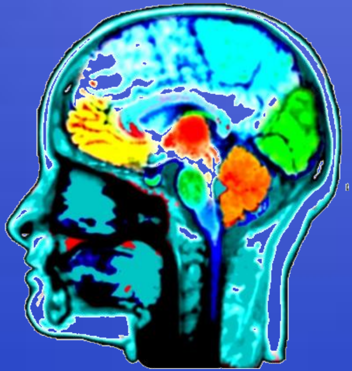
Introduction To Data Mining

Isfahan University of Technology (IUT)
Farvardin 1401



Frequent Pattern Mining

Dr. Hamidreza Hakim
hamid.hakim.u@gmail.com

بسم الله الرحمن الرحیم

What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?

چه کالاهایی باهم تکرار شدند؟ یا توی یک گرافی میخاهیم نودها با ویژگی های خاصی رو شناسایی کنیم مثلا ترتیب و تکرار ژن های مختلف در افراد باعث ویژگی های متفاوتشون میشه

- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

توالی اتفاق؟

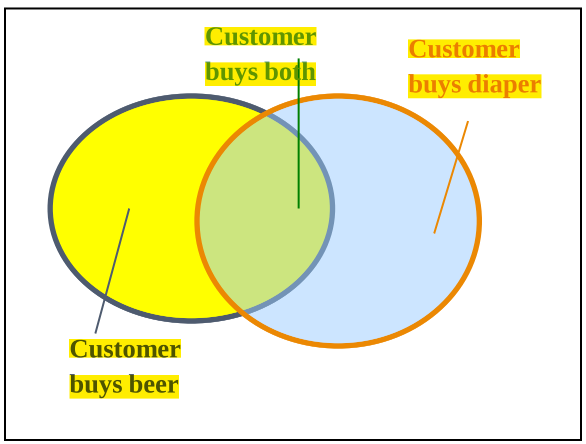
ما یک متنی داریم و میخایم ببینیم چه کلمات پرتکراری توی متن وجود داره؟ چه دنباله ای از کلمات پشت سر هم خیلی تکرار شده؟

Why Is Freq. Pattern Mining Important?

- Freq. pattern: An intrinsic and important property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

Basic Concepts: Frequent Patterns

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$

ینی مجموعه ی ما
kتا عضو داره

- ***(absolute) support*, or, *support count* of X:** Frequency or occurrence of an itemset X

تعداد تکرار یه
جموعه در کل لیست
ایتم هامون

- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)

مینیمم ساپورت چقدره؟
مثلا میگیم الگوهای تکراری رو پیدا
کن که حداقل دوبار رخ دادن

- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

مثلا خرید شیرو ماست به طور
۱۰درصد مواقع در خرید ها باهم
خریده شدن که بهش ساپورت
نسبی میگیم

Basic Concepts: Frequent Patterns

ما با تراکنش ها کار داریم!

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E

1-itemset

Support count ({C}) = 3
Support ratio ({C})= 3/4

2-itemset

Support count ({B, C}) = 2
Support ratio ({B,C})= 2/4

3-itemset

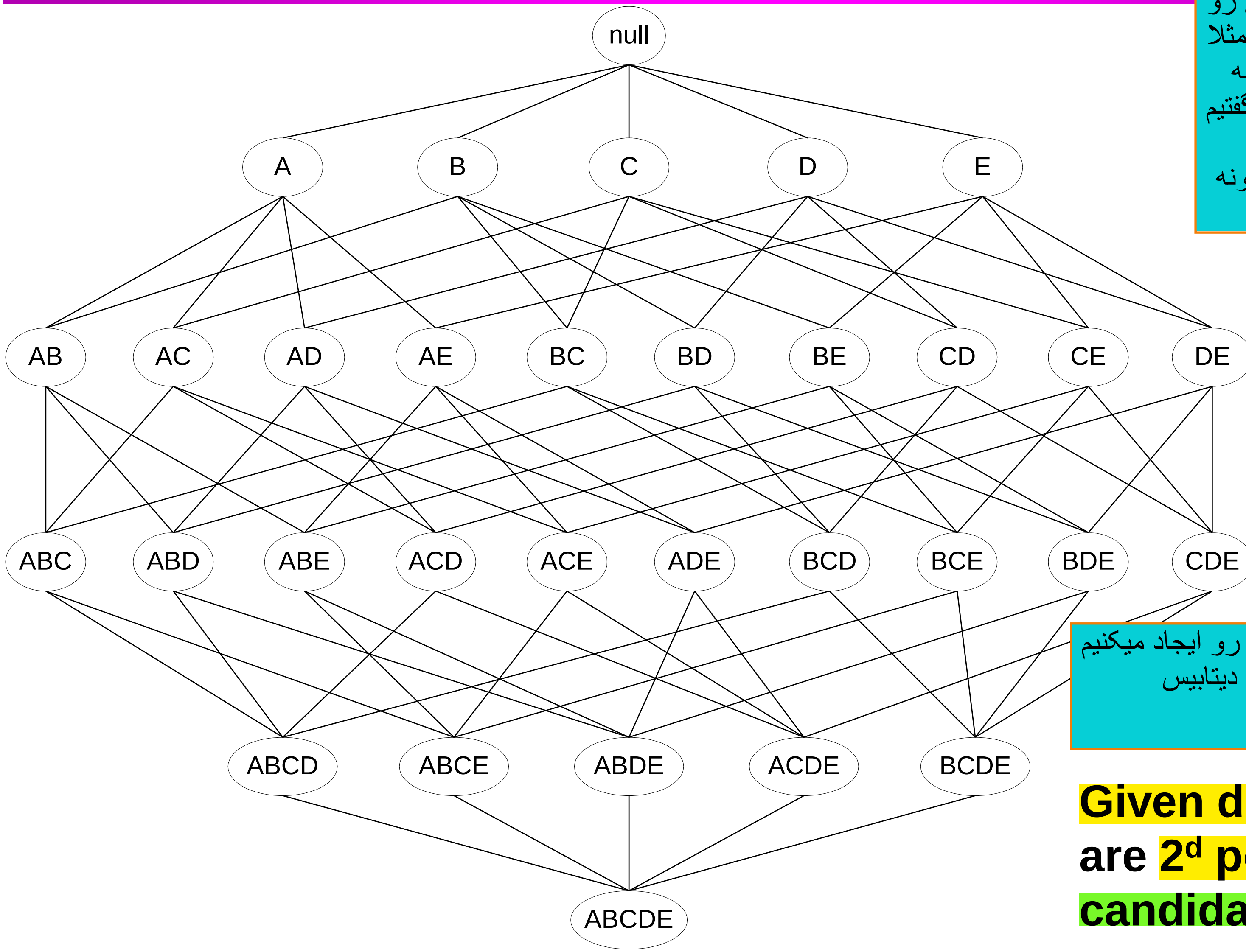
Support count ({B, C, E}) = 2
Support ratio({B,C,E}) = 2/4

یعنی ۵۰درصد تراکنش های ما، این مجموعه رو توی خودشون دارن

If minsup = 0.7 → {C} is a Frequent itemset

این مجموعه یک الگوی پرتکرار است

Frequent Itemset Generation



ما اگه بخایم الگوهای پرتکرار رو برای ۴تا ایتم ببینیم باید همچین درختی رو ررسی کنیم چون نگفتیم مثلا الگوهای پرتکرار یه دونه ی یا دوتایی یا ۳تایی و گفتیم الگوهای پرتکراری که میتونه اتفاق بیفته،که میتونه از ۱ تایی باشه تا۴تایی

رابطه ی تعداد الگوهای پرتکرار با تعداد ایتم ها نمایی است!

ول همه ی الگوهای پرتکرار رو ایجاد میکنیم بعد تعداد تکرارشون رو توی دیتابیس میشماریم

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

Frequent Itemset Generation

- Brute-force approach:

بررسی همه ی حالت ها

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

Transactions

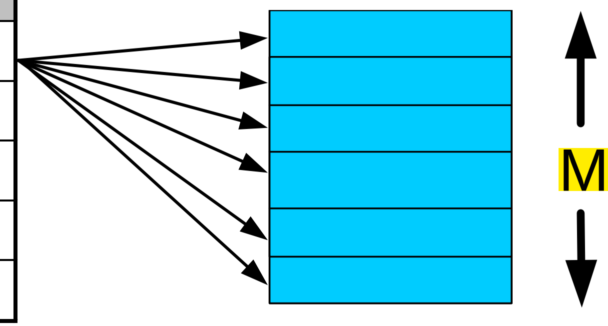
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

N

w

طول بزرگترین لیست خرید یا بیشترین تعداد ایتم خریداری شده در یک تراکنش

List of Candidates



  – Match each transaction against every candidate

نمایی میشه

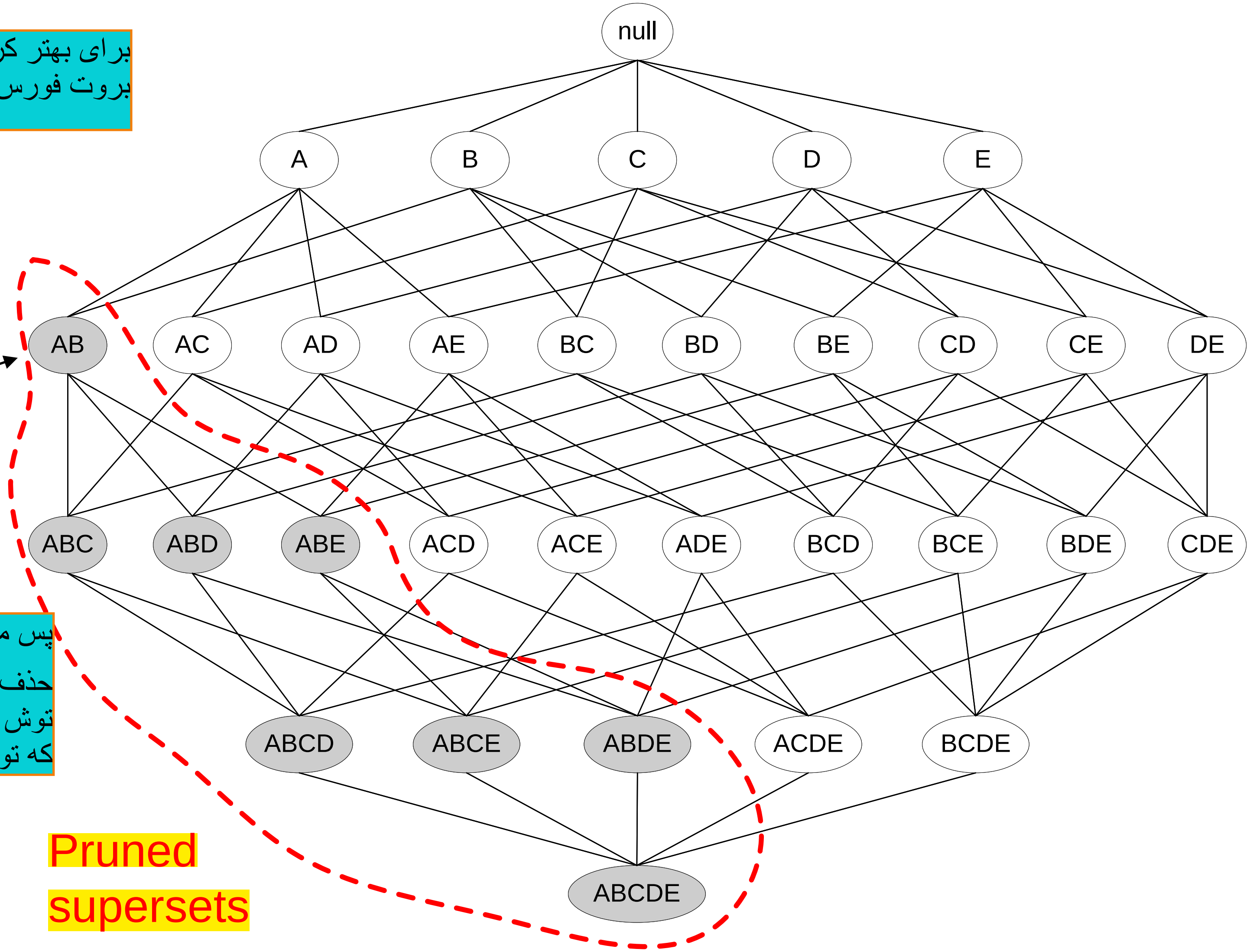
  – Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Illustrating Apriori Principle

برای بهتر کردن این عملیات سرچ و
بروت فورس چیکار میتونیم بکنیم؟

وقتی بدونیم پدر یه مجموعه
کم تکراره میشه نتیجه گرفت
همه ی بچه هاش هم، کم
تکراره پس از بررسی بچه
هاشون صرف نظر میکنیم و
حرس شون میکنیم

Found to be Infrequent



پس مجموعه ی سوپرست ABرو
حذف میکنیم: مجموعه ای که AB
توش هست که همین نودهایی میشه
که تو دایره قرمزه است

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

در کل ۶تا ایتم داریم

Items (1-itemsets)

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

Minimum Support = 3

چه ایتم هایی حذف میشوند؟ اونایی که تعداد تکرارشون کمتر از ۳ باشه: تخم مرغ ونوشابه

ساخت ایتم های یک تایی: انتخاب یک از ۶ : ۶
ساخت ایتم های دوتایی : انتخاب ۲ از ۶ = ۱۵
به همین ترتیب...

If every subset is considered,

$$^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

$$C(n, r) = \frac{n!}{r!\,(n-r)!}$$

انتخاب ۲ از ۴

انتخاب ۳ از ۴

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

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3$

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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

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

If every subset is considered,

$$^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

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**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

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

Minimum Support = 3

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3$

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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**

Pairs (2-itemsets)

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

(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}

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3$

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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**

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
{ Beer, Diaper, Milk}	**2**
{ Beer,Bread, Diaper}	**2**
{Bread, Diaper, Milk}	**2**
{Beer, Bread, Milk}	**1**

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

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

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**

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3$

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

$6 + 6 + 1 = 13$

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
    1. **Candidate Generation**: Generate L_{k+1} from F_k
    2. **Candidate Pruning**: Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
    3. **Support Counting**: Count the support of each candidate in L_{k+1} by scanning the DB
    4. **Candidate Elimination**: Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}

Candidate Generation: 1-Brute-force method

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

Item
Beer
Bread
Cola
Diapers
Eggs
Milk

Candidate Generation

Itemset
{Beer, Bread, Cola}
{Beer, Bread, Diapers}
{Beer, Bread, Eggs}
{Beer, Bread, Milk}
{Beer, Cola, Diapers}
{Beer, Cola, Eggs}
{Beer, Cola, Milk}
{Beer, Diapers, Eggs}
{Beer, Diapers, Milk}
{Beer, Eggs, Milk}
{Bread, Cola, Diapers}
{Bread, Cola, Eggs}
{Bread, Cola, Milk}
{Bread, Diapers, Eggs}
{Bread, Diapers, Milk}
{Bread, Eggs, Milk}
{Cola, Diapers, Eggs}
{Cola, Diapers, Milk}
{Cola, Eggs, Milk}
{Diapers, Eggs, Milk}

Candidate Pruning

Itemset
{Bread, Diapers, Milk}

Frequent 2-itemset

Itemset
{Beer, Diapers}
{Bread, Diapers}
{Bread, Milk}
{Diapers, Milk}

Figure 5.6. A brute-force method for generating candidate 3-itemsets.

Candidate Generation: 2-Merge Fk-1 and F1 itemsets

Frequent
2-itemset

Itemset
{Beer, Diapers}
{Bread, Diapers}
{Bread, Milk}
{Diapers, Milk}

Frequent
1-itemset

Item
Beer
Bread
Diapers
Milk

Candidate Generation

Itemset
{Beer, Bread, Diapers}
{Beer, Diapers, Milk}
{Bread, Diapers, Milk}
{Beer, Bread, Milk}

Candidate
Pruning

Itemset
{Bread, Diapers, Milk}

Figure 5.7. Generating and pruning candidate k-itemsets by merging a frequent $(k-1)$-itemset with a frequent item. Note that some of the candidates are unnecessary because their subsets are infrequent.

Candidate Generation: 3-F_{k-1} x 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}

Candidate Generation:3-Fk-1 x Fk-1 Method

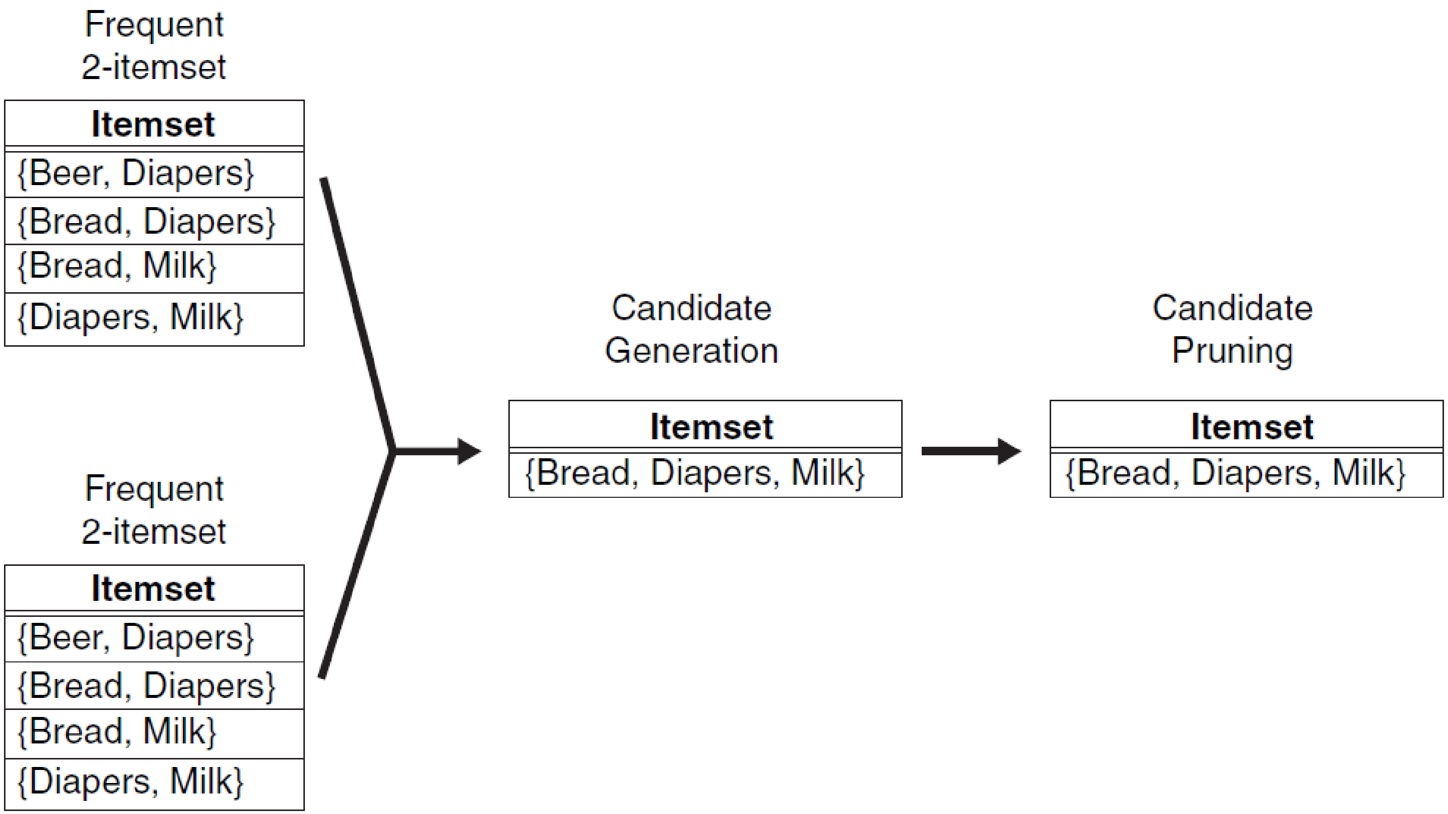


Figure 5.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

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

If every subset is considered,

$^6C_1 + ^6C_2 + ^6C_3$

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

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**<u>BC</u>**, **<u>BC</u>**D) = A**<u>BC</u>**D
  - Merge(A**<u>BD</u>**, **<u>BD</u>**E) = A**<u>BD</u>**E
  - Merge(A**<u>CD</u>**, **<u>CD</u>**E) = A**<u>CD</u>**E
  - Merge(B**<u>CD</u>**, **<u>CD</u>**E) = B**<u>CD</u>**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}

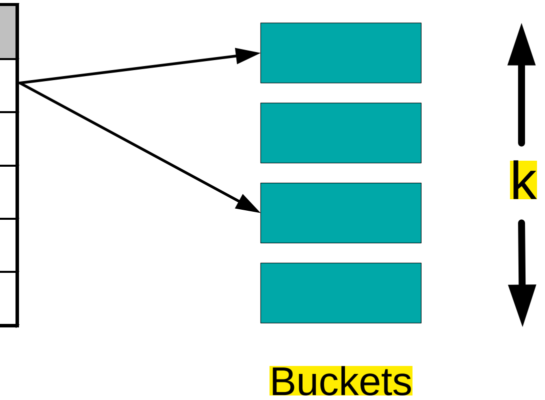
Support Counting of Candidate Itemsets

- 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 **Hash Structure**

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

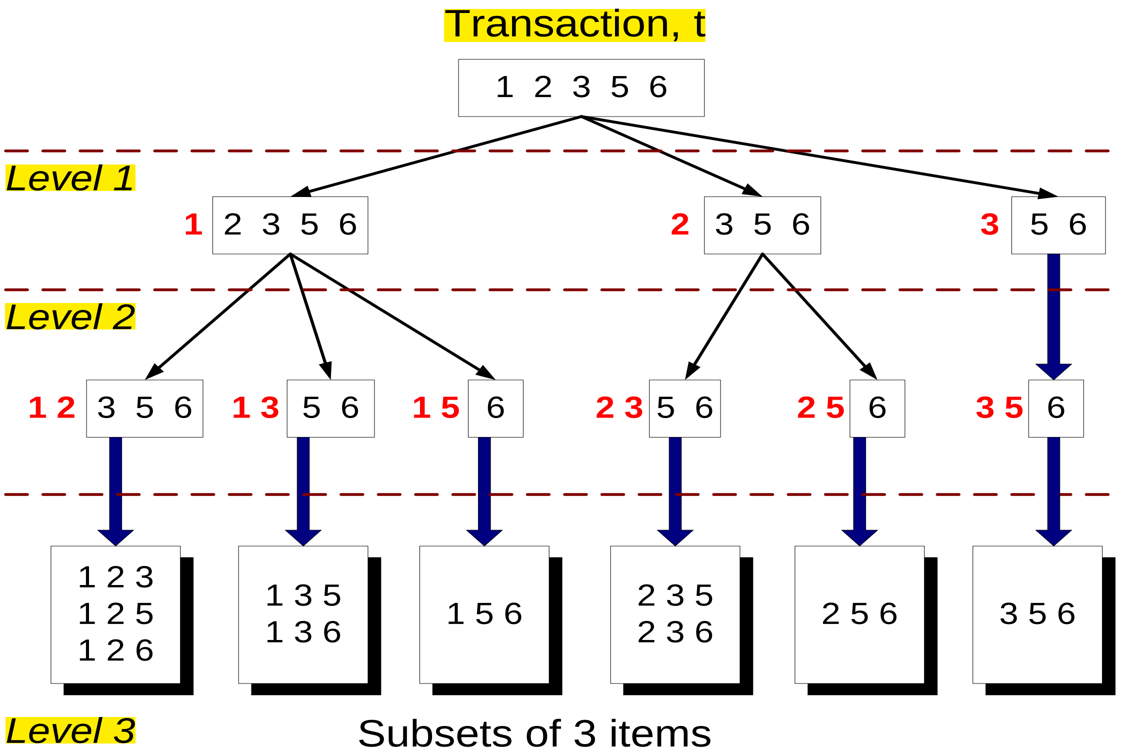


Support Counting: An Example

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)?



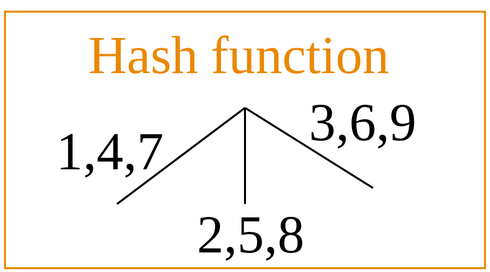
Support Counting Using a Hash Tree

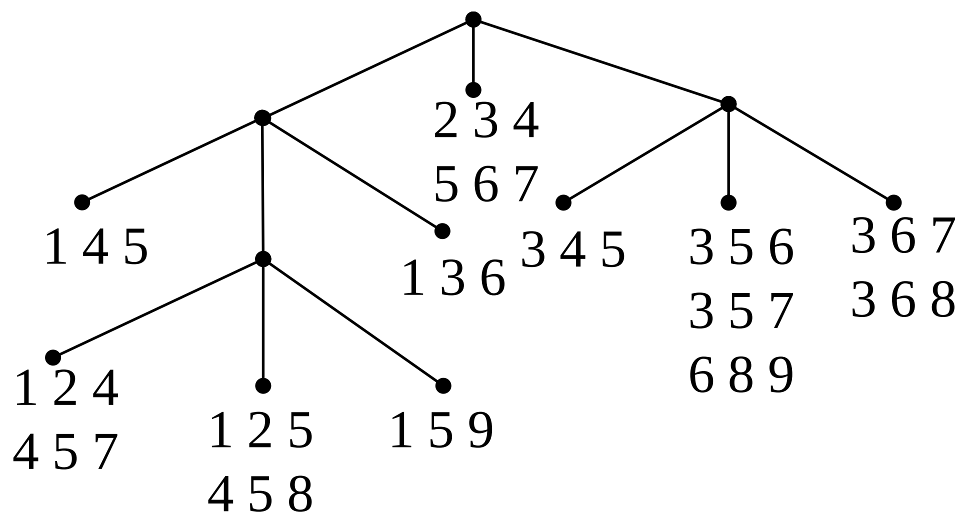
Suppose you have 15 candidate itemsets of length 3:

{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}

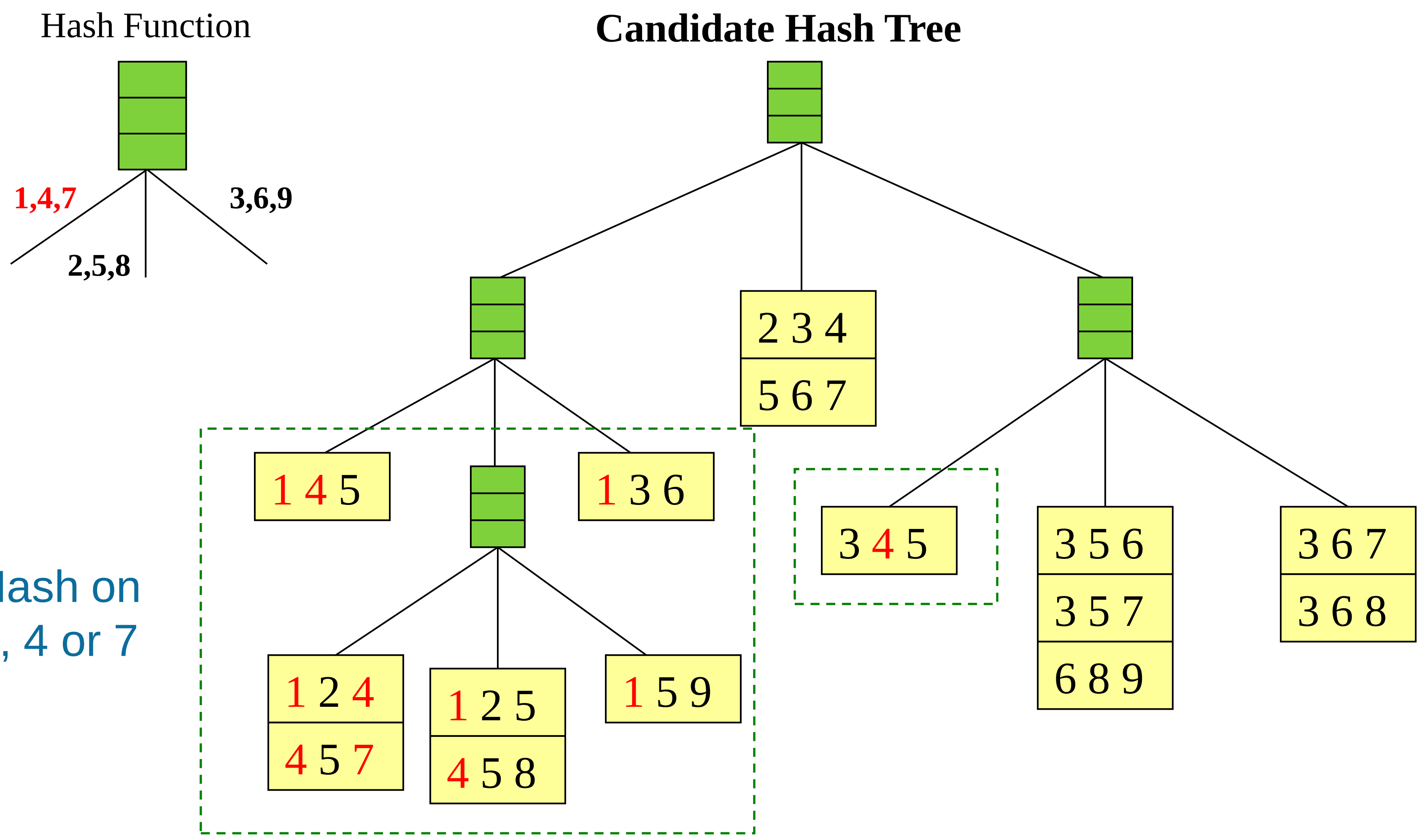
You need:

- **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)**



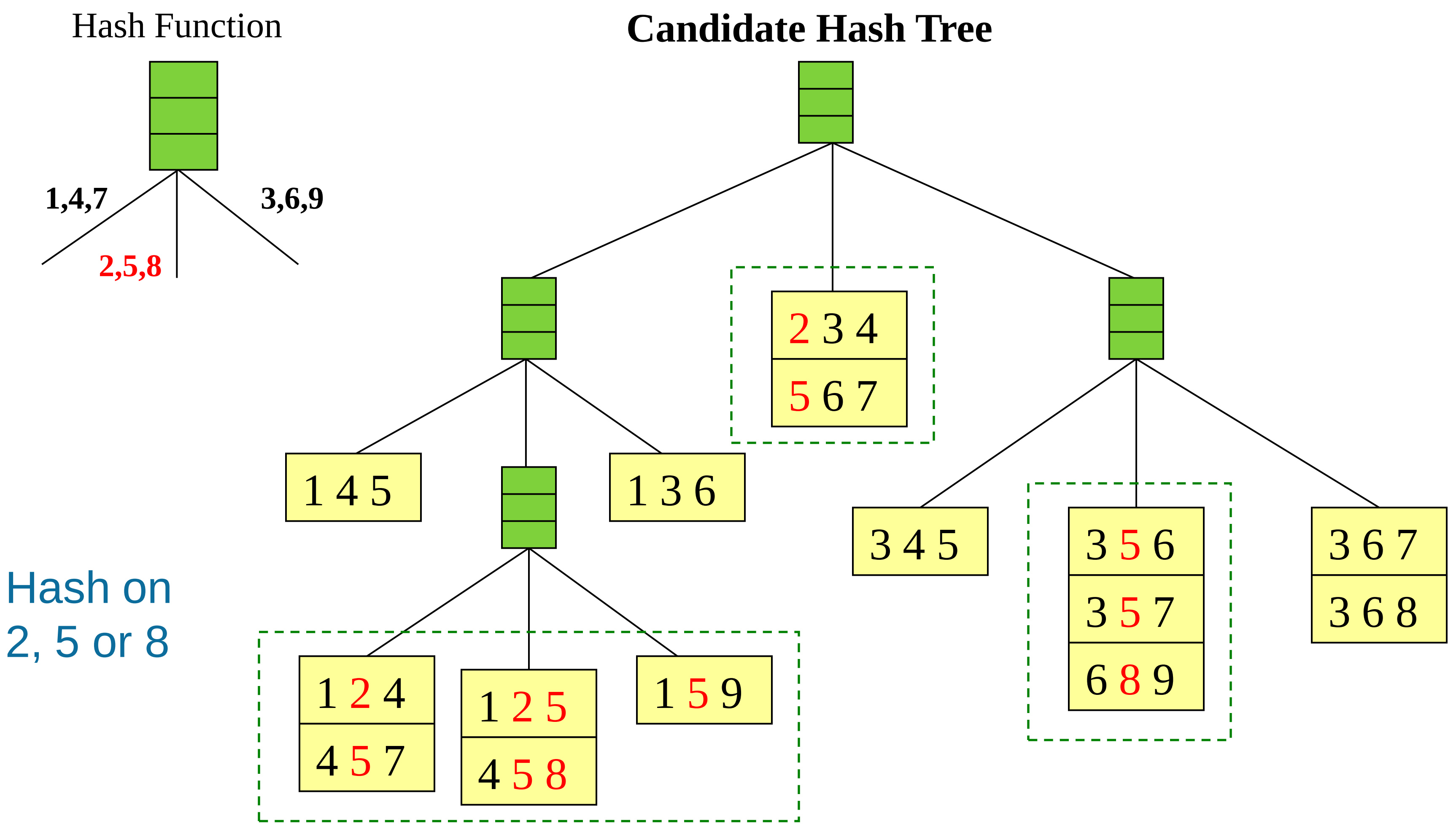


Support Counting Using a Hash Tree

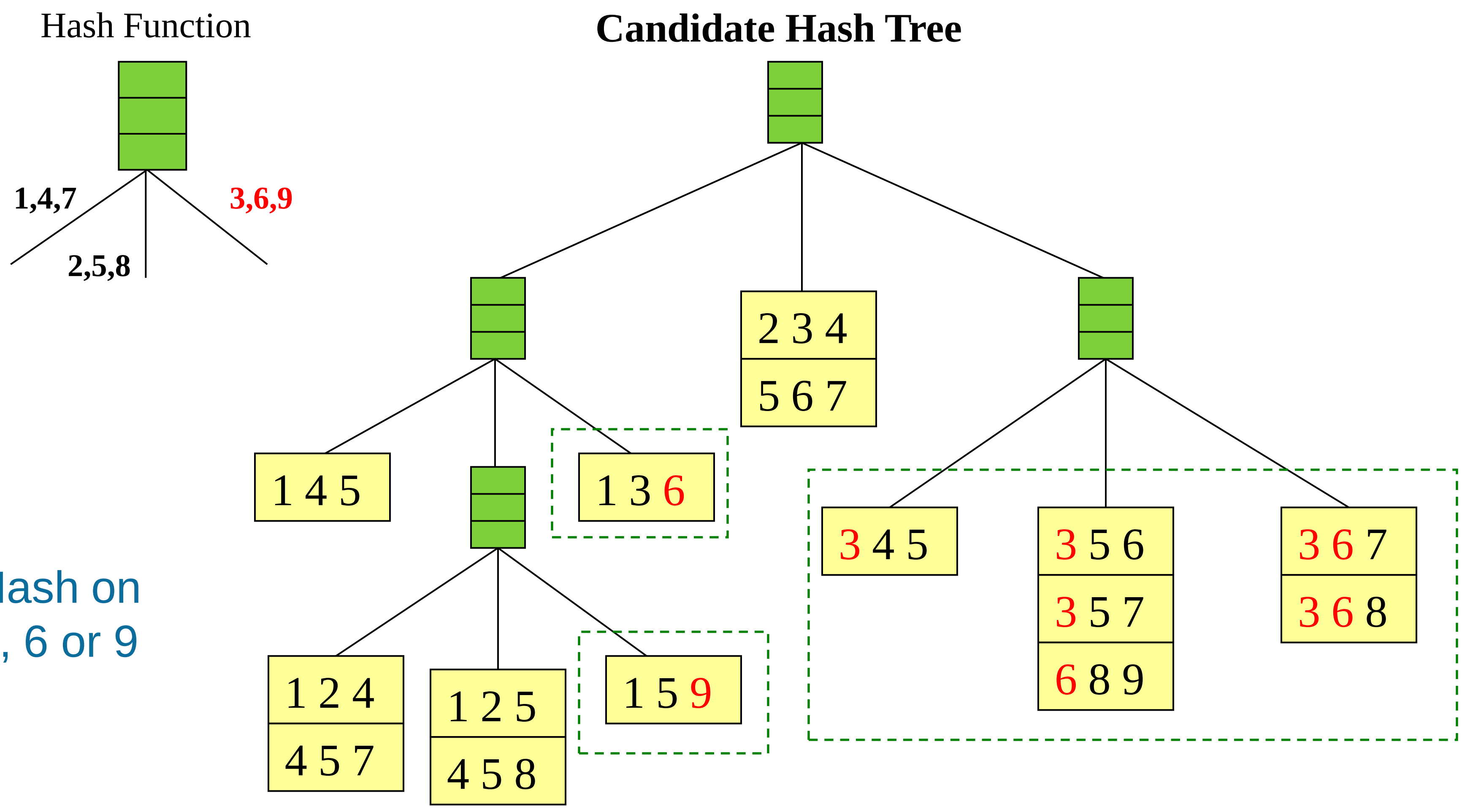


Hash on
1, 4 or 7

Support Counting Using a Hash Tree

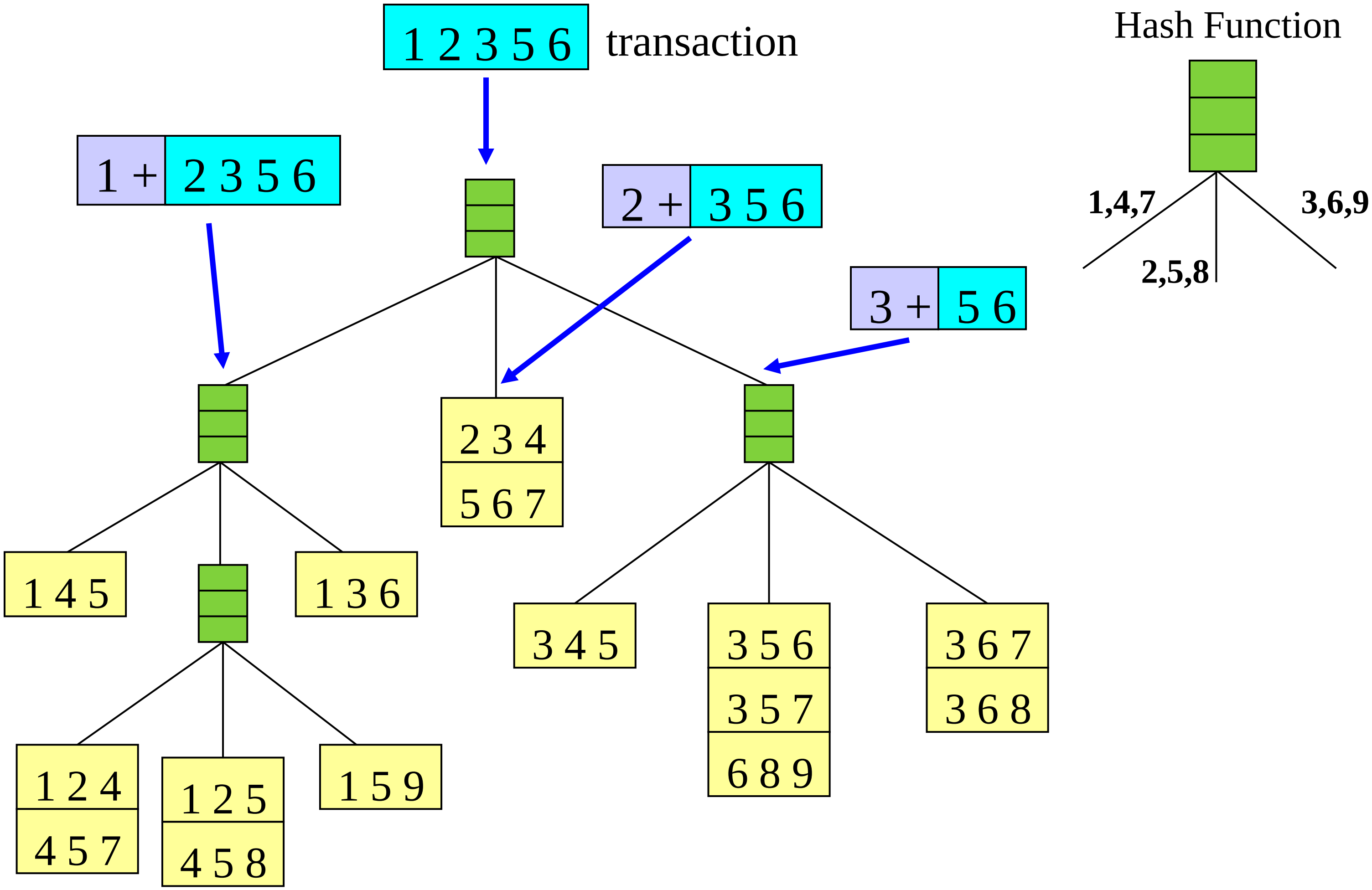


Support Counting Using a Hash Tree

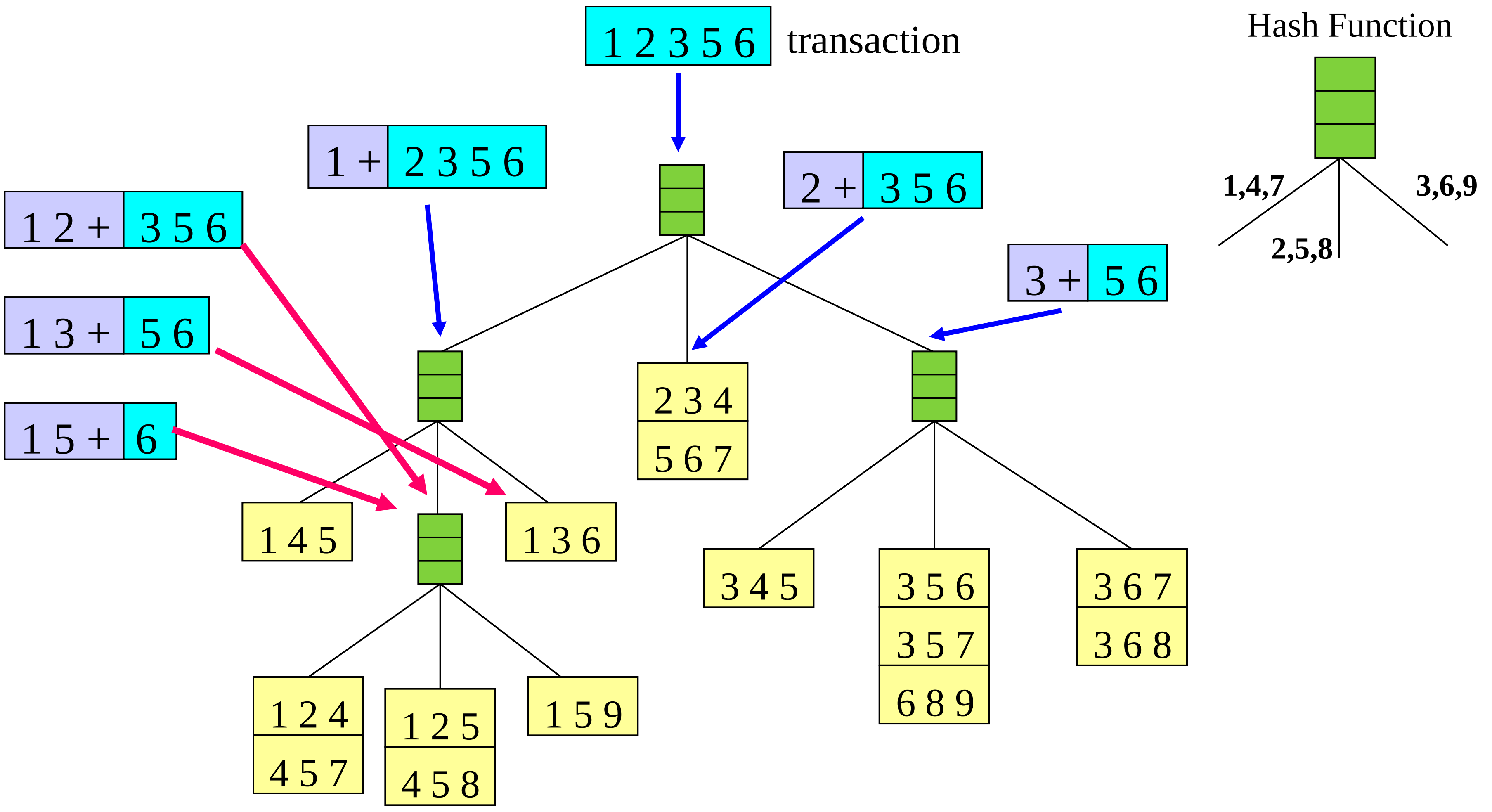


Hash on
3, 6 or 9

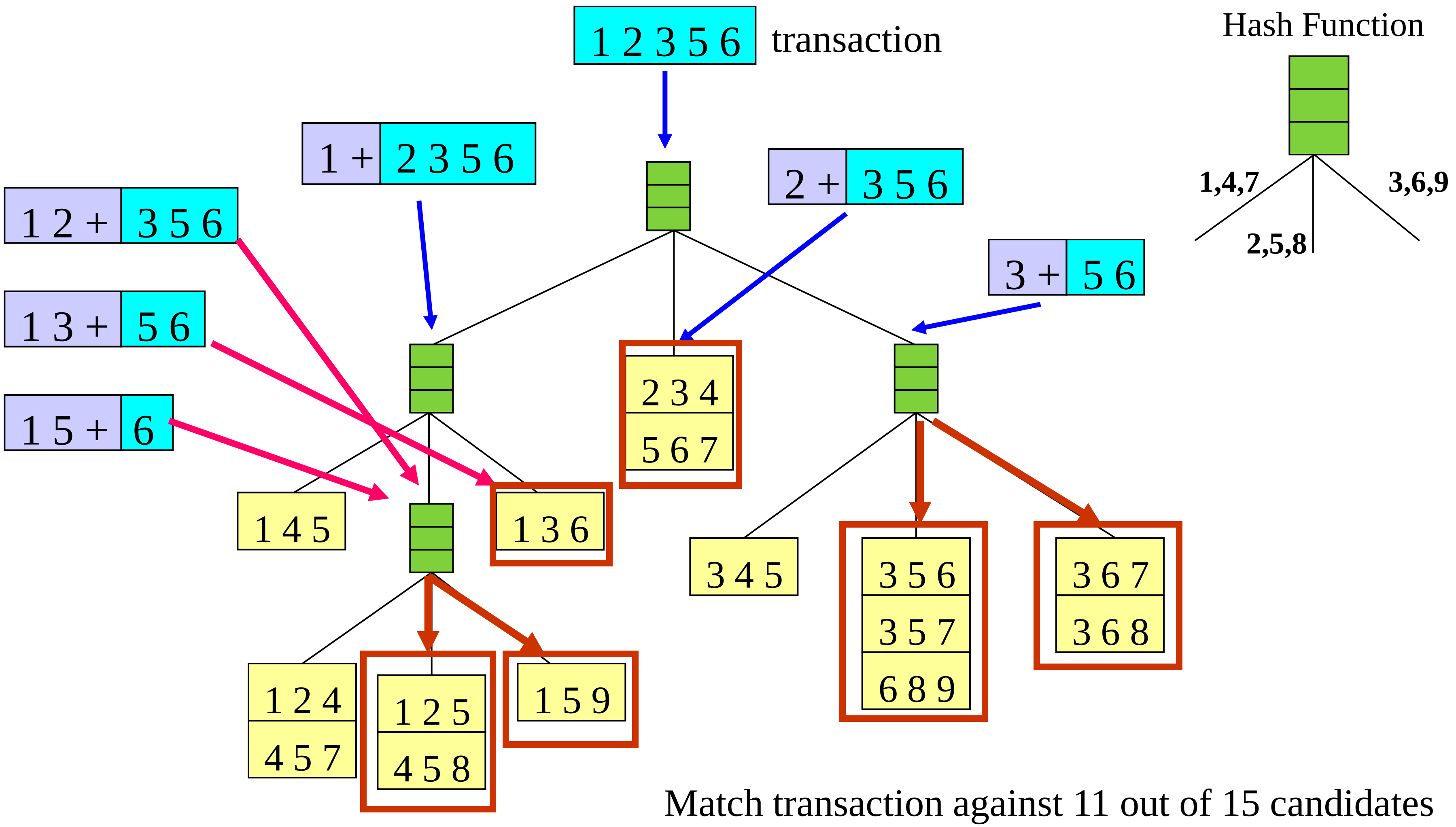
Support Counting Using a Hash Tree



Support Counting Using a Hash Tree



Support Counting Using a Hash Tree



Match transaction against 11 out of 15 candidates

ASSOCIATION RULES

Association Rule Mining

استخراج قوانین

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

Implication means co-occurrence, not causality!

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

سطح اطمینان این قانون چقدره؟ چقدر این قانون مطمئنه؟

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

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:

$$\overbrace{\{\text{Milk}, \text{Diaper}\}}^{\text{X}} \Rightarrow \overbrace{\{\text{Beer}\}}^{\text{Y}}$$

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

$$c = \frac{\overbrace{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}^{\text{X,Y}}}{\underbrace{\sigma(\text{Milk}, \text{Diaper})}_{\text{X}}} = \frac{2}{3} = 0.67$$

Mining Association Rules(Example)

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

Support Vs Confidence

I. Support and confidence are both high.

I
A, B
A, B, C
A, B, D
A, B
A, B, C, D

⇨

$A \rightarrow B$
sup = 1
conf = 1

II. Support and confidence are both low.

II
A
B
A, C
B, C
C, D

⇨

$A \rightarrow B$
sup = 0
conf = 0

Support Vs Confidence

III. Confidence is high and support is low.

III
A, B, D
A, C, D
A, D, E
B, E, F
B, C, D, E, F
G, A

$\Rightarrow$

$G \rightarrow A$
$sup = \frac{1}{6}$
$conf = 1$

IV. Confidence is low and support is high.

It is impossible because:
$Sup \leq Conf$

$\Downarrow$

$$Conf(A \rightarrow B) = \frac{P(A,B)}{P(A)} = \frac{Sup(A \rightarrow B)}{P(A)}$$

$$P\ (A) \leq 1$$

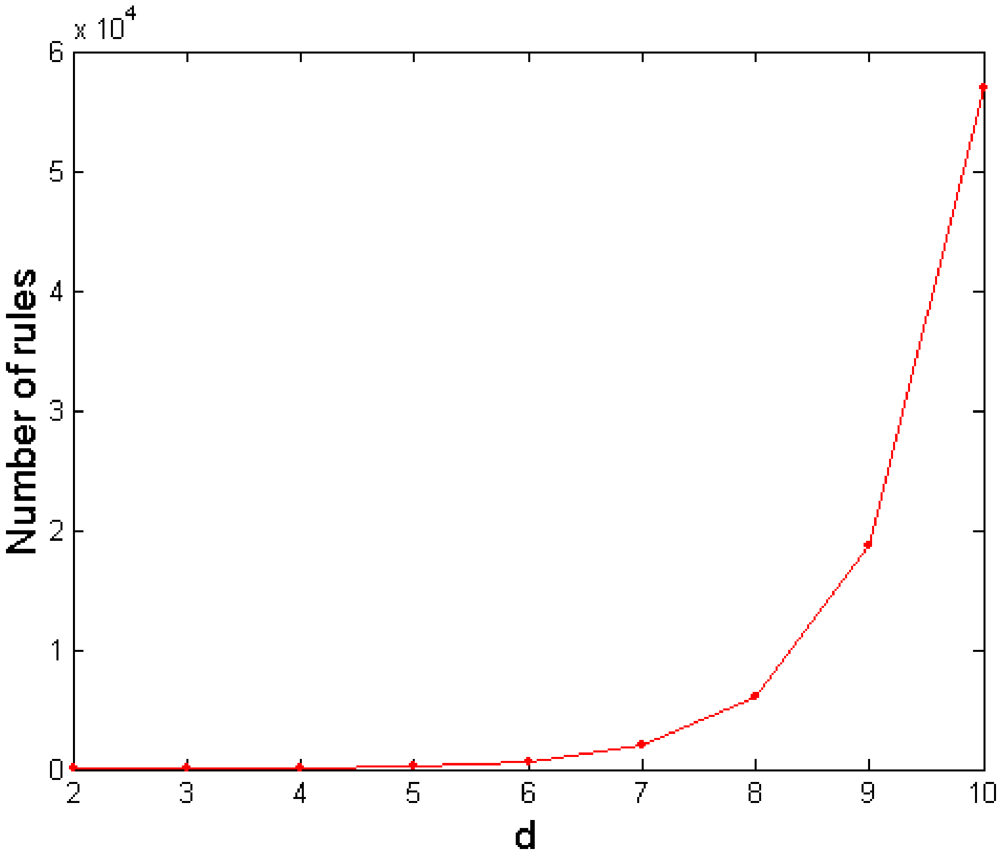
Association Rule Mining

- 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!

Computational Complexity

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



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k}\times\sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$
$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Mining Association Rules by Frequent Itemset

- Two-step approach:
  1. Frequent Itemset Generation
     – Generate all itemsets whose support ≥ 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

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 $\emptyset \rightarrow L$)

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

TID	Items
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3
Minsup = 2, minconf = %70	

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

L_3

Itemset	Sup. count
I1, I2, I3	2
I1, I2, I5	2

Itemset	Rule	Confidence
I1, I2	I1 → I2	$\text{conf} = \frac{4}{6}$ ✗
	I2 → I1	$\text{conf} = \frac{4}{7}$ ✗
I1, I3	I1 → I3	$\text{conf} = \frac{4}{6}$ ✗
	I3 → I1	$\text{conf} = \frac{4}{6}$ ✗
I2, I5	I2 → I5	$\text{conf} = \frac{2}{7}$ ✗
	I5 → I2	$\text{conf} = 1$ ✓

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Itemset	Rule	Confidence	
I1, I2, I3	I1 → I2 I3	conf $= \frac{2}{6}$	✗
	I2 → I1 I3	conf $= \frac{2}{7}$	✗
	I3 → I1 I2	conf $= \frac{2}{6}$	✗
	I1 I2 → I3	conf $= \frac{2}{4}$	✗
	I1 I3 → I2	conf $= \frac{2}{4}$	✗
	I2 I3 → I1	conf $= \frac{2}{4}$	✗
I1, I2, I5	I5 → I1 I2	conf = 1	✓
	I1 I5 → I2	conf = 1	✓
	I2 I5 → I1	conf = 1	✓

L_1

Itemset	Sup.count
I1	6
I2	7
I3	6
I4	2
I5	2

L_2

Itemset	Sup.count
I1, I2	4
I1, I3	4
I1, I5	2
I2, I3	4
I2, I4	2
I2, I5	2

Rule Generation

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

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

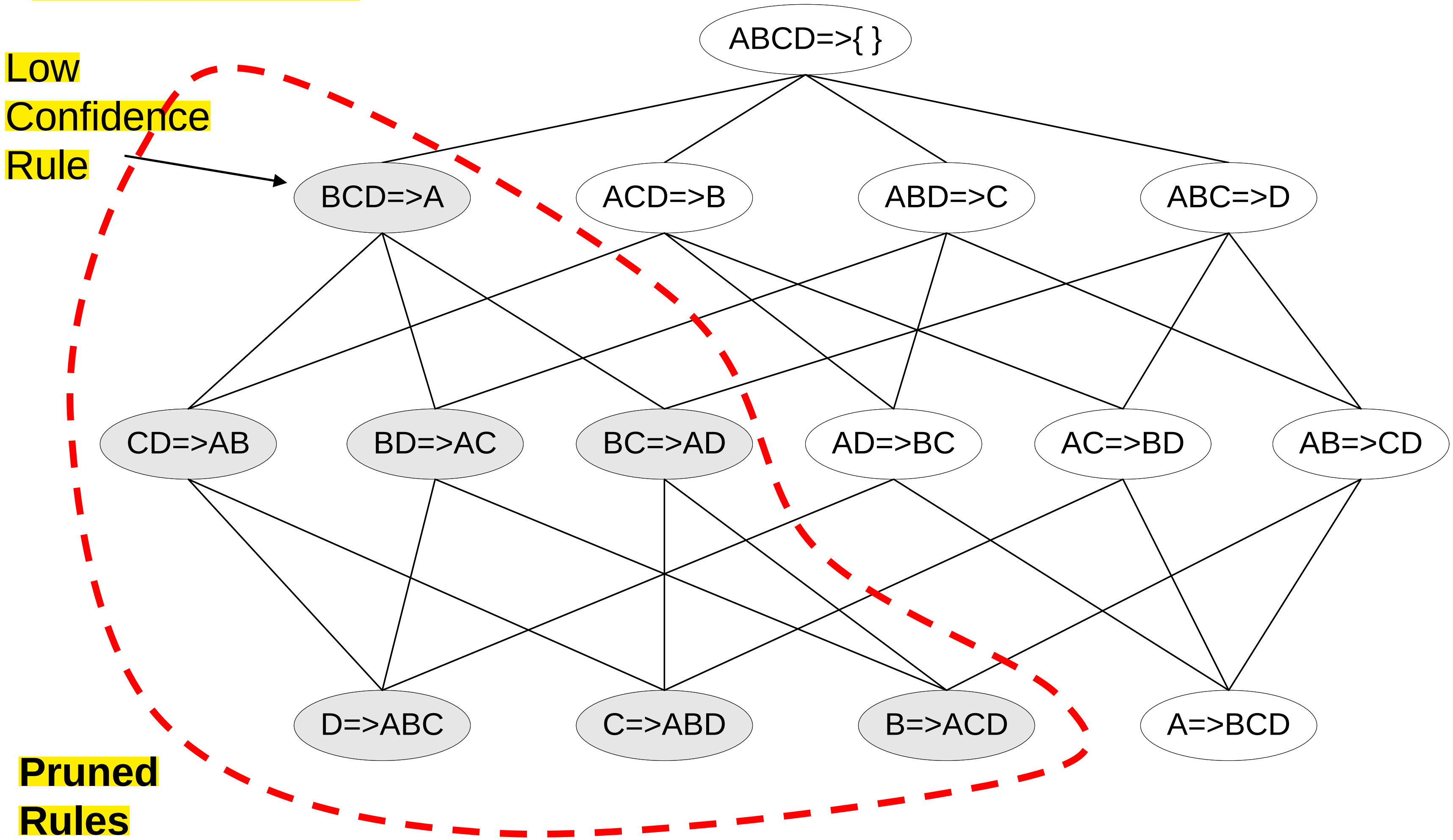
- But confidence of rules generated from the same itemset has an anti-monotone property
  - E.g., Suppose {A,B,C,D} is a frequent 4-itemset:

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

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

Rule Generation for Apriori Algorithm

Lattice of rules



Example

ID	Basketball	Cereal consumption
…	…	…
…	…	…

C \ B	YES	NO	
YES	2000	1750	3750
NO	1000	250	1250
	3000	2000	5000

Example

ID	Basketball	Cereal consumption
…	…	…
…	…	…

C \ B	YES	NO	
YES	2000	1750	3750
NO	1000	250	1250
	3000	2000	5000

Basketball → Cereal consumption

$$\text{sup} = \frac{2000}{5000} = \%\ 40$$

$$\text{conf} = \frac{2000}{3000} = \%\ 66$$

$$P\ (\text{Cereal consumption}) = \frac{3750}{5000} = \%\ 75$$

Basketball → $\overline{\text{Cereal consumption}}$

$$\text{sup} = \frac{1000}{5000} = \%\ 20$$

$$\text{conf} = \frac{1000}{3000} = \%\ 33.3$$

$$P\ (\overline{\text{Cereal consumption}}) = \%\ 25$$

Example

Is Symptom → Disease a valid rule?

S \ D	YES	NO	
YES	80	40	120
NO	20	10	30
	100	50	150

Example

Is Symptom $\rightarrow$ Disease a valid rule?

S \ D	YES	NO	
YES	80	40	120
NO	20	10	30
	100	50	150

$S \rightarrow D$
$\text{sup} = \frac{80}{15000} = \%\ 53$
$\text{conf} = \frac{80}{120} = \%\ 66$

But S and D are independent!
$P(D|S) = P(D) = 0.67$

Lift Measure

Strong Rules are not necessarily interesting.
We need more measures to evaluate rules.

$$\text{Lift}(A \rightarrow B) = \frac{P(A,B)}{P(A)P(B)} = \frac{\text{conf}\ (A \rightarrow B)}{P\ (B)} = \text{Lift}(B \rightarrow A)$$

Lift < 1	$P\ (B \mid A) < P\ (B)$	Negative Correlation
Lift = 1	$P\ (B \mid A) = P\ (B)$	Independent
Lift > 1	$P\ (B \mid A) > P\ (B)$	Positive Correlation

Lift Measure

C \ B	YES	NO	
YES	2000	1750	3750
NO	1000	250	1250
	3000	2000	5000

Basketball → Cereal consumption

$$\text{Lift} = \frac{\frac{2000}{5000}}{\frac{3000}{5000} \times \frac{3750}{5000}} = \frac{100}{3 \times 375} = 0.88$$

Basketball → $\overline{\text{Cereal consumption}}$

$$\text{Lift} = \frac{\frac{1000}{5000}}{\frac{3000}{5000} \times \frac{1250}{5000}} = \frac{500}{3 \times 125} = 1.33$$

Lift Measure

Lift measure is not null-invariant

	B	$\overline{B}$	
C	100	1000	1100
$\overline{C}$	1000	null count	
	1100		

If null count = 100000

$$\text{Lift }(B,C) = \frac{P(B,C)}{P(B)P(C)} = \frac{\frac{100}{102100}}{\frac{1100}{102100} \times \frac{1100}{102100}} = 8.44 \gg 1$$

If null count = 100

$$\text{Lift }(B,C) = \frac{P(B,C)}{P(B)P(C)} = \frac{\frac{100}{2200}}{\frac{1100}{2200} \times \frac{1100}{2200}} = 0.18 \ll 1$$

All Confidence

$$\text{All-confidence}(A,B) = \frac{P(A,B)}{\max(P(A),P(B))}$$

$$0 \leq \text{All-confidence} \leq 1$$

Other Measure

symbol	measure	range	formula
ϕ	ϕ-coefficient	-1…1	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 … 1	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 … 1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 … 1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-Shapiro's	-0.25 … 0.25	$P(A,B)-P(A)P(B)$
F	Certainty factor	-1 … 1	$\max(\frac{P(B\mid A)-P(B)}{1-P(B)}, \frac{P(A\mid B)-P(A)}{1-P(A)})$
AV	added value	-0.5 … 1	$\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
K	Klosgen's Q	-0.33 … 0.38	$\sqrt{P(A,B)}\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
g	Goodman-kruskal's	0 … 1	$\frac{\Sigma_j \max_k P(A_j,B_k)+\Sigma_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 … 1	$\frac{\Sigma_i \Sigma_j P(A_i,B_j) \log \frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\Sigma_i P(A_i) \log P(A_i) \log P(A_i), -\Sigma_i P(B_i) \log P(B_i) \log P(B_i))}$
J	J-Measure	0 … 1	$\max(P(A,B)\log(\frac{P(B\mid A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\mid A)}{P(\overline{B})}))$ $P(A,B)\log(\frac{P(A\mid B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\mid B)}{P(\overline{A})})$
G	Gini index	0 … 1	$\max(P(A)[P(B\mid A)^2+P(\overline{B}\mid A)^2]+P(\overline{A}[P(B\mid \overline{A})^2+P(\overline{B}\mid\overline{A})^2]-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\mid B)^2+P(\overline{A}\mid B)^2]+P(\overline{B}[P(A\mid\overline{B})^2+P(\overline{A}\mid\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 … 1	$P(A,B)$
c	confidence	0 … 1	$max(P(B\mid A), P(A\mid B))$
L	Laplace	0 … 1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 … 1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 … 1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 … 1	$\frac{P(A,B)}{\max(P(A),P(B))}$
o	odds ratio	0 … ∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 … ∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 … ∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 … ∞	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
χ^2	χ^2	0 … ∞	$\Sigma_i \frac{(P(A_i)-E_i)^2}{E_i}$