بسمه تعالی

داده کاوی

Data mining

مدرس

فاطمه دارائی

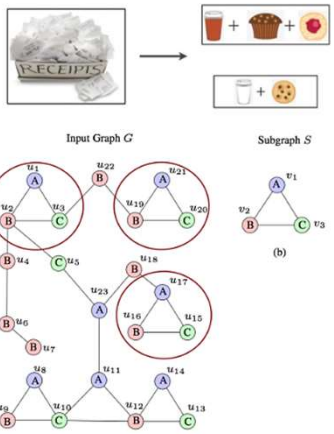
f_daraei@semnan.ac.ir

https://fdaraei.profile.semnan.ac.ir

1

فصل سوم: الگوهای مکرر و قوانین انجمنی

الگوی مکرر: الگویی (مجموعه‌ای از آیتم‌ها، زیرترتیب‌ها، زیرساختارها و غیره) که به طور مکرر در یک مجموعه داده اتفاق می‌افتد.



- چه محصولاتی اغلب با هم خریداری شده‌اند؟
- پس از خرید یک کامپیوتر، خریدهای بعدی چه چیزهایی هستند؟
- چه نوع DNA به این داروی جدید حساس است؟

- اولین بار در سال ۱۹۹۳ توسط Agrawal در زمینه
- **کاوش مجموعه آیتم‌های مکرر** پیشنهاد شد.

2

دانشگاه سمنان
Semnan University
پردیس فرزانگان

کاربردها

- تحلیل سبد خرید (Market Basket Analysis)
- بازاریابی متقاطع (Cross-marketing)
- مدیریت قفسه (Shelf management)
- تحلیل وب‌لاگ (تحلیل جریان کلیک)
- تحلیل توالی (DNA (DNA sequence analysis)
- استخراج ارتباطات کلمات
- همکاران و کلمات کلیدی در مقالات علمی

3

دانشگاه سمنان
Semnan University
پردیس فرزانگان

مفاهیم پایه

مجموعه آیتم: Itemset یک مجموعه از آیتم‌ها

مجموعه kآیتم: k-itemset یک مجموعه آیتم با kآیتم.

تعداد پشتیبانی: Support count تعداد تراکنش‌هایی که شامل یک مجموعه آیتم هستند.

نسبت پشتیبانی: Support ratio کسری از تراکنش‌هایی که شامل یک مجموعه آیتم هستند.

مجموعه آیتم مکرر: Frequent itemset یک مجموعه آیتم که پشتیبانی آن بزرگتر یا مساوی یک آستانه حداقل پشتیبانی (minsup threshold) است.

جدول ۱-۴: نمونه‌ای از یک پایگاه داده‌ی تراکنشی فروشگاه

TID	Items
1	{Bread, Milk}
2	{Bread, Egg, Butter, Cheese}
3	{Milk, Butter, Cheese, Chicken}
4	{Bread, Cheese, Chicken}
5	{Bread, Milk, Butter, Cheese}

4

دانشگاه سمنان
Semnan University

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

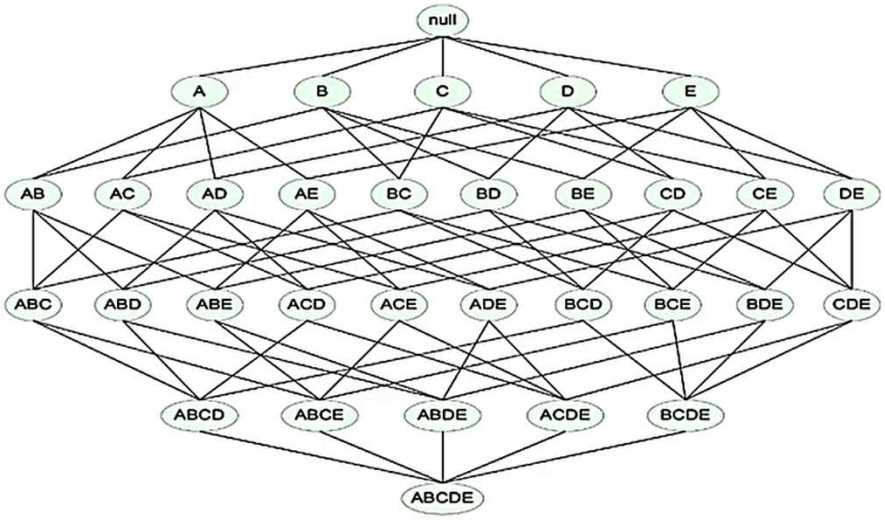
1-itemset	2-itemset	3-itemset
Support count ({C}) = 3 Support ratio ({C})= 3/4	Support count ({B, C}) = 2 Support ratio ({B,C})= 2/4	Support count ({B, C, E}) = 2 Support ratio({B,C,E}) = 2/4

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

5

دانشگاه سمنان
Semnan University

مجموعه آیتم مکرر: Frequent itemset



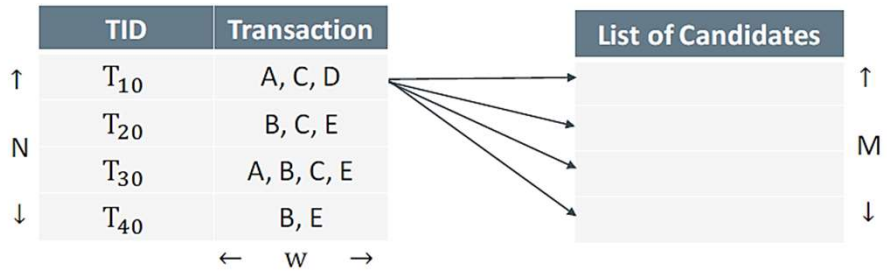
6

رویکرد Brute-force

هر مجموعه اقلام در شبکه یک مجموعه اقلام مکرر کاندید است.

با اسکن و جستجوی پایگاه داده تعداد پشیبانی هر نامزد و کاندید را بشمارید.

هر تراکنش را با هر کاندید مطابقت دهید – پیچیدگی ~ O(NMw) <= گران از مرتبه $M = 2^d$ ☹



7

الگوریتم Apriori

الگوریتم دو ورودی دیتابیس و minsub را دارد.

اگر یک مجموعه اقلام مکرر نباشد، subset های آن حتما مکرر نمی باشد.

- در ابتدا، یک بار دیتابیس را اسکن کنید تا مجموعه ۱ موردی مکرر frequent 1-itemset را دریافت کنید.
- k = 1 را تنظیم کنید. (شروع الگوریتم)
- مجموعه آیتم های کاندید با طول (k+1) را از مجموعه آیتم های k مکرر ایجاد کنید.
- کاندیداها را در دیتابیس آزمایش کنید تا مجموعه های مکرر (k+1)-اقلام را بدست آورید.
- به k یکی اضافه کنید. (بدنه تکرار)
- زمانی خاتمه دهید که هیچ مجموعه ای مکرر یا کاندید ایجاد نشود. (شرط توقف)

8

دانشگاه سمنان
Semnan University

مثال ۱ الگوریتم Apriori

TID	Transaction
T_{10}	A, C, D
T_{20}	B, C, E
T_{30}	A, B, C, E
T_{40}	B, E
minsup = 2	

Scan D to count each item →

C_1

Itemset	Sup.count
A	2
B	3
C	3
D	1
E	3

Compare supports with minsup →

L_1

Itemset	Sup.count
A	2
B	3
C	3
E	3

9

دانشگاه سمنان
Semnan University

ادامه مثال

Generate C_2 candidates from L_1 →

C_2

Itemset
A B
A C
A E
B C
B E
C E

Scan D for count of each candidate →

C_2

Itemset	Sup. count
A B	1
A C	2
A E	1
B C	2
B E	3
C E	2

Compare supports with minsup →

L_2

Itemset	Sup.count
A C	2
B C	2
B E	3
C E	2

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

دانشگاه سمنان
Semnan University

ادامه مثال

Generate C_3 candidates from L_2 →

C_3

Itemset
B C E

→ Scan D for count of each candidate →

C_3

Itemset	Sup. count
B C E	2

→ Compare support with minsup →

L_3

Itemset	Sup. count
B C E	2

⇨

Frequent Itemsets

$L_1 \cup L_2 \cup L_3$

11

دانشگاه سمنان
Semnan University

چگونه در الگوریتم Apriori کاندیدها را تولید کنیم؟

۱) ادغام با خودش L_{k-1} self joining

C_k با ادغام و پیوستن L_{k-1} با خودش تولید می شود.

۲) هرس کردن pruning

هر (k-1) مجموعه آیتم که پرکاربرد نیست، نمی‌تواند زیرمجموعه‌ای از یک k مجموعه آیتم پرکاربرد باشد.

"اصل هرس آپریوری."

Example

L_3={abc, abd, acd, ace, bcd}
Self-joining: L_3*L_3
abcd from abc and abd
acde from acd and ace
Pruning:
acde is removed because ade is not in L_3
C_4 = {abcd}

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

12

دانشگاه سمنان
Semnan University

ادامه مثال ۲

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	

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 with minsup count →

L_1

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

13

دانشگاه سمنان
Semnan University

ادامه مثال ۲

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 with minsup count →

L_2

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

14

دانشگاه سمنان
Semnan University

ادامه مثال ۲

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 with minsup count →

L_3

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

Generate C_4 candidates from L_3 →

C_4

Itemset
∅

⇨

Frequent Itemsets
$L_1 \cup L_2 \cup L_3$

15

دانشگاه سمنان
Semnan University

بهبود الگوریتم Apriori

اصلی ترین چالش های محاسباتی

مرور چندباره داده های تراکنشی

تعداد زیاد کاندیدهای تولید شده

محاسبه support برای همه کاندیدها زمانبر است.

بهبود دادن Apriori: ایده های کلی

کاهش تعداد مرورهای پایگاه داده

کاهش تعداد کاندیدها

تسهیل شمارش support کاندیدها

16

افزایش بهره وری الگوریتم Apriori

1. استفاده از توابع درهم سازی (Hash based)
2. کاهش تراکنش (Transaction Reduction)
3. پارتیشن بندی (Partitioning)
4. نمونه برداری (Sampling)
5. شمارش پویای Itemset ها (Dynamic itemset counting)

17

فرمت داده افقی Horizontal Data Format

تبدیل فرمت در دیتابیس باعث می شود محاسبات کمتر و ساده تر شود.

گاهی اوقات می توان قالب داده ها را عوض کرد و یا در فرمت ها مختلفی نمایش داد.

در حالت عادی برای هر تراکنش، مجموع کالاها را به صورت سطری مقابل آن داریم.

به این چینش اطلاعات، فرمت داده افقی می گوییم.

TID	Transaction
T_1	A, B, C
T_2	B
T_3	B, C, D
T_4	A, D

Horizontal Format 1

TID	Item
T_1	A
T_1	B
T_1	C
T_2	B
T_3	B
T_3	C
…	…

Horizontal Format 2

TID	Transaction			
	A	B	C	D
T_1	1	1	1	0
T_2	0	1	0	0
T_3	0	1	1	1
T_4	1	0	0	1

Horizontal Format 3

18

فرمت داده عمودی Vertical Data Format

این فرمت به طور گسترده ای در موتورهای جستجو مورد استفاده قرار می‌گیرد.

Horizontal: document → words
Vertical: words → document

TID	Transaction
T_1	A, B, C
T_2	B
T_3	B, C, D
T_4	A, D

Horizontal Format 1

item	Transaction				
		T_1	T_2	T_3	T_4
A	T_1 , T_4	1	0	0	1
B	T_1 , T_2 , T_3	1	1	1	0
C	T_1 , T_3	1	0	1	0
D	T_3 , T_4	0	0	1	1

Vertical Format

19

الگوریتم Eclat

ECLAT (Equivalence Class Transformation) (تبدیل کلاس هم‌ارز) مشابه الگوریتم Apriori است،

- اما از فرمت داده‌های عمودی استفاده می‌کند.
- ابتدا، با یک‌بار پیمایش پایگاه داده، فرمت افقی را به فرمت عمودی تبدیل می‌کنیم.
- تعداد support یک مجموعه آیتم به سادگی برابر با طول مجموعه است = دستیابی به مجموعه آیتم‌های مکرر ۱آیتمی.
- مرحله تولید کاندیدها مشابه الگوریتم Apriori است.

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

Horizontal

vs

Vertical

Item	TID
I1	T1, T4, T5, T7, T8, T9
I2	T1, T2, T3, T4, T6, T8, T9
I3	T3, T5, T6, T7, T8, T9
I4	T2, T4
I5	T1, T8

20

لیست‌های مربوط به مجموعه آیتم‌های مکرر kآیتمی را با هم اشتراک می‌گیریم تا لیست‌های مربوط به مجموعه آیتم‌های (k+1)- آیتمی را به دست آوریم.

Item	TID
I1	T1, T4, T5, T7, T8, T9
I2	T1, T2, T3, T4, T6, T8, T9
I3	T3, T5, T6, T7, T8, T9
I4	T2, T4
I5	T1, T8

$L_1 = \{I1, I2, I3, I4, I5\}$

TID	Items
I1, I2	T1, T4, T8, T9
I1, I3	T5, T7, T8, T9
I1, I4	T4
I1, I5	T1, T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8
I3,I4	∅
I3, I5	T8
I4,I5	∅

$L_2 = \{I_1I_2, I_1I_3, I_1I_5, I_2I_3, I_2I_4, I_2I_5\}$

$C_3 = \{I_1I_2I_3, I_1I_2I_5\}$

TID	Items
I1, I2, I3	T8, T9
I1, I2, I5	T1, T8

$C_4 = \emptyset$

21

قوانین انجمنی

با توجه به مجموعه ای از تراکنش های T، هدف از استخراج قوانین انجمنی این است که همه قوانین قوی پیدا شود.

$$\text{support} \geq \text{minsup threshold}$$
$$\text{confidence} \geq \text{minconf threshold}$$

Support: درصدی از تراکنش ها که حاوی A و B است.

$$\text{Support}(A \rightarrow B) = \text{Support}(A \cup B) = \text{Support}(B \rightarrow A)$$

Confidence: درصدی از تراکنش هایی است که اگر حاوی A باشند، B نیز در آنها وجود داشته باشد.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

Support پوشش قانون را نشان می دهد و Confidence اطمینان، دقت و درستی قانون را.

22

دانشگاه سمنان
Semnan University
پردیس فرزانگان

مثال

TID	Items
1	A, B
2	A, C, D, E
3	B, C, D, F
4	A, B, C, D
5	A, B, C, F

{B, C} → {D} (sup = 0.4, conf = 0.67)

{B, D} → {C} (sup = 0.4, conf = 1.0)

{C} → {B, D} (sup = 0.4, conf = 0.5)

{B} → {C} (sup = 0.6, conf = 0.75)

23

دانشگاه سمنان
Semnan University
پردیس فرزانگان

مقایسه Support وConfidence

I. Support and confidence are both high.

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

A → B
sup = 1
conf = 1

II. Support and confidence are both low.

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

A → B
sup = 0
conf = 0

24

دانشگاه سمنان
Semnan University

مقایسه Support وConfidence

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

$G \rightarrow A$

$sup = \frac{1}{6}$

$conf = 1$

IV. Confidence is low and support is high.

It is impossible because:
$Sup \leq Conf$

$$Conf(A \rightarrow B) = \frac{P(A,B)}{P(A)} = \frac{Sup(A \rightarrow B)}{P(A)}$$
$$P(A) \leq 1$$

25

دانشگاه سمنان
Semnan University

استخراج قوانین انجمنی

رویکرد Brute-force:

- تمام قوانین ارتباط ممکن را فهرست کنید.
- پشتیبانی و اطمینان برای هر قانون را محاسبه کنید.
- قوانینی را که در آستانه minsup و minconf ناموفق هستند، هرس کنید.

غیر عملی است، زیرا ما می توانیم قوانین متفاوتی را برای هر کدام ایجاد کنیم و مجموعه آیتم ها، و تعداد نمایی از مجموعه آیتم ها وجود دارد!

26

استخراج قوانین انجمنی

داده کاو در هنگام استخراج قوانین دو آستانه زیر را تعیین می نماید:

min-sub

min-conf

قوانینی که درجه پشتیبانی و اطمینان آنها بیش از این دو مقدار باشند، به عنوان قوی strong شناخته می شوند.

****

استفاده ازمجموعه الگوهای مکرر

- همه مجموعه موارد را با $support \geq minsup$ ایجاد کنید.
- تولید قوانین با اطمینان بالا از هر مجموعه آیتم های مکرر، که در آن هر قانون یک پارتیشن بندی باینری از مجموعه آیتم های مکرر است.

27

مثال

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

دانشگاه سمنان
Semnan University

ادامه مثال

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

29

دانشگاه سمنان
Semnan University

ادامه مثال

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