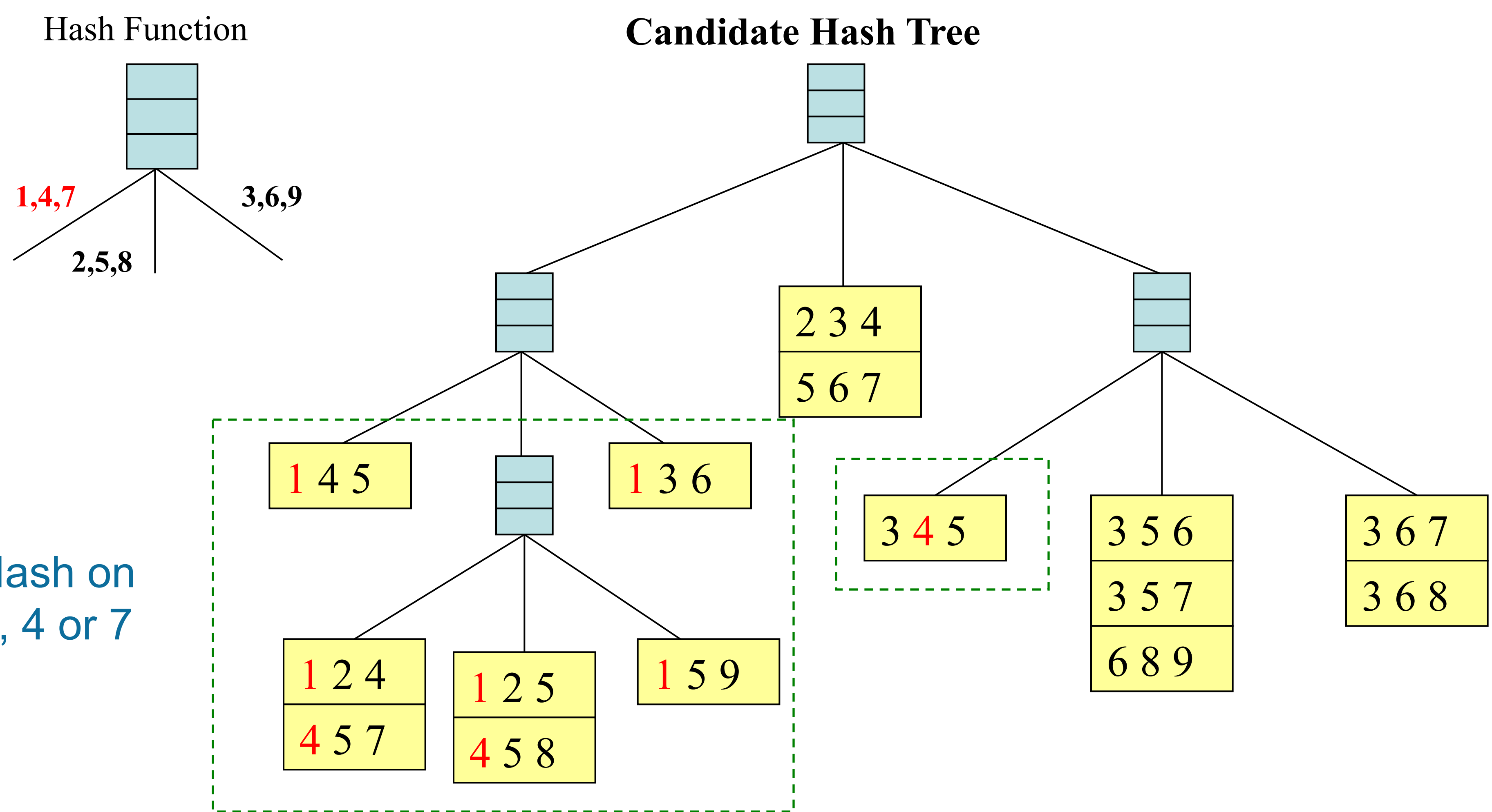
Suppose you have 15 candidate itemsets of length 3:

{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}



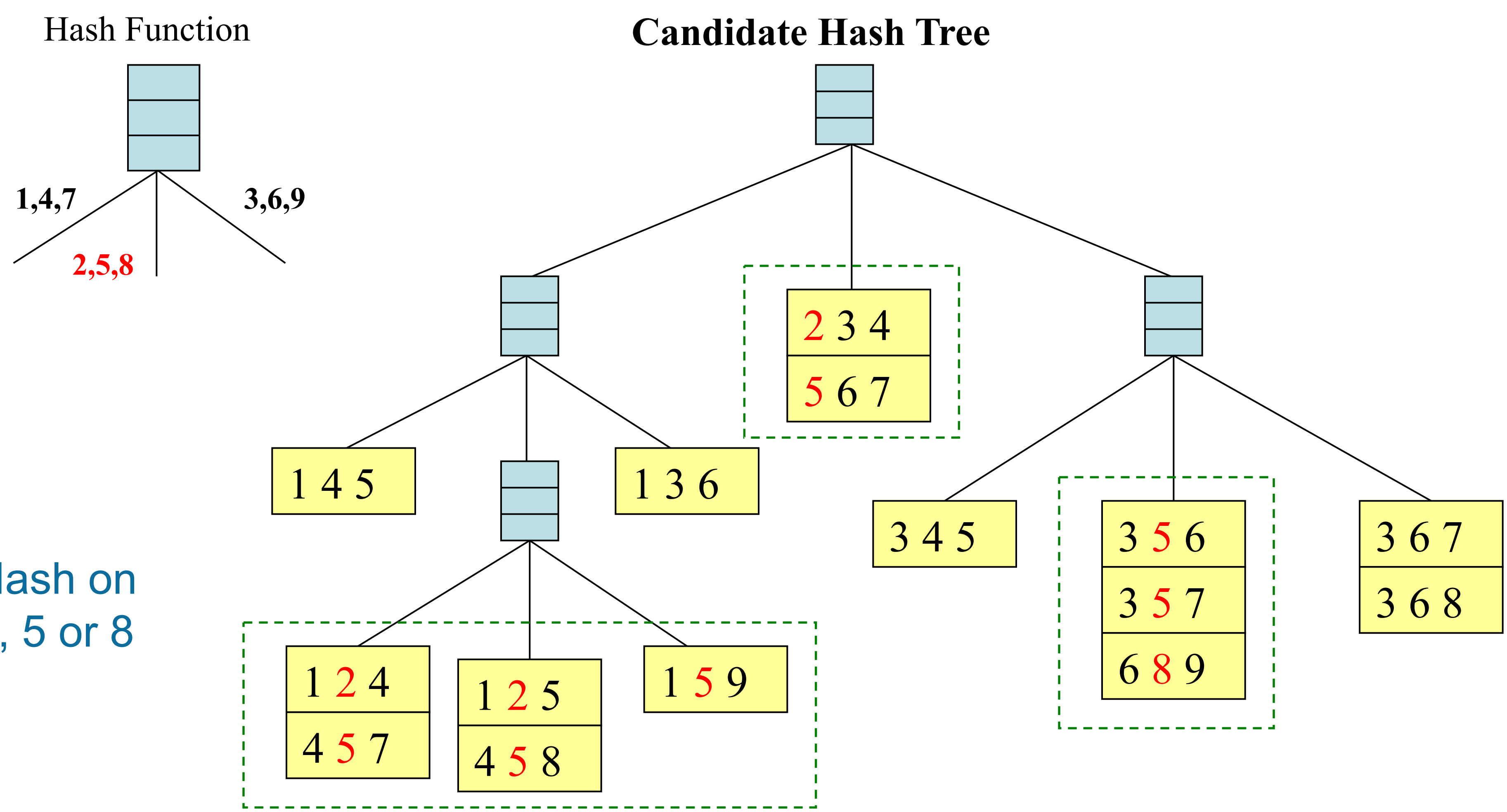


Association Rule Discovery: Hash tree



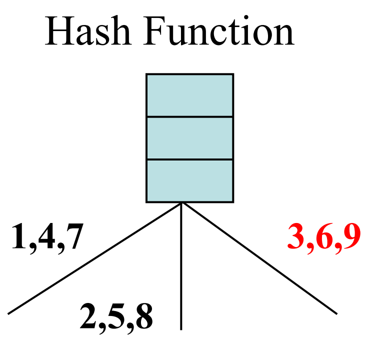
Hash on 1, 4 or 7

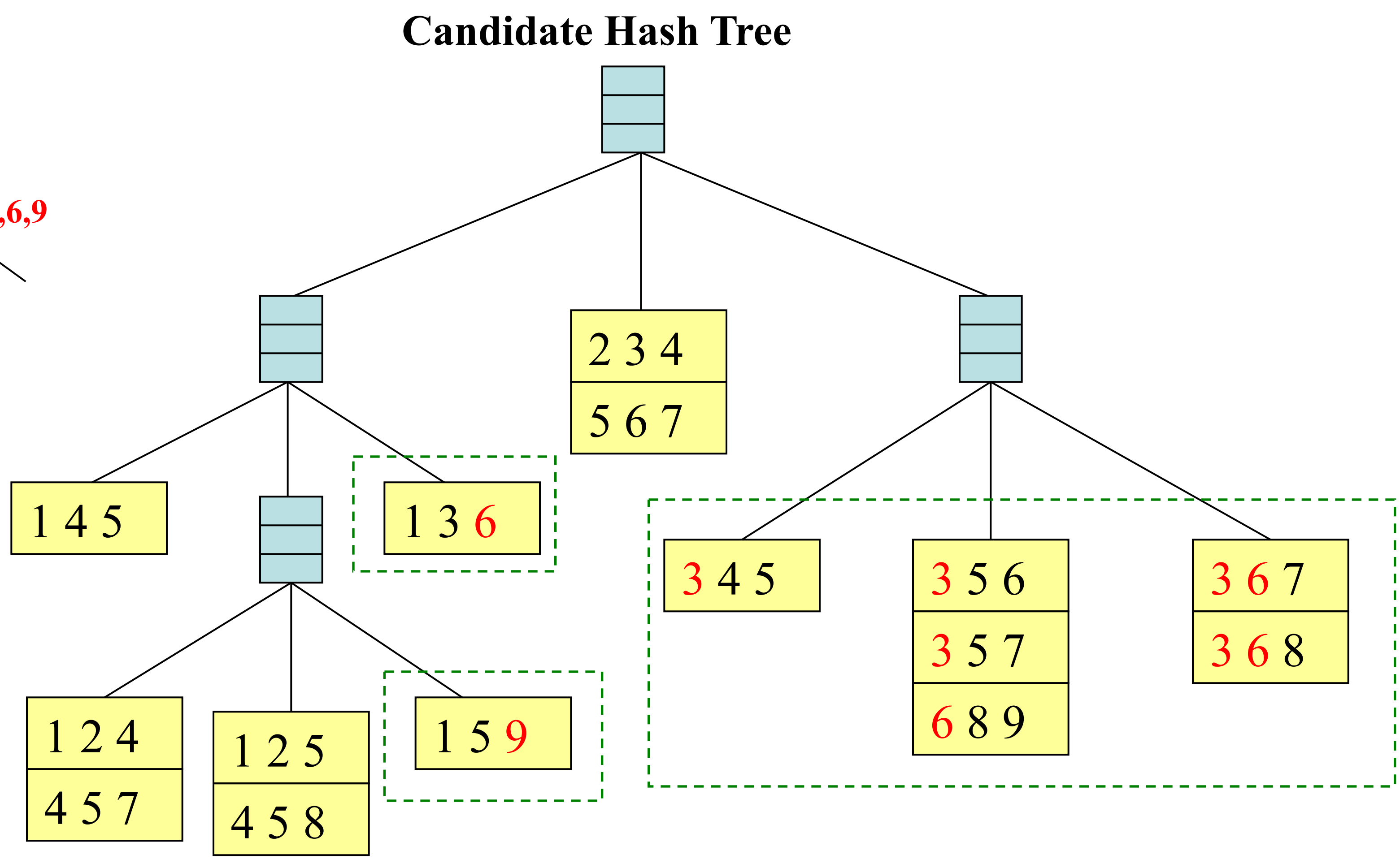
Association Rule Discovery: Hash tree



Hash on
2, 5 or 8

Association Rule Discovery: Hash tree

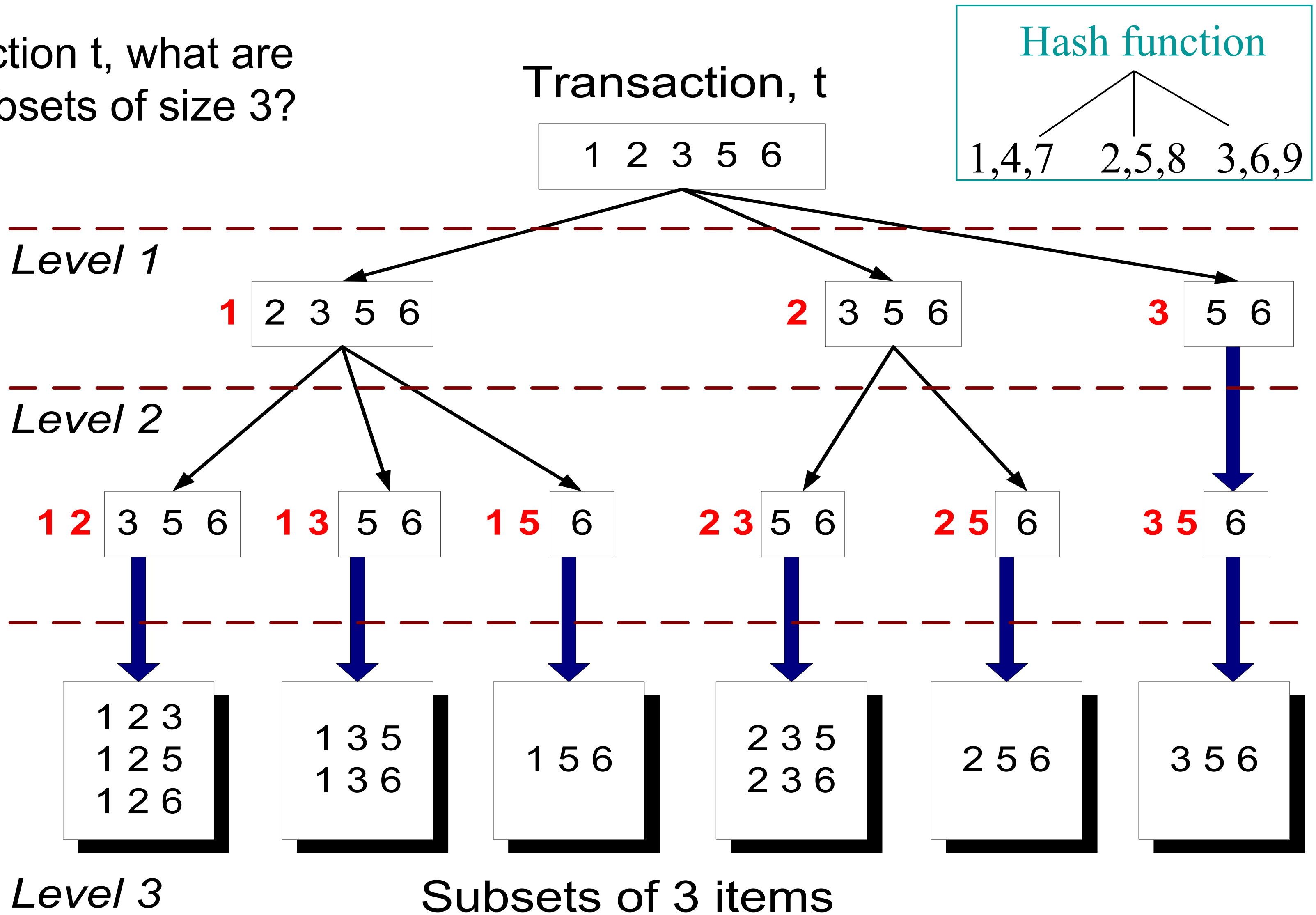




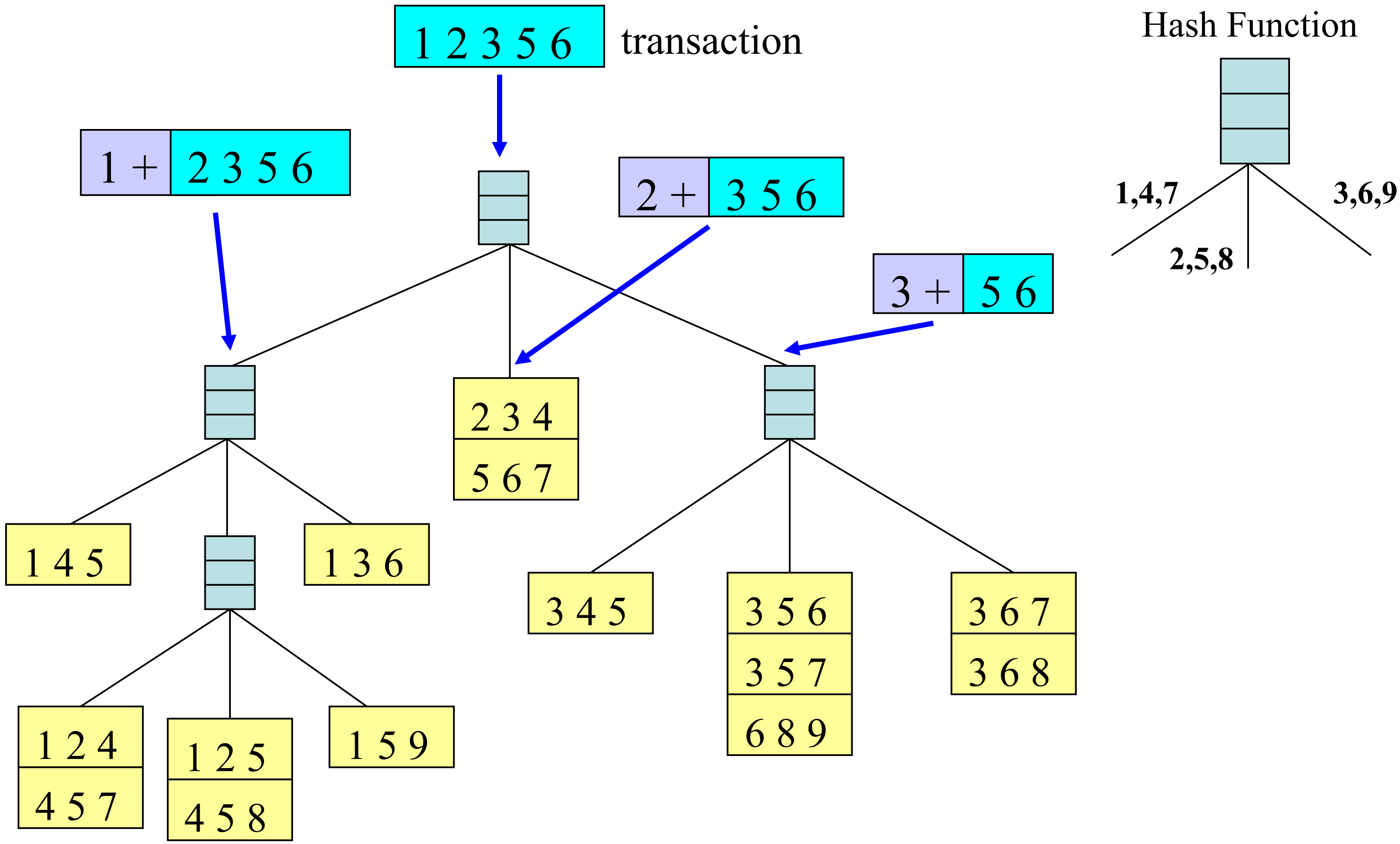
Hash on
3, 6 or 9

Subset Operation

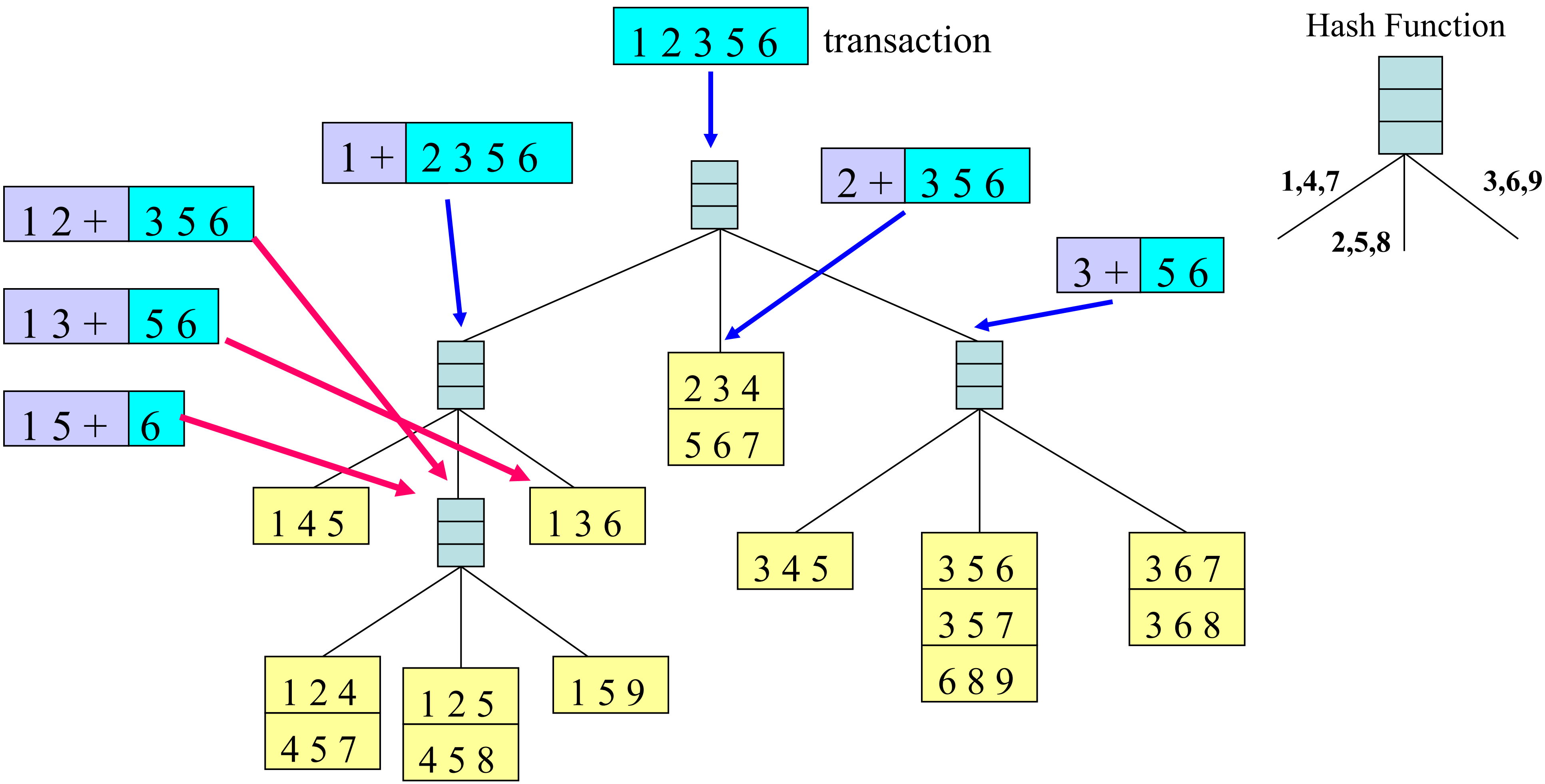
Given a transaction t, what are the possible subsets of size 3?



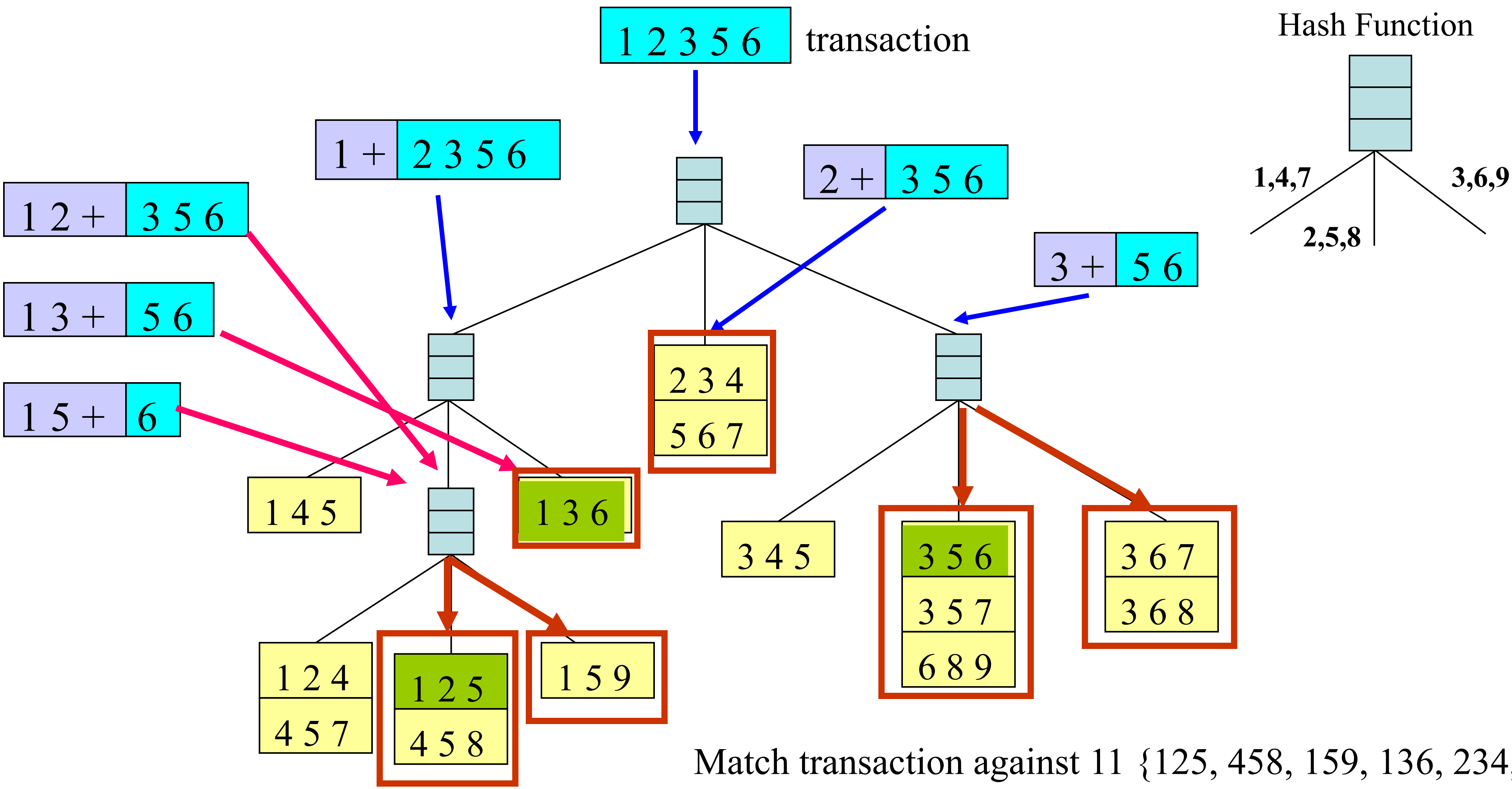
Subset Operation Using Hash Tree



Subset Operation Using Hash Tree



Subset Operation Using Hash Tree



Match transaction against 11 {125, 458, 159, 136, 234, 567, 356, 357, 689, 367, 368} out of 15 candidates.

Step 2: Rule Generation

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement

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

    $A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$

    $AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BC \rightarrow AD$,
    $BD \rightarrow AC$, $CD \rightarrow AB$,

    $ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$,

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

Step 2: Rule Generation

How to efficiently generate rules from frequent itemsets?

- In general, confidence does not have an anti-monotone property

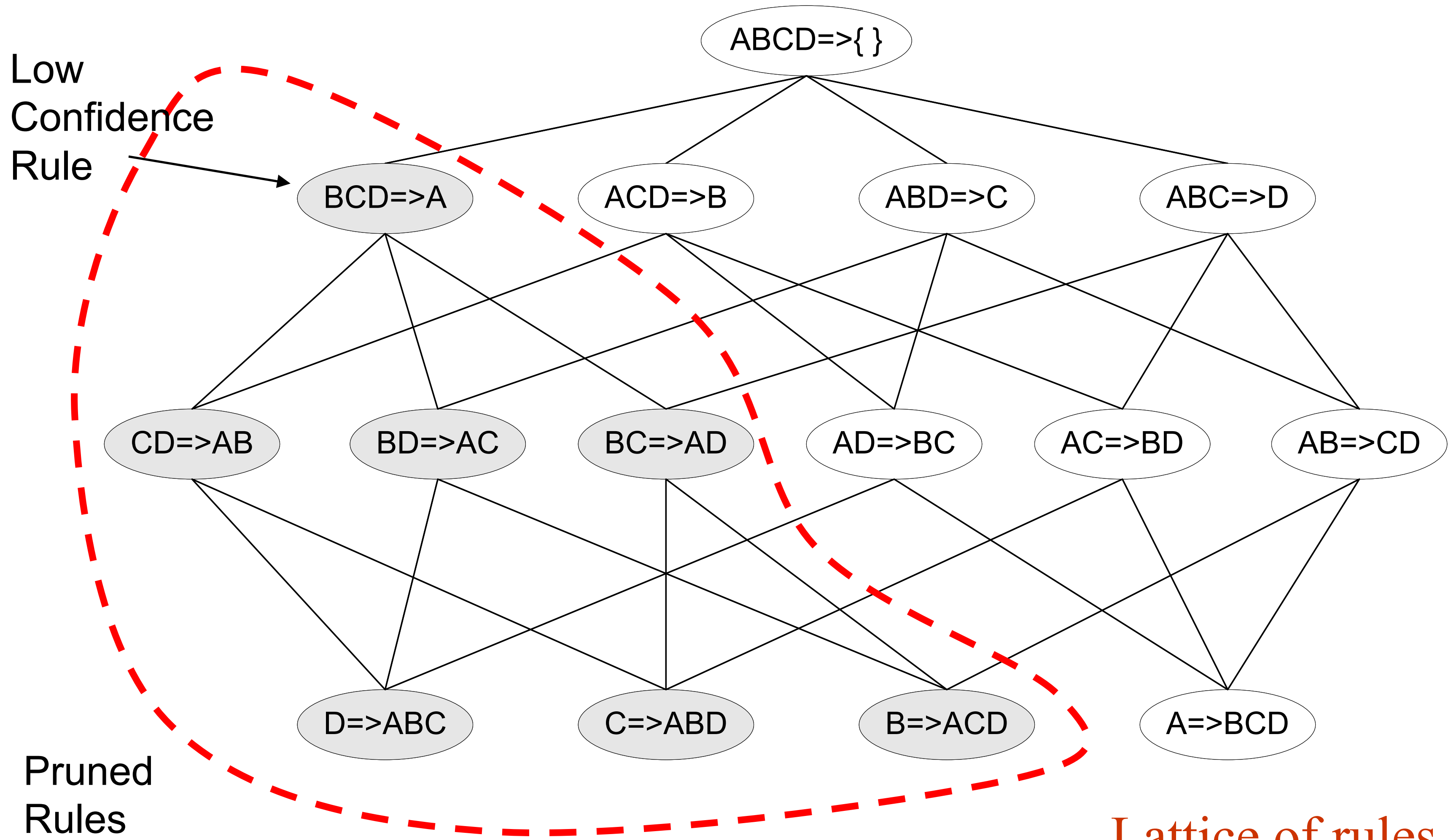
  $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$

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

  e.g., $L = \{A,B,C,D\}$:

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

Rule Generation for Apriori Algorithm



Lattice of rules

Limitations of Association Rules

- Only applicable to nominal attributes
- Comprehensibility of association rules
- Rule redundancy
- Need for secondary evaluation of genuine interestingness of the rule
- Are the association rules what we want?

Interpreting Association rules

Useful?

"*Customers who purchase maintenance agreements also purchase large appliances*"

Rules can be classified as

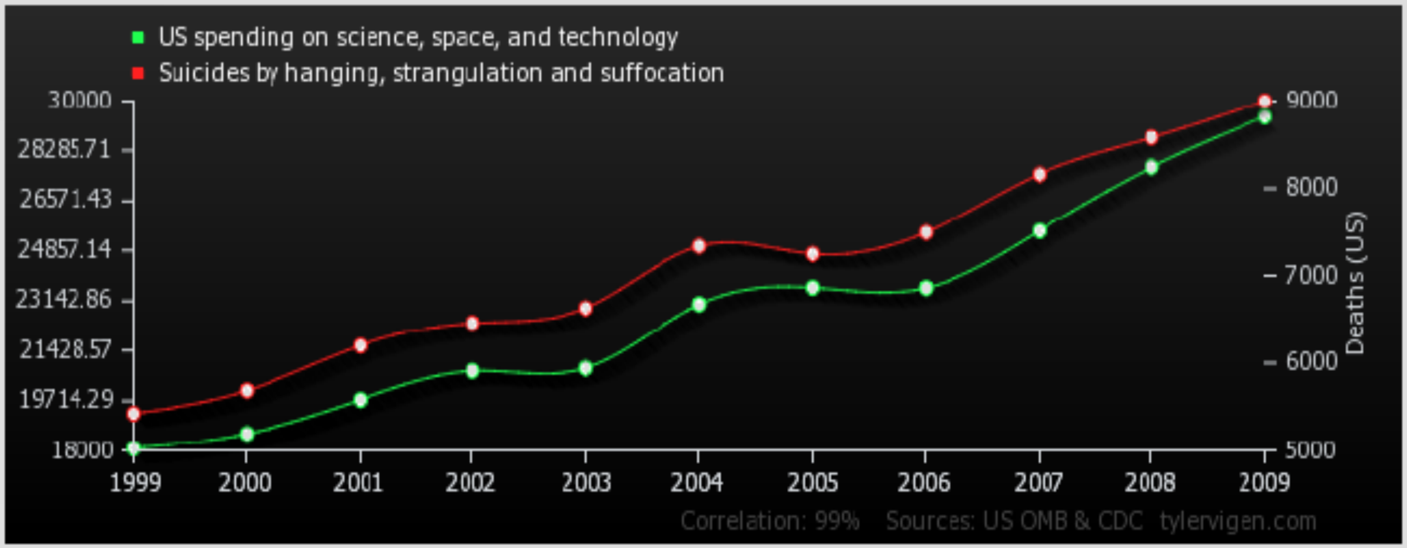
- useful: high quality, actionable information
- trivial: already known to anyone familiar with the context (business)
- inexplicable: this which have no apparent explanation

Correlation is not Causation

US spending on science, space, and technology

correlates with

Suicides by hanging, strangulation and suffocation



	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
US spending on science, space, and technology *Millions of todays dollars (US OMB)*	18,079	18,594	19,753	20,734	20,831	23,029	23,597	23,584	25,525	27,731	29,449
Suicides by hanging, strangulation and suffocation *Deaths (US) (CDC)*	5,427	5,688	6,198	6,462	6,635	7,336	7,248	7,491	8,161	8,578	9,000

Correlation: 0.992082

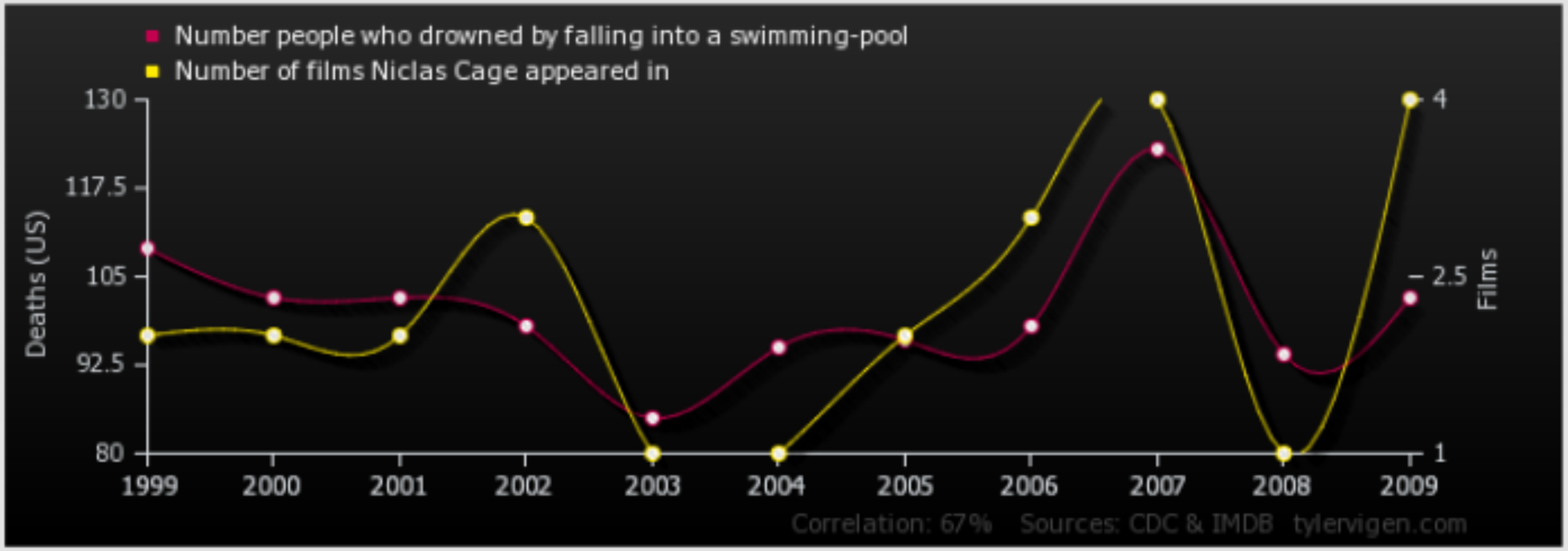
http://www.tylervigen.com/

Correlation is not Causation

Number people who drowned by falling into a swimming-pool

correlates with

Number of films Nicolas Cage appeared in



	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Number people who drowned by falling into a swimming-pool Deaths (US) (CDC)	109	102	102	98	85	95	96	98	123	94	102
Number of films Nicolas Cage appeared in Films (IMDB)	2	2	2	3	1	1	2	3	4	1	4

Correlation: 0.666004

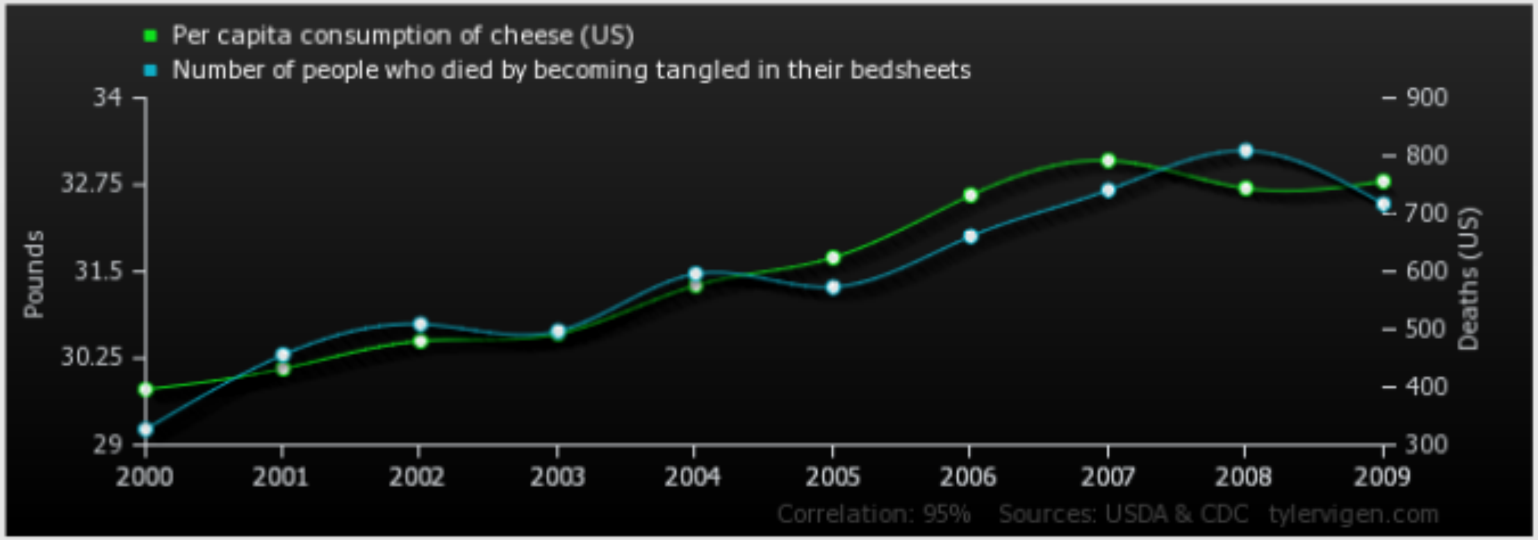
http://www.tylervigen.com/

Correlation is not Causation

Per capita consumption of cheese (US)

correlates with

Number of people who died by becoming tangled in their bedsheets



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Per capita consumption of cheese (US) *Pounds (USDA)*	29.8	30.1	30.5	30.6	31.3	31.7	32.6	33.1	32.7	32.8
Number of people who died by becoming tangled in their bedsheets *Deaths (US) (CDC)*	327	456	509	497	596	573	661	741	809	717

Correlation: 0.947091

http://www.tylervigen.com/

Summary

- What are association rules and how do we evaluate them?
- Discuss the relationship between support and confidence in association rule mining
- Detail the Apriori algorithm for mining association rule

Reference:

http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf