Chapter 5

Association Analysis: Basic Concepts

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

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

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

Implication means co-occurrence, not causality!

Find groups of items which are frequently purchased together

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 $X \rightarrow Y$, where X and Y are itemsets
  - Example:
    $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

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

- **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

Example:

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

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

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

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

  ⇒ Computationally prohibitive!

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:
  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

Basic Apriori Algorithm

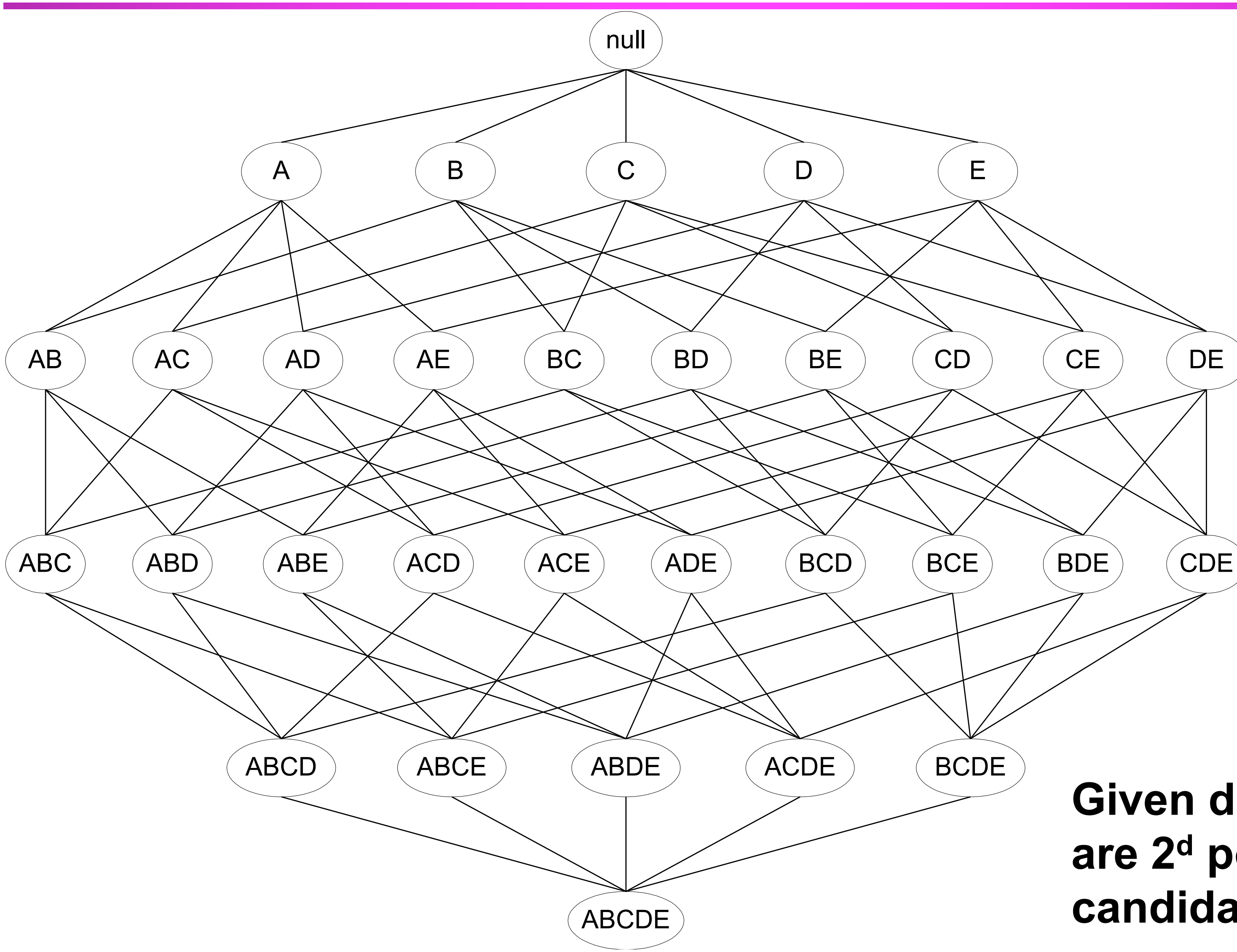
Problem Decomposition

① Find the *frequent itemsets*: the sets of items that satisfy the support constraint

- A subset of a frequent itemset is also a frequent itemset, i.e., if {*A*,*B*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
- Iteratively find frequent itemsets with cardinality from 1 to *k (k*-itemset*)*

② Use the frequent itemsets to generate association rules.

Frequent Itemset Generation



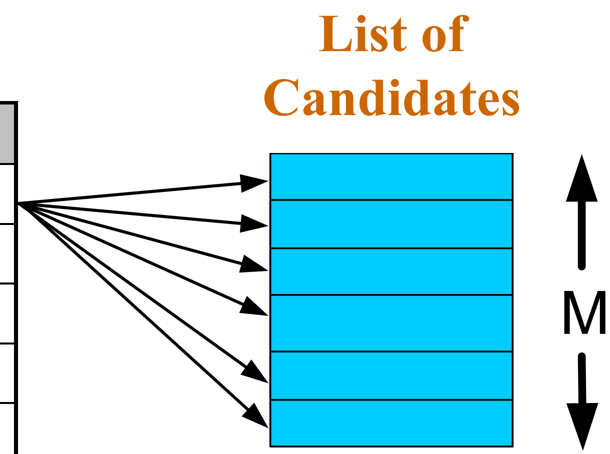
Given d items, there are 2^d possible candidate itemsets

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

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



  – Match each transaction against every candidate
  – Complexity ~ O(NMw) => 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
- 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

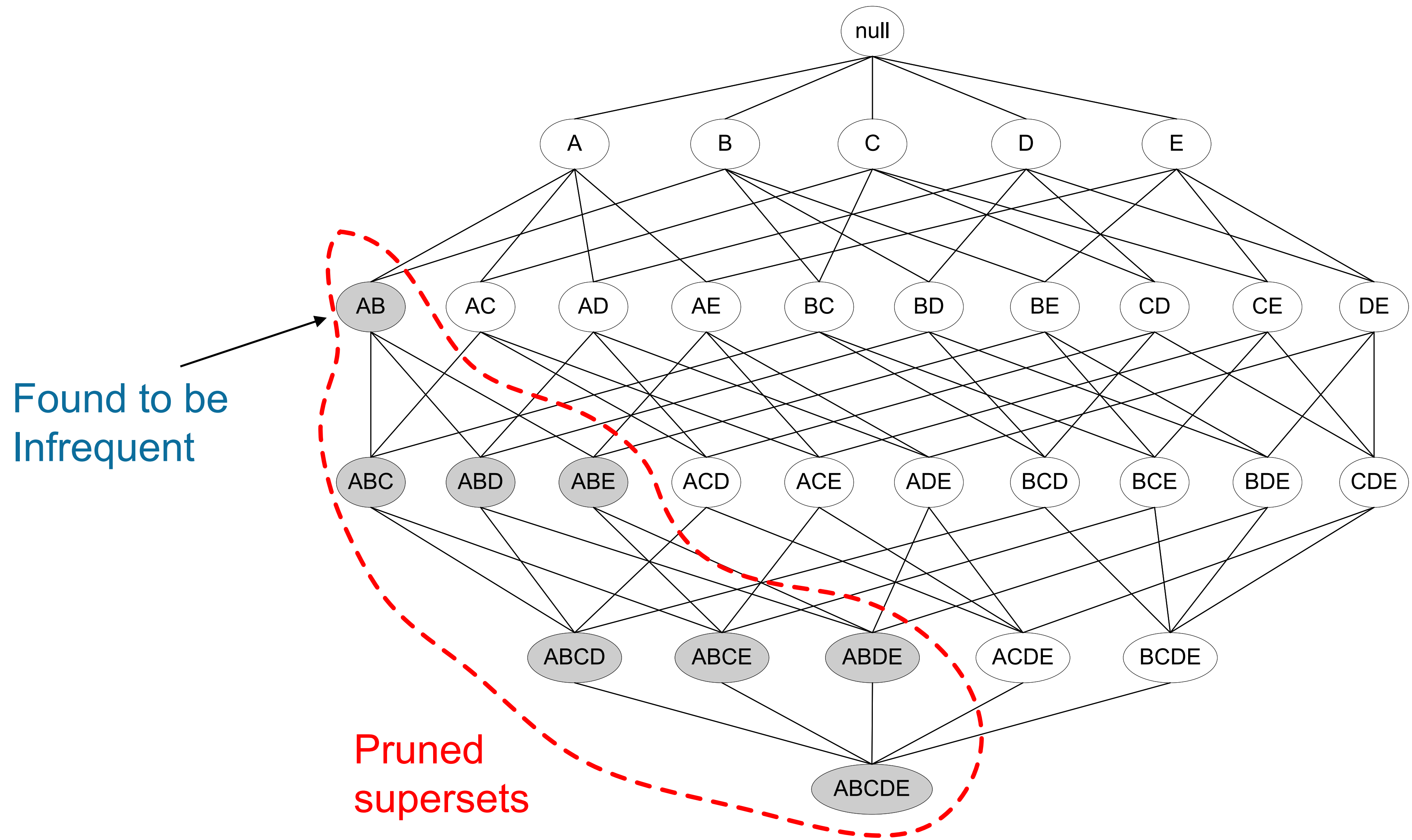
- Apriori principle:
  – If an itemset is frequent, then all of its subsets must also be frequent

- 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

TID	*Items*
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

Illustrating Apriori Principle

TID	*Items*
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

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

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

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
{ 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**

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

Apriori Algorithm

- F_k: frequent k-itemsets
- L_k: candidate k-itemsets

- Algorithm
  - Let k=1
  - Generate F_1 = {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 => F_{k+1}

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

- Merge two frequent (k-1)-itemsets if their first (k-2) items are identical

- F_3 = {ABC,ABD,ABE,ACD,BCD,BDE,CDE}
  - Merge(**AB**C, **AB**D) = **AB**CD
  - Merge(**AB**C, **AB**E) = **AB**CE
  - Merge(**AB**D, **AB**E) = **AB**DE
  - Do not merge(**A**BD,**A**CD) 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}

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)

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

Triplets (3-itemsets)

Itemset	Count
{Bread, Diaper, Milk}	**2**

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

Alternate F_{k-1} x 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(A**BC**, **BC**D) = A**BC**D
  - Merge(A**BD**, **BD**E) = A**BD**E
  - Merge(A**CD**, **CD**E) = A**CD**E
  - Merge(B**CD**, **CD**E) = B**CD**E

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}

Support Counting of Candidate Itemsets

- Scan the database of transactions to determine the support of each candidate itemset
  - Must match every candidate itemset against every transaction, which is an expensive operation

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

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

Support Counting of Candidate Itemsets

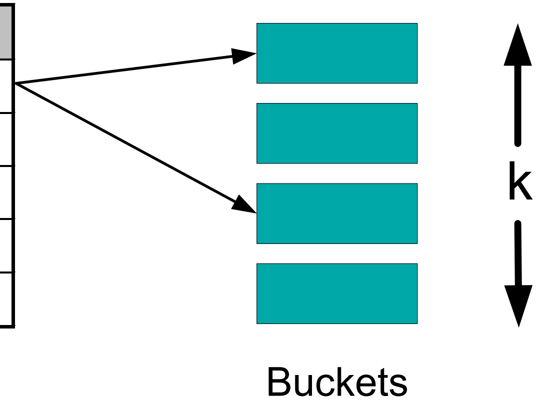
- To reduce number of comparisons, store the candidate itemsets in a hash structure
  – Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets

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



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:

$ABC \rightarrow D$,	$ABD \rightarrow C$,	$ACD \rightarrow B$,	$BCD \rightarrow A$,
$A \rightarrow BCD$,	$B \rightarrow ACD$,	$C \rightarrow ABD$,	$D \rightarrow ABC$
$AB \rightarrow CD$,	$AC \rightarrow BD$,	$AD \rightarrow BC$,	$BC \rightarrow AD$,
$BD \rightarrow AC$,	$CD \rightarrow AB$,		

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

Rule Generation

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

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

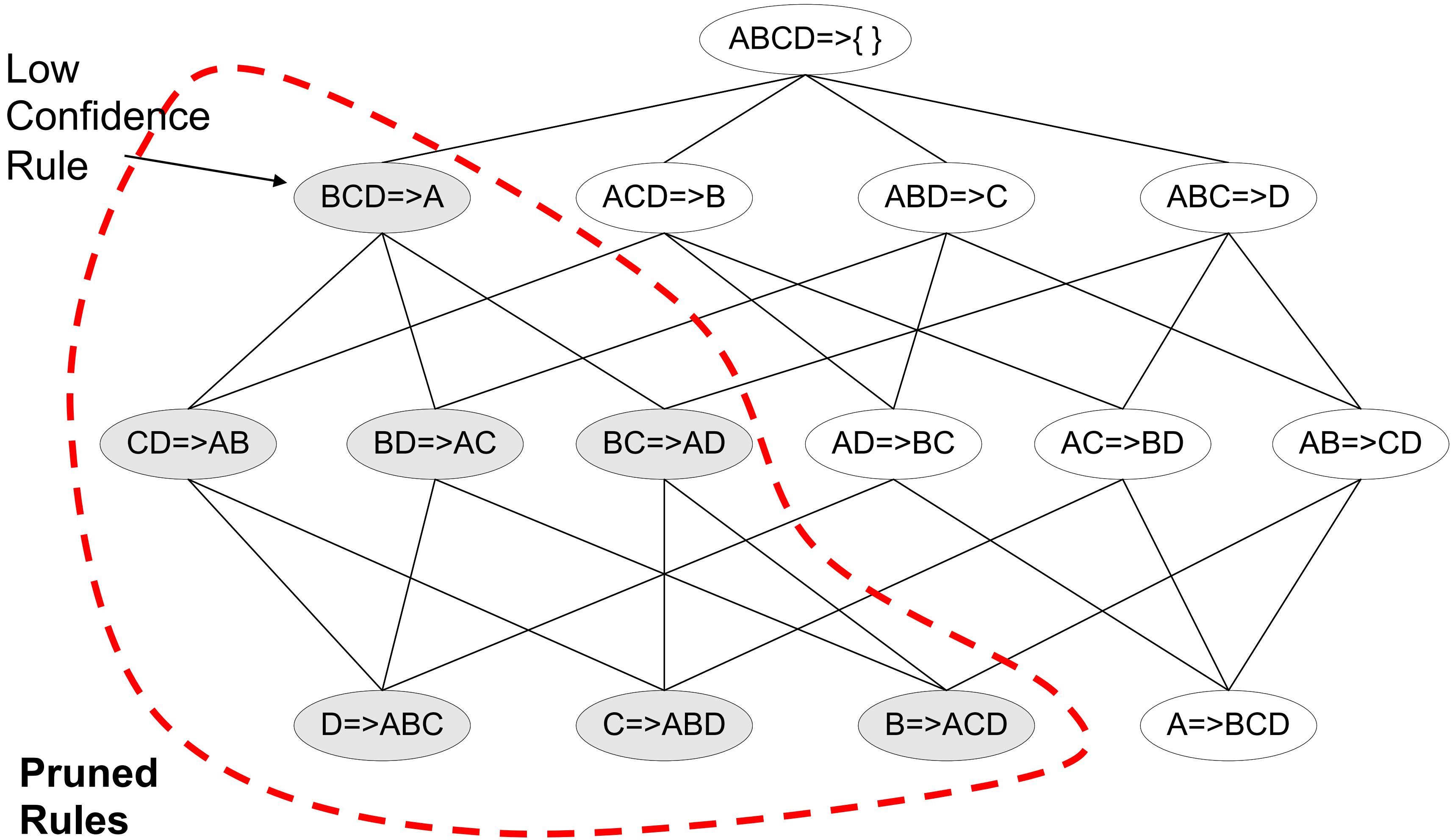
- But confidence of rules generated from the same itemset has an anti-monotone property
  - E.g., Suppose {A,B,C,D} is a frequent 4-itemset:

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

  - Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

Rule Generation for Apriori Algorithm

Lattice of rules

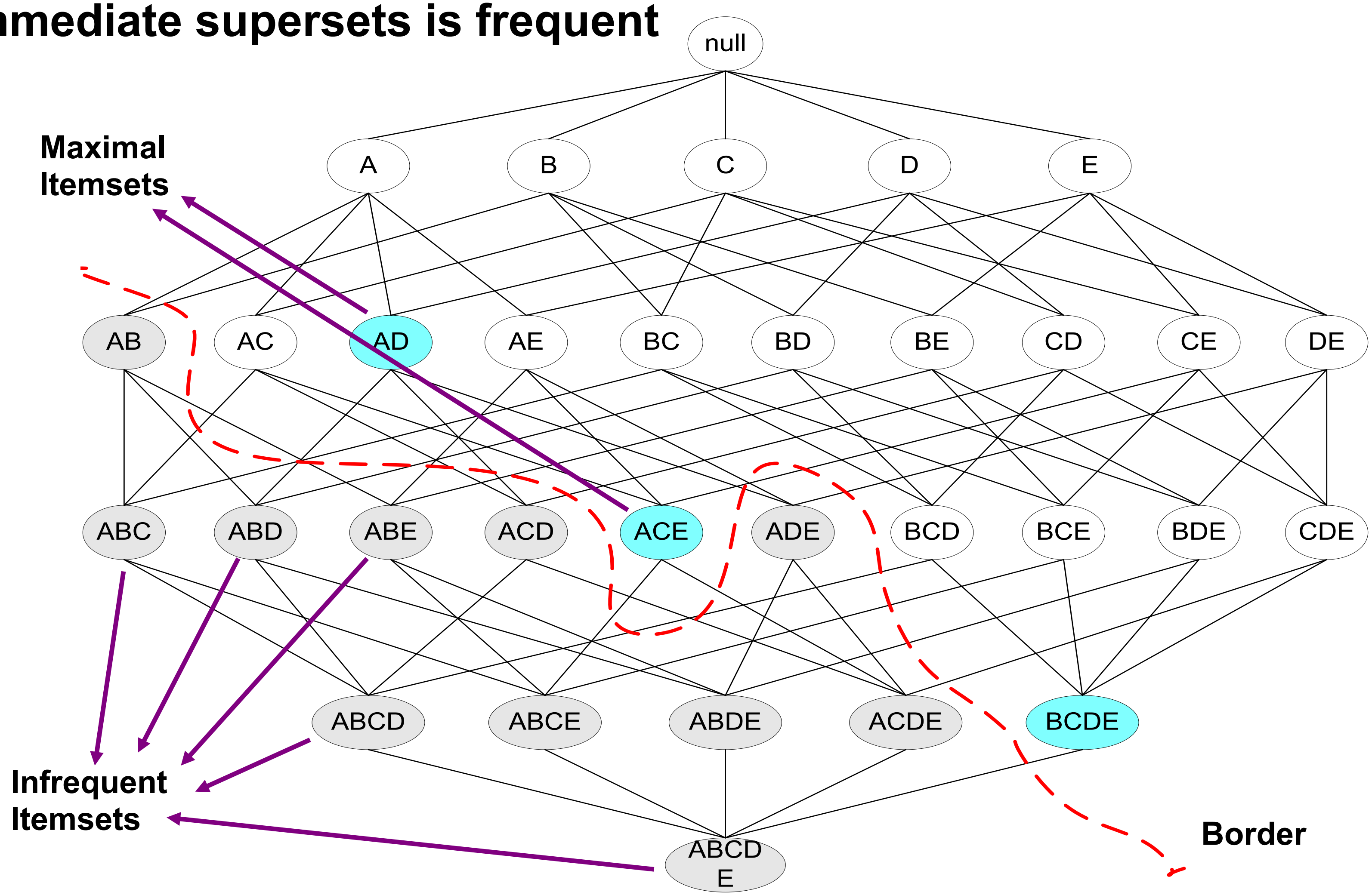


Factors Affecting Complexity of Apriori

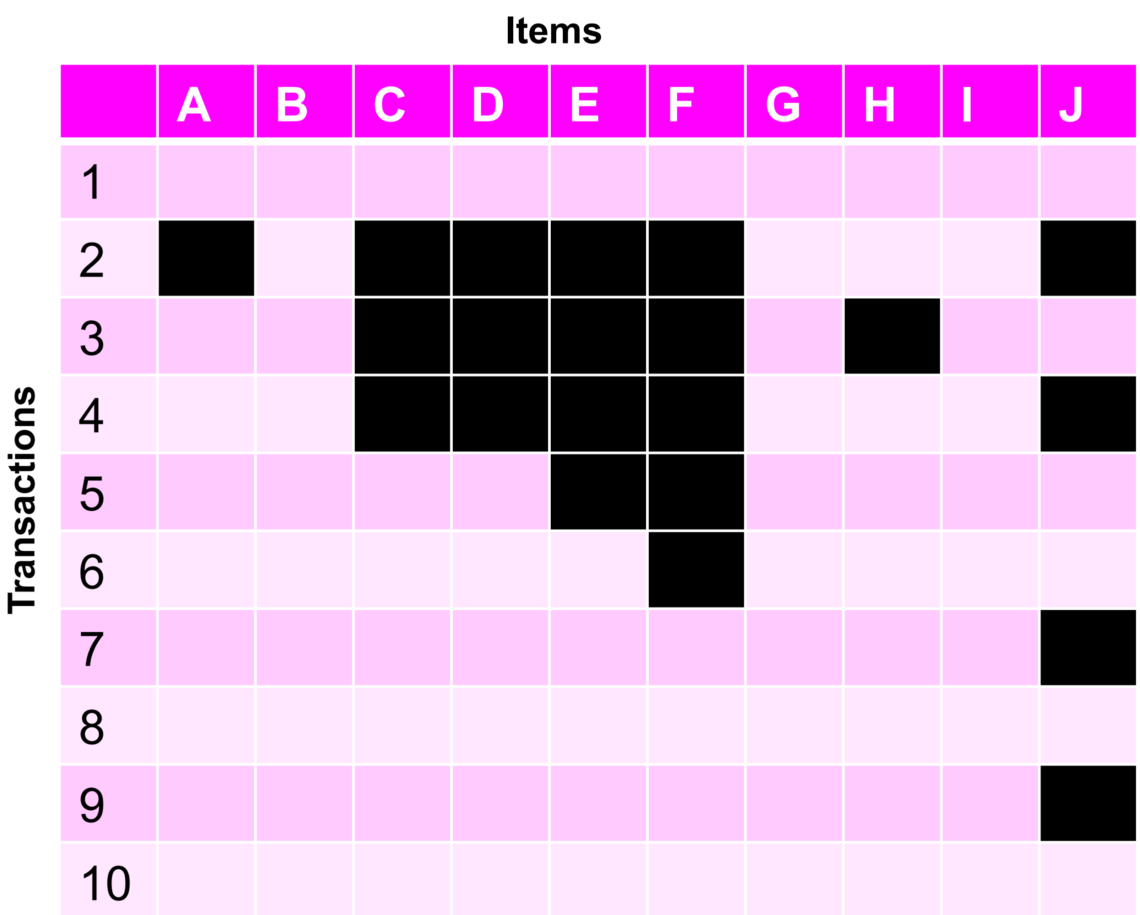
- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
  - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - More space is needed to store support count of itemsets
  - if number of frequent itemsets also increases, both computation and I/O costs may also increase
- Size of database
  - run time of algorithm increases with number of transactions
- Average transaction width
  - transaction width increases the max length of frequent itemsets
  - number of subsets in a transaction increases with its width, increasing computation time for support counting

Maximal Frequent Itemset

An itemset is maximal frequent if it is frequent and none of its immediate supersets is frequent



An illustrative example

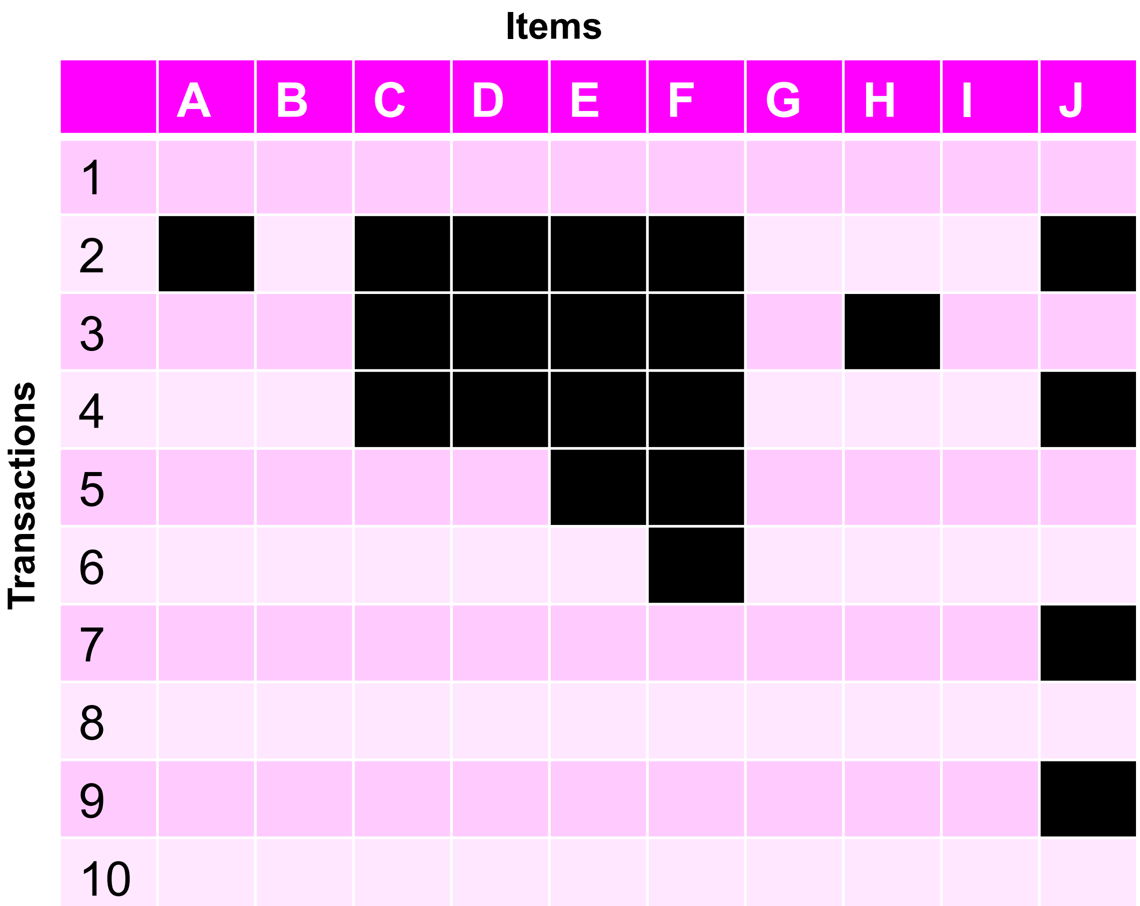


Support threshold (by count) : 5

Frequent itemsets: ?

Maximal itemsets: ?

An illustrative example



Support threshold (by count) : 5

Frequent itemsets: {F}

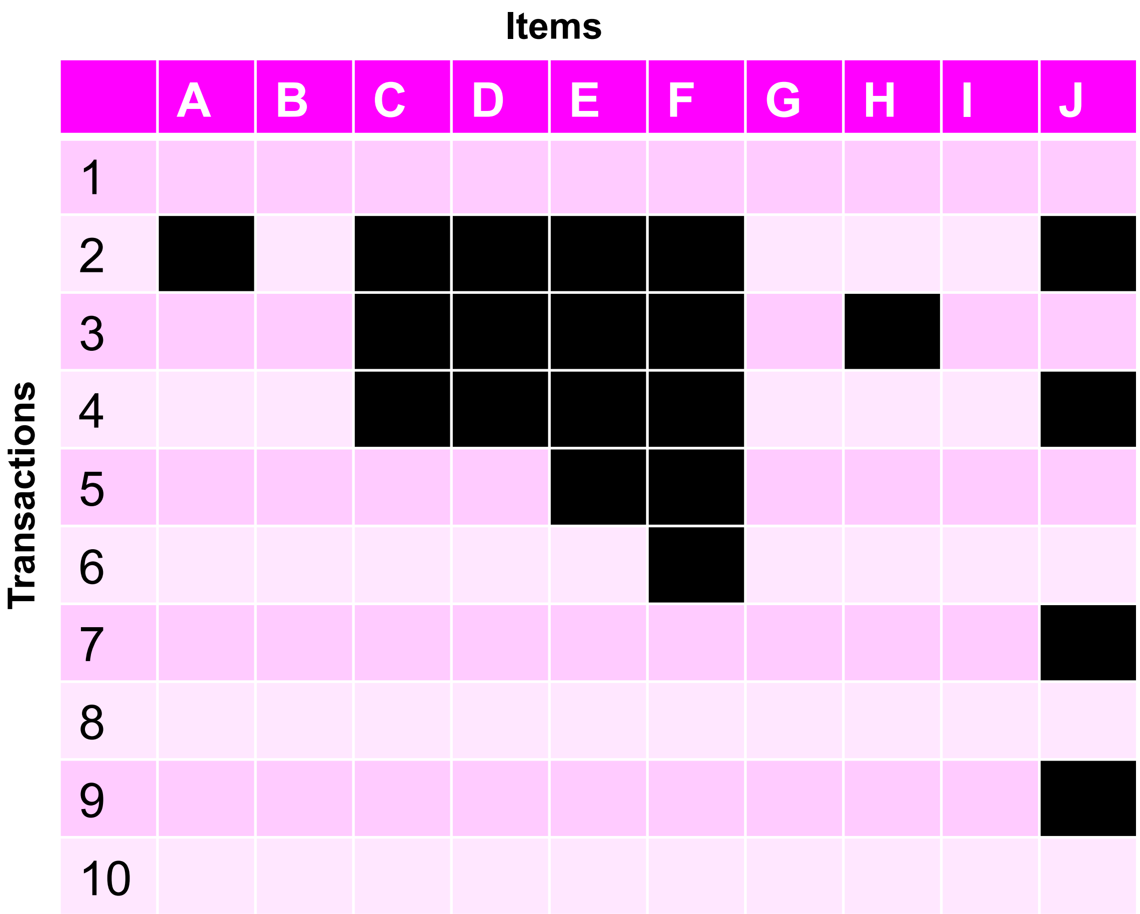
Maximal itemsets: {F}

Support threshold (by count): 4

Frequent itemsets: ?

Maximal itemsets: ?

An illustrative example



Support threshold (by count) : 5

Frequent itemsets: {F}

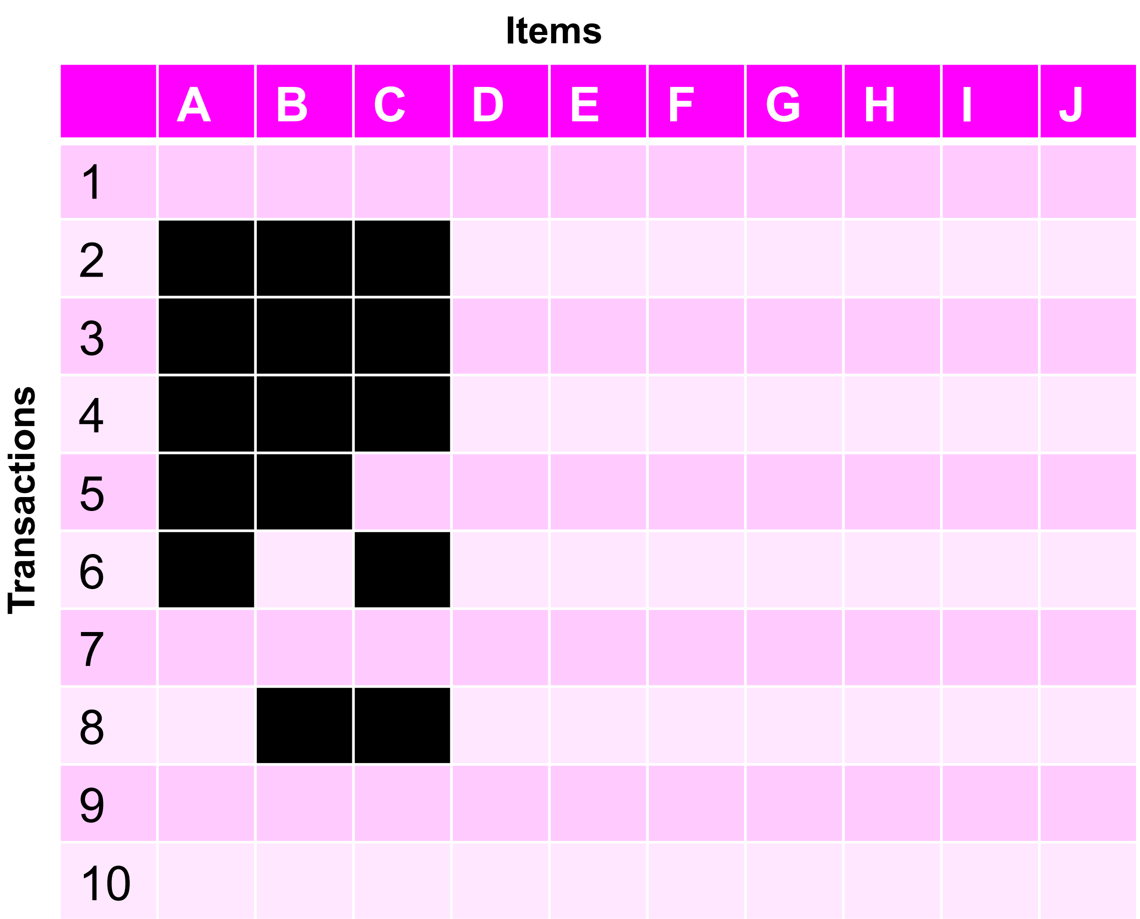
Maximal itemsets: {F}

Support threshold (by count): 4

Frequent itemsets: {E}, {F}, {E,F}, {J}

Maximal itemsets: {E,F}, {J}

Another illustrative example



Support threshold (by count) : 5
Maximal itemsets: {A}, {B}, {C}

Support threshold (by count): 4
Maximal itemsets: {A,B}, {A,C},{B,C}

Support threshold (by count): 3
Maximal itemsets: {A,B,C}

Closed Itemset

- An itemset X is closed if none of its immediate supersets has the same support as the itemset X.
- X is not closed if at least one of its immediate supersets has support count as X.

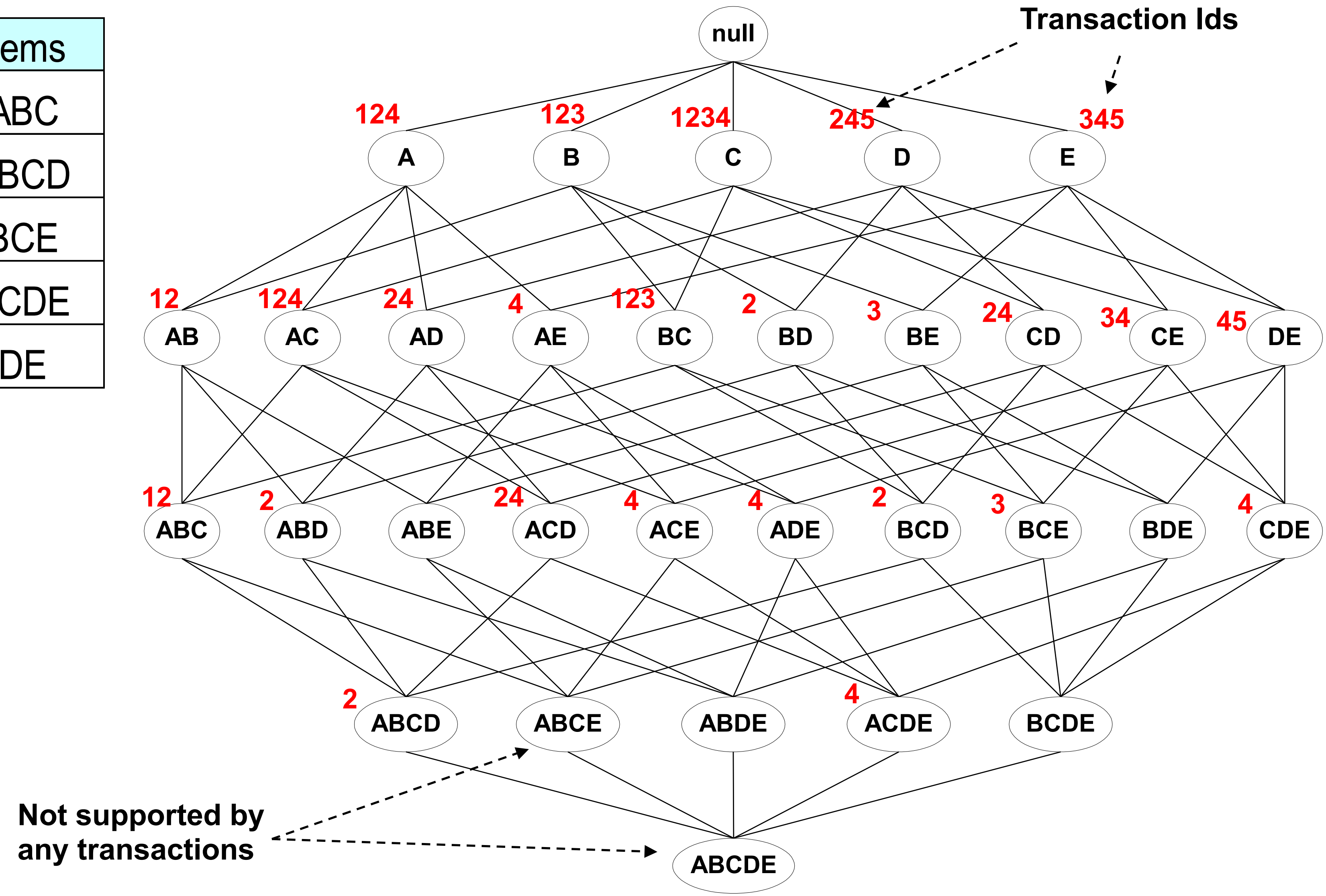
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,B,C,D}
4	{A,B,D}
5	{A,B,C,D}

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3

Itemset	Support
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	2
{A,B,C,D}	2

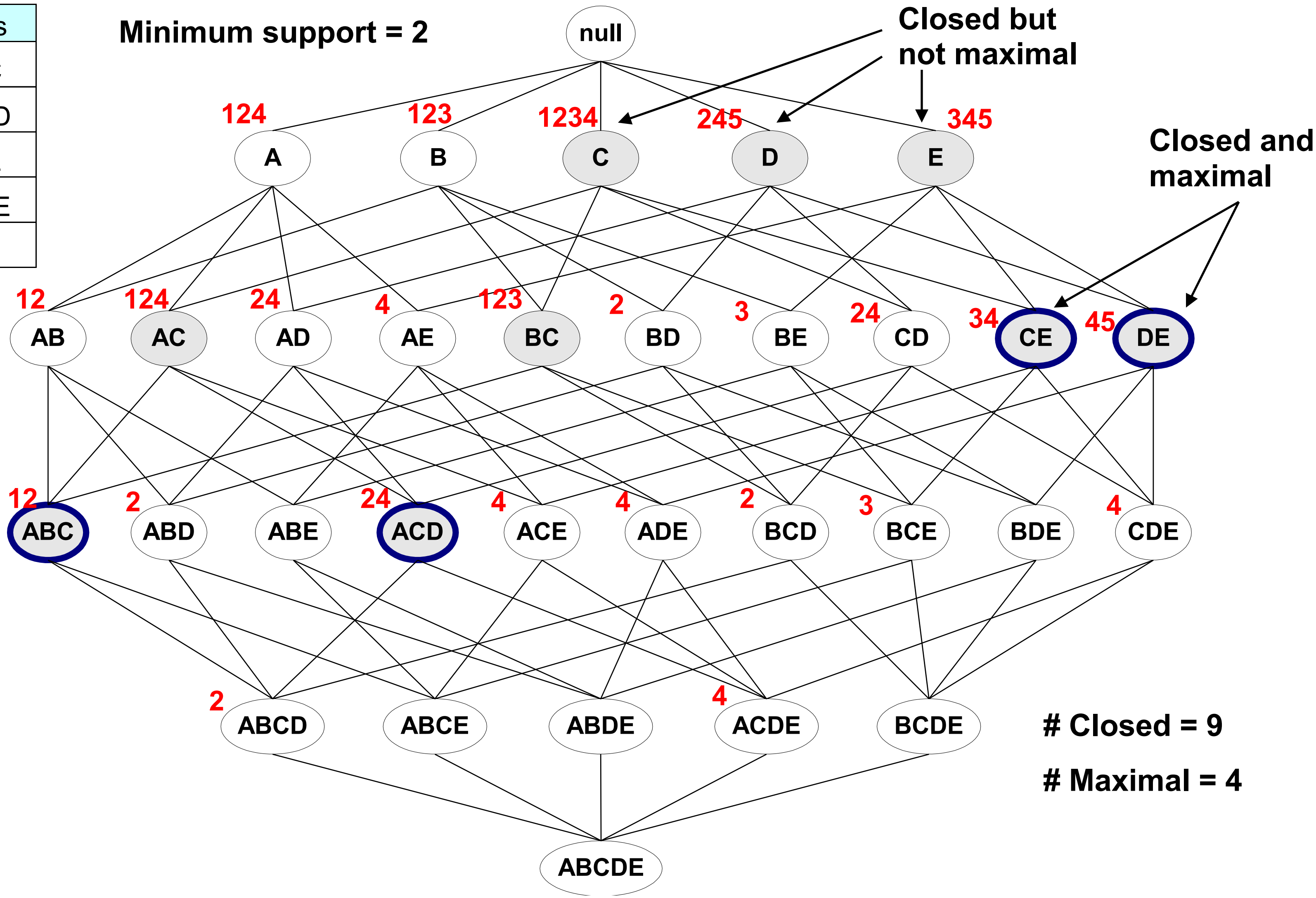
Maximal vs Closed Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

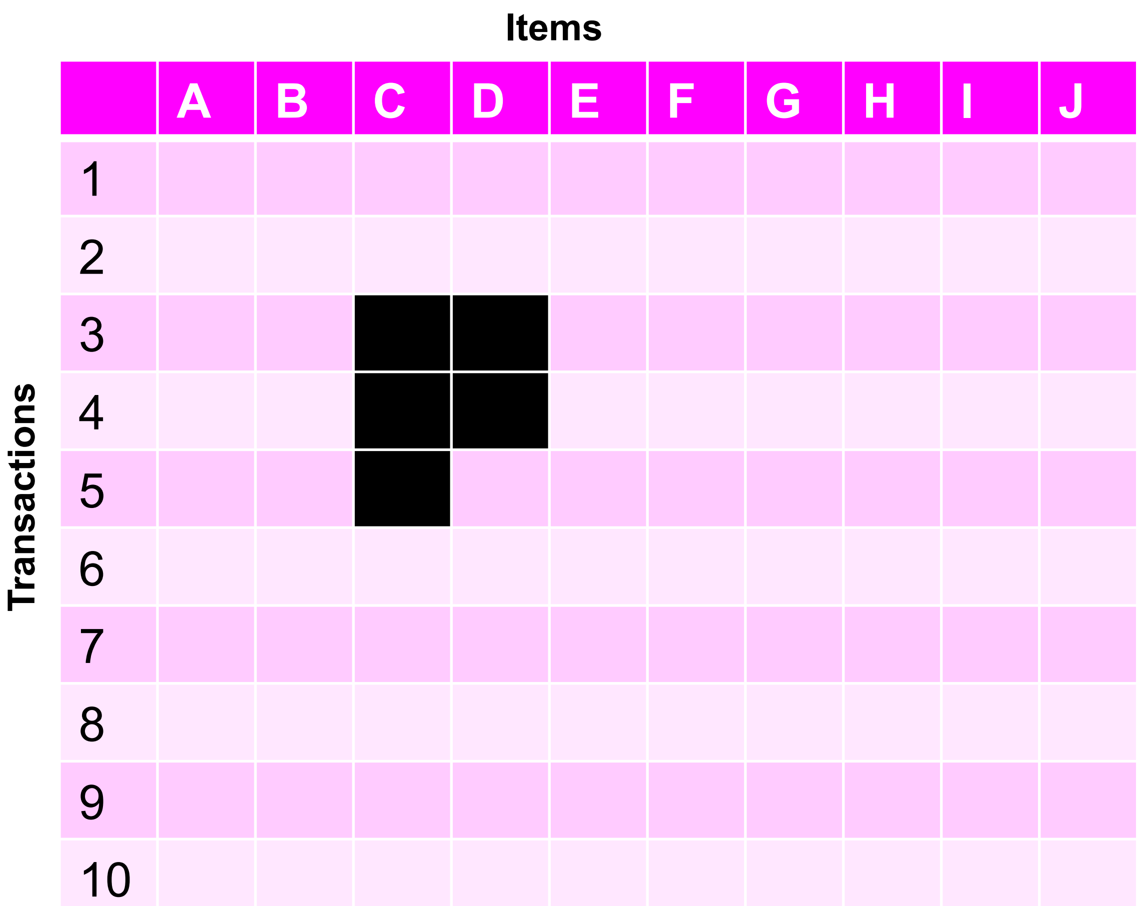


Maximal Frequent vs Closed Frequent Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

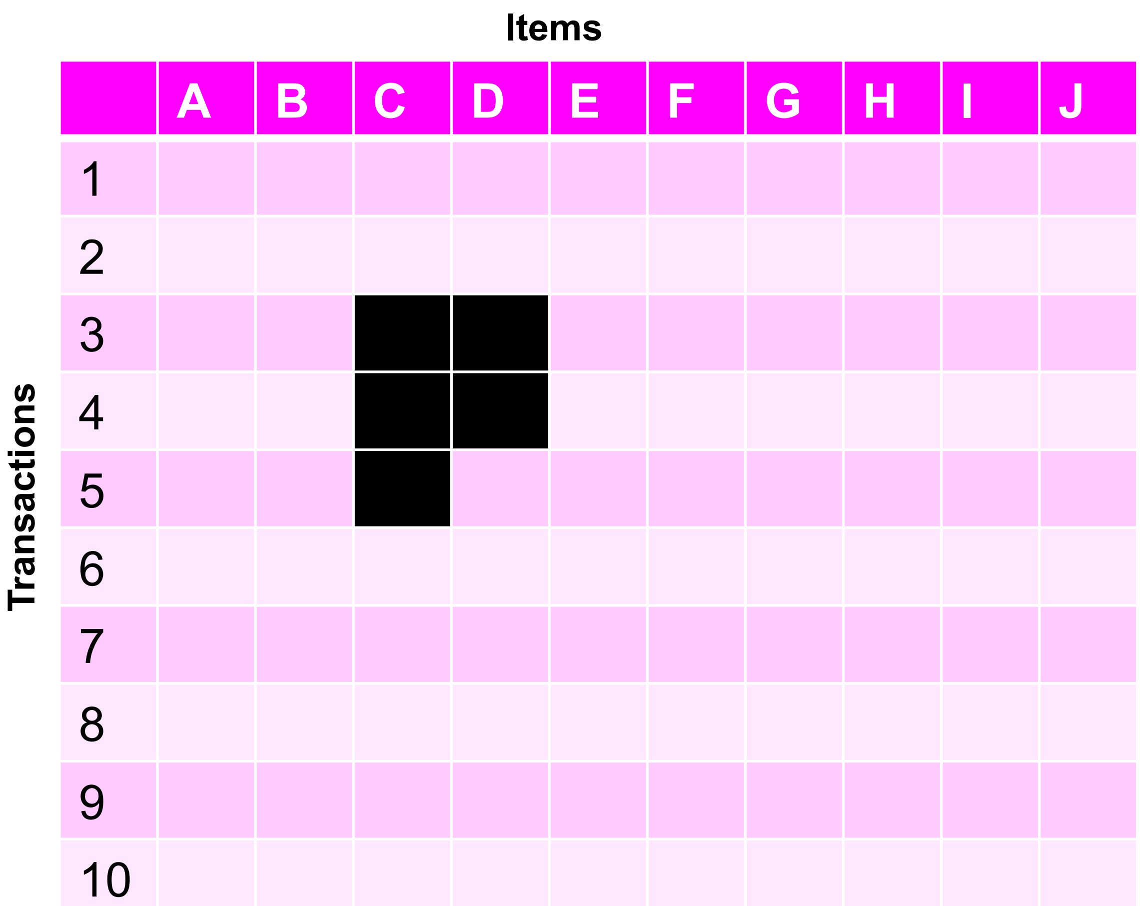


Example 1



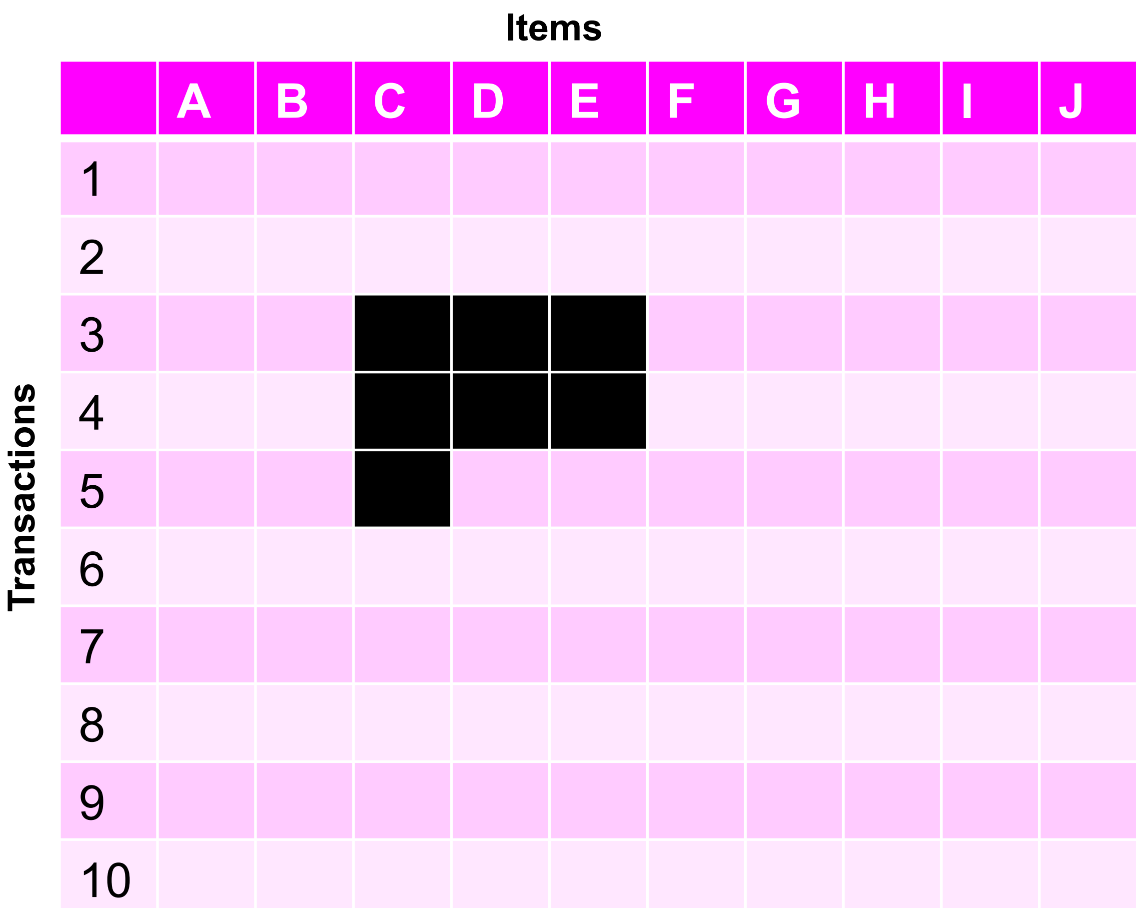
Itemsets	Support (counts)	Closed itemsets
{C}	3	
{D}	2	
{C,D}	2	

Example 1



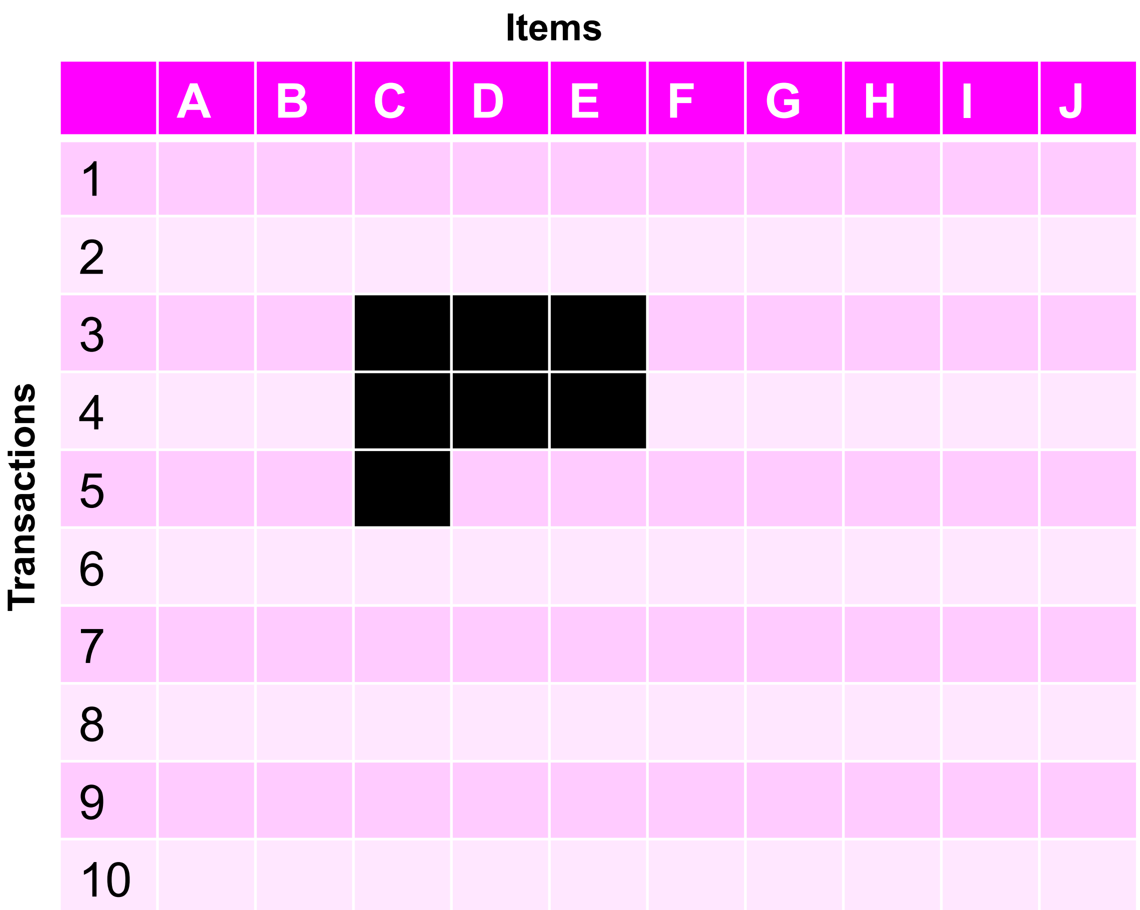
Itemsets	Support (counts)	Closed itemsets
{C}	**3**	✓
{D}	2	
{C,D}	**2**	✓

Example 2



Itemsets	Support (counts)	Closed itemsets
{C}	3	
{D}	2	
{E}	2	
{C,D}	2	
{C,E}	2	
{D,E}	2	
{C,D,E}	2	

Example 2



Itemsets	Support (counts)	Closed itemsets
{C}	**3**	✓
{D}	2	
{E}	2	
{C,D}	2	
{C,E}	2	
{D,E}	2	
{C,D,E}	**2**	✓

Maximal vs Closed Itemsets

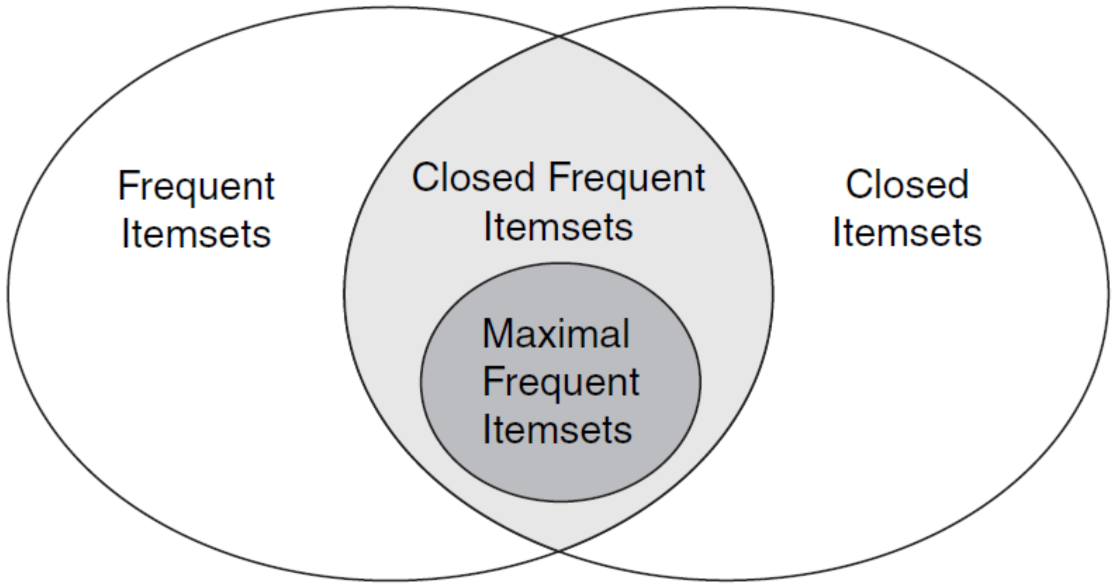


Figure 5.18. Relationships among frequent, closed, closed frequent, and maximal frequent itemsets.

Pattern Evaluation

- Association rule algorithms can produce large number of rules

- Interestingness measures can be used to prune/rank the patterns
  - In the original formulation, support & confidence are the only measures used

Computing Interestingness Measure

- Given $X \rightarrow Y$ or $\{X,Y\}$, information needed to compute interestingness can be obtained from a contingency table

Contingency table

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

f_{11}: support of X and Y
f_{10}: support of X and $\overline{Y}$
f_{01}: support of $\overline{X}$ and Y
f_{00}: support of $\overline{X}$ and $\overline{Y}$

Used to define various measures

- support, confidence, Gini, entropy, etc.

Drawback of Confidence

Customers	Tea	Coffee	…
C1	0	1	…
C2	1	0	…
C3	1	1	…
C4	1	0	…
…			

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

Association Rule: Tea $\rightarrow$ Coffee

Confidence ≅ P(Coffee|Tea) = 150/200 = 0.75

Confidence > 50%, meaning people who drink tea are more likely to drink coffee than not drink coffee

So rule seems reasonable

Drawback of Confidence

	Coffee	$\overline{\text{Coffee}}$	
Tea	150	50	200
$\overline{\text{Tea}}$	650	150	800
	800	200	1000

Association Rule: Tea → Coffee

Confidence= P(Coffee|Tea) = 150/200 = 0.75

but P(Coffee) = 0.8, which means knowing that a person drinks tea reduces the probability that the person drinks coffee!

⇒ Note that P(Coffee|$\overline{\text{Tea}}$) = 650/800 = 0.8125

Drawback of Confidence

Customers	Tea	Honey	…
C1	0	1	…
C2	1	0	…
C3	1	1	…
C4	1	0	…
…			

	$Honey$	$\overline{Honey}$	
Tea	100	100	200
$\overline{Tea}$	20	780	800
	120	880	1000

Association Rule: Tea $\rightarrow$ Honey

Confidence $\cong$ P(Honey|Tea) = 100/200 = 0.50

Confidence = 50%, which may mean that drinking tea has little influence whether honey is used or not

So rule seems uninteresting

But P(Honey) = 120/1000 = .12 (hence tea drinkers are far more likely to have honey

Measure for Association Rules

- So, what kind of rules do we really want?
  - Confidence($X \rightarrow Y$) should be sufficiently high
    - To ensure that people who buy X will more likely buy Y than not buy Y
  - Confidence($X \rightarrow Y$) > support(Y)
    - Otherwise, rule will be misleading because having item X actually reduces the chance of having item Y in the same transaction
    - Is there any measure that capture this constraint?
      - Answer: Yes. There are many of them.

Statistical Relationship between X and Y

- The criterion

  $$\text{confidence}(X \rightarrow Y) = \text{support}(Y)$$

  is equivalent to:
  - $P(Y|X) = P(Y)$
  - $P(X,Y) = P(X) \times P(Y)$ (X and Y are independent)

  If $P(X,Y) > P(X) \times P(Y)$: X & Y are positively correlated

  If $P(X,Y) < P(X) \times P(Y)$: X & Y are negatively correlated

Measures that take into account statistical dependence

$$Lift = \frac{P(Y \mid X)}{P(Y)}$$

$$Interest = \frac{P(X,Y)}{P(X)P(Y)}$$

lift is used for rules while interest is used for itemsets

$$PS = P(X,Y) - P(X)P(Y)$$

$$\phi - coefficient = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1-P(X)]P(Y)[1-P(Y)]}}$$

Example: Lift/Interest

	Coffee	$\overline{\text{Coffee}}$	
Tea	150	50	200
$\overline{\text{Tea}}$	650	150	800
	800	200	1000

Association Rule: Tea $\rightarrow$ Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.8

$\Rightarrow$ Interest = 0.15 / (0.2×0.8) = 0.9375 (< 1, therefore is negatively associated)

There are lots of measures proposed in the literature

Measure (Symbol)	Definition
Correlation (ϕ)	$\frac{N f_{11} - f_{1+} f_{+1}}{\sqrt{f_{1+} f_{+1} f_{0+} f_{+0}}}$
Odds ratio (α)	$(f_{11} f_{00}) / (f_{10} f_{01})$
Kappa (κ)	$\frac{N f_{11} + N f_{00} - f_{1+} f_{+1} - f_{0+} f_{+0}}{N^2 - f_{1+} f_{+1} - f_{0+} f_{+0}}$
Interest (I)	$(N f_{11}) / (f_{1+} f_{+1})$
Cosine (IS)	$(f_{11}) / (\sqrt{f_{1+} f_{+1}})$
Piatetsky-Shapiro (PS)	$\frac{f_{11}}{N} - \frac{f_{1+} f_{+1}}{N^2}$
Collective strength (S)	$\frac{f_{11} + f_{00}}{f_{1+} f_{+1} + f_{0+} f_{+0}} \times \frac{N - f_{1+} f_{+1} - f_{0+} f_{+0}}{N - f_{11} - f_{00}}$
Jaccard (ζ)	$f_{11} / (f_{1+} + f_{+1} - f_{11})$
All-confidence (h)	$\min \left[\frac{f_{11}}{f_{1+}}, \frac{f_{11}}{f_{+1}} \right]$

Continuous and Categorical Attributes

How to apply association analysis to non-asymmetric binary variables?

Gender	···	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	···	26	90K	20	4	Yes
Male	···	51	135K	10	2	No
Male	···	29	80K	10	3	Yes
Female	···	45	120K	15	3	Yes
Female	···	31	95K	20	5	Yes
Male	···	25	55K	25	5	Yes
Male	···	37	100K	10	1	No
Male	···	41	65K	8	2	No
Female	···	26	85K	12	1	No
···	···	···	···	···	···	···

Example of Association Rule:

{Gender=Male, Age $\in$ [21,30)} $\rightarrow$ {No of hours online $\geq$ 10}

Handling Categorical Attributes

- Example: Internet Usage Data

Gender	Level of Education	State	Computer at Home	Online Auction	Chat Online	Online Banking	Privacy Concerns
Female	Graduate	Illinois	Yes	Yes	Daily	Yes	Yes
Male	College	California	No	No	Never	No	No
Male	Graduate	Michigan	Yes	Yes	Monthly	Yes	Yes
Female	College	Virginia	No	Yes	Never	Yes	Yes
Female	Graduate	California	Yes	No	Never	No	Yes
Male	College	Minnesota	Yes	Yes	Weekly	Yes	Yes
Male	College	Alaska	Yes	Yes	Daily	Yes	No
Male	High School	Oregon	Yes	No	Never	No	No
Female	Graduate	Texas	No	No	Monthly	No	No
...	...	...	...	...	...	...	...

{Level of Education=Graduate, Online Banking=Yes}
$\rightarrow$ {Privacy Concerns = Yes}

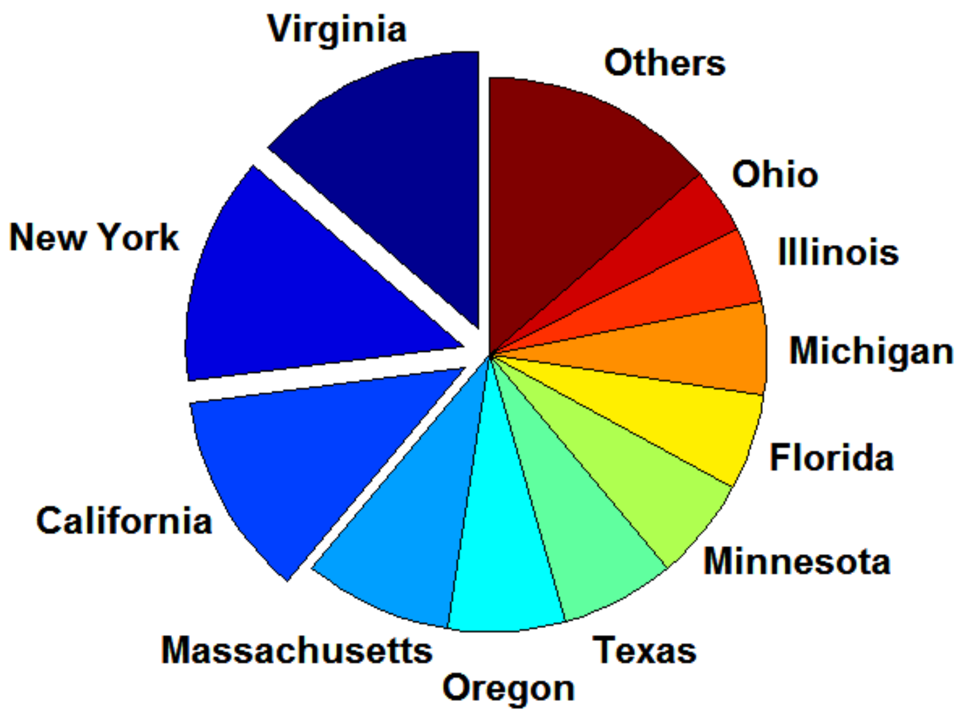
Handling Categorical Attributes

- Introduce a new "item" for each distinct attribute-value pair

Male	Female	Education = Graduate	Education = College	Education = High School	...	Privacy = Yes	Privacy = No
0	1	1	0	0	...	1	0
1	0	0	1	0	...	0	1
1	0	1	0	0	...	1	0
0	1	0	1	0	...	1	0
0	1	1	0	0	...	1	0
1	0	0	1	0	...	1	0
1	0	0	0	0	...	0	1
1	0	0	0	1	...	0	1
0	1	1	0	0	...	0	1
...	...	...	...	...	...	...	...

Handling Categorical Attributes

- Some attributes can have many possible values
  - Many of their attribute values have very low support
    - Potential solution: Aggregate the low-support attribute values



Handling Continuous Attributes

- Different methods:
  - Discretization-based
  - Statistics-based
  - Non-discretization based
    - minApriori

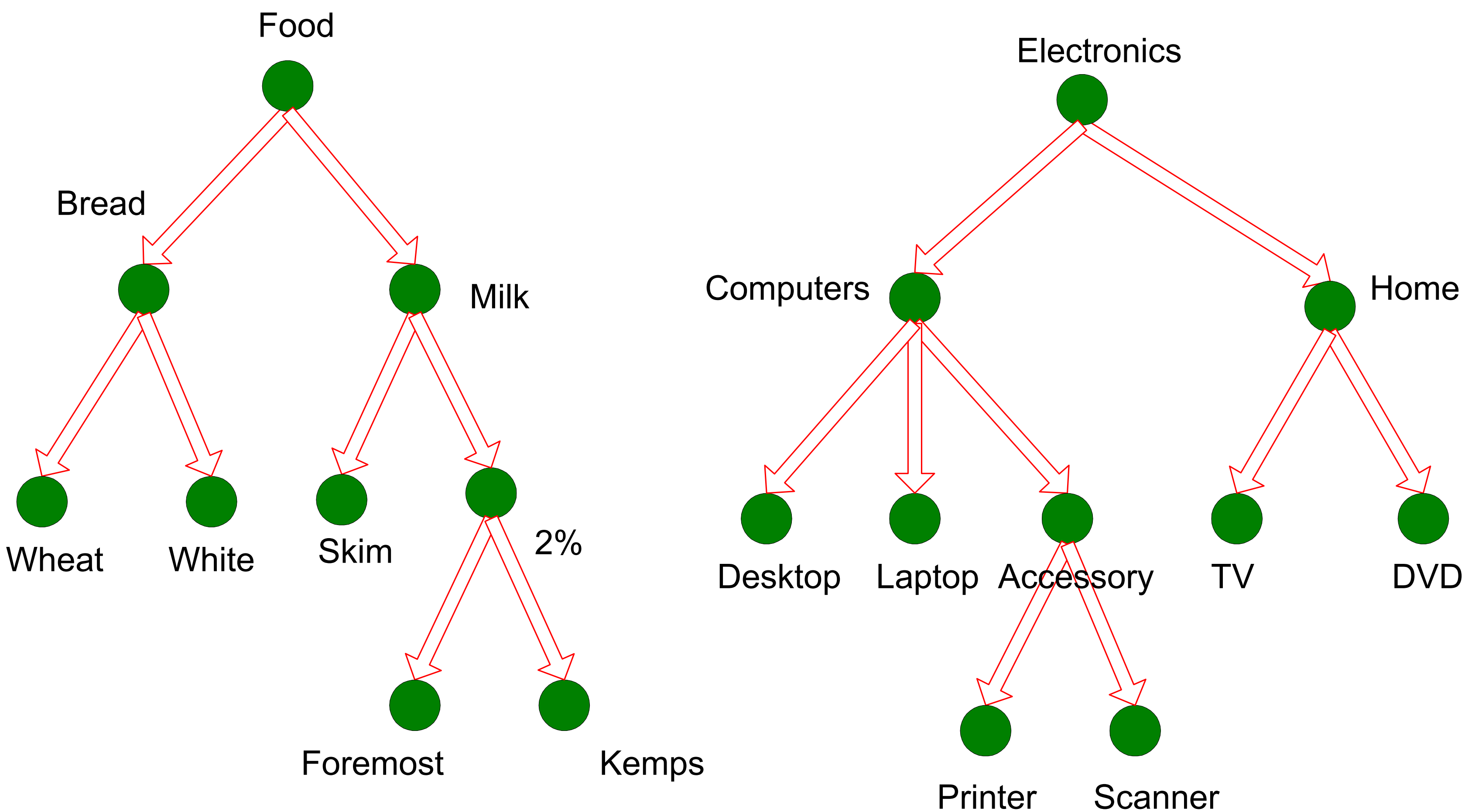
- Different kinds of rules can be produced:
  - {Age $\in$[21,30), No of hours online $\in$[10,20)} $\rightarrow$ {Chat Online =Yes}
  - {Age $\in$[21,30), Chat Online = Yes} $\rightarrow$ No of hours online: μ=14, σ=4

Discretization-based Methods

Gender	···	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	···	26	90K	20	4	Yes
Male	···	51	135K	10	2	No
Male	···	29	80K	10	3	Yes
Female	···	45	120K	15	3	Yes
Female	···	31	95K	20	5	Yes
Male	···	25	55K	25	5	Yes
Male	···	37	100K	10	1	No
Male	···	41	65K	8	2	No
Female	···	26	85K	12	1	No
···	···	···	···	···	···	···

Male	Female	··· ···	Age < 13	Age $\in [13, 21)$	Age $\in [21, 30)$	··· ···	Privacy $=$ Yes	Privacy $=$ No
0	1	···	0	0	1	···	1	0
1	0	···	0	0	0	···	0	1
1	0	···	0	0	1	···	1	0
0	1	···	0	0	0	···	1	0
0	1	···	0	0	0	···	1	0
1	0	···	0	0	1	···	1	0
1	0	···	0	0	0	···	0	1
1	0	···	0	0	0	···	0	1
0	1	···	0	0	1	···	0	1
···	···	···	···	···	···	···	···	···

Concept Hierarchies



Multi-level Association Rules

- Why should we incorporate concept hierarchy?
  - Rules at lower levels may not have enough support to appear in any frequent itemsets
  - Rules at lower levels of the hierarchy are overly specific
    - e.g., skim milk → white bread, 2% milk → wheat bread, skim milk → wheat bread, etc.

    are indicative of association between milk and bread
  - Rules at higher level of hierarchy may be too generic

Multi-level Association Rules

- How do support and confidence vary as we traverse the concept hierarchy?
  - If X is the parent item for both X1 and X2, then $\sigma(X) \leq \sigma(X1) + \sigma(X2)$
  - If $\sigma(X1 \cup Y1) \geq$ minsup,
and X is parent of X1, Y is parent of Y1
then $\sigma(X \cup Y1) \geq$ minsup, $\sigma(X1 \cup Y) \geq$ minsup
$\sigma(X \cup Y) \geq$ minsup
  - If conf$(X1 \Rightarrow Y1) \geq$ minconf,
then conf$(X1 \Rightarrow Y) \geq$ minconf

Multi-level Association Rules

- Approach 1:
  - Extend current association rule formulation by augmenting each transaction with higher level items

  Original Transaction: {skim milk, wheat bread}

  Augmented Transaction:
  {skim milk, wheat bread, milk, bread, food}

- Issues:
  - Items that reside at **higher levels have much higher support** counts
    - if support threshold is low, too many frequent patterns involving items from the higher levels
  - **Increased dimensionality** of the data

Multi-level Association Rules

- Approach 2:
  - Generate frequent patterns at highest level first
  - Then, generate frequent patterns at the next highest level, and so on
- Issues:
  - I/O requirements will increase dramatically because we need to perform more passes over the data
  - **May miss some potentially interesting cross-level association patterns**

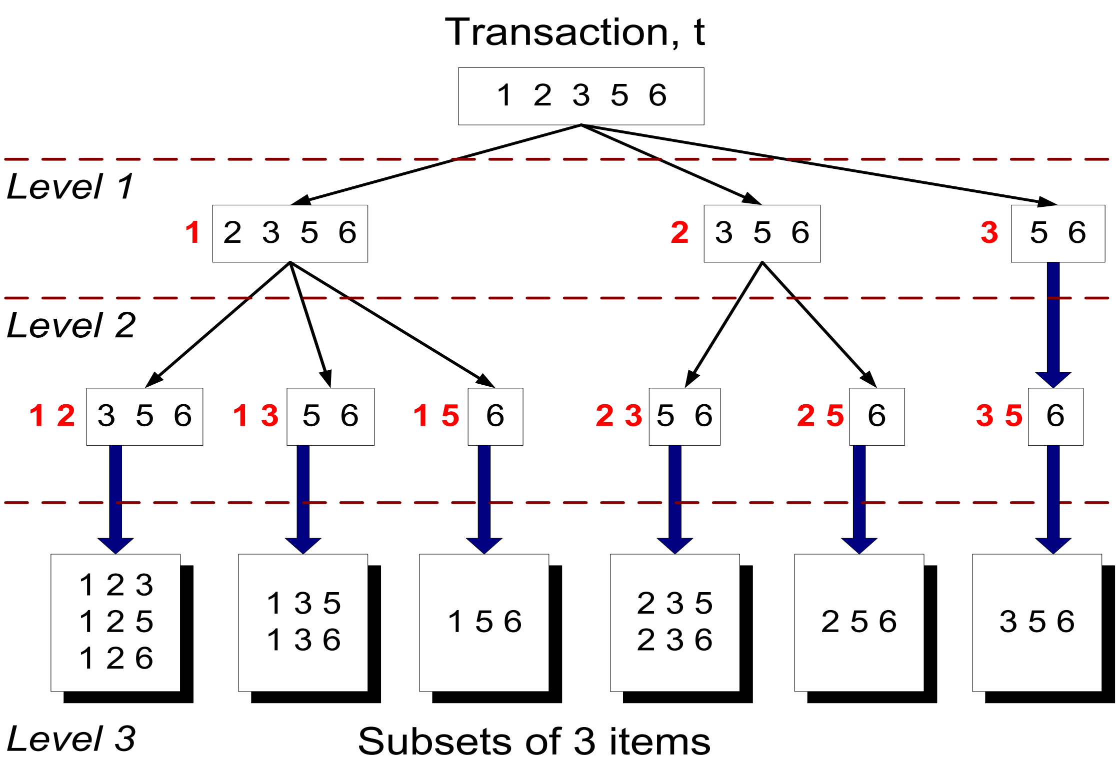
Support Count strategy

Support Counting: An Example

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}

How many of these itemsets are supported by transaction (1,2,3,5,6)?



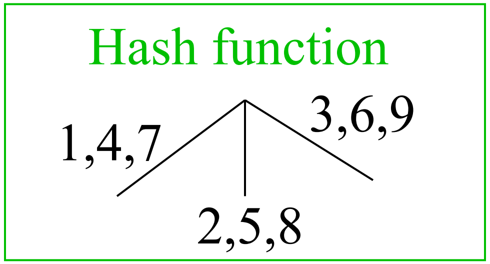
Support Counting Using a Hash Tree

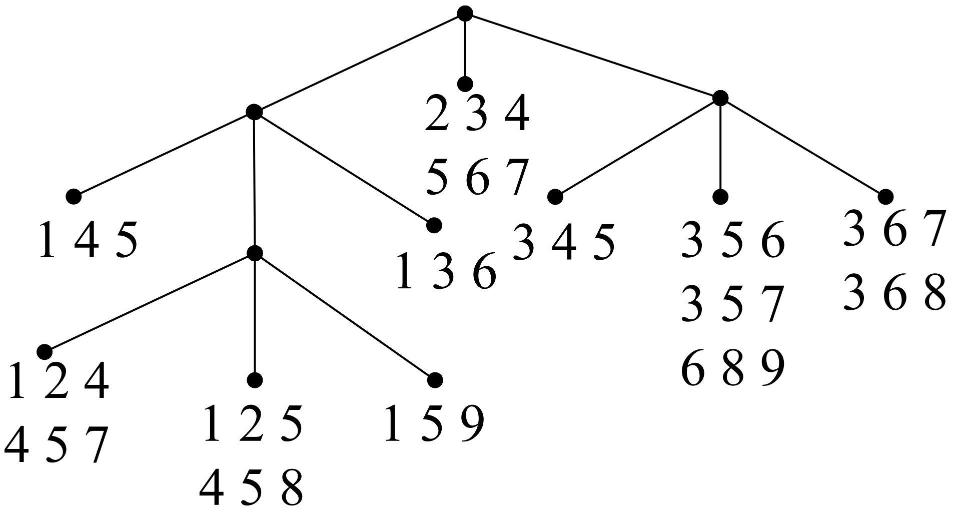
Suppose you have 15 candidate itemsets of length 3:

{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}

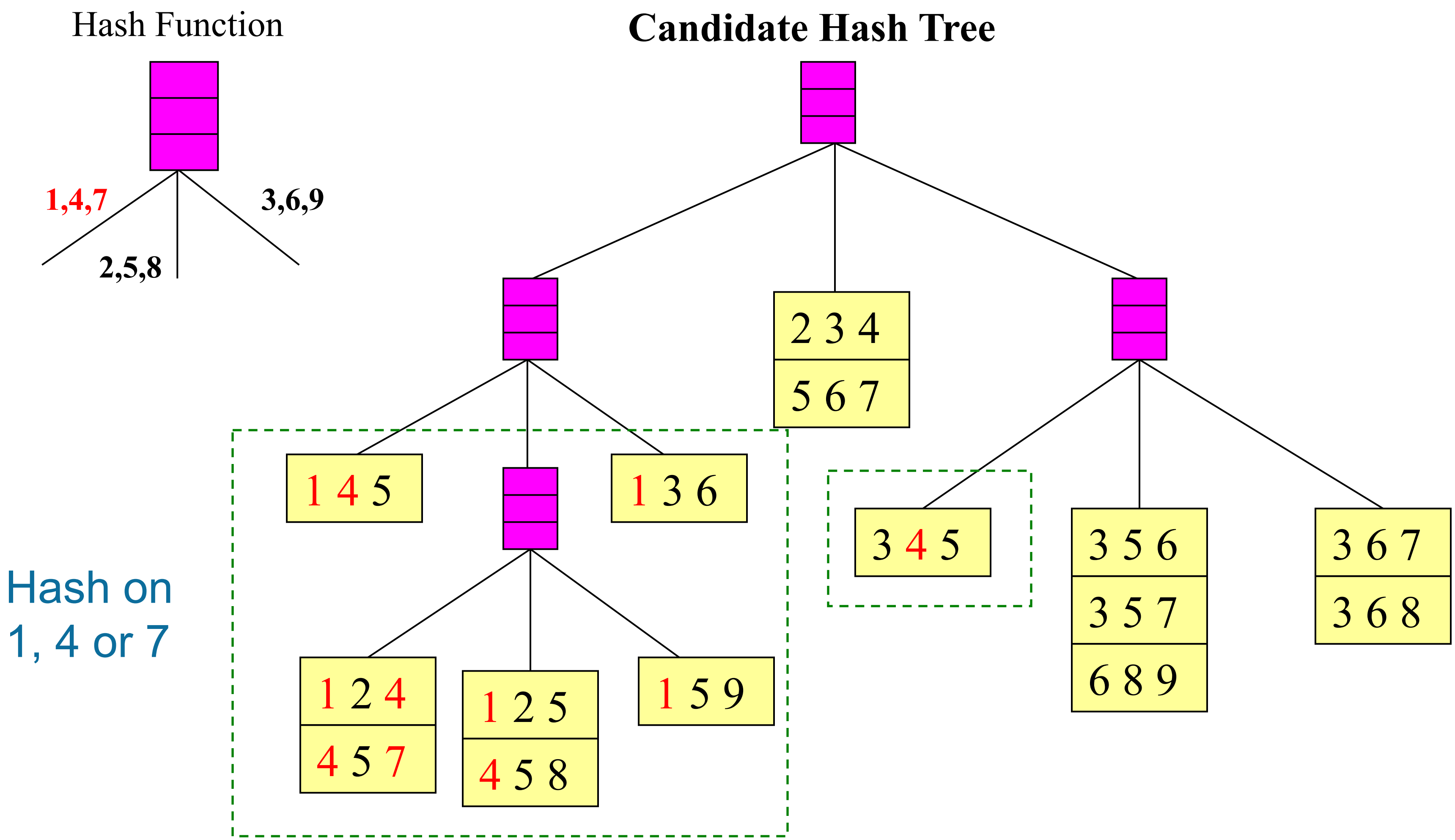
You need:

- **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)**

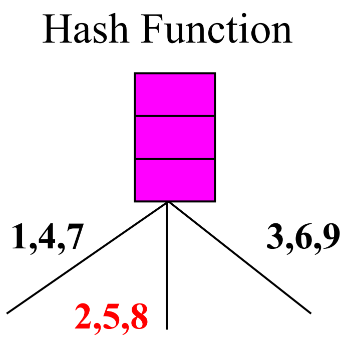


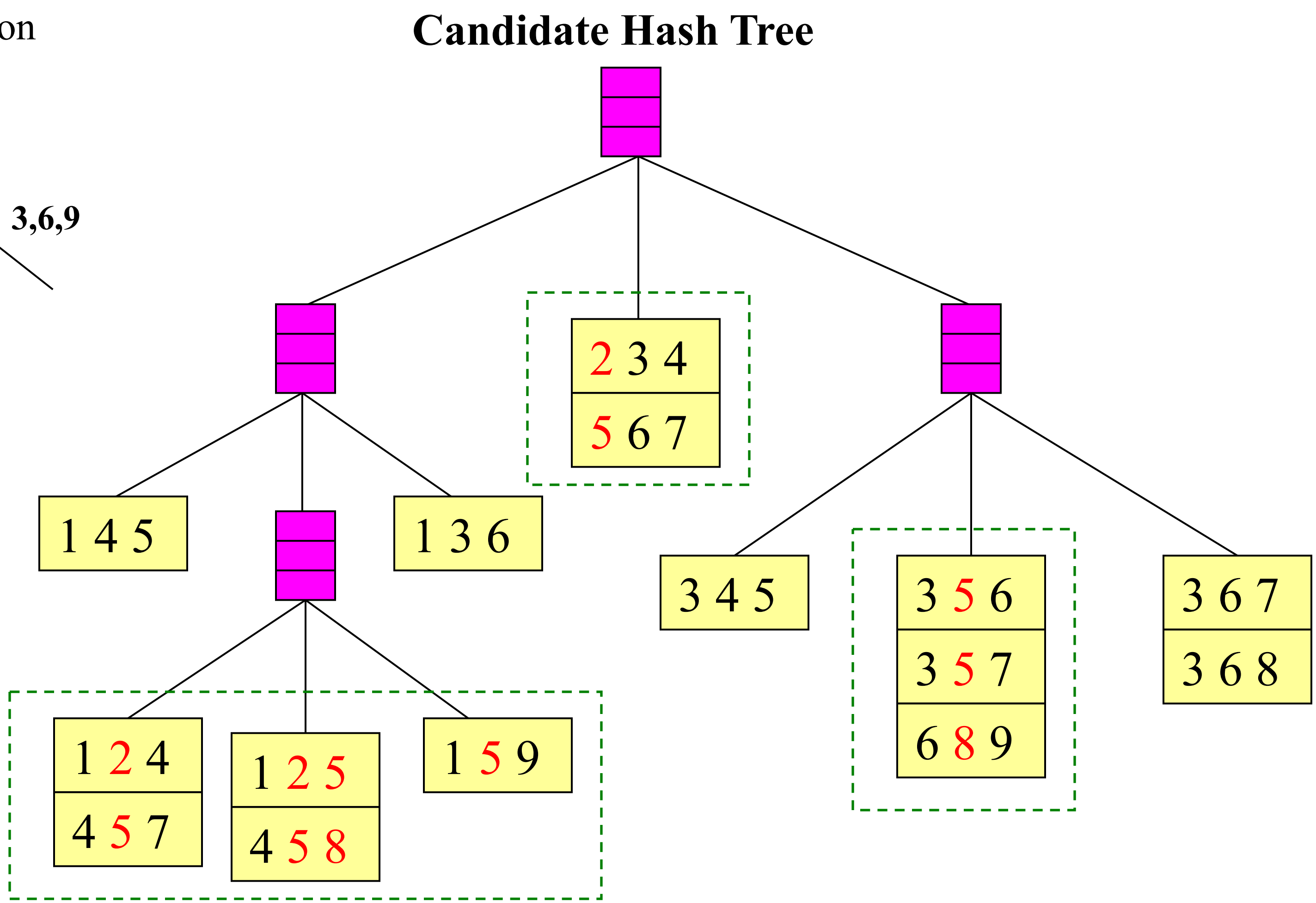


Support Counting Using a Hash Tree



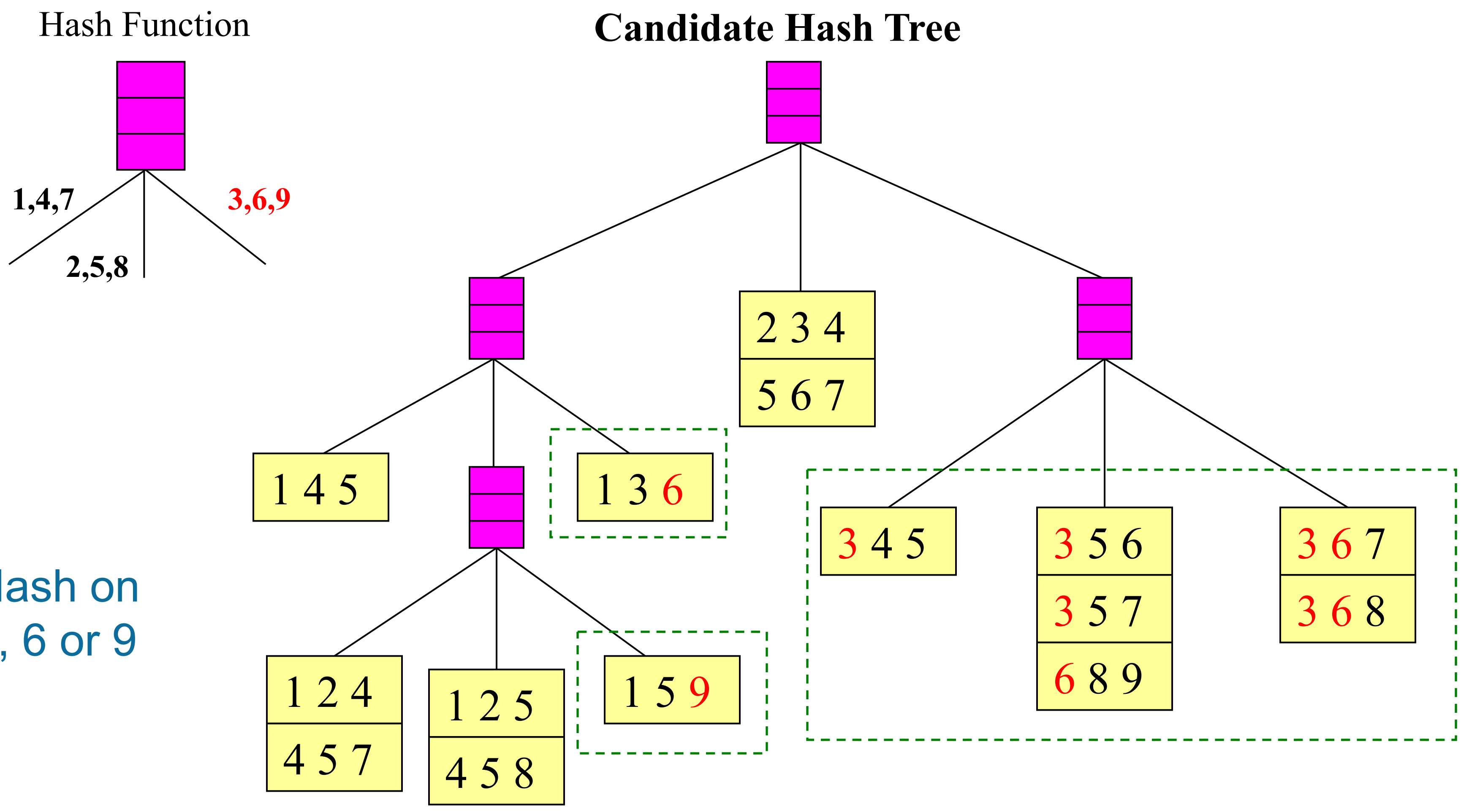
Support Counting Using a Hash Tree





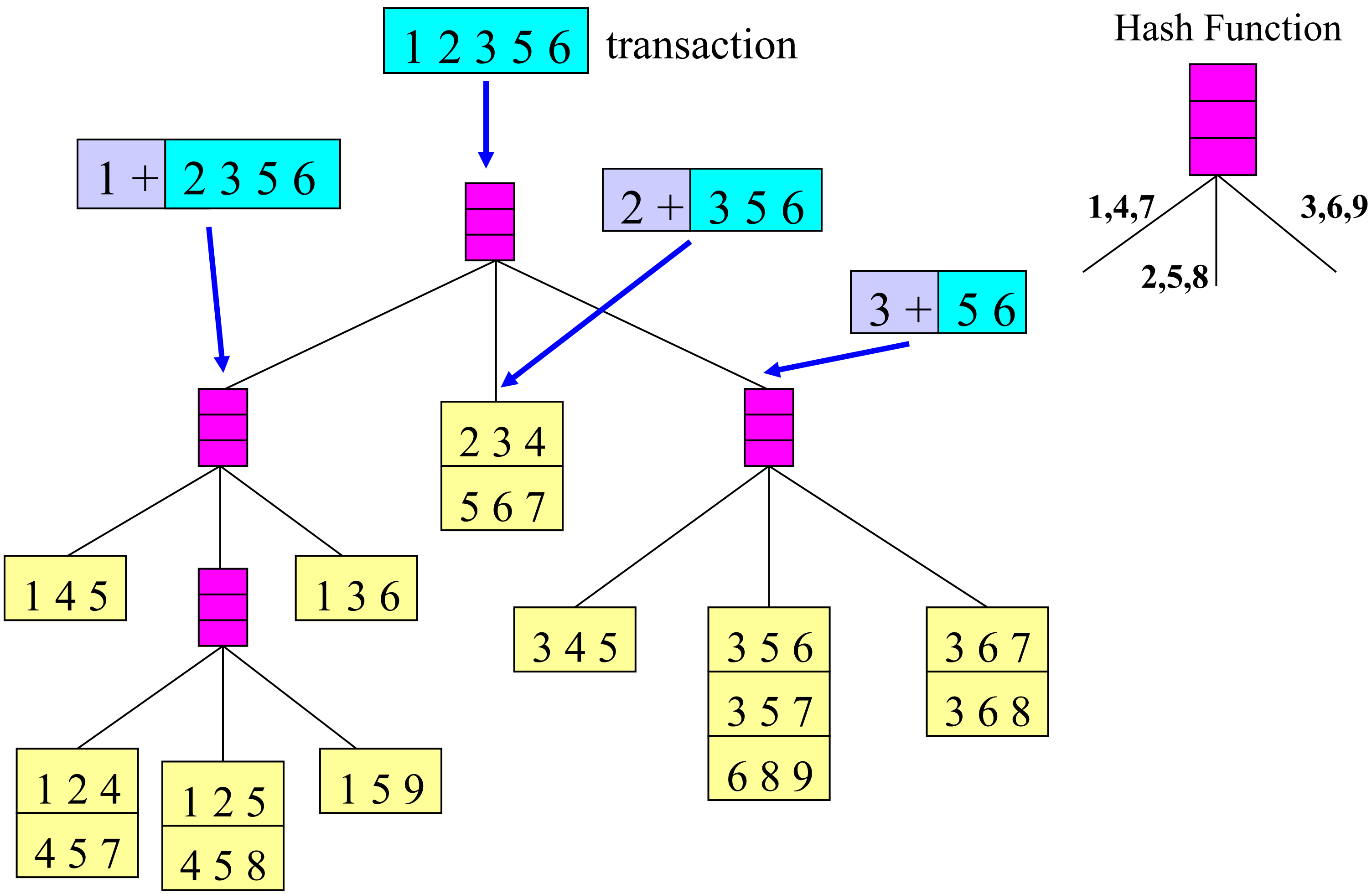
Hash on
2, 5 or 8

Support Counting Using a Hash Tree

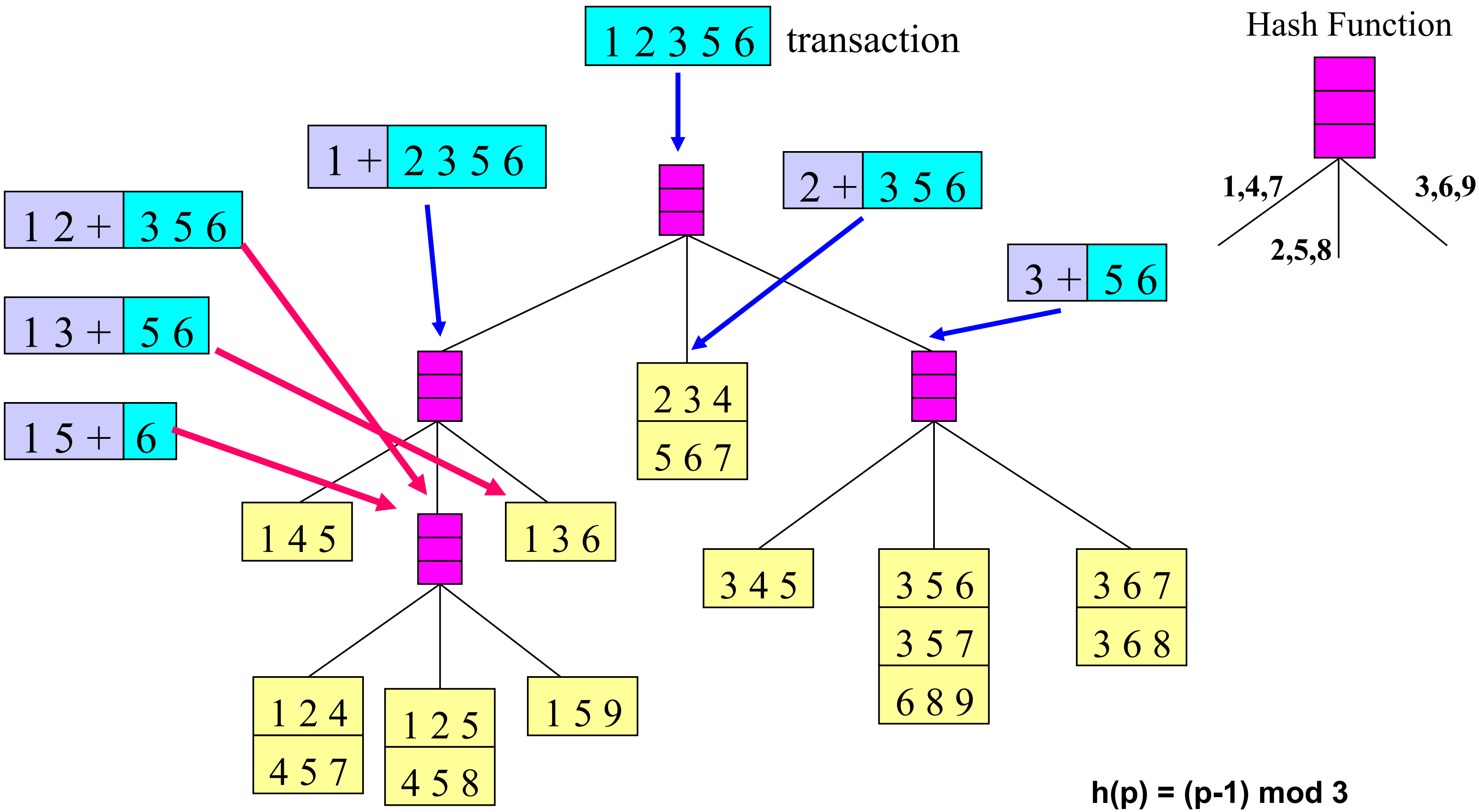


Hash on
3, 6 or 9

Support Counting Using a Hash Tree

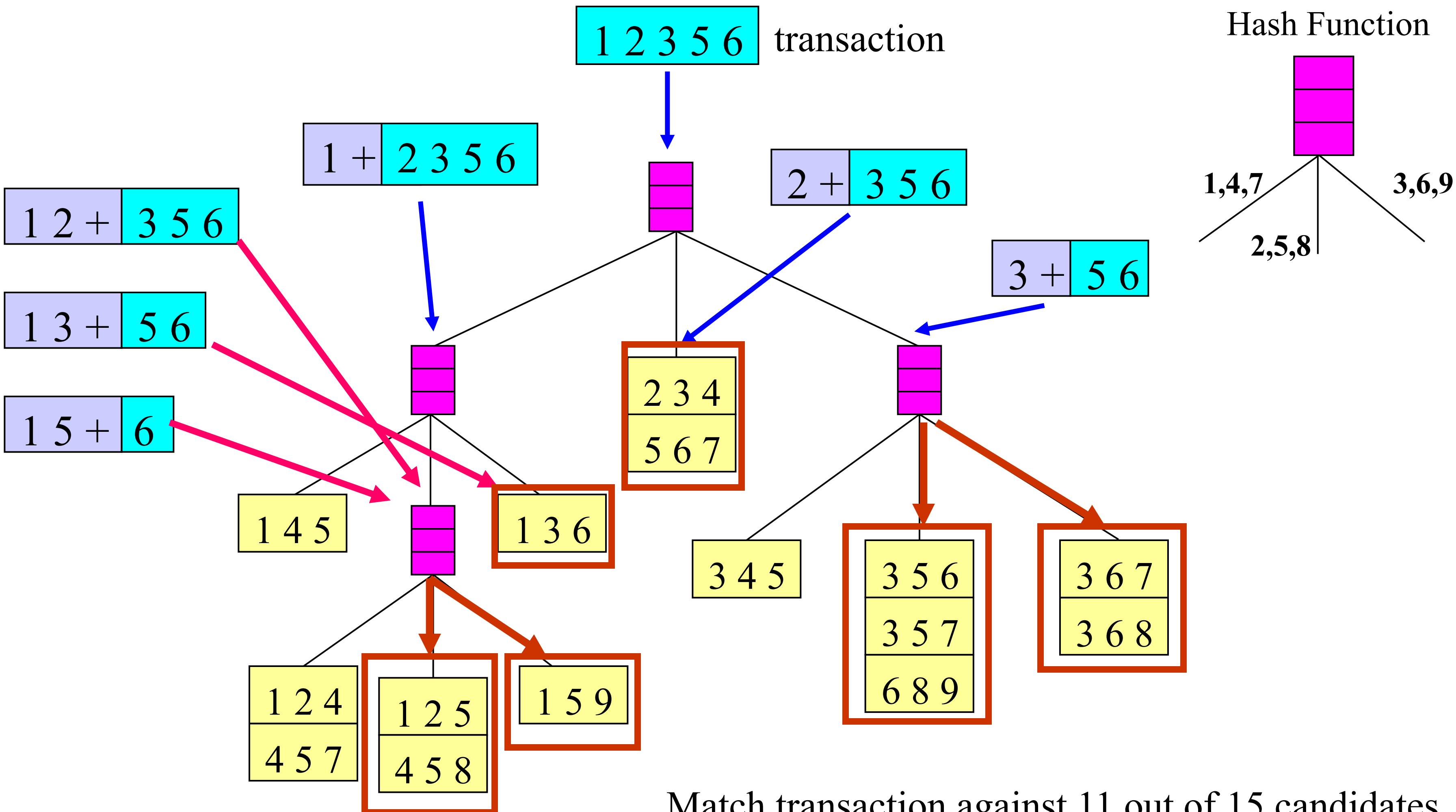


Support Counting Using a Hash Tree



h(p) = (p-1) mod 3

Support Counting Using a Hash Tree



Match transaction against 11 out of 15 candidates