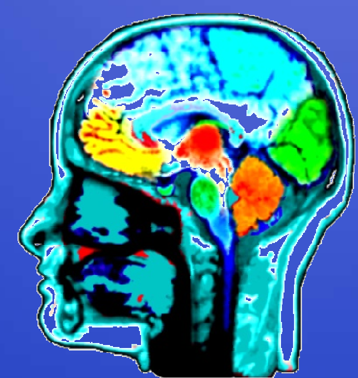
Introduction To Data Mining

Isfahan University of Technology (IUT)



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

شناسایی الگوهای پرتکرار:

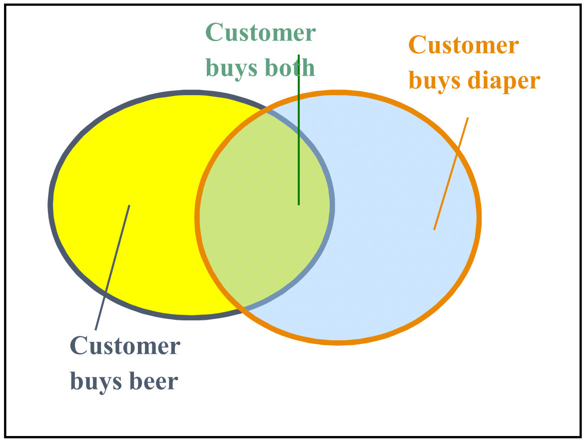
وقتی مثلا شیر رو خرید کیک هم بخره ینی اینارو کنار هم بذاریم

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 = $\{x_1, \ldots, x_k\}$: a Itemset with k items
- ***(absolute) support*, or, *support count* of X:** Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

دیتایی که جمع اوری میشه توی این فضا یک دنباله تراکنش است
Itemset: یک مجموعه ای که از چندتا ایتم تشکیل شده
k-itemset: از k تا المان متمایز ساخته شده
support: تعداد رخداد یک ایتم ست است
چه زمانی میگیم ایتم ست پرتکرار است؟ وقتی که از یک minsup بیشتر باشه

Basic Concepts: Frequent Patterns

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

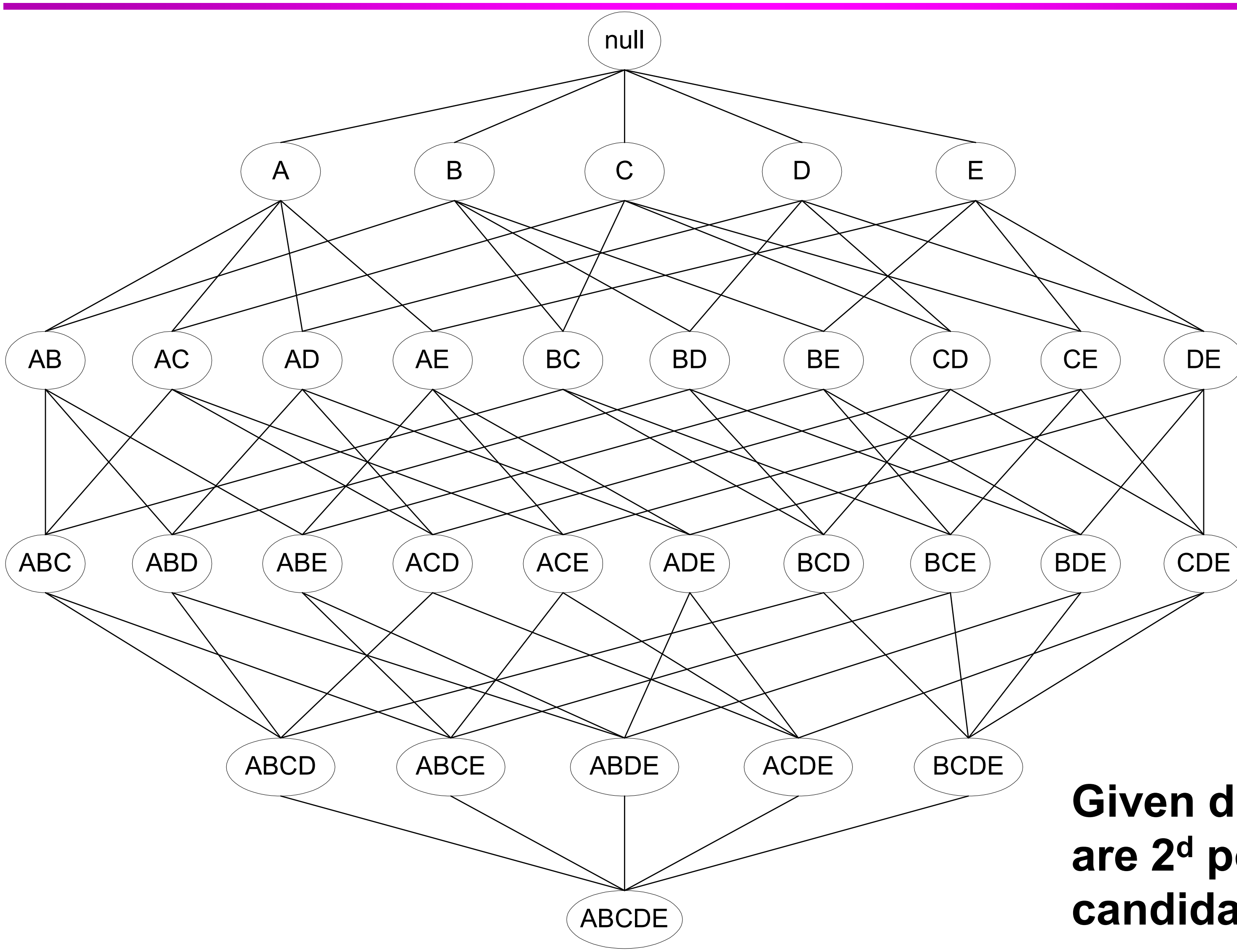
همزمان C , B توی چندتا تراکنش وجود دارن

1-itemset	2-itemset	3-itemset
Support count ({C}) = 3 Support ratio ({C})= 3/4	Support count ({B, C}) = 2 Support ratio ({B,C})= 2/4	Support count ({B, C, E}) = 2 Support ratio({B,C,E}) = 2/4

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

اگر minsup باشه 0.7 بگو frequent ایتم چندتا است؟
c میشه چون c تعداد تکرارهاش بیشتر از 70 درصد است

Frequent Itemset Generation



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

پرتکرارها توی این ایتم ست چیا است؟ با توجه به این که مشخص نکرده که ایتم ست های 1 یا 2 یا 3 یا 4 یا 5 تایی می خواد ما باید همه حالت ها رو در نظر بگیریم ینی فضای حالت میشه 2 به توان d

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

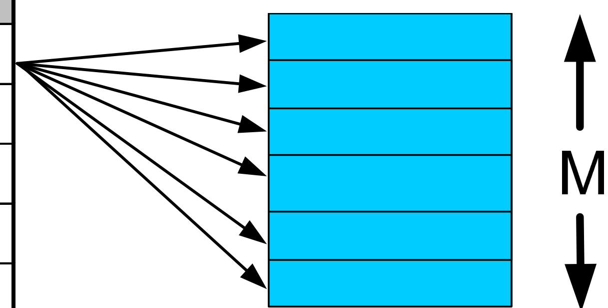
Transactions

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

N

w

List of Candidates

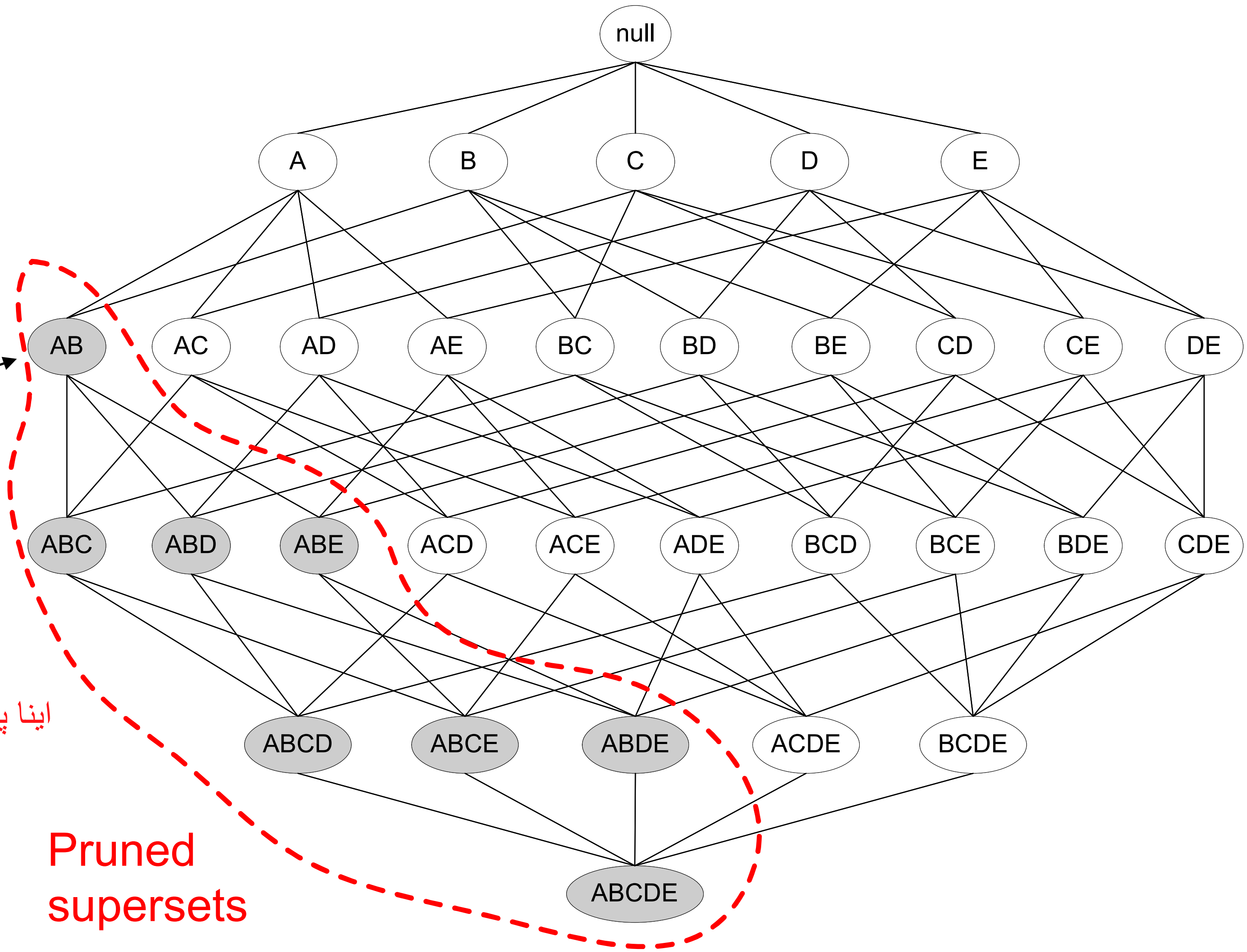


  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

رویکرد بی رحمانه:

- هر مجموعه اقلام در شبکه یک مجموعه اقلام مکرر نامزد است
– با اسکن پایگاه داده حمایت هر نامزد را بشمارید
- هر تراکنش را با هر نامزد مطابقت دهید
- پیچیدگی ~ O(NMw) => گران از M = 2d !!!

Illustrating Apriori Principle



Found to be Infrequent

اینا پرتکرار نیستن

Pruned supersets

ایده پشت Apriori:
وقتی داریم راجع به الگوهای پرتکرار می گردیم
اگه به یک ایتمی ستی رسیدیم به نام ایتم ست a و رفتی اینو شمردی و دیدی این ایتم ست اصلا پرتکرار نیست ینی از اون حداقلی که بهمون دادن کمتر است دیگه نیازی نیست زیرمجموعه a , b رو بریم بگردیم و نیاز نیست این کاندیدها بررسی بشه و می تونیم این بخشو هرس کنیم و اینجوری فضای جستجو کاهش پیدا میکنه

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,

$6 + 6 + 4 = 16$

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

اگر بروت فرس می خواستیم بریم:

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,

$6 + 6 + 4 = 16$

طبق الگوریتم Apriori:
از 6 تا 4 تا داریم الان

الگوریتم Apriori همین عملیات هرس کردن رو برای هر گام انجام میده الان ما فقط برای ایتم ست 1 دونه ای این کار رو کردیم

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)

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

(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

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

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

Triplets (3-itemsets)

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

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,

$$6 + 6 + 4 = 16$$

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,

$$ 6 + 6 + 4 = 16 $$

$$ 6 + 6 + 1 = 13 $$

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}

الگوریتم Apriori چیه؟
برای اینکه این الگوریتم اجرا بشه دوتا مجموعه تعریف میکنه:
پرتکرار: Fk --> این میشه k ایتم ست پرتکرار
Lk: ینی مجموعه ای که ما می خوایم سرچ بکنیم که ایا اینها پرتکرار هستن یا نه --> k ایتم ستی که کاندید هستن و ما باید این ها رو بررسی بکنیم

این الگوریتم سه قسمت داره:
تنظیمات اولیه
فرایند تکراری
جواب نهایی رو گزارش میده

رویه الگوریتم:
اول سرچ میکنه ببینه تک ایتم های پرتکرار چیا هستن این میشه مجموعه f1
بعد ایتم های ست های دوتایی: اول باید ایتم ست های مرتبه بالاتر رو کاندیدهاش رو ایجاد میکنه و بعد هرس میکنه و بعد شمارش و بعد حذف کاندید های که کم هستن
و به همین صورت می ره جلو
------*
1. Candidate Generation: Lk+1 را از Fk تولید کنید
2. هرس کاندید: مجموعههای اقلام کاندید در Lk+1 حاوی زیرمجموعههای طول k که نادر هستند را هرس کنید.
3. شمارش پشتیبانی: با اسکن DB، حمایت هر نامزد را در L k+1 بشمارید
4. حذف نامزدها: نامزدهایی را در Lk+1 که نادر هستند حذف کنید و فقط آنهایی را که مکرر هستند باقی بگذارید => Fk+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.

مسئله Generation کردن:

سه تا روش برای جنریت کردن وجود داره:

1- Brute-force که اصلا راه خوبی نیست

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}

الان فقط یک کاندید داریم
که همین باید سرچ بشود

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.

هرس کردن:
بدون دیتاست اولیه

توی ایتم ست 2 تایی می بینیم beer با bread اصلا با هم نیومدن پس توی ایتم ست 3 تایی اینا پرتکرار نیستن
و...

روش دوم: اینه

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

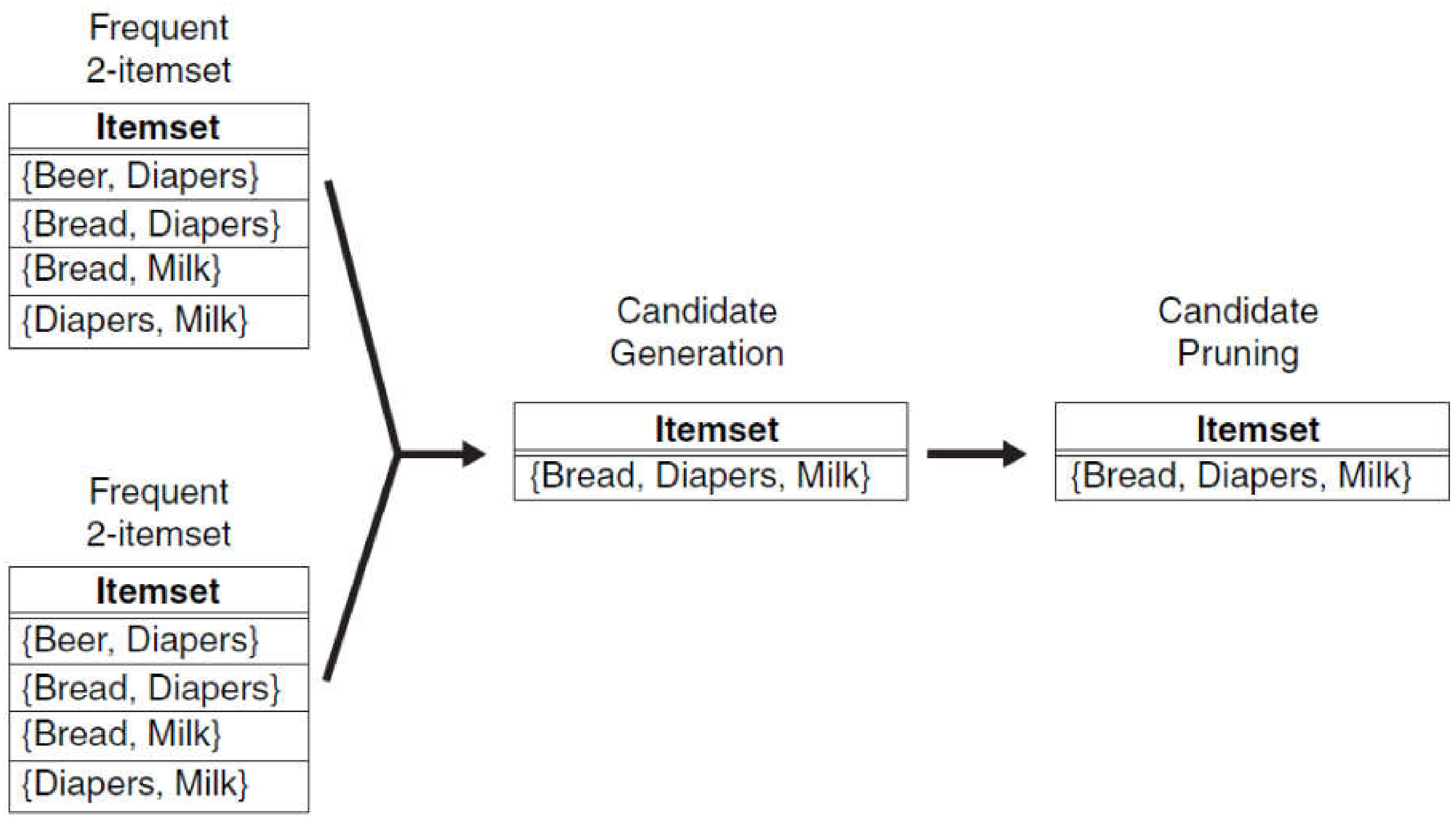


Figure 5.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

روش سوم:

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 (Self-Join)

- 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

  می تونیم جوین بکنیم به شرطی که
  پیشوندهاش مثل هم باشه

  - 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

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

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

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 $F_{k-1} \times F_{k-1}$ 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}

پشتیبانی از شمارش مجموعه اقلام نامزد
پایگاه داده تراکنش ها را اسکن کنید تا حمایت هر مجموعه اقلام نامزد را مشخص کنید
- باید هر مجموعه اقلام نامزد را در برابر هر تراکنش که یک عملیات گران است، مطابقت دهد

Support Counting of Candidate Itemsets

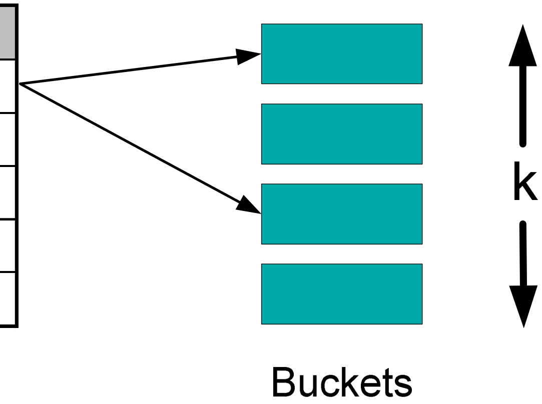
- To reduce number of comparisons, store the candidate itemsets in a hash structure
  - Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets

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

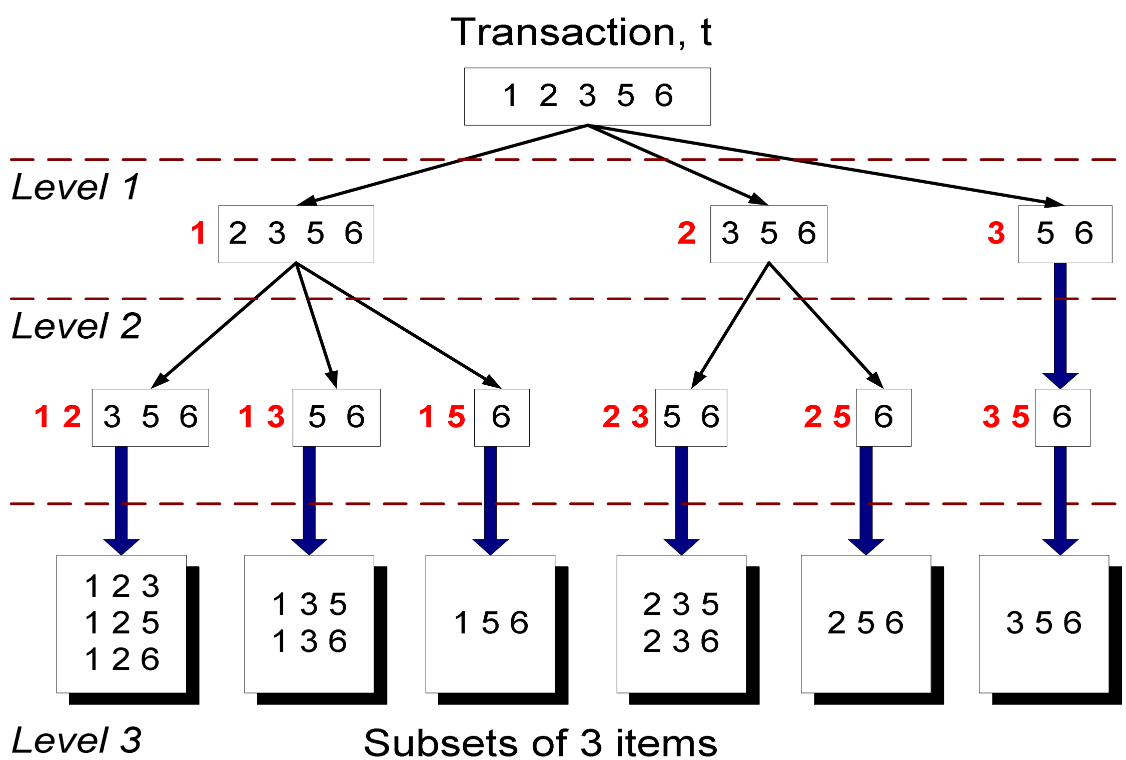


Support Counting: An Example

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)?



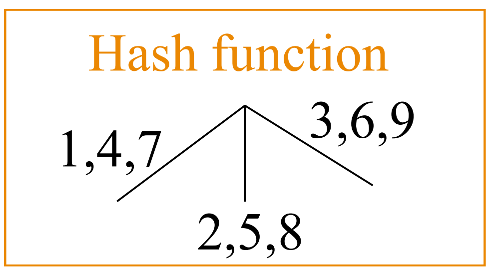
Support Counting Using a Hash Tree

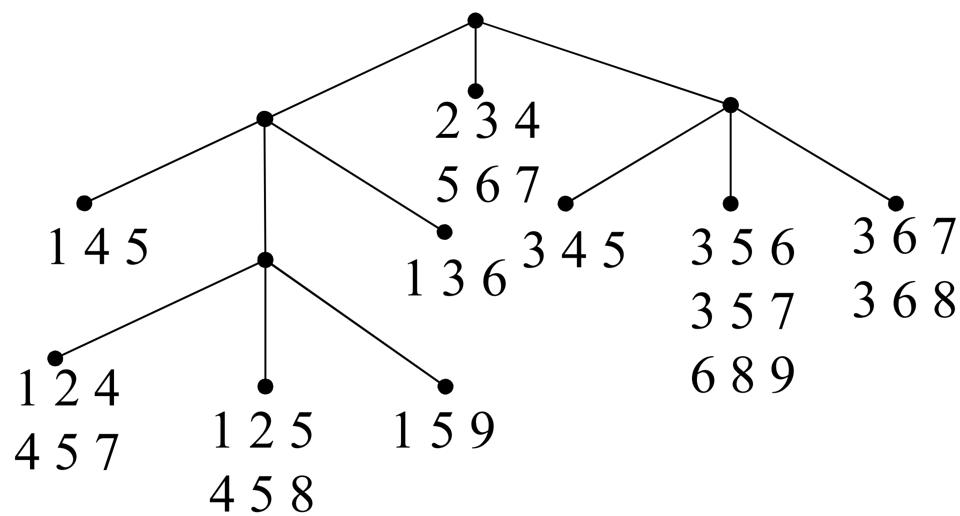
Suppose you have 15 candidate itemsets of length 3:

{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}

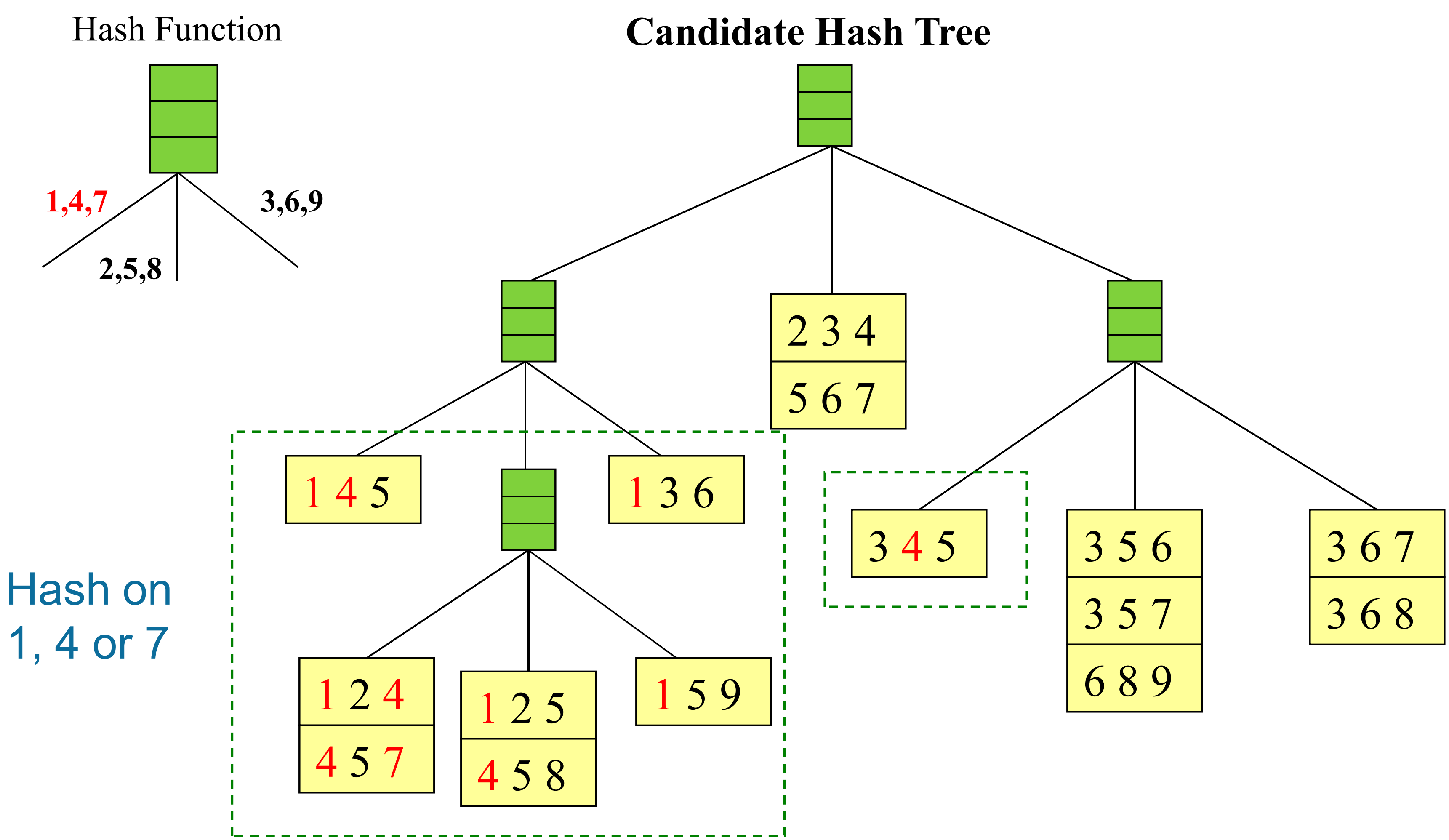
You need:

- **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)**

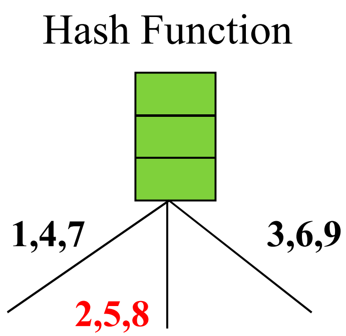


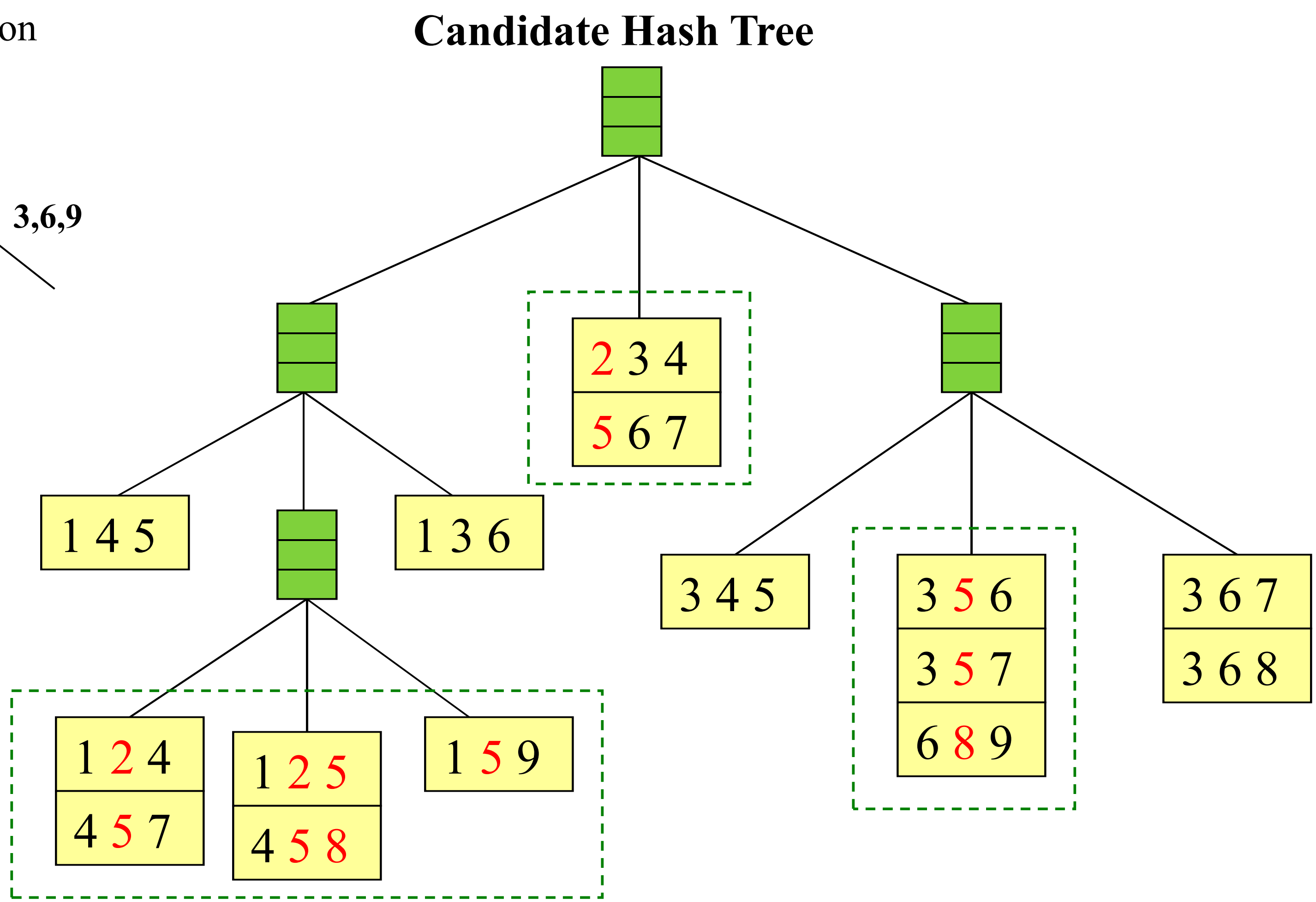


Support Counting Using a Hash Tree



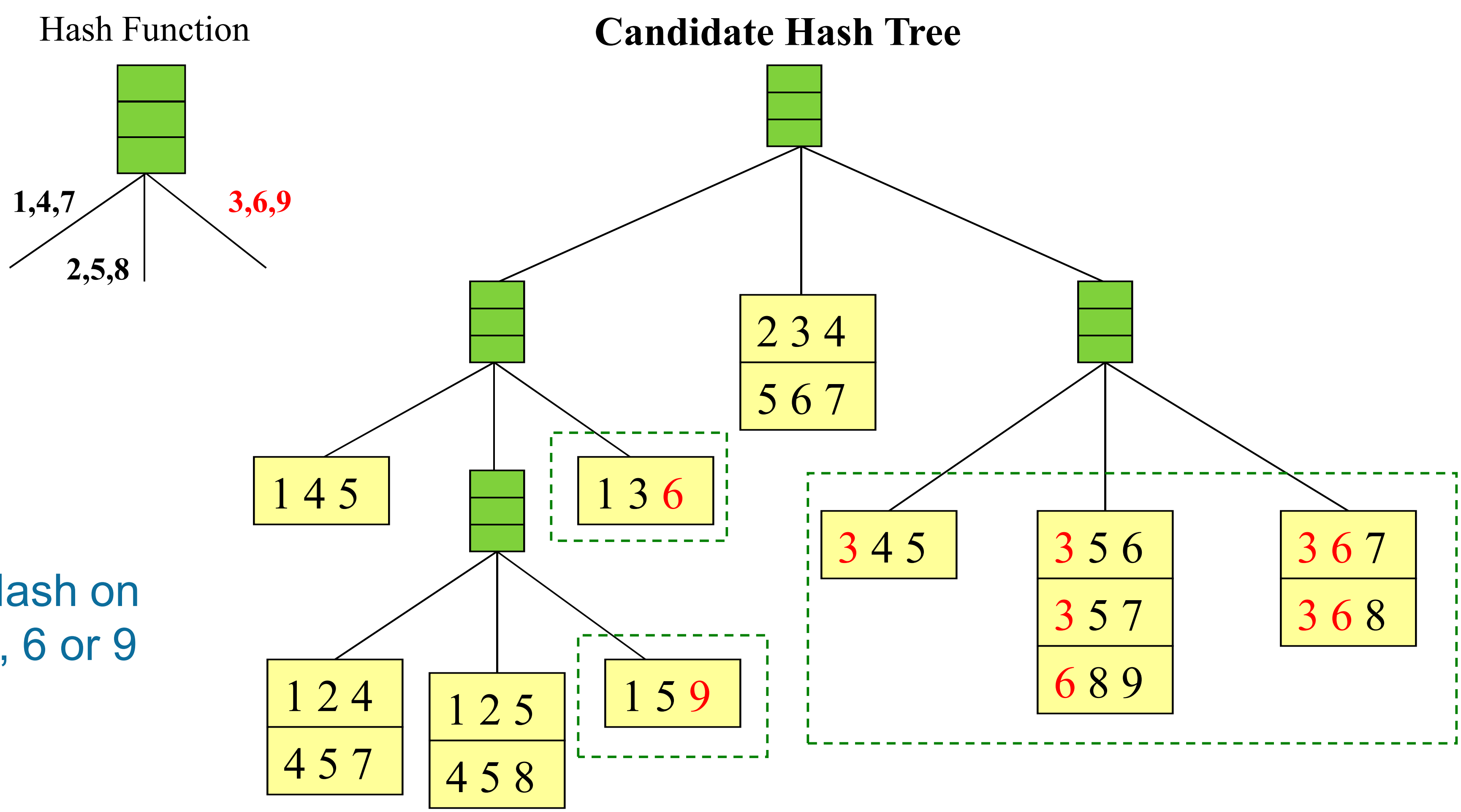
Support Counting Using a Hash Tree





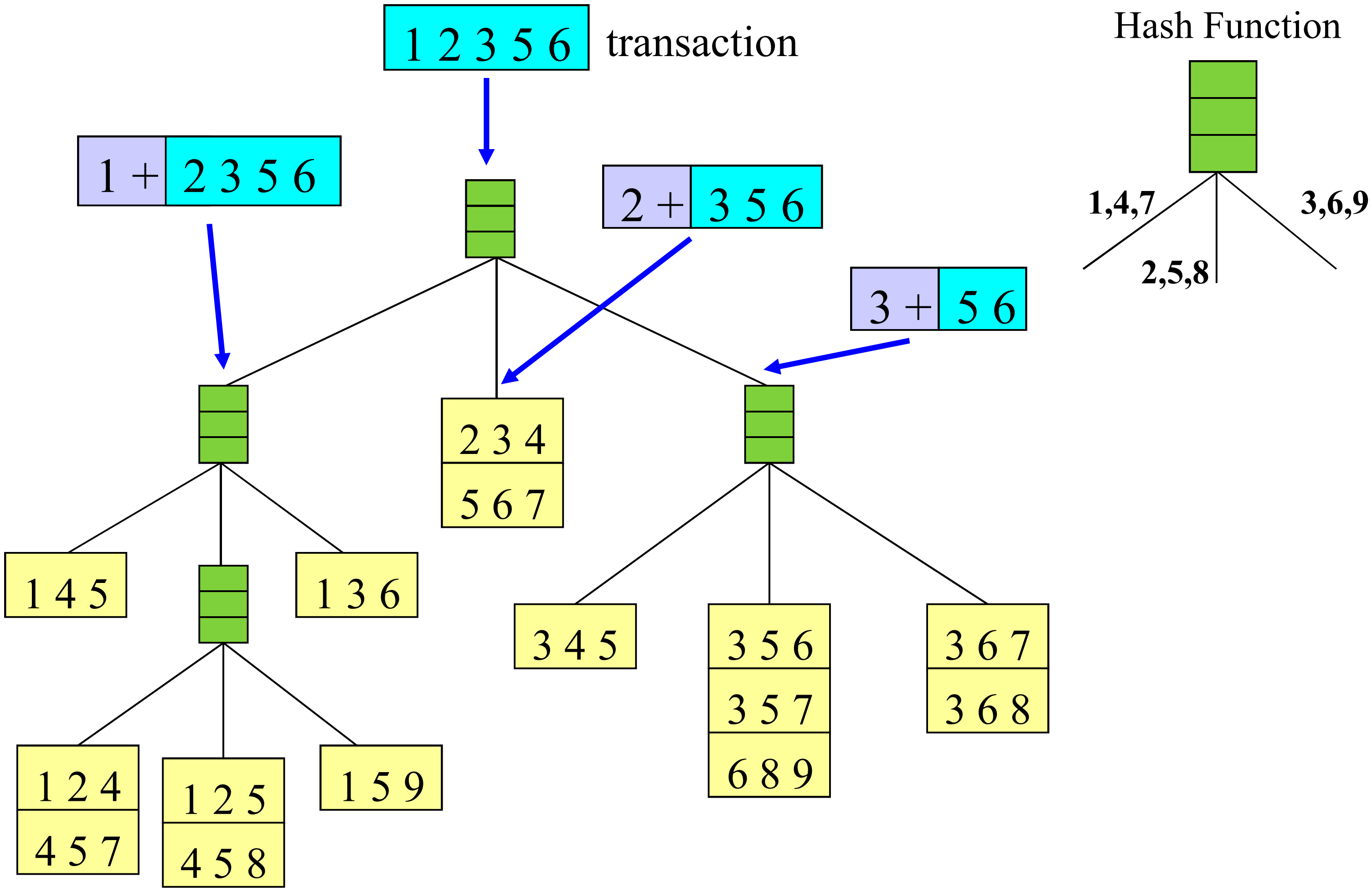
Hash on
2, 5 or 8

Support Counting Using a Hash Tree

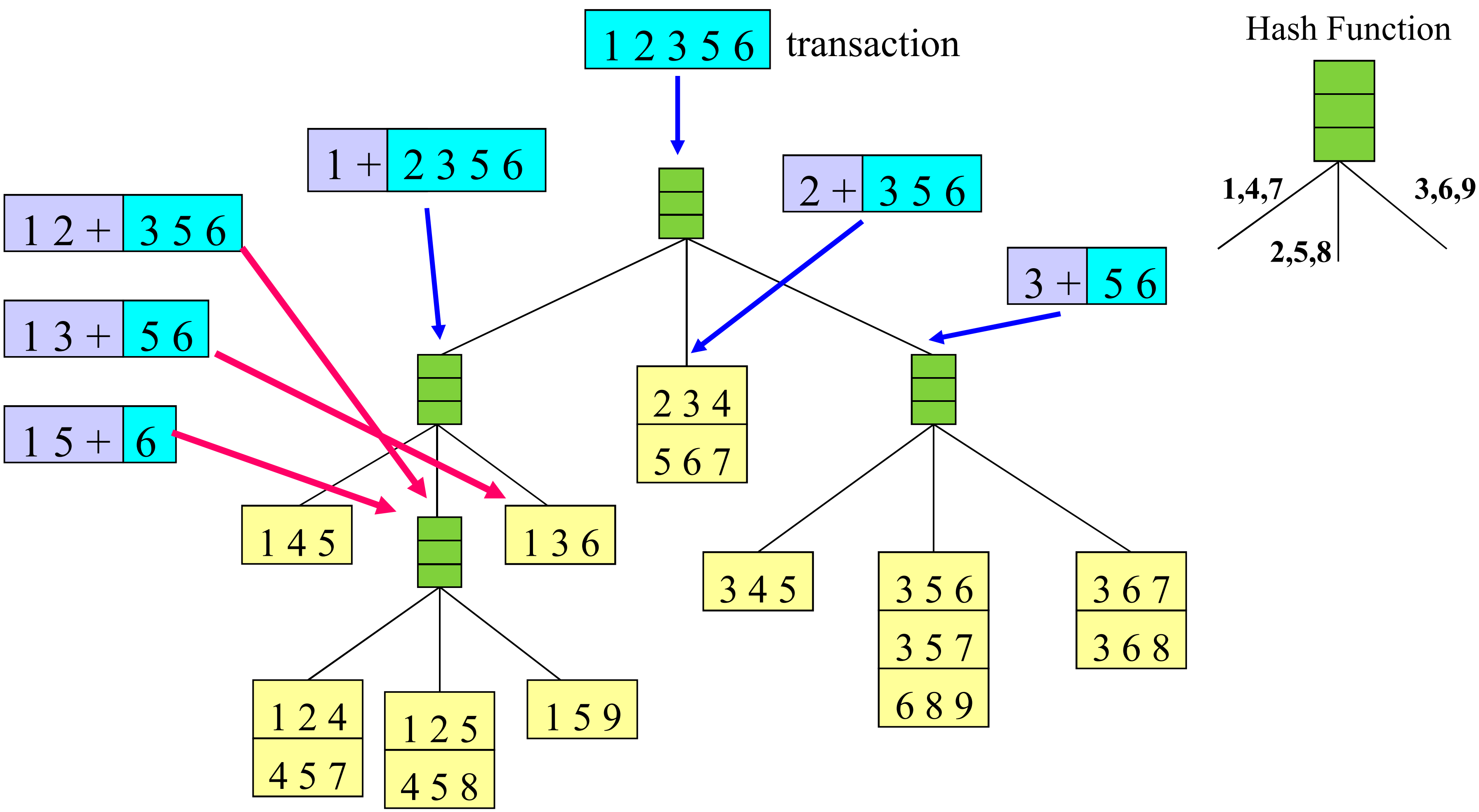


Hash on
3, 6 or 9

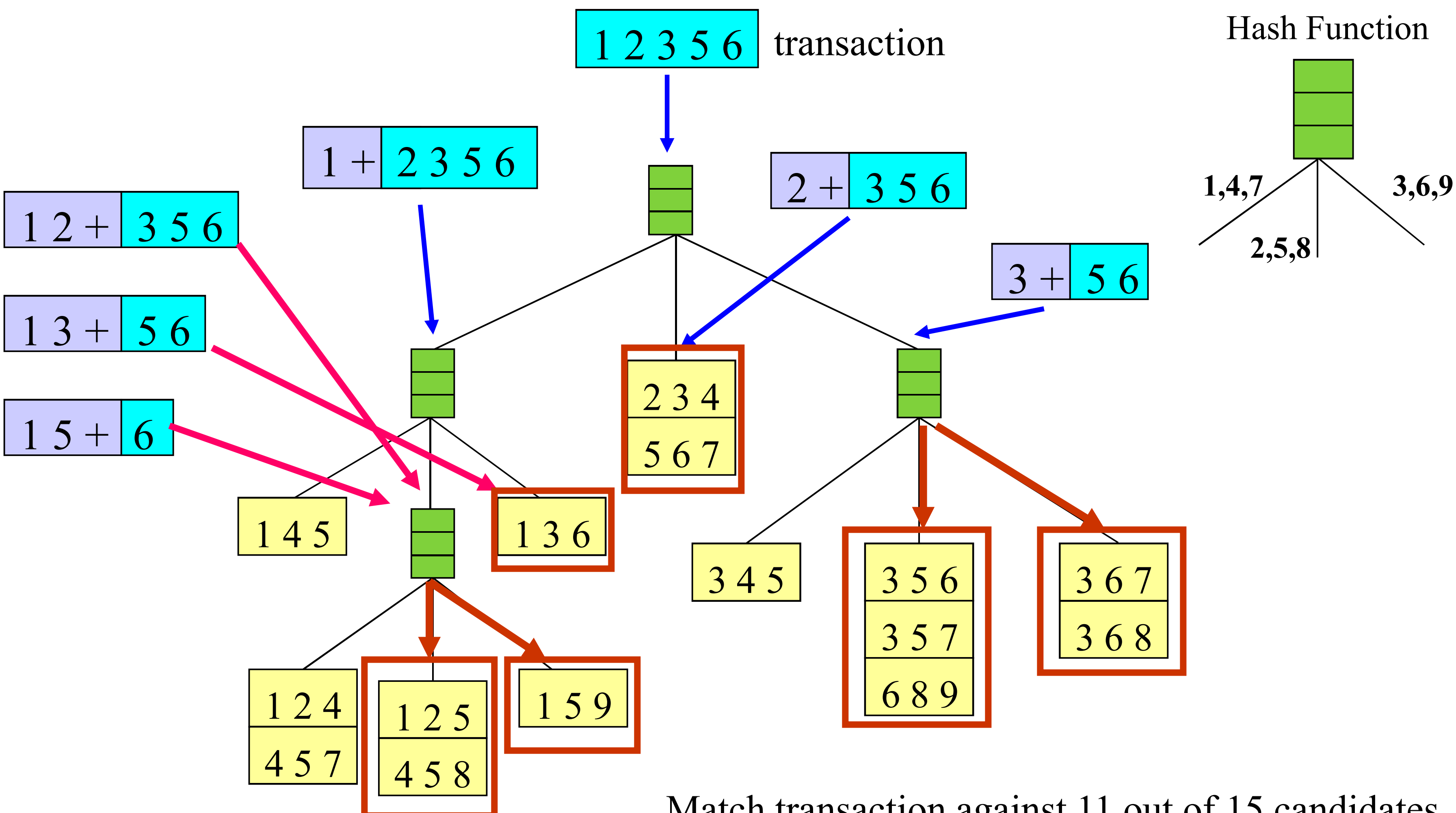
Support Counting Using a Hash Tree



Support Counting Using a Hash Tree



Support Counting Using a Hash Tree



Match transaction against 11 out of 15 candidates

ASSOCIATION RULES

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!

قوانین انجمنی:
به دنبال این قوانین هستیم

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having

Strong Rule

  – support ≥ *minsup* threshold
  – confidence ≥ *minconf* threshold

برای استخراج این قوانین:

minsup: این قانونی که میدیم باید قانون پرتکراری باشه

اگر مقدم قانون رو داشتیم ؟؟

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

استخراج قوانین پرتکرار از همون ایتم های پرتکرار شروع میشه
اول باید ایتم های پرتکرار رو ایجاد بکنیم
support: از کل قوانینی که می تونه وجود داشته باشه از کل این سه تایی یا 4 تایی ها یا .. اینا چند بار توی کل تراکنش بودن
confidence: از کل دفعاتی که x اومده یا y اومده ؟؟؟؟

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:

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

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

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

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

-

چندتا حالت:

هم s بزرگه و هم c بزرگه

هم s کوچیکه و هم c کوچیکه

c بزرگه و s کوچیکه

c کوچیکه و s بزرگه: این حالت ایا پیش میاد؟

نه این حالت هیچ وقت پیش نمیاد چرا؟ چون s همیشه مقدار مثبتی داره و همیشه بین صفر و یک است و مقدار کسر c همیشه از مخرج بیشتره پس c از s همیشه بزرگتر است

Support Vs Confidence

I. Support and confidence are both high.

II. Support and confidence are both low.

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

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

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

⇒

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

IV. Confidence is low and support is high.

It is impossible because:
$\text{Sup} \leq \text{Conf}$

⇓

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

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

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!

-

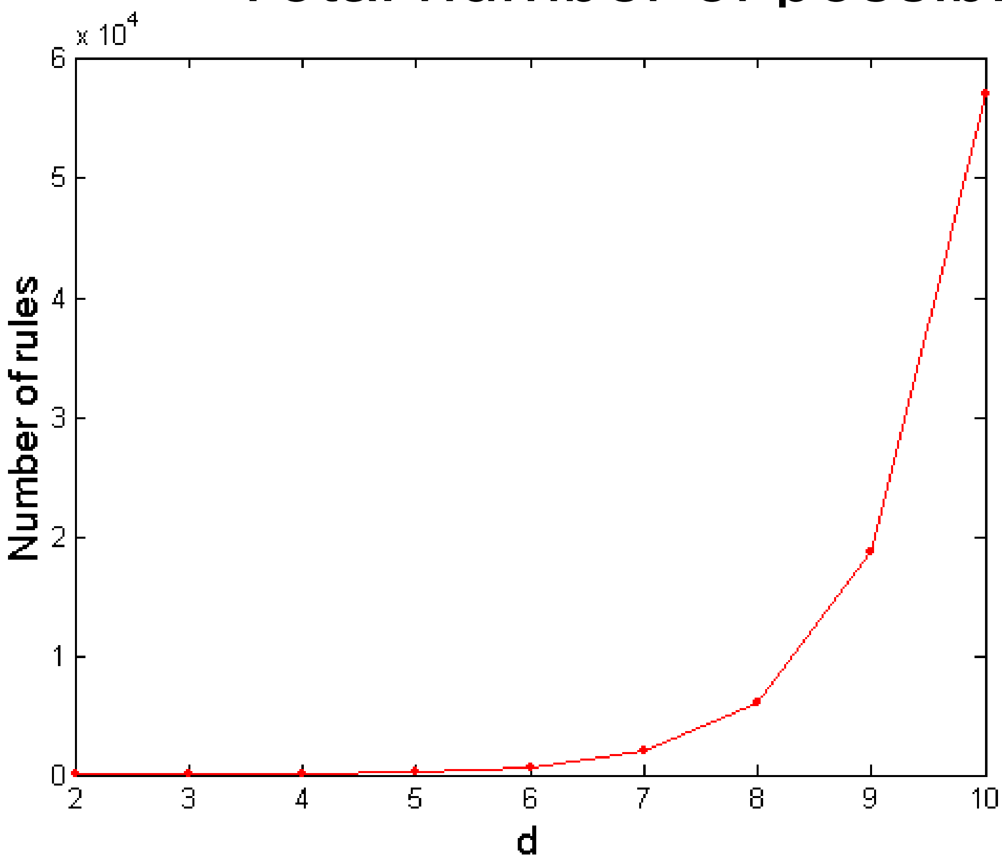
چطوری قوانین انجمنی رو استخراج بکنیم؟

Brute-force-1: ینی همه قانون هایی که می تونه توسط این سیمبل ها تولید بشه رو بچینیم و بعد شروع بکنیم به شمار s , c که این ساده ترین راه کار است و غیر ممکن ترین

تعداد قانون ها اینجا خیلی زیاده میشه و این راهکار اصلا شدنی نیست و اردر زمانیش خیلی بالا میشه

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

-

چطوری قوانین انجمنی رو استخراج بکنیم؟

2- قوانین قوی رو استخراج بکنیم

اول کار بیایم ایتم ست های پرتکرار رو پیدا بکنیم و یه تعداد الگو می مونن و یه تعداد حذف میشن و بعد از این الگوهایی که می مونن بریم دنبال قانون پیدا کردن و استخراج کردن بگردیم

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 $\emptyset \rightarrow L$)

سراغ این حالت نمی ریم که تهی بده یه چیزی یا یه چیزی بده تهی

L_1

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

L_3

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

دنبال قوانینی هستیم که minconfشون 70 درصده و minsupشون 2
حالا از این سه تا می خوایم قانون استخراج بکنیم --> از این ایتم ست های تکی
که قانون استخراج نمیشه پس کاری به اینا نداریم

I1, I2	$I1 \rightarrow I2$	$conf = \frac{4}{6}$
	$I2 \rightarrow I1$	$conf = \frac{4}{7}$
I1, I3	$I1 \rightarrow I3$	$conf = \frac{4}{6}$
	$I3 \rightarrow I1$	$conf = \frac{4}{6}$
I2, I5	$I2 \rightarrow I5$	$conf = \frac{2}{7}$
	$I5 \rightarrow I2$	$conf = 1$

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

برای ایتم ست های دوتایی 12 تا قانون داریم
و برای ایتم ست های 3 تایی هم 12 تا قانون داریم
پس در کل 24 تا قانون رو باید تست بکنیم و برای
هر قانون باید c رو حساب بکنیم

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

تفاوت اینا:
همشون صورتشون یکی است و تفاوت توی مخرج هاست
سه تایی پایین مخرجشون یکی است و بالایی ها دوتایی

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

---

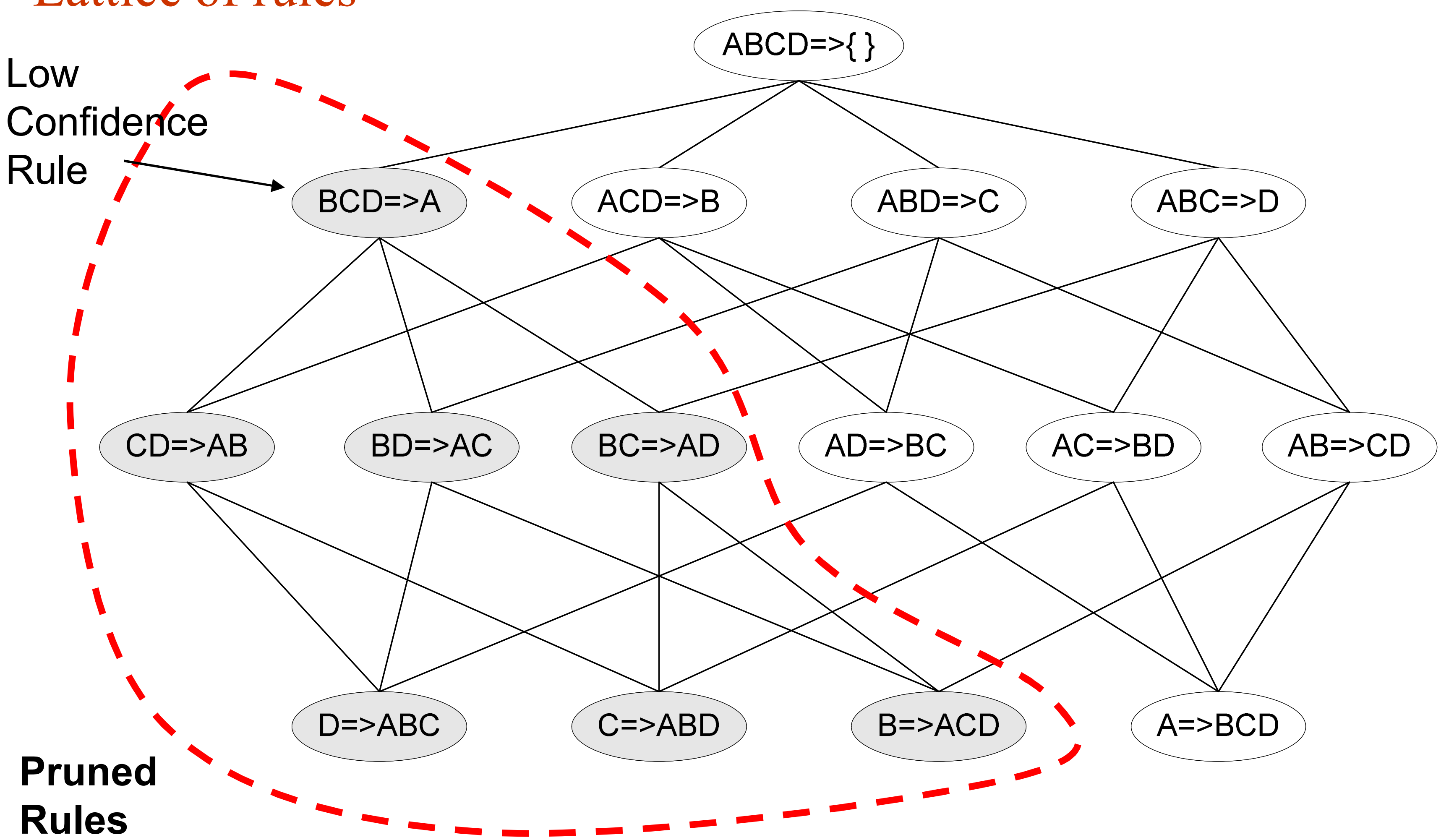
۲) قانون ها را از روی الگوهای پرتکرار تولید می کنیم. هرچی تعداد متغیرهای A بیشتر شود $P(A)$ کمتر می شود

15 $\#sym$ in $A \uparrow$ $P(A) \downarrow \Rightarrow con(A \to B) \uparrow$ پس اگر $C(ABC \to D)$ خوب نبود نیازی نیست

$C(AB \to CD)$ و $C(A \to BCD)$ را حساب کنیم. خود به خود اونا حذف میشن. اسلاید ۱۵۳:

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

غلات

بسكتبال

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

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

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

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

از این جدول می خوایم یکسری قانون استخراج بکنیم:
ایا بسکتبال بازی کردن نتیجه می دهد که طرف غلات مصرف میکنه یا برعکسش

...

4 تا قانون می تونیم بگیم

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 → 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

مژر Lift --> مشکل قبلی رو نداره
ینی زمانی که داریم قوانین قوی رو استخراج می کنیم فقط نیا به s , c نگاه بکن بیا به Lift هم نگاه کن --> این میاد بحث استقلال رو می سنجه

اگر Lift =1 باشه --> در این حالت a , b مستقل از هم هستند

این Lift هیچ ربطی به جهت قانون نداره ینی A -> B or B-> A هیچ ربطی به این جهت نداره

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		

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

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

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

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

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

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

If null count = 100000: $\text{All-conf}(B,C) = \dfrac{\frac{100}{102100}}{\max(\frac{1100}{102100}, \frac{1100}{102100})} = \dfrac{1}{11}$

If null count = 100: $\text{All-conf}(B,C) = \dfrac{\frac{100}{2200}}{\max(\frac{1100}{2200}, \frac{1100}{2200})} = \dfrac{1}{11}$

$$all\_conf(A,B) = \frac{sup(A \cup B)}{max\{sup(A), sup(B)\}} = min\{P(A|B), P(B|A)\},$$

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…∞	$\Sigma_i\frac{(P(A_i)-E_i)^2}{E_i}$