داده کاوی

فصل ششم

تهیه شده توسط
محمد پورزعفرانی
Pourzaferani.ir
دانشگاه اصفهان

دانشگاه اصفهان

تشکر

- آقای باباخانی
- خانم خسروی
- خانم رضایی
- خانم سبکتکین
- آقای مسعودیان
- آقای مقیمی

فصل ششم: الگوهای مکرر و قوانین انجمنی

- مفاهیم اولیه
- روش‌های کاوش مجموعه اقلام مکرر ($Frequent\ Itemset$)
- چه الگوهایی از دید ما جذاب هستند؟ (روش‌های ارزیابی الگوها)
- خلاصه فصل

تحلیل الگوهای پرتکرار

- الگوی پرتکرار (Frequent Pattern): یک الگو که غالبا در یک مجموعه داده رخ می دهد. این الگوها می‌توانند
  - مجموعه‌ای از ایتم‌ها : به طور مثال نان و شیر
  - زیر مجموعه‌ای از توالی‌ها : پس از دوربین کارت حافظه خرید می‌شود
  - زیرساختارها : مانند زیرگراف، زیردرخت
  - و غیره باشند

- نخستین بار توسط Agrawal، Imielinski و Swami [AIS93] در زمینه مجموعه آیتم‌های پرتکرار (frequent itemset) و کاوش قوانین انجمنی پیشنهاد شد.

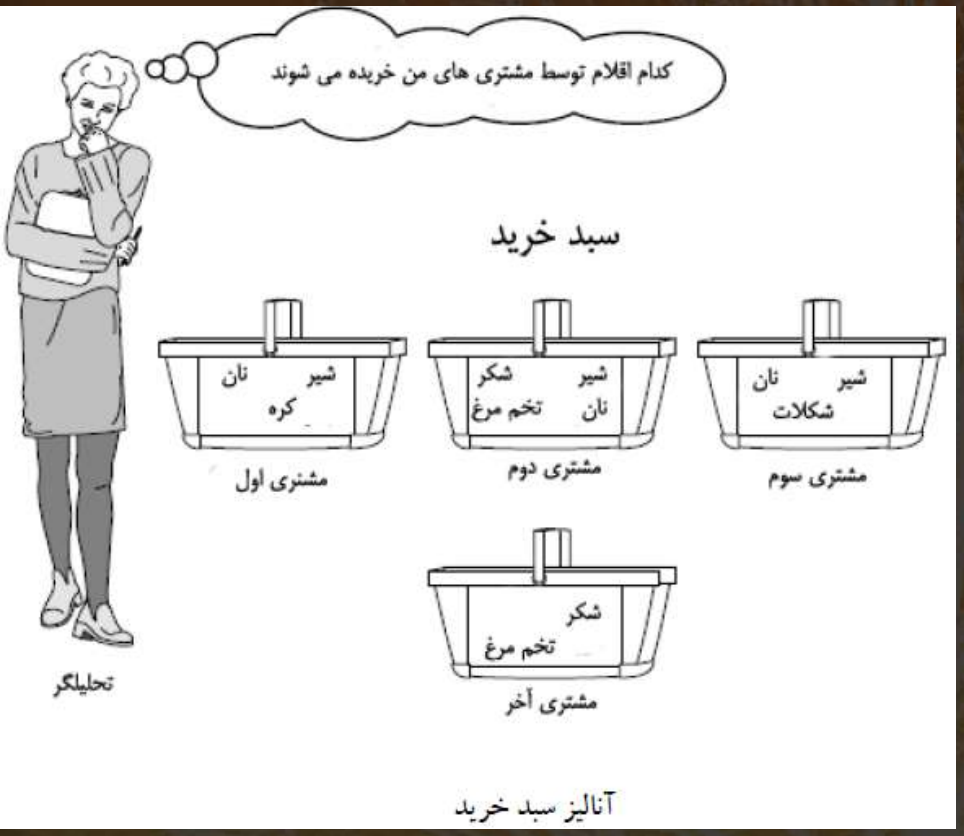
تحلیل الگوهای پرتکرار

- انگیزه (Motivation):
  - پیدا کردن نظم و قاعده ذاتی موجود در داده‌ها
  - چه محصولاتی اغلب با هم خریداری شده‌اند –کاغذ و خودکار؟
  - خریدهای بعدی بعد از خرید کامپیوتر چیست؟
  - چه نوع از DNA حساس به این داروی جدید است؟
  - آیا ما می‌توانیم به طور اتوماتیک اسناد وب را طبقه بندی کنیم؟
- برنامه‌های کاربردی
  - آنالیز سبد خرید، بازاریابی، طراحی کاتالوگ، آنالیز سلسله عملبات فروش، آنالیز وبلاگ (جریان کلیک کردن) .

چرا کاوش الگوهای مکرر مهم است؟

- پایه و اساس بسیاری از کارهای داده کاوی است
  - برای آنالیز قواعد انجمنی، همبستگی و آنالیز علیت (causality)
  - آنالیز الگوهای ترتیبی، ساختاری (به عنوان مثال، زیر گراف)
  - آنالیز الگوی در فضا و زمان، چند رسانه‌ای، سری‌های زمانی و جریان داده
  - دسته بندی: تبعیض یا تمیز، آنالیز الگوی مکرر
  - آنالیز خوشه‌ای: خوشه‌بندی مبتنی بر الگوهای مکرر

آنالیز سبد خرید



آنالیز سبد خرید

1) با پیدا کردن ارتباط زیاد فروش دو محصول می‌توان آنها را با فاصله از هم قرار داد و به فروش محصولات دیگر کمک کرد

2) به خرده‌فروش کمک می‌کند که چه محصولی را فروش ویژه کند، به طور مثال کامپیوتر و پرینتر با همدیگر خرید می‌شوند و فروش ویژه پرینتر به فروش هر دو کمک می‌کند

آنالیز سبد خرید

- پس از بررسی مجموعه داده به رابطه‌ای مانند زیر می‌رسیم

$$\boldsymbol{computer \Rightarrow anivirus\ [support = 2\%, Confidence = 60\%]}$$

- فرض کنید $I = \{I1, I2, \dots, Im\}$ مجموعه اقلام کالا باشد. هر تراکنش با نام T مجموعه‌ای از آیتم هاست که در آن $T \subseteq I$ است

- فرض کنید A مجموعه‌ای از آیتم ها باشد. گفته می شود تراکنش T شامل A است اگر و فقط اگر $A \subseteq T$.
- یک قانون انجمنی به صورت $A \Rightarrow B$ بیان می‌شود
- بطوریکه $A \subset I$، $B \subset I$ و $A \cap B = \emptyset$

مفاهیم پایه

- **Support**: فرض کنید قانون در مجموعه تراکنش‌های D با $support = s$ است.
- در این صورت s درصدی از تراکنش‌های D است که حاوی A و B است. این درصد را بصورت احتمال $P(A\ U\ B)$ بیان می‌کنیم
- $support(A \Rightarrow B) = P(A \cup B)$

- **Confidence**: قانون $(A \Rightarrow \mathrm{B})$ دارای $confidence = c$ در مجموعه تراکنش D است که در آن c درصد تراکنش‌هایی است که اگر حاوی A باشند B نیز در آن‌ها وجود داشته باشد

- $confidence(A \Rightarrow \mathrm{B}) = P(B|A)$
- $confidence(A \Rightarrow \mathrm{B}) = \frac{support(A \cup B)}{support(A)} = \frac{support_Count(A \cup B)}{support_Count(A)}$

مفاهیم پایه

- داده کاو در هنگام استخراج قوانین دو آستانه زیر را تعیین می‌نماید

$$min - sup -$$
$$min - conf -$$

- قوانینی که درجه پشتیبانی و اطمینان آن‌ها بیش از این دو مقدار باشند به عنوان قوی $(Strong)$ شناخته می‌شوند

مفاهیم پایه

- یک مجموعه از آیتم‌ها (اقلام کالا) را با عنوان ***itemset*** می‌شناسیم
- itemset که حاوی k آیتم باشد را ***K-itemset*** می‌نامیم
- فرکانس تکرار یک itemset را تعداد تراکنش‌های در نظر می‌گیریم که حاوی آن مجموعه اقلام باشند
- فراوانی رخداد یک itemset برابر است با تعداد تراکنش‌هایی که شامل itemset باشند.
- این درجه پشتیبانی را با $absolute\ support$ می‌شناسیم
- رابطه اسلاید پیش را با $relative\ suport$ می‌شناسیم

اقلام پرتکرار و قوانین انجمنی

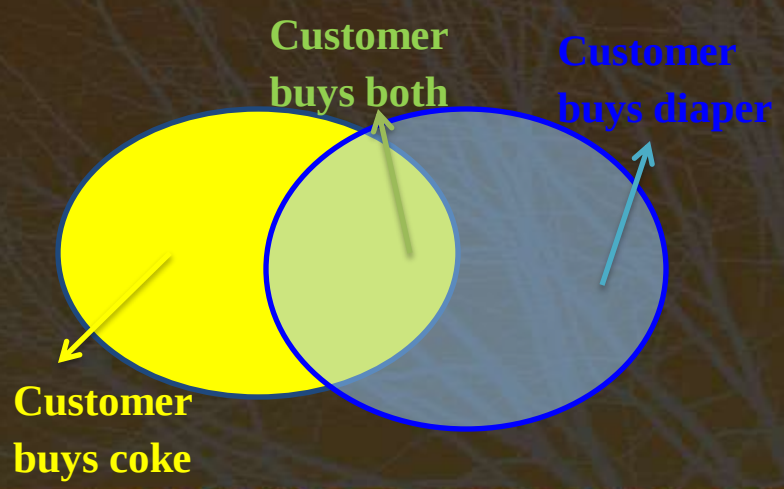
- برای استخراج قوانین انجمنی از مجموعه اطلاعات به دو قدم زیر نیاز داریم

1) پیدا نمودن تمام الگوهای پرتکرار
   - طبق تعریف، در قدم اول لازم است تمام مجموعه‌های پرتکرار (درجه پشتیبانی آن‌ها بیش از min-sup) مشخص شوند

2) تولید قوانین انجمنی از روی مجموعه‌های پرتکرار
   - در این مرحله تمام قوانینی که دو مقدار آستانه (min-sup و min-conf) را ارضا می‌کنند تولید می‌شوند

تبیین مفاهیم پایه (با یک مثال)

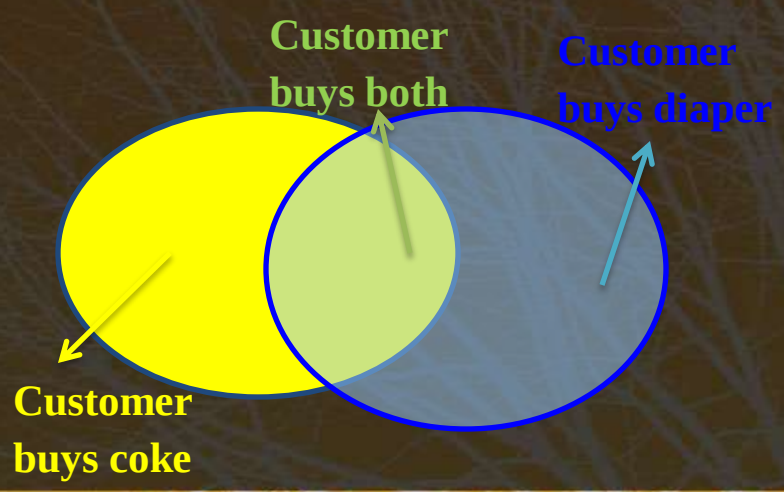
Tid	Items bought
10	coke, Nuts, Diaper
20	coke, Coffee, Diaper
30	coke, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- itemset: مجموعه‌ای از یک یا چند آیتم
- $k - itemset \rightarrow X = \{x_1, \dots, x_k\}$
- support(مطلق) : فرکانس یا تعداد رخداد مجموعه آیتم‌ها X
- support(نسبی): درصدی از تراکنش های است که حاوی X می باشد (احتمال این که یک تراکنش حاوی X باشد.)
- مجموعه آیتم‌ها X پرتکرار است اگر پشتیبانی X ها بیشتر از یک حد آستانه minsup باشد.

تبیین مفاهیم پایه (با یک مثال)

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



- یافتن تمام قوانین $X \rightarrow Y$ با حداقل support و confidence
  - ***support***، S، احتمال این که یک خرید شامل X ∪ Y
  - ***confidence***، C، احتمال شرطی است که یک تراکنش با داشتن X همچنین حاوی Y باشد.
  - *Let minsup = 50%, minconf = 50%*

    Freq. Pat.: coke:3, Nuts:3, Diaper:4, Eggs:3, {coke, Diaper}:3
- $Association\ rules: (many\ more!)$
  - $coke \rightarrow Diaper\ (60\%, 100\%)$
  - $Diaper \rightarrow coke\ (60\%, 75\%)$

یک مشکل

- فرض کنید یک الگوی پرتکرار شامل مجموعه $\{a_1, \dots, a_{100}\}$باشد
- نکته: زیرمجموعه‌ها یک مجموعه پرتکرار حتما پرتکرار هستند
- بنابراین داریم:

$$\binom{100}{1} + \binom{100}{2} + \cdots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \times 10^{30}$$

- این تعداد زیاد itemsetها هم از لحاظ محاسبه و هم فضای ذخیره‌سازی مشکل است
- راه‌حل: معرفی دو مفهوم جدید

$$Closed\ frequent\ itemset -$$

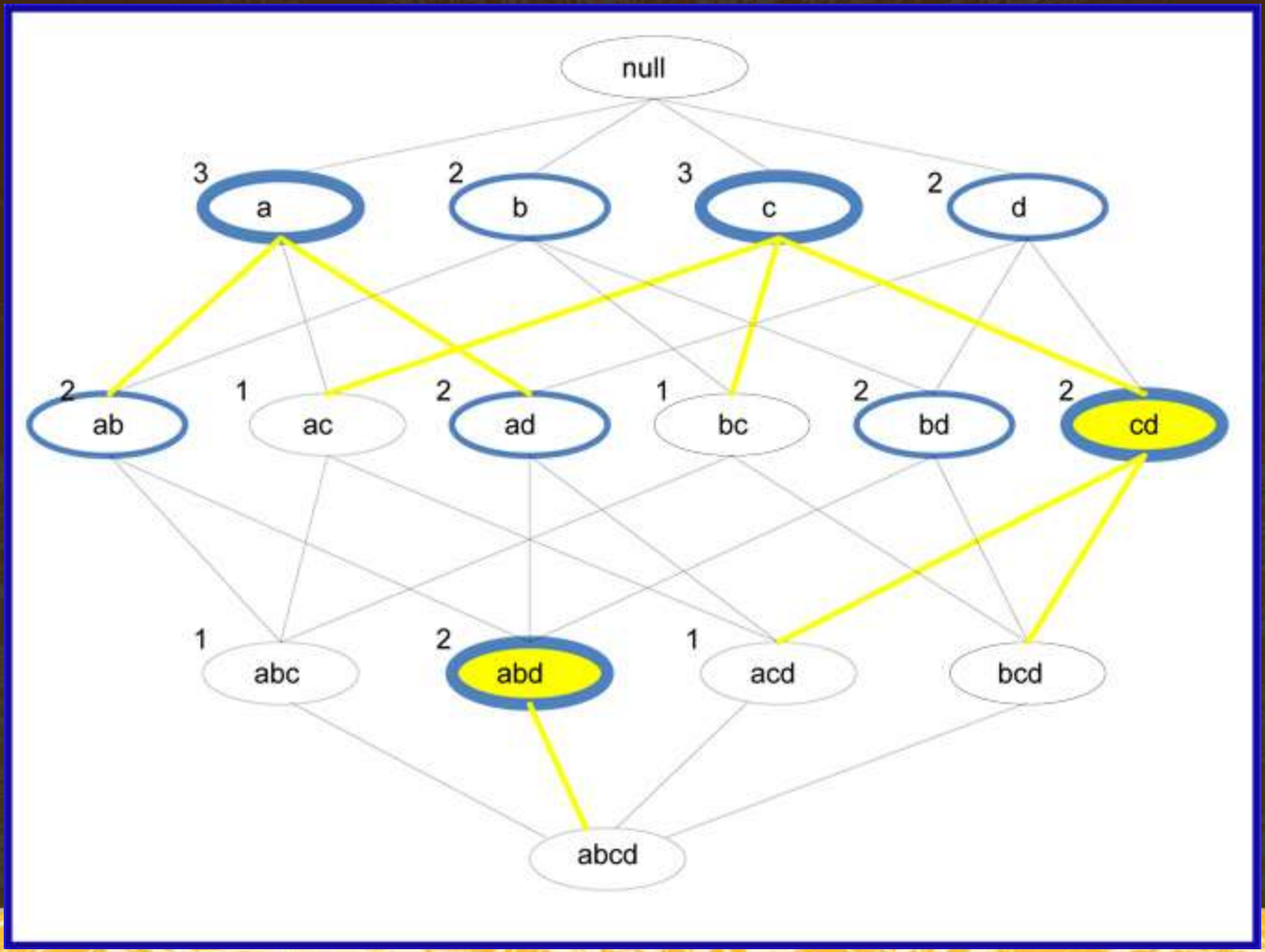
$$Maximal\ frequent\ itemset -$$

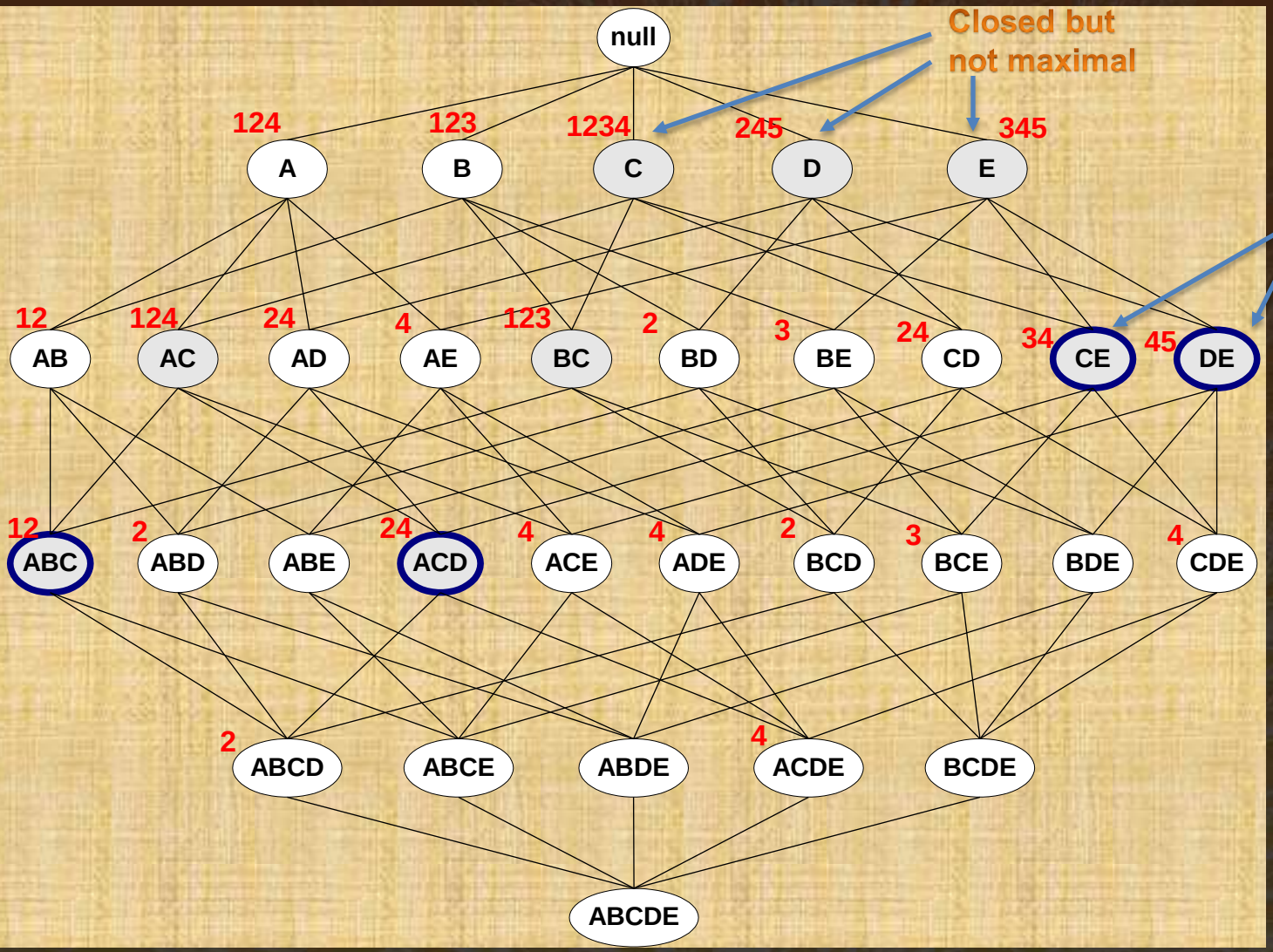
Closed frequent itemset

- تعریف ۱: یک itemset مانند X را بسته (closed) گوییم هرگاه هیچ super-itemset مانند Y وجود نداشته باشدکه support آن مشابه X باشد

- تعریف ۲: یک itemset را closed frequent itemset گوییم هرگاه
  - بسته باشد
  - پرتکرار باشد

Closed frequent itemset

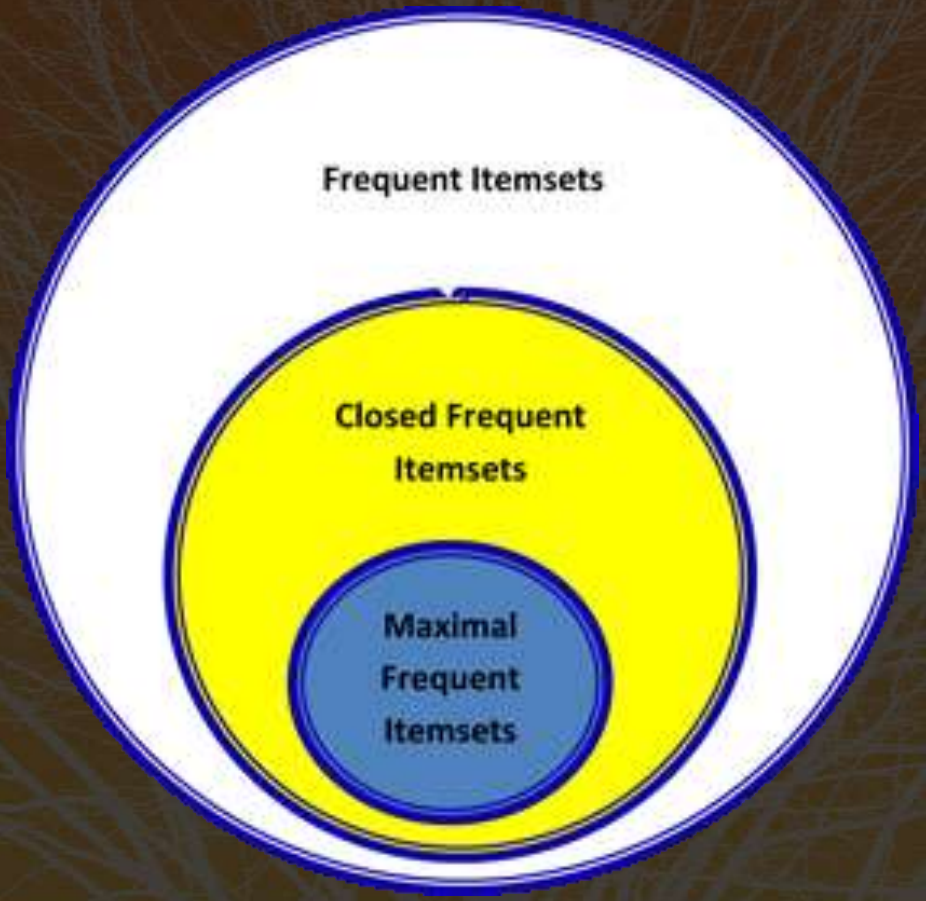




Minimum support = 2

Closed = 9
Maximal = 4

ارتباط این سه مفهوم



فصل ششم: الگوهای مکرر و قوانین انجمنی

- مفاهیم اولیه
- روش‌های کاوش مجموعه اقلام مکرر ($Frequent\ Itemset$)
- چه الگوهایی از دید ما جذاب هستند؟ (روش‌های ارزیابی الگوها)
- خلاصه فصل

روش‌های کاوش مجموعه آیتم‌های مکرر

- **Apriori** : روشی است مبتنی بر تولید-و-آزمایش مجموعه‌های کاندید
- افزایش بهره‌وری الگوریتم **Apriori**
- **FPGrowth**: روشی برای ساخت الگوی مکرر از طریق گسترش آن‌ها

(Frequent Pattern-Growth Approach)

- **ECLAT**: کاوش الگوی مکرر بر اساس چیدمان داده‌ها به صورت عمودی

ویژگی Downward Closure

- ویژگی downward closure از الگوهای مکرر
- یک زیرمجموعه از یک مجموعه اقلام مکرر خود مکرر است
- به طور مثال: اگر {coke, diaper, nuts} مکرر باشد آنگاه {coke, diaper} نیز مکرر است
- یعنی هر تراکنشی که {coke, diaper, nuts} را داشته باشد آنگاه {coke, diaper} را نیز خواهد داشت

- روش‌های استخراج : سه روش عمده

- **Apriori** (Agrawal & Srikant@VLDB'94)
- **Freq. pattern growth** (FPgrowth—Han, Pei & Yin @SIGMOD'00)
- **Vertical data format approach** (Charm—Zaki & Hsiao @SDM'02)

الگوریتم Apriori

- نام آن به این دلیل است که الگوریتم از دانش پیشین (prior knowledge) استفاده می‌کند
- در واقع الگوریتم از دانش موجود در تولید k-itemset برای تولید (k+1)itemset استفاده می‌کند
- مراحل الگوریتم:

1) در ابتدا مجموعه تراکنش‌ها را یکبار اسکن می‌کنیم و 1-itemsetهای پرتکرار را استخراج می‌کنیم
2) مجموعه تولید شده را با L_1 نشان می‌دهیم
3) در مرحله بعدی از L_1 استفاده می‌شود تا L_2 را بدست آوریم
4) اینکار را تا زمانی ادامه می‌دهیم که هیچ k-itemset پرتکراری پیدا نشود

الگوریتم Apriori (ادامه)

- با راهکار قبلی لازم است برای هر Lk مجموعه تراکنش‌ها یک اسکن کامل شود
- برای تولید هوشمند مجموعه‌های پرتکرار می‌توان از خاصیت Apriori property استفاده نمود و فضای جستجو را کاهش داد

- Apriori property: هر زیر مجموعه ناتهی از یک مجموعه پرتکرار، خود پرتکرار است
- در واقع اگر یک itemset مانند I کمتر از min_sup باشد، آنگاه P(I) < min_sup
- فرض کنید کالای A را به مجموعه اضافه کنیم آنگاه $(I \cup A)$ پرتکرارتر از I نخواهد بود
- بنابراین $P(I \cup A) < \min_sup$
- چگونه از این ویژگی استفاده نماییم؟؟

دو مرحله‌ای بودن الگوریتم

- گام اتصال (join Step):
- برای بدست آوردن Lk که معرف مجموعه کاندید k-itemset است، Lk-1 را با خودش اتصال می‌دهیم
- مجموعه ایجاد شده را با Ck نشان می‌دهیم و با نام مجموعه کاندید می‌شناسیم

- گام هرس کردن (prune step):
- هر (k-1)-itemset که پرتکرار نباشد نمی‌تواند در k-itemset شرکت داشته باشد

یک مثال از Apriori

$Sup_{min} = 2$

Database TDB

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

1st scan →

C_1

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

مثالی دیگر از Apriori

- چگونگی کاندیدها را تولید کنیم؟
- گام اول: self-joining L_k
- گام دوم: pruning
- مثالی از تولید کاندیدها (نامزد نسل)

– L_3={*abc, abd, acd, ace, bcd*}
– **Self-joining: $L_3 * L_3$**
  - *Abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
– **Pruning:(هرس کردن)**
  - ***acde* is removed because *ade* is not in L_3**
– C_4 = {*abcd*}

یک مثال واقعی از Apriori

- داده‌های یک فروشگاه الکترونیکی

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

The Apriori Algorithm—مثال

Scan D for count of each candidate →

C_1

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

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 روی مجموعه اقلام مکرر

- زمانی که itemsetهای مکرر از روی پایگاه‌داده استخراج شدند، تولید قواعد انجمنی قوی کاری آسان خواهد بود
- قاعده‌ای قوی بود که minimum support و minimum confidence را ارضا کند.
- قوانین انجمنی را می‌توان به شرح زیر تولید کرد:
- برای هر آیتم‌ست پرتکرار مانند L تمام زیر مجموعه‌های ناتهی آن ایجاد می‌شوند
- برای هر زیرمجموعه غیرتهی S از L قانون به صورت زیر تولید می‌شود
- $s \rightarrow (l - s) \quad if \ \frac{support_count(l)}{support_count(s)} \geq min_conf$
- که min_conf حداقل آستانه اعتماد است

تولید قواعد انجمنی از آیتم‌ست‌های پرتکرار(مثال)

- **داده‌های شامل آیتم‌های پرتکرار** X ={I1, I2, I5}
- قوانین انجمنی تولید شده از X چه هستند ؟

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

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

روش‌های کاوش مجموعه آیتم‌های مکرر

- **Apriori** : روشی است مبتنی بر تولید-و-آزمایش مجموعه‌های کاندید

افزایش بهره‌وری الگوریتم **Apriori**

- **FPGrowth**: روشی برای ساخت الگوی مکرر از طریق گسترش آن‌ها

(Frequent Pattern-Growth Approach)

- **ECLAT**: کاوش الگوی مکرر بر اساس چیدمان داده‌ها به صورت عمودی

بهبود روش Apriori

- **اصلی‌ترین چالش‌های محاسباتی**
- مرور چندباره پایگاه داده‌های تراکنشی
- تعداد زیاد کاندیدهای تولید شده
- محاسبه support برای تمام کاندیدها زمان‌گیر است
- **بهبوددادن Apriori: ایده‌های کلی**
- کاهش تعداد مرورهای پایگاه داده
- کاهش تعداد کاندیدها
- تسهیل شمارش support کاندیدها

افزایش بهره وری Apriori

1) استفاده از توابع درهم‌سازی (Hash-based)

2) کاهش تراکنش (Transaction Reduction)

3) پارتیشن‌بندی (Partitioning)

4) نمونه‌برداری (Sampling)

5) شمارش پویای itemsetها (Dynamic itemset counting)

۱. درهم‌سازی

- از این روش برای کاهش مجموعه‌های کاندید و اسکن مجموعه داده استفاده می‌شود
- درواقع روش درهم‌سازی سایز مجموعه Ck را کاهش می‌دهد
- به طور مثال برای 2-itemsetها می‌توانیم در زمان تولید 1-itemsetها آن‌ها را ایجاد کنیم
- در مرحله بعدی 2-itemsetها را با استفاده از یک تابع درهم‌سازی، هش می‌کنیم
- هر 2-itemset در یک bucket قرار می‌گیرد
- هر باکت که تعداد اعضای آن کمتر از support باشد، مکرر نخواهد بود

۱. درهم‌سازی

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

- اگر min-sup = 3 باشد، آنگاه

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

H_2

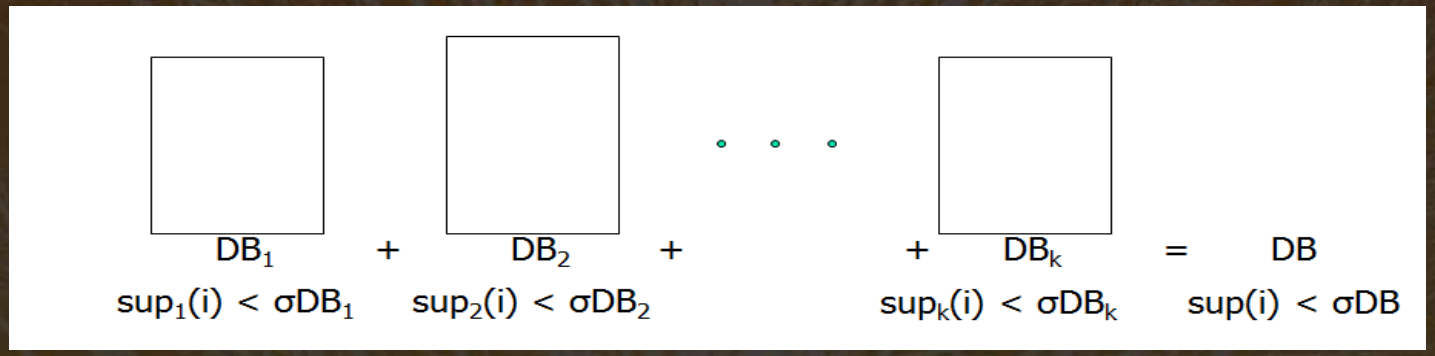
bucket address	0	1	2	3	4	5	6
bucket count	2	2	4	2	2	4	4
bucket contents	{I1, I4} {I3, I5}	{I1, I5} {I1, I5}	{I2, I3} {I2, I3} {I2, I3} {I2, I3}	{I2, I4} {I2, I4}	{I2, I5} {I2, I5}	{I1, I2} {I1, I2} {I1, I2} {I1, I2}	{I1, I3} {I1, I3} {I1, I3} {I1, I3}

۲. کاهش تراکنش

- در این روش در هر مرحله از تعداد تراکنش‌های که در مراحل بعدی باید اسکن شوند، کم می‌شود
- ایده: یک تراکنش که شامل هیچ k-itemset پرتکرار نباشد، نمی‌تواند (k+1)-itemset پرتکرار داشته باشد.
- بنابراین،این تراکنش می‌تواند برای سنجش‌های بعدی حذف شود.
- در نتیجه برای تولید j-itemsets زمانیکه j > k باشد، نیازی به در نظر گرفتن این تراکنش نیست.

۳. پارتیشن‌بندی: مرور پایگاه داده تنها دوبار

- از دو فاز تشکیل می‌شود
- فاز اول:
- در این روش داده‌ها به چند بخش تقسیم می‌شوند به طوریکه هر بخش در بتواند در حافظه اصلی قرار گیرد



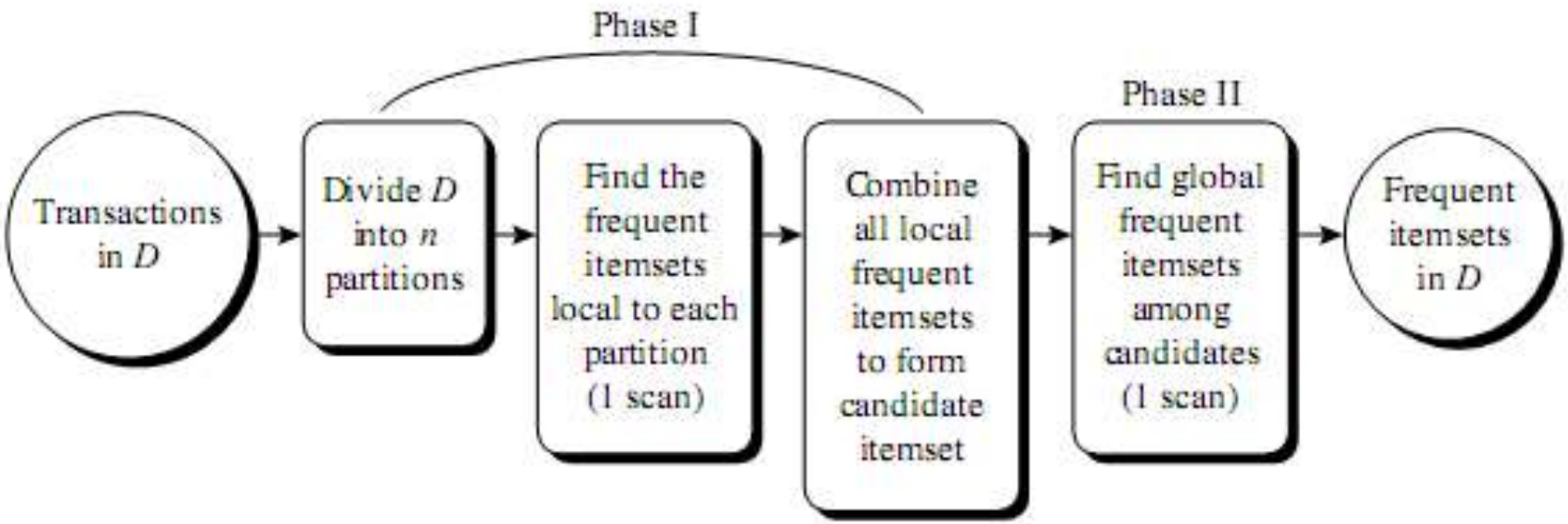
- اگر درجه پشتیبانی نسبی برابر D باشد، آنگاه مینیمم تعداد تکرار برای هر پارتیشن برابر است با (min-sup*تعداد تراکنش‌های یک بخش)
- برای هر بخش، الگوهای پرتکرار را می‌یابیم

پارتیشن‌بندی

1) الگوهای پرتکرار هر بخش (الگوهای محلی) ممکن است نسبت به کل پایگاه داده پرتکرار باشند یا نباشند
2) اما هر itemset که در پایگاه داده پرتکرار باشد، باید حداقل در یکی از پارتیشن‌های پایگاه داده پرتکرار باشد
3) نتیجه : هر itemset پرتکرار در یک بخش محلی می‌تواند در مجموعه کاندید کل پایگاه داده باشد

- فاز دوم:
- دومین مرور پایگاه داده انجام می‌گیرد
- برای هر itemset کاندید، Support واقعی آن محاسبه می‌شود تا پرتکرار بودن آن مشخص شود

پارتیشن‌بندی: ادامه

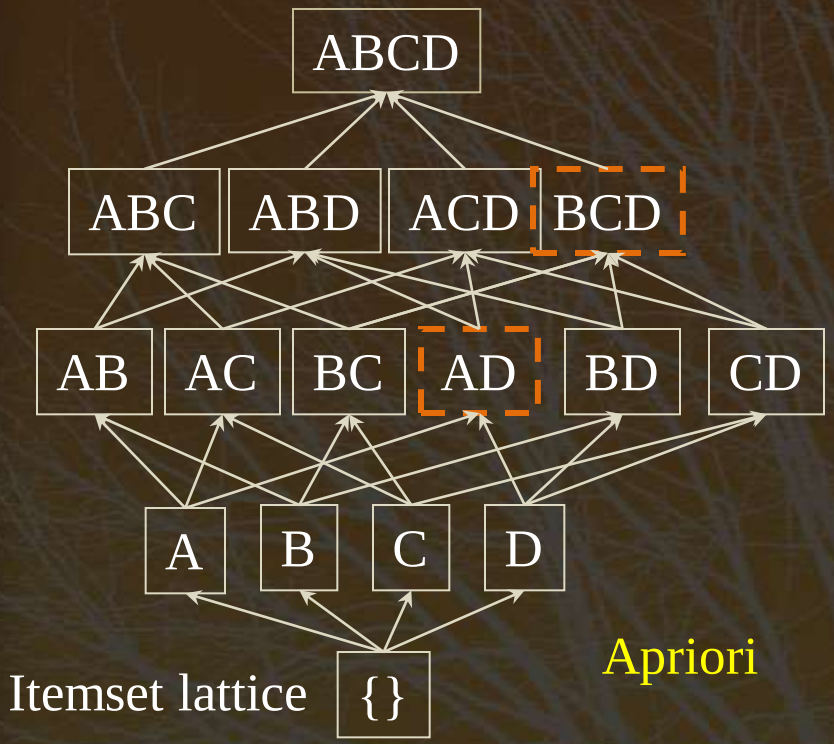


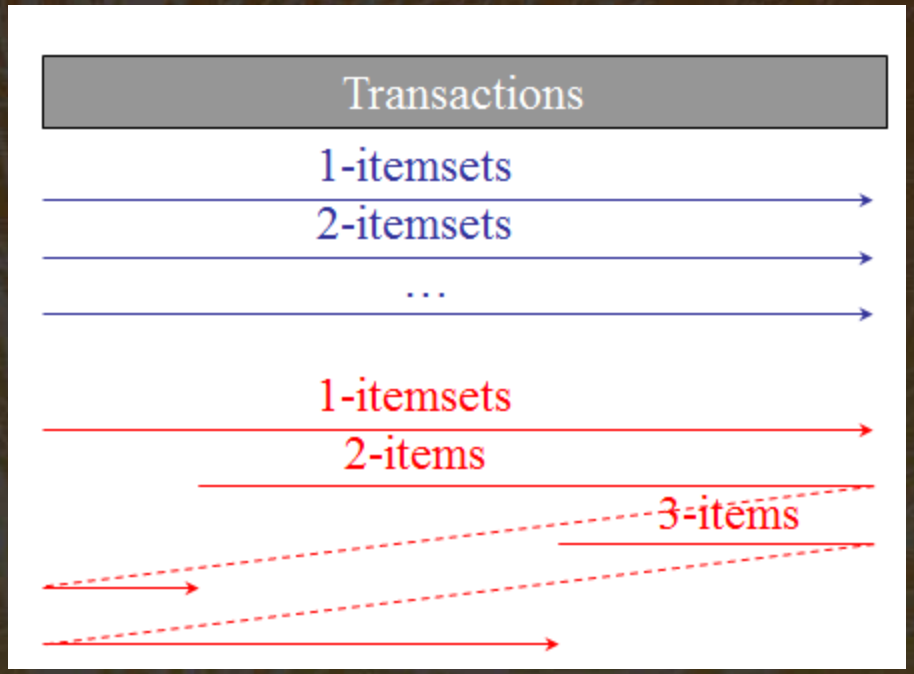
۴. نمونه‌برداری

- یک نمونه (s) از پایگاه داده اصلی انتخاب شده و الگوهای پرتکرار از این نمونه استخراج می‌شود.
- نمونه انتخابی به گونه‌ای است که در حافظه اصلی جا شده و تنها یک مرور پایگاه داده نیاز داشته باشیم
- عیب آن این است که ممکن است تعدادی از itemsetهای پرتکرار را از دست بدهیم

۵. شمارش پویا (DIC): کاهش تعداد اسکن‌ها

- زمانیکه پرتکرار بودن A و D مشخص شد، شمارش AD آغاز می‌شود.
- زمانیکه تمامی زیرمجموعه‌های به طول ۲ از BCD پرتکرار بودند، شمارش BCD آغاز می‌شود.





Apriori

روش‌های کاوش مجموعه آیتم‌های مکرر

- **Apriori** : روشی است مبتنی بر تولید-و-آزمایش مجموعه‌های کاندید
- افزایش بهره‌وری الگوریتم **Apriori**
- **FPGrowth**: روشی برای ساخت الگوی مکرر از طریق گسترش آن‌ها

(Frequent Pattern-Growth Approach)

- **ECLAT**: کاوش الگوی مکرر بر اساس چیدمان داده‌ها به صورت عمودی

روش رشد–الگو (Pattern-Growth)

- **این الگوریتم به صورت افزایشی رشد می‌کند (Growth)**
- **هدف:** یافتن itemsetهای پرتکرار بدون تولید مجموعه‌های کاندید

- **معایب روش Apriori**
  - جستجوی اول سطح (Breadth-first search)
  - تولید مجموعه‌های کاندید و تست آن‌ها
    - بعضی مواقع تعداد کاندیدها بسیار زیاد است

- **روش FP-growth**
  - جستجوی اول عمق (Depth-first search)
  - از تولید صریح مجموعه‌های کاندید جلوگیری می‌کند

الگوریتم FP-growth

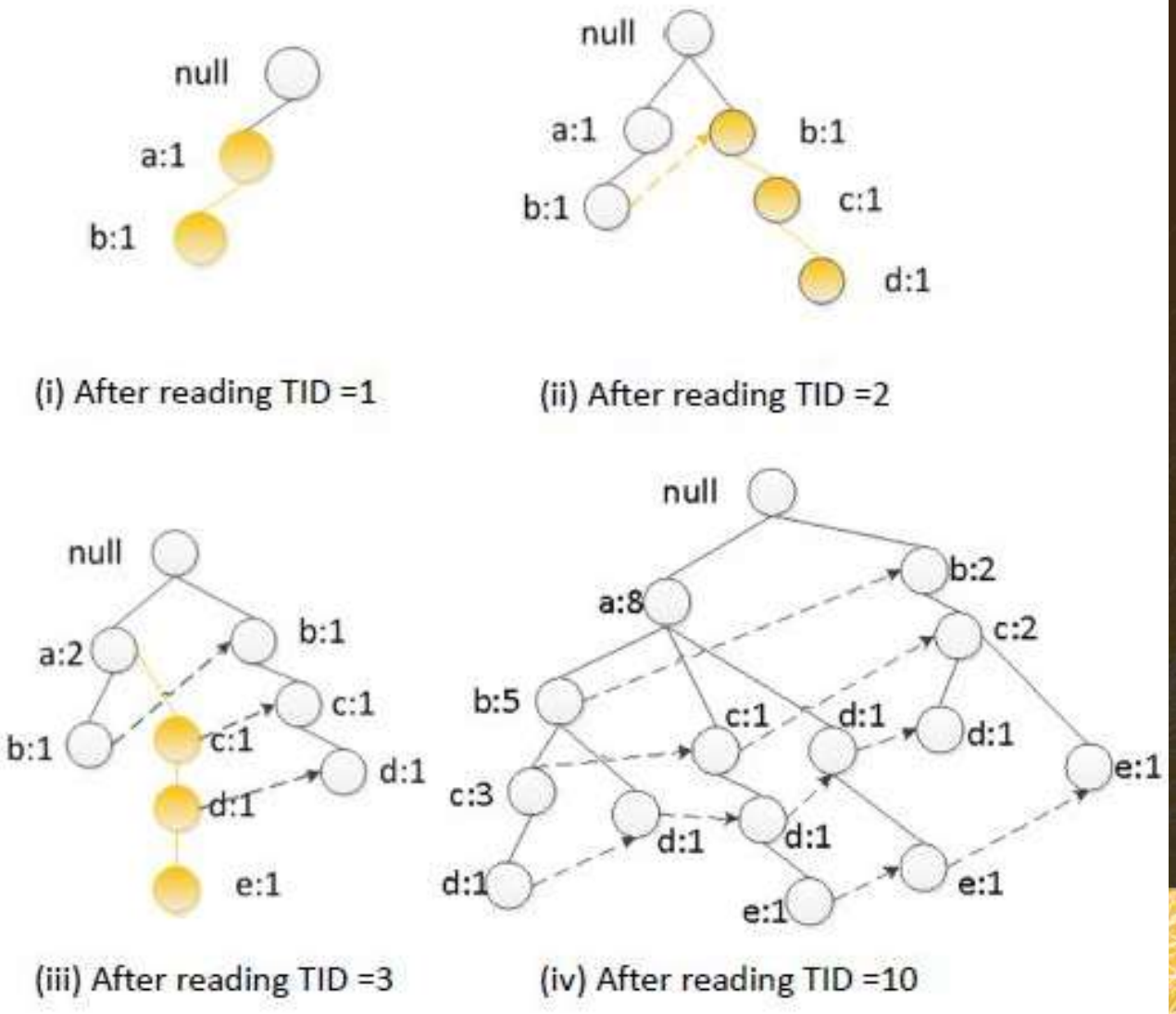
- الگوریتم در دو فاز انجام می‌گیرد
  ۱. تولید درخت FP-tree
  ۲. از روی درخت FP-tree تعدادی Conditional FP-tree ساخته می‌شود

۱. ساخت درخت FP-tree

a) در مرحله اول مانند Apriori مجموعه‌های 1-itemset را ایجاد می‌شود
b) این مجموعه بر اساس support count آن‌ها مرتب می‌شود
c) برای تمام تراکنش‌ها اعمال زیر را انجام دهید
  – اقلام موجود در هر تراکنش را بر اساس ترتیب Support count آن‌ها مرتب کنید
  – درخت FP-tree را تکمیل کنید

الگوریتم FP-growth (مثال)

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}



(i) After reading TID =1

(ii) After reading TID =2

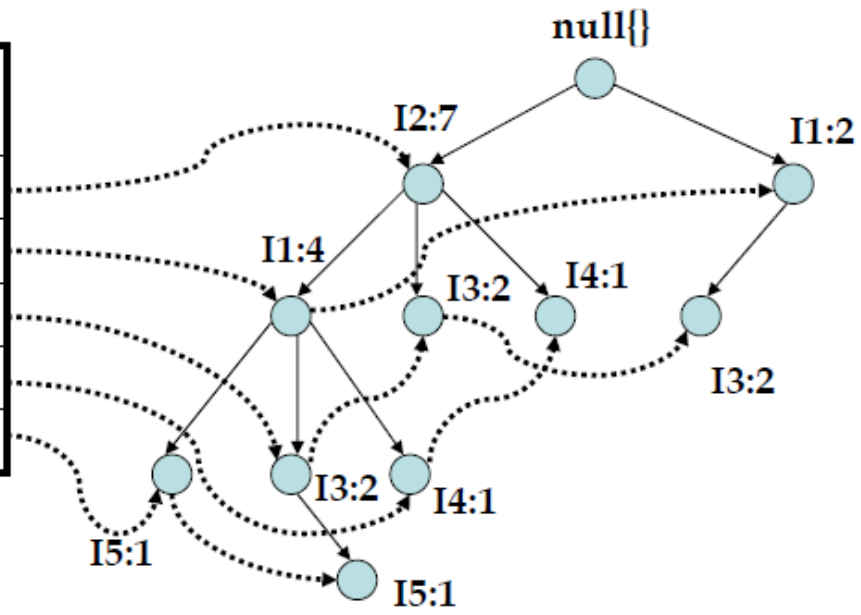
(iii) After reading TID =3

(iv) After reading TID =10

مثالی دیگر

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

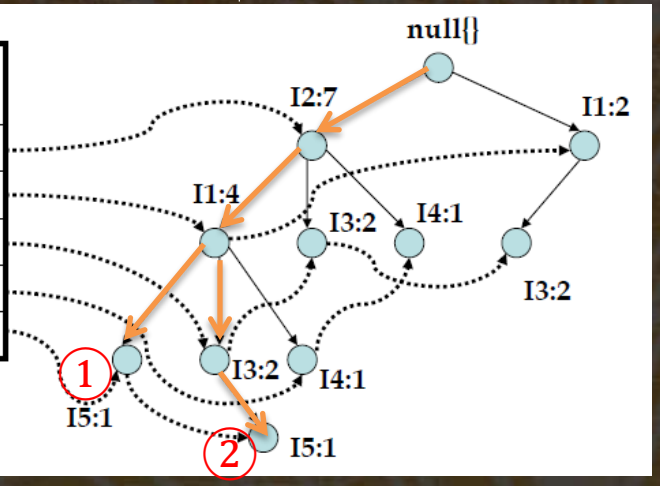
Item Id	Sup Count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



فاز دوم الگوریتم FP-Growth

- فاز دوم الگوریتم استخراج درخت‌های شرطی و الگوهای مکرر از آن‌ها است
- مراحل به شرح زیر است
- از برگ‌ها درخت FP-tree شروع می‌کنیم و همه برگ‌ها را بررسی می‌کنیم
- به طور مثال برای رسیدن به I5 دو مسیر در درخت داریم
- 1 : (I2, I1, I3, I5:1)
- 2 : (I2, I1, I5:1)
- حال I5 را پسوند الگو می‌نامیم و
- (I2,I1)و (I2, I1, I3)
- تشکیل Conditional pattern base را میدهند

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

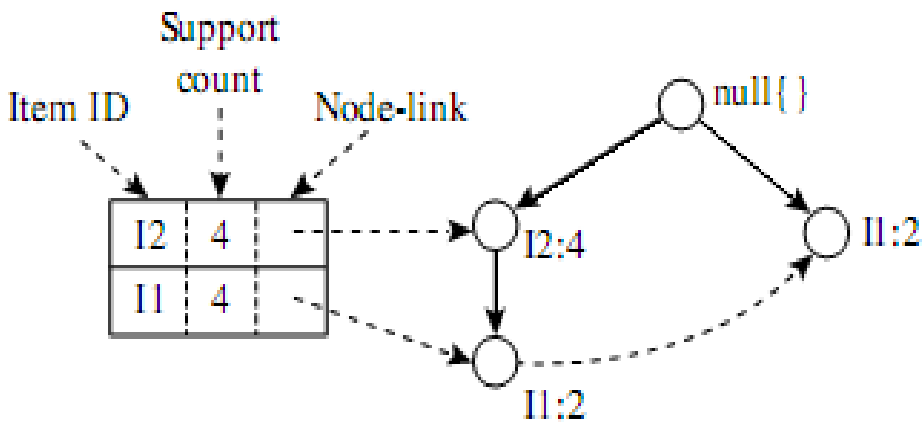


ادامه فاز دوم

- اینبار دو مقدار (I2, I1, I3, I5:1) و (I2, I1, I5:1) به عنوان تراکنش در نظر می‌گیریم و الگوریتم را ادامه می‌دهیم
- درخت حاصله دارای مسیر (I2:2 , I1:2) است
- I3 حذف شده است. **چرا؟**
- چون در دو تراکنش بالا کمتر از **min-sup** بوده است
- حال برای تولید **itemset**های مکرر کافی است ترکیبات مختلف با **I5** را در نظر بگیریم
- {I2, I5 : 2}
- {I1, I5 : 2}
- {I2, I1, I5 : 2}

مثالی دیگر

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}



- I3's conditional FP-tree

روش‌های کاوش مجموعه آیتم‌های مکرر

- **Apriori** : روشی است مبتنی بر تولید-و-آزمایش مجموعه‌های کاندید
- افزایش بهره‌وری الگوریتم **Apriori**
- **FPGrowth**: روشی برای ساخت الگوی مکرر از طریق گسترش آن‌ها

(Frequent Pattern-Growth Approach)

- **ECLAT**: کاوش الگوی مکرر بر اساس چیدمان داده‌ها به صورت عمودی

الگوریتم Eclat

- گاهی اوقات می‌توان قالب داده‌ها را عوض نمود
- در حالت عادی برای هر تراکنش مجموع کالاها را به صورت سطری مقابل آن داریم
- به این چینش اطلاعات، قالب افقی (horizontal) می‌گوییم
- در مقابل آن قالب عمودی (Vertical) به صورت زیر است

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}

الگوریتم Eclat

- برای مشخص کردن itemsetهای پرتکرار کافیست تعداد اعضای مجموعه شمرده شوند (support برابر تعداد اعضای مجموعه است)
- برای ساخت 2-itemset کافیست دو مجموعه 1-itemset را با یکدیگر ترکیب کنیم (اشتراک بگیریم)

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}

الگوریتم Eclat

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 مجموعه‌های متراکم و با طول itemsetهای بلند از قابلیت diffset استفاده می‌شود
- در این روش به جای ذخیره‌سازی اشتراک دو مجموعه، اختلاف آن‌ها ذخیره می‌شود

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

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

اختلاف آن‌ها فضای کمتری لازم دارد

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

فصل ششم: الگوهای مکرر و قوانین انجمنی

- مفاهیم اولیه
- روش‌های کاوش مجموعه اقلام مکرر ($Frequent\ Itemset$)
- چه الگوهایی از دید ما جذاب هستند؟ (روش‌های ارزیابی الگوها)
- خلاصه فصل

قوانین جذاب

- تا اینجا تنها تعریف قوانین قوی را داریم
- متاسفانه گاهی اوقات قوانین قوی از دیدگاه داده‌کاو جذاب نیستند
- به خصوص زمانی که min-sup و min-conf کم باشد
- بنابراین به روش‌های ارزیابی بهتری برای قوانین تولید شده داریم

قوانین قوی لزوما جذاب نیستند

- به طور مثال فرض کنید مجموعه داده شامل
- ۱۰،۰۰۰ تراکنش است
- ۶۰۰۰ ترکانش شامل بازی‌های کامپیوتری
- ۷۵۰۰ تراکنش شامل فیلم ویدیویی
- ۴۰۰۰ تراکنش شامل هر دو مورد است
- فرض کنید

- $Min - \sup = 30\% \quad Min - conf = 60\%$
- $Buys(X, \text{“}Computer\ games\text{”}) \Rightarrow Buys(X, \text{videos})$
- $[support = 40\%, \quad Confidence = 66\%]$

قوانین قوی لزوما جذاب نیستند

- بنابراین قانون ایجاد شده یک قانون قوی است
- ولی...
- **قانون گمراه کننده است**
- **زیرا احتمال خرید فیلم ویدیویی ۷۵٪ است که از ۶۶٪ بیشتر است**
- **در واقع خرید این دو محصول باهم رابطه عکس دارند**
- **بنابراین خرید یک محصول احتمال خرید محصول دوم را کاهش می‌دهد**
- **بدون دانستن چنین موضوعی، یک تصمیم اقتصادی نادرست گرفته می‌شود**

معیارهای دیگر

- برای بهبود چارچوب support-confidence می‌توان از آنالیز همبستگی استفاده نمود
- $A \Rightarrow B\ [support, confidence, correlation]$
- بنابراین یک قانون انجمنی نه تنها با درجه پشتیبانی و اطمینان سنجیده می‌شود بلکه با همبستگی بین اقلام هم سنجیده می‌شود

- انواع معیارهای همبستگی
- معیار همبستگی $lift$
- معیار همبستگی χ^2
- معیار همبستگی $All - confidence$
- معیار همبستگی cosine

استقلال آماری

- جمعیت ۱۰۰۰ دانشجو
  - ۶۰۰ دانش‌آموز شنا کردن را می‌دانند.(S)
  - ۷۰۰ دانش‌آموز دوچرخه سواری را می‌دانند.(B)
  - ۴۲۰ دانش‌آموز هم دوچرخه سواری و هم شنا می‌دانند.(S,B)
- $P(S \cap B) = 420/1000 = 0.42$
- $P(S) * P(B) = 0.6 * 0.7 = 0.42$

- $P(S \cap B) = P(S) \times P(B)$ => استقلال آماری
- $P(S \cap B) > P(S) \times P(B)$ => ارتباط مثبت
- $P(S \cap B) < P(S) \times P(B)$ => ارتباط منفی

معیار همبستگی lift

- معیار lift با استفاده از رابطه زیر بدست می‌آید
- $lift(A,B) = \frac{P(A \cup B)}{P(A).P(B)}$
- اگر نتیجه رابطه کمتر از ۱ باشد آنگاه پیشامد A با پیشامد B همبستگی منفی دارند
- اگر بزرگتر از ۱ باشد آنگاه A و B همبستگی مثبت دارند و به این معناست که پیشامد یکی بر پیشامد دیگری دلالت دارد
- اگر نتیجه برابر ۱ باشد، آنگاه A و B مستقل هستند و هیچ همبستگی بین آن‌ها وجود ندارد

#	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)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
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)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$