داده کاوی

مفاهیم و تکنیک ها

– فصل 6 –

کاوش الگوهای مکرر، مشارکت ها و همبستگی ها: مفاهیم پایه و روش ها

- مفاهیم اولیه
- روش های کاوش الگوهای مکرر
- چه الگوهایی جذاب هستند؟ (روش های ارزیابی الگوها)
- خلاصه

تحلیل الگو چیست؟

- الگوی پرتکرار: یک الگو(یک مجموعه از اقلام، زیرتوالی، زیرساختارو غیره) که بصورت مکرر در یک مجموعه داده رخ میدهد.
  - مجموعه اقلام: مثل خرید توام نان وشیر در بسیاری از تراکنش ها
  - زیرتوالی ها: مثلا معمولا بعد از دوربین کارت حافظه خریده می شود.
  - زیرساختارها: مانند زیر گراف یا زیر درخت
  - ...
- نخستین بار توسط Agrawal, Imielinski, and Swami [AIS93] درزمینه مجموعه آیتم های پرتکرار (frequent itemsets) و(association rule mining) پیشنهاد شد.

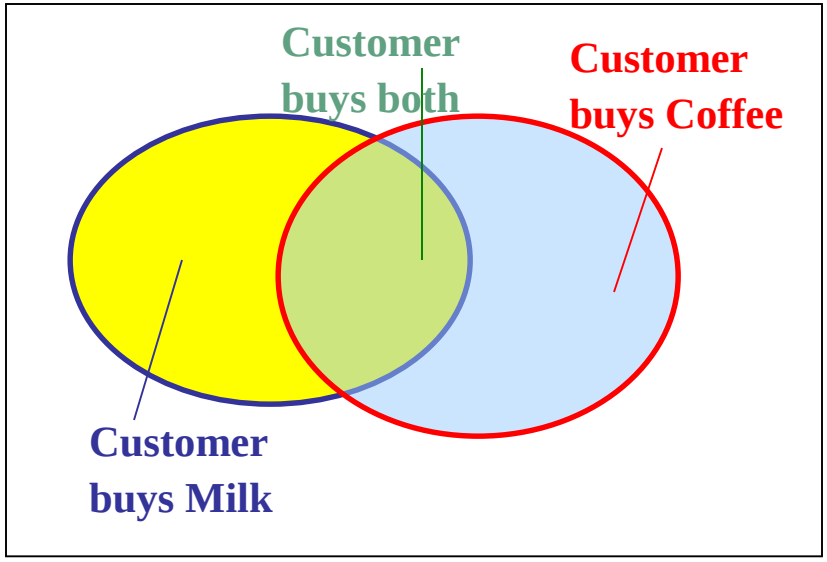
- انگیزه: پیدا کردن نظم ذاتی در داده ها
  - چه محصولاتی معمولا با هم خریده می شوند؟
  - خرید بعدی بعد از خریدن یک کامپیوتر چیست؟
  - چه انواعی از DNA به این دارو حساس است؟
  - آیا می توانیم بصورت اتوماتیک اسناد وب را دسته بندی کنیم؟
- کاربردها:
  - تحلیل سبد خرید، بازاریابی، طراحی کاتالوگ، آنالیز سلسله عملیات فروش، تحلیل جریان کلیک در وب، تحلیل توالی DNA

چرا کاوش الگو مهم است؟

- الگوهای مکرر یک ویژگی ذاتی و مهم مجموعه داده ها است.
- اساس بسیاری از عملیات مهم در داده کاوی است
  - تحلیل قواعد انجمنی، همبستگی و علیت
  - الگوهای متوالی، ساختاری (به عنوان مثال، زیر گراف)
  - تجزیه و تحلیل الگوهای مکانی، چند رسانه ای، سری زمانی، و جریان داده ها
  - دسته بندی: تحلیل الگوی مکرر متمایزکننده
  - آنالیز خوشه ای: خوشه بندی بر اساس الگوهای مکرر
  - انبارسازی داده: iceberg cube و cube-gradient
  - فشرده سازی داده های معنایی
  - کاربردهای گسترده

مفاهیم پایه: الگوهای پرتکرار

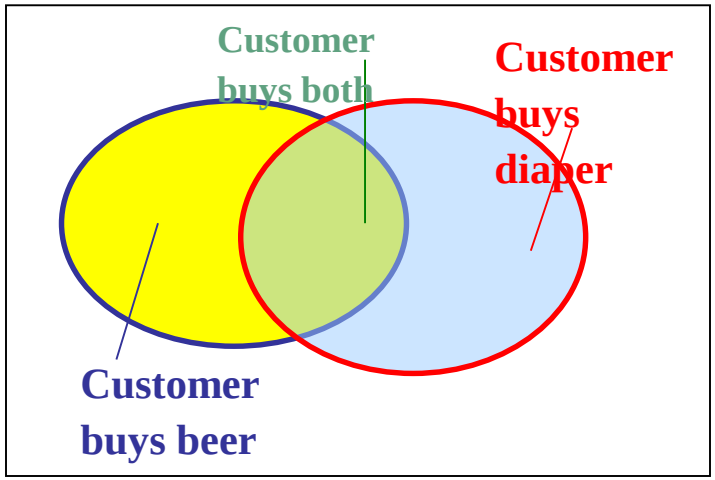
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- مجموعه اقلام: مجموعه ای از یک یا چند آیتم (مثلا کالا)
- k-itemset: مجموعه اقلام شامل K آیتم
- $X = \{x_1, \ldots, x_k\}$
- *(absolute) support* یا *support count*: تعداد تکرار یک مجموعه آیتم
- *(relative) support*: *درصد تراکنش های حاوی X یا احتمال اینکه یک تراکنش حاوی X باشد.*
- یک itemset پرتکرار است اگر support آن از حد *minsup* پایین تر نباشد.

مفاهیم پایه: قوانین انجمنی

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



- پیدا کردن همه قوانین $X \rightarrow Y$ که دو مقدار آستانه (min sup, min conf) را ارضا می کنند.
  - Support به معنی احتمال اینکه تراکنش حاوی $X \cup Y$ باشد.
  - confidenceاحتمال شرطی که یک تراکنش حاوی X حاوی Y هم باشد.
  - فرض کنید

minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

Association rules: (many more!)

- *Beer* → *Diaper* (60%, 100%)
- *Diaper* → *Beer* (60%, 75%)

Max-Patterns و Closed Patterns

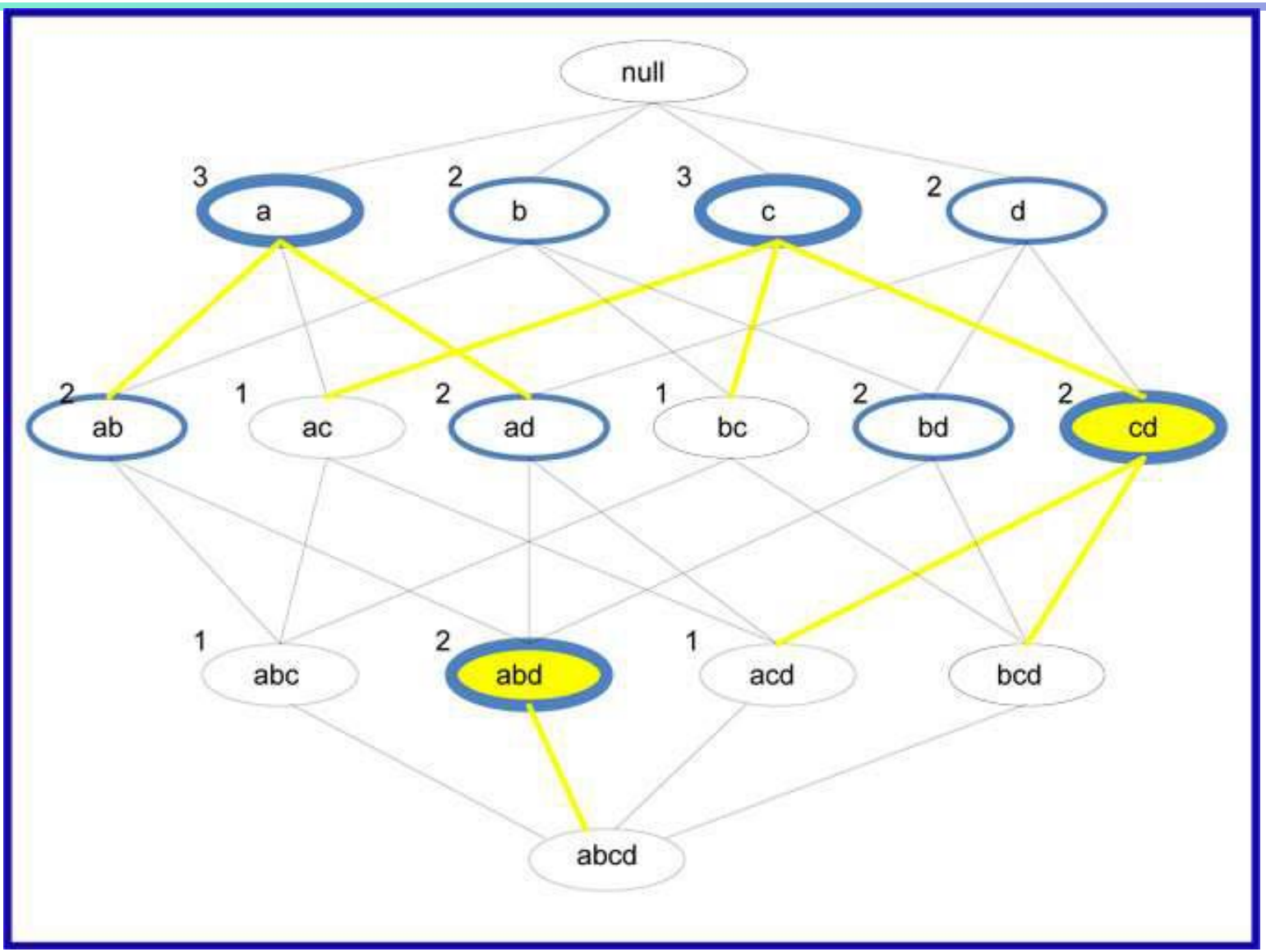
- یک الگوی طولانی شامل تعداد زیادی زیر الگو است. مثلا $\{a_1, ..., a_{100}\}$ شامل $\binom{100}{1} + \binom{100}{2} + ... + \binom{1}{1}\binom{0}{0}\binom{0}{0} = 2^{100} - 1 = 1.27*10^{30}$ زیرالگو می باشد.

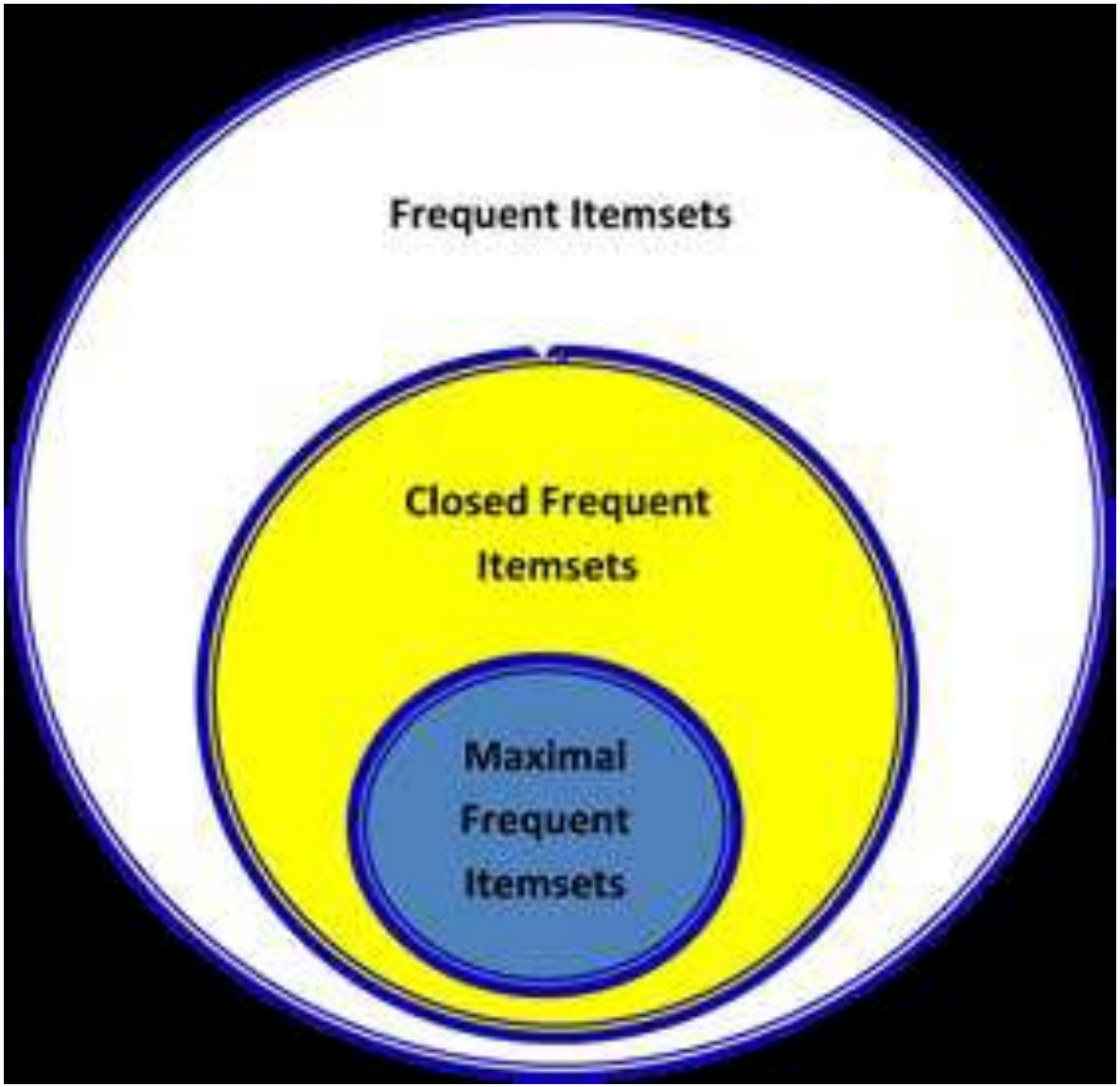
- راه حل : کاوش *closed patterns* و *max-patterns* *بجای همه الگوها*

- مجموعه اقلام X را *closed pattern* می نامیم اگر X پرتکرار باشد و هیچیک از *super-pattern*های *آن support معادل آن را نداشته باشند*

- مجموعه اقلام X را *max-pattern* می نامیم اگر X پرتکرار باشد و هیچیک از *super-pattern*های *آن پرتکرار نباشد*.

مثال



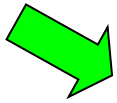

Frequent Itemsets
Closed Frequent Itemsets
Maximal Frequent Itemsets

پیچیدگی محاسباتی کاوش الگوهای مکرر

- در بدترین حالت چند itemset ممکن است تولید شوند؟
  - تعداد الگوهای مکرر تولید شده وابسته به minsup است.
  - اگر minsup پایین باشد تعداد الگوهای پرتکرار نمایی خواهد بود.
  - بدترین حالت: اگر M تعداد اقلام و N ماکزیمم طول تراکنش ها باشد تعداد الگوها M^N خواهد بود.

- پیچیدگی در بدترین حالت در مقابل احتمال رخداد یک الگو:
  - فرض کنید فروشگاهی 10^4 نوع محصول متفاوت دارد.
    - شانس انتخاب یکی از محصولات : 10^{-4}
    - شانس انتخاب یک مجموعه حاوی 10 محصول مشخص: $\sim 10^{-40}$
  - شانس اینکه این مجموعه خاص به اندازه 10^3بار در 10^9 تراکنش تکرار شده باشد چیست؟

کاوش الگوهای مکرر، مشارکت ها و همبستگی ها: مفاهیم پایه و روش ها

- مفاهیم اولیه
- روش های کاوش الگوهای مکرر 
- چه الگوهایی جذاب هستند؟ (روش های ارزیابی الگوها)
- خلاصه

روشهای مقیاس پذیر کاوش الگوهای مکرر

- Apriori: یک روش مبتنی بر تولید و آزمایش مجموعه های کاندید
- FPGrowth: ساخت الگوهای مکرر از طریق گسترش آنها
- ECLAT: کاوش الگوهای مکرر بر اساس چیدمان عمودی داده ها

ویژگی **Downward Closure** و روش های مقیاس پذیر

- ویژگی downward closure الگوهای مکرر
  - هر زیرمجموعه یک الگوی پرتکرار لزوما پرتکرار است.
  - اگر **{beer, diaper, nuts}** پرتکرار است **{beer, diaper}** هم پر تکرار است.
  - همه تراکنش های حاوی اولی دومی را هم در خود دارد.
- سه روش عمده مقیاس پذیر برای کاوش الگوهای مکرر:
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: یک روش تولید و تست الگوهای کاندید

- اساس هرس کردن روش Apriori: اگر یک مجموعه اقلام پرتکرار نباشد هیچ superset آن نباید تولید و آزمایش شود.
- روش کار:
  - پایگاه داده در ابتدا برای پیدا کردن مجموعه های تک عضوی پرتکرار اسکن می شود.
  - کاندیداهای با طول K+1 از الگوهای پرتکرار با طول K تولید می شوند.
  - کاندیداها تست می شوند.
  - هرگاه مجموعه پرتکرار یا کاندیدای دیگری نبود خاتمه می یابد.

مثال

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

$Sup_{min} = 2$

C_1

1st scan

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$;

پیاده سازی **Apriori**

- چگونه کاندیداها تولید می شوند؟

- Step 1: self-joining L_k
- Step 2: pruning

- مثال برای تولید کاندیدا

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*

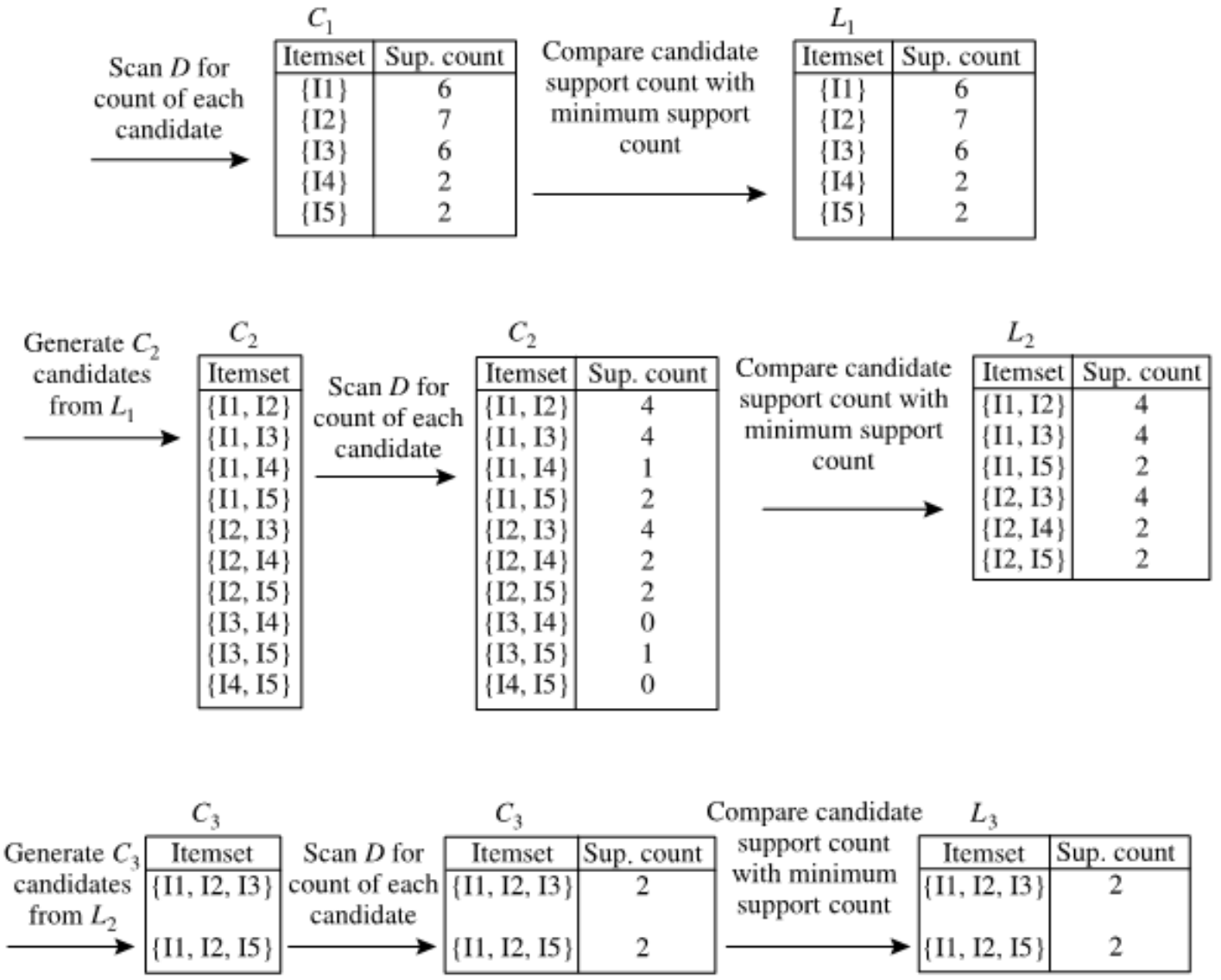
- هرس کردن:

  - *acde* is removed because *ade* is not in L_3
- C_4 = {*abcd*}

مثال دیگر

Table 6.1 Transactional Data for an *AllElectronics* Branch

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3


C_1
Itemset | Sup. count
{I1} 6
{I2} 7
{I3} 6
{I4} 2
{I5} 2
Scan D for count of each candidate
Compare candidate support count with minimum support count
L_1
Itemset | Sup. count
{I1} 6
{I2} 7
{I3} 6
{I4} 2
{I5} 2
Generate C_2 candidates from L_1
C_2
Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}
Scan D for count of each candidate
C_2
Itemset | Sup. count
{I1, I2} 4
{I1, I3} 4
{I1, I4} 1
{I1, I5} 2
{I2, I3} 4
{I2, I4} 2
{I2, I5} 2
{I3, I4} 0
{I3, I5} 1
{I4, I5} 0
Compare candidate support count with minimum support count
L_2
Itemset | Sup. count
{I1, I2} 4
{I1, I3} 4
{I1, I5} 2
{I2, I3} 4
{I2, I4} 2
{I2, I5} 2
Generate C_3 candidates from L_2
C_3
Itemset
{I1, I2, I3}
{I1, I2, I5}
Scan D for count of each candidate
C_3
Itemset | Sup. count
{I1, I2, I3} 2
{I1, I2, I5} 2
Compare candidate support count with minimum support count
L_3
Itemset | Sup. count
{I1, I2, I3} 2
{I1, I2, I5} 2

تولید قوانین انجمنی از مجموعه های پرتکرار

- شرط قوی بودن قواعد انجمنی تامین آستانه تعیین شده برای min-support و min-confidence است
- در الگوهای پر تکرار min-support تامین است.
- قوانین انجمنی به شکل زیر تولید می شوند:

برای هر مجموعه اقلام پرتکرار مانند L تمام زیر مجموعه های غیر تهی آن ایجاد می شوند. برای هر زیر مجموعه غیر تهی s قانون به صورت زیر تولید می شود:

$$s \rightarrow (l - s) \; if \; support\_count(l)/support\_count(s) \geq min\_conf$$

مثال

- فرض کنید X={I1,I2,I5} پرتکرار باشد.
- قوانین انجمنی تولید شده از x :

$$\{I1, I2\} \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$\{I1, I5\} \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$\{I2, I5\} \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow \{I2, I5\}, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow \{I1, I5\}, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow \{I1, I2\}, \quad confidence = 2/2 = 100\%$$

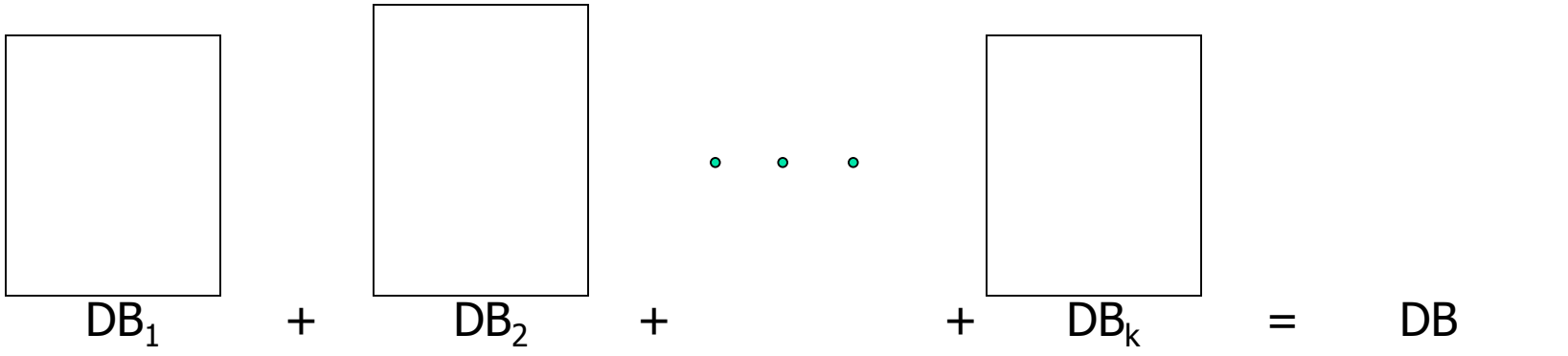
- در صورتی که min-conf=70% باشد قواعد دوم و سوم و ششم انتخاب خواهند شد.

بهبود روش Apriori

- مهمترین چالش های محاسباتی
  - مرور چندباره پایگاه داده تراکنش ها
  - تعداد زیاد کاندیداها
  - حجم زیاد کار شمارش تعداد تکرار کاندیداها
- اصلاح Apriori: ایده های کلی
  - کاهش تعداد مرورهای پایگاه داده تراکنش ها
  - کاهش تعداد کاندیداها
  - تسهیل شمارش تعداد تکرار کاندیداها

پارتیشن بندی: فقط دو بار مرور پایگاه داده

- هر itemset که امکان پرتکرار بودن در کل پایگاه داده را داشته باشد باید حداقل در یکی از پارتیشن ها به نسبت پرتکرار باشد.
  - مرور اول: تقسیم پایگاه داده و پیدا کردن الگوهای مکرر محلی
  - مرور دوم: شمارش و تعیین الگوهای مکرر سراسری از میان کاندیداهای مرحله قبل
- A. Savasere, E. Omiecinski and S. Navathe, *VLDB'95*



تکنیک مبتنی بر درهم سازی: کاهش تعداد کاندیداها

Table 6.1 Transactional Data for an *AllElectronics* Branch

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

$$h(x, y) = ((order\ of\ x) \times 10 + (order\ of\ y)) \bmod 7$$

- همزمان با شمارش مجموعه اقلام یک عضوی جدول درهم سازی از ترکیب های دوتایی پر می شود.

مجموعه اقلام دوتایی که مجموعه تعداد باکت آنها کمتر از min-sup باشد از کاندیداهای مرحله بعد حذف می شوند.

H_2

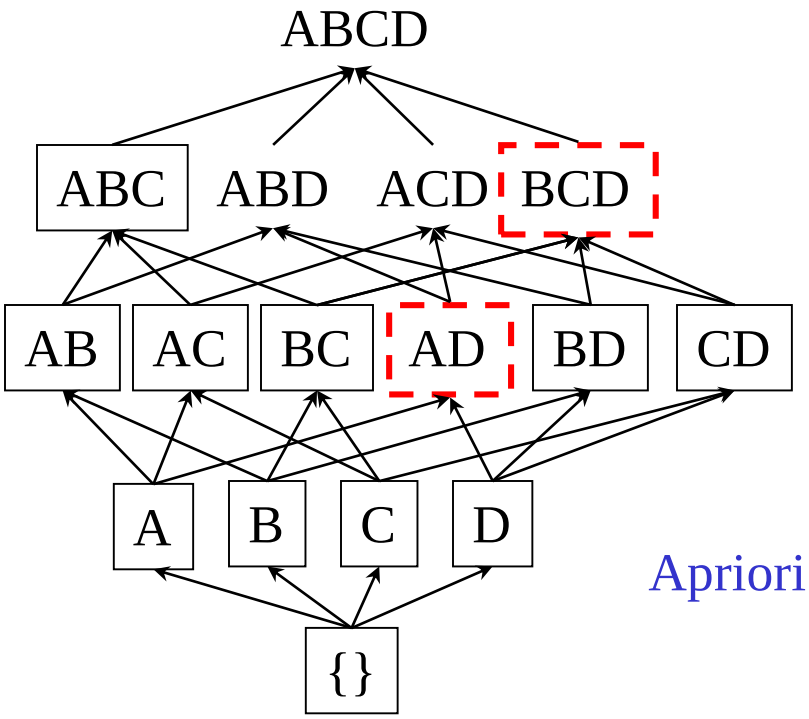
bucket address	0	1	2	3	4	5	6
bucket count	2	2	4	2	2	4	4
bucket contents	{I1, I4} {I3, I5}	{I1, I5} {I1, I5}	{I2, I3} {I2, I3} {I2, I3} {I2, I3}	{I2, I4} {I2, I4}	{I2, I5} {I2, I5}	{I1, I2} {I1, I2} {I1, I2} {I1, I2}	{I1, I3} {I1, I3} {I1, I3} {I1, I3}

Hash Table

نمونه گیری برای الگوهای پرتکرار

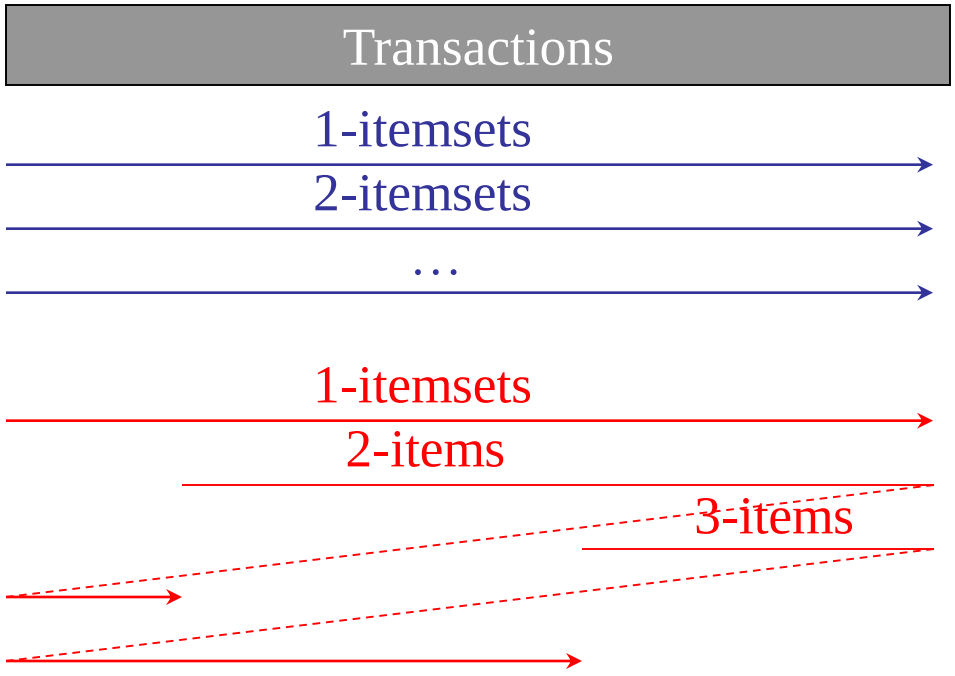
- انتخاب یک نمونه از پایگاه داده اصلی که در حافظه اصلی قرار گیرد، جستجوی الگوهای مکرر در داخل نمونه با استفاده از Apriori
- پیمایش کل پایگاه برای شمارش تعداد کاندیداها در کل (با توجه به تقریبی بودن روش،برای اینکه تعداد کمتری الگوی پرتکرار از دست بروند کاندیداها را از حد آستانه پایین تری انتخاب می کنیم.)
- H. Toivonen. Sampling large databases for association rules. In *VLDB'96*

شمارش پویا (DIC):کاهش تعداد پیمایش ها



Itemset lattice

- در طی شمارش هر جا پرتکرار بودن A و D محرز شد شمارش AD از همان محل شروع میشود
- هرجا پرتکرار بودن همه زیرمجموعه های دوتایی BCD مشخص شد از همان نقطه شمارش تعداد تکرار BCD آغاز می شود.



S. Brin R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD'97*

مثال

Items		*Database*	
		Transaction ID (tid)	*Items bought*
Bread	*a*	1	*a b d e*
Butter	*b*	2	*b c e*
Milk	*c*	3	*a b d e*
cheese	*d*	4	*a b c e*
coke	*e*	5	*a b c d e*
		6	*b c d*

C_1

pattern	support	state
a	0/0	PI
b	0/0	PI
c	0/0	PI
d	0/0	PI
e	0/0	PI

reading objects 1-3

C_1

pattern	support	state
a	2/3	PF
b	3/3	PF
c	2/3	PF
d	2/3	PF
e	3/3	PF

C_2

pattern	support	state
ab	0/0	PI
ac	0/0	PI
ad	0/0	PI
ae	0/0	PI
bc	0/0	PI
bd	0/0	PI
be	0/0	PI
cd	0/0	PI
ce	0/0	PI
de	0/0	PI

reading objects 4-6

C_1

pattern	support	state
a	4/6	CF
b	6/6	CF
c	4/6	CF
d	4/6	CF
e	5/6	CF

C_2

pattern	support	state
ab	2/3	PF
ac	1/3	PF
ad	1/3	PF
ae	2/3	PF
bc	2/3	PF
bd	2/3	PF
be	2/3	PF
cd	1/3	PF
ce	1/3	PF
de	1/3	PF

C_3

pattern	support	state
abc	0/0	PI
abd	0/0	PI
abe	0/0	PI
acd	0/0	PI
ace	0/0	PI
ade	0/0	PI
bcd	0/0	PI
bce	0/0	PI
bde	0/0	PI
cde	0/0	PI

C_2

pattern	support	state
ab	4/6	CF
ac	2/6	CF
ad	3/6	CF
ae	4/6	CF
bc	4/6	CF
bd	4/6	CF
be	5/6	CF
cd	2/6	CF
ce	3/6	CF
de	3/6	CF

C_3

pattern	support	state
abc	1/3	PF
abd	2/3	PF
abe	2/3	PF
acd	1/3	PF
ace	1/3	PF
ade	2/3	PF
bcd	1/3	PF
bce	1/3	PF
bde	2/3	PF
cde	1/3	PF

C_4

pattern	support	state
abcd	0/0	PI
abce	0/0	PI
abde	0/0	PI
acde	0/0	PI
bcde	0/0	PI

reading objects 4-6

C_3

pattern	support	state
abc	2/6	CF
abd	3/6	CF
abe	4/6	CF
acd	1/6	CI
ace	2/6	CF
ade	3/6	CF
bcd	2/6	CF
bce	2/6	CF
bde	3/6	CF
cde	1/6	CI

C_4

pattern	support	state
abcd	0/3	PI
abce	1/3	PF
abde	1/3	PF
acde	0/3	PI
bcde	0/3	PI

reading objects 1-3

C_4

pattern	support	state
abcd	1/6	CI
abce	2/6	CF
abde	3/6	CF
acde	1/6	CI
bcde	1/6	CI

FPGrowth: روشی برای کاوش الگوهای مکرر بدون تولید کاندیداها

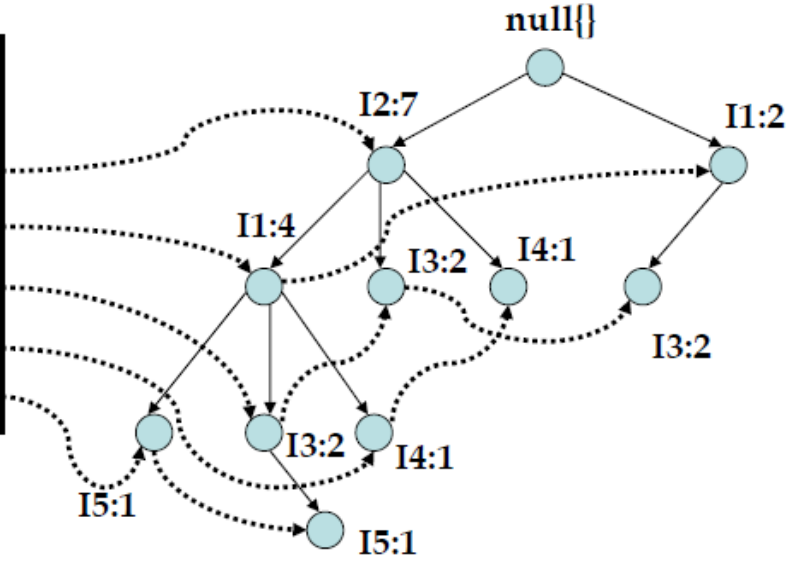
- معایب روش Apriori
  - جستجوی اول سطح
  - تولید مجموعه های کاندید و تست آن ها
    - اغلب تعداد کاندیداها بسیار زیاد است.
- روش FPGrowth (J. Han, J. Pei, and Y. Yin, SIGMOD' 00)
  - جستجوی اول عمق
  - پرهیز از تولید صریح کاندیداها
- فلسفه اصلی: رشد الگوهای طولانی از الگوهای کوتاه با استفاده از الگوهای مکرر محلی
  - "abc" is a frequent pattern
  - Get all transactions having "abc", i.e., project DB on abc: DB|abc
  - "d" is a local frequent item in DB|abc → abcd is a frequent pattern

فاز اول: ساخت **FP-tree** از یک پایگاه داده تراکنش

1. پایگاه داده را یکبار مرور کنید و الگوهای مکرر تک عضوی را بیابید.
2. الگوهای مکرر یافته شده را بر مبنای تعداد تکرار بصورت نزولی مرتب کنید، f-list
3. یکبار دیگر پایگاه را مرور کنید اقلام موجود در هر تراکنش را بر اساس تعداد تکرار آنها مرتب کنید و FP-tree را بسازید.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Item Id	Sup Count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



فاز دوم: استخراج درخت های شرطی و الگوهای مکرر

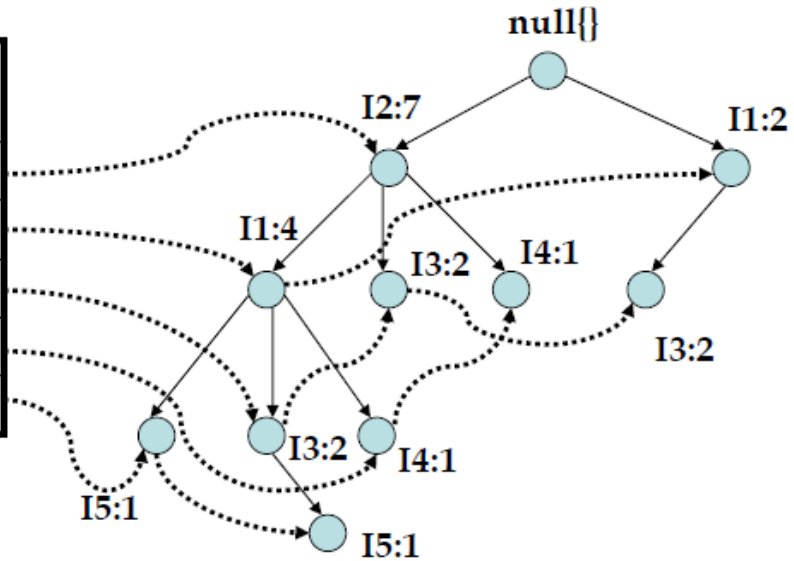
- از آخرین عنصر فهرست یعنی I5 شروع می کنیم. دو مسیر در درخت به I5 ختم می شود:

1 : (I2, I1, I3, I5:1)

2 : (I2, I1, I5:1)

- I5 را پسوند الگو و (I2,I1) (I2, I1, I3) را پایگاه الگوی شرطی یا Conditional pattern base می نامیم.

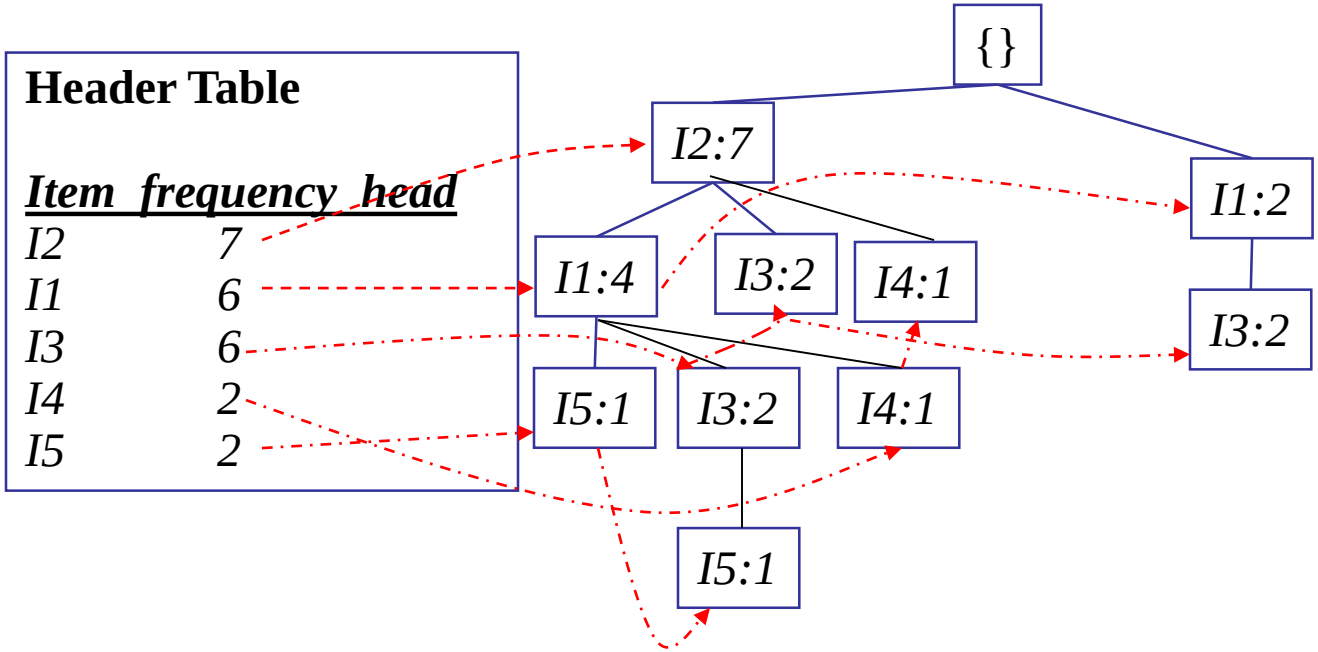
Item Id	Sup Count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



- دو مقدار (I2, I1, I3, I5:1) و (I2, I1, I5:1) را به عنوان تراکنش در نظر می گیریم و الگوریتم را ادامه می دهیم.
- با استفاده از این دو تراکنش درخت شرطی I5 ساخته می شود که حاوی تنها یک مسیر <I2:2 , I1:2> است. I3 به دلیل اینکه تعداد تکرار آن کمتر از min-sup است حذف می شود.
- ترکیبات الگوهای مکرر با استفاده از این تک مسیر ایجاد می شود:

- {I2, I5 : 2}
- {I1, I5 : 2}
- {I2, I1, I5 : 2}

Item	conditional pattern base	conditional FP-tree	frequent patterns generated
I5	{(I2 I1: 1),(I2 I1 I3:1)}	<I2:2, I1:2>	I2 I5:2, I1 I5:2, I2 I1 I5:2
I4	{(I2 I1:1), (I2:1)}	<I2: 2>	I2 I4:2
I3	{(I2 I1: 2),(I2:2),(I1:2)}	<I2:4,I1:2>,<I1:2>	I2 I3:4, I1 I3:4 I2 I1 I3:2
I1	{(I2: 4)}	<I2: 4>	I2 I1:4



ECLAT: کاوش مجموعه اقلام مکرر با استفاده از قالب عمودی داده ها

- در این الگوریتم قالب نگهداری داده ها عوض می شود و بجای نگهداری تراکنش ها اطلاعات به شکل زیر نگهداری می شود:

The Vertical Data Format of the Transaction Data Set *D* of Table 6.1

itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

- برای پیدا کردن 1-itemset های پر تکرار کافیست تعداد اعضای مجموعه ها شمرده شوند.
- برای پیدا کردن 2-itemset های پر تکرار لازم است بین مجموعه ها اشتراک گرفته شود و تعداد اعضای مجموعه حاصل شمرده شود.
- ...

The Vertical Data Format of the Transaction Data Set D of Table 6.1

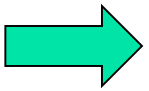
itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

2-Itemsets in Vertical Data Format

itemset	*TID_set*
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

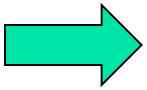
The Vertical Data Format of the Transaction Data Set D of Table 6.1

itemset	*TID_set*
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}



2-Itemsets in Vertical Data Format

itemset	*TID_set*
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}



3-Itemsets in Vertical Data Format

itemset	*TID_set*
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

بهبود الگوریتم ECLAT

- برای مجموعه های متراکم و با اشتراکات زیاد از قابلیت diffset استفاده می شود.
- در این روش بجای ذخیره سازی اشتراک دو مجموعه، اختلاف آن ها ذخیره می شود:

{I1} = {T100, T400, T500, T700, T800, T900}

{I1,I2} = {T100, T400, T800, T900}

Diffset({I1},{I1,I2}) = {T500,T700}

در این روش فضای حافظه کمتری مصرف می شود اما الگوریتم ها متفاوت و پیچیده تر است.

کاوش الگوهای مکرر، مشارکت ها و همبستگی ها: مفاهیم پایه و روش ها

- مفاهیم اولیه
- روش های کاوش الگوهای مکرر
- چه الگوهایی جذاب هستند؟ (روش های ارزیابی الگوها)
- خلاصه

قوانین جالب

قوانینی که تابحال با استفاده از معیارهای support و confidence بدست آوردیم قوانین قوی هستند اما لزوما جالب نیستند.

بنابراین به روش های ارزیابی بهتری برای قوانین تولید شده نیاز داریم.

مثال

- به طور مثال فرض کنید مجموعه داده شامل 10000 تراکنش است.
- 6000 تراکنش شامل بازی های کامپیوتری
- 7500 تراکنش شامل فیلم ویدیویی
- 4000 تراکنش شامل هر دو مورد
- فرض کنید

- $Min - \text{sup} = 30\%$ $Min - conf = 60\%$
- $Buys\ Computer\ games \Rightarrow\ Buys$ videos
- $support = 40\%$, $Confidence = 66\%$

قوانین قوی لزوما جالب نیستند

- قانون ایجاد شده در مثال قبل یک قانون قوی است ولی جالب نیست. بلکه یک قانون گمراه کننده است.

- احتمال خرید فیلم ویدیویی 75 درصد است که از 66 درصد بیشتر است.

- در واقع خرید این دو محصول با هم رابطه عکس دارند. یعنی خرید بازی احتمال خرید فیلم را کم می کند.

سایر معیارها

برای بهبود چارچوب support-confidence می توان از آنالیز همبستگی استفاده کرد:

$$A \Rightarrow B \ (support, confidence, correlation)$$

استقلال آماری

- جمعیت 1000 دانشجو
  - 600 دانش آموز شنا کردن را می دانند. (S)
  - 700 دانش آموز دوچرخه سواری را می دانند. (B)
  - 420 دانشجو هم دوچرخه سواری و هم شنا می دانند. (S,B)

$P(S \cap B) = 420/1000 = 0.42$

$P(S) * P(B) = 0.6 * 0.7 = 0.42$

$P(S \cap B) = P(S) * P(B)$ => استقلال آماری

$P(S \cap B) > P(S) * P(B)$ => ارتباط مثبت

$P(S \cap B) < P(S) * P(B)$ => ارتباط منفی

معیار همبستگی lift

- معیار lift با استفاده از رابطه زیر بدست می آید:

$$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$

- اگر نتیجه رابطه کمتراز 1 باشد آنگاه پیشامد A با پیشامد B همبستگی منفی دارند.
- اگر نتیجه بزرگتر از 1 باشد همبستگی مثبت است و به این معناست که پیشامد یکی بر پیشامد دیگری دلالت دارد.
- اگر نتیجه برابر 1 باشد آنگاه A و B مستقل هستند و همبستگی بین آنها وجود ندارد.

سایر معیارها

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\left(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),\right.$ $\left.P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\right)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$

کاوش الگوهای مکرر، مشارکت ها و همبستگی ها: مفاهیم پایه و روش ها

- مفاهیم اولیه
- روش های کاوش الگوهای مکرر
- چه الگوهایی جذاب هستند؟ (روش های ارزیابی الگوها)
- خلاصه

خلاصه

- مفاهیم پایه: association rules, support-confident framework, closed and max-patterns
- روش های مقیاس پذیر کاوش الگوهای مکرر
  - Apriori (Candidate generation & test)
  - Projection-based (FPgrowth, CLOSET+, ...)
  - Vertical format approach (ECLAT, CHARM, ...)
- کدام الگوها جالب هستند؟
  - روش های ارزیابی الگوها

Ref: Basic Concepts of Frequent Pattern Mining

- (Association Rules) R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93
- (Max-pattern) R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98
- (Closed-pattern) N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99
- (Sequential pattern) R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95

Ref: Apriori and Its Improvements

- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95
- J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95
- H. Toivonen. Sampling large databases for association rules. VLDB'96
- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98

Ref: Depth-First, Projection-Based FP Mining

- R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. J. Parallel and Distributed Computing, 2002.
- G. Grahne and J. Zhu, Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. FIMI'03
- B. Goethals and M. Zaki. An introduction to workshop on frequent itemset mining implementations. *Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03),* Melbourne, FL, Nov. 2003
- J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD' 00
- J. Liu, Y. Pan, K. Wang, and J. Han. Mining Frequent Item Sets by Opportunistic Projection. KDD'02
- J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. ICDM'02
- J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. KDD'03

Ref: Vertical Format and Row Enumeration Methods

- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. DAMI:97.
- M. J. Zaki and C. J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining, SDM'02.
- C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. KDD’02.
- F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. Zaki , CARPENTER: Finding Closed Patterns in Long Biological Datasets. KDD'03.
- H. Liu, J. Han, D. Xin, and Z. Shao, Mining Interesting Patterns from Very High Dimensional Data: A Top-Down Row Enumeration Approach, SDM'06.

Ref: Mining Correlations and Interesting Rules

- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94.
- R. J. Hilderman and H. J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic, 2001.
- C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98.
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02.
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE'03.
- T. Wu, Y. Chen, and J. Han, "Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework", Data Mining and Knowledge Discovery, 21(3):371-397, 2010