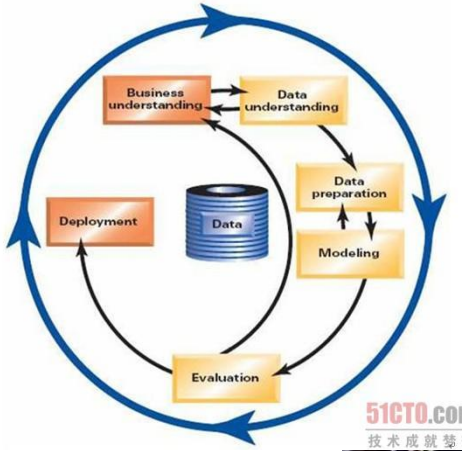
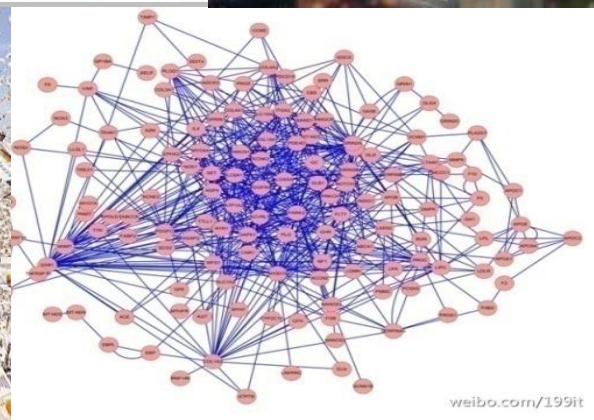




Chapter 12 Association Rules

2025 Autumn

Lei Sun

01 Support and Confidence

02 Apriori Property

03 Apriori Algorithm

04 Lift

05 Elcat

Rakesh Agrawal

Fast Algorithms for Mining Association Rules

Basic Idea

The Association rule is very useful in analyzing datasets.

The data is collected using bar-code scanners in supermarkets. Such databases consists of a large number of transaction records which list all items bought by a customer on a single purchase.

So the manager could know if certain groups of items are consistently purchased together and use this data for adjusting store layouts, cross-selling, promotions based on statistics.

bread and milk → pop and peanut butter [s=50%,c=90%]

Basic Idea

Used in many recommender systems

Basic Idea

Applicable beyond retail

- Optional services purchased by telecom customers (call waiting, speed calling, ISDN, etc) help determine how to bundle these services to maximize revenue.
- Banking services used by customers (money market accounts, CDs, car loans, etc) to help identify customers likely to want other services.
- Unusual combinations of insurance claims can be a sign of fraud.
- Medical patient histories can provide indications of complications based on certain combinations of treatments.

Basic Idea

Given the rule: “If Barbie then candy”

We have several possible actions

- Give candy coupon with Barbie purchase
- Create a Barbie candy with picture of Barbie on candy
- Put candy and Barbie close together
- Increase price of candy during Barbie sale
- Package Barbies and candies together
- Put candy and Barbie far apart

02

Concept

✓ Association rules can be automatically generated:

- Find frequent patterns by sifting through the data
- Rules are usually of the form “If A then B”

✓ In the rule “if A then B”, the condition A is the antecedent(前件) and the B is the consequent（后件）.

- Antecedent and consequent are disjoint (i.e., have no items in common) $A \cap B = \phi$

bread and milk $\Rightarrow$ pop and peanut butter [s=50%,c=90%]

[s=50%,c=90%]

02

Concept

Transaction → A collection of items that occur together

Itemset(项集) → a set of items / the items (e.g., products) comprising the antecedent or consequent

K-itemset (K项集) → an itemset containing K items. E.g. {egg,milk} is a 2-itemset.

Frequent itemset (频繁项集) → itemset is frequent if the corresponding support count is greater than the minimum support count.

Measures

customers	Items purchased
1	OJ, soda
2	Milk, OJ, window cleaner
3	OJ, detergent
4	OJ, detergent, soda
5	window cleaner, soda

A B
If a customer purchases soda, then he also purchases OJ

Support: – the ratio of the number of times an item occurs in the transactions to the total number of transactions.
支持度

$$\text{support}\ (A \Rightarrow B) = P(A \cap B) = \frac{2}{5} = 40\%$$

means total 40% of transactions in the database follow the rule.

Confidence: The percentage of times that the items in the antecedent appear in the same transaction as the items in the consequent.
置信度

fraction of those with Soda (A) who also have OJ (B):

$$P(B \mid A) = \frac{2}{3} \approx 67\% \qquad P(B \mid A) = \frac{\text{support}(A \Rightarrow B)}{\text{sup } port(A)}$$

67% of the customers who purchased soda also bought OJ

Measures

customers	Items purchased
1	OJ, soda
2	Milk, OJ, window cleaner
3	OJ, detergent
4	OJ, detergent, soda
5	window cleaner, soda

If a customer purchases soda, then he also purchases OJ

$$confidence(soda) \rightarrow OJ) = \frac{support(soda \rightarrow OJ)}{support(soda)} = \frac{2}{3}$$

A B

If a customer purchases OJ, then he also purchases soda

$$\text{support}\ (\text{OJ} \Rightarrow \text{soda}) = 40\%$$

$$\text{confidence}\,(\text{OJ} \Rightarrow \text{soda}) = \frac{\text{support}(A \Rightarrow B)}{\text{sup}\ port(A)} = \frac{2}{4}$$

Measures

Support	Confidence(conditional possibility)
a measure of the number of times an item set appears in a dataset.	a measure of the likelihood that an itemset will appear if another itemset appears.
Support is used to identify itemsets that occur frequently in the dataset.	Confidence is used to evaluate the strength of a rule.
Support is interpreted as the percentage of transactions in which an item set appears.	Confidence is interpreted as the percentage of transactions in which the right itemset appears given that the left itemset appears.

If a rule satisfies both minimum support and minimum confidence, it is a strong rule.

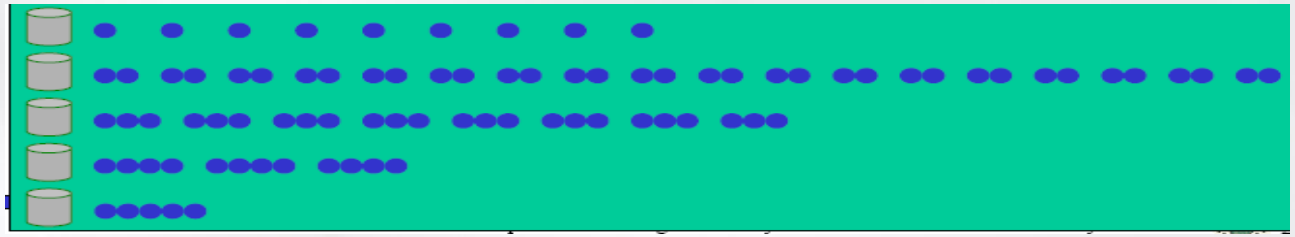
Apriori Algorithm

Two Main steps

① **Frequent itemsets generation:** Find all itemsets that have minimum support (frequent itemsets)

-- Generate frequent k-itemsets

-- Generate candidate (k+1)-itemsets from frequent k-itemsets



② **Expand and then prune itemsets**

--Check all subsets of an itemset are frequent or not and if not frequent remove that itemset using Apriori property.

③ **generate rules:** Use frequent itemsets to generate rules based on confidence

Apriori Algorithm

Apriori Property(Antimonotonicity)

All subsets of a frequent itemset must be frequent(Apriori property).
If an itemset is infrequent, all its supersets will be infrequent.

- If an itemset G does not satisfy the min-support threshold (P(G) < min-sup), then it is not frequent.
- If an item A is added to the itemset G, then the resulting itemset($G \cup A$) can not occur more frequently than G. Therefore, is not frequent either; that is $P(G \cup A)$<min_sup.

Apriori Algorithm

Join Step

Every itemset must be sorted.

① This step generates 2 itemset from 1-itemsets by joining each item with itself.

② **Condition:** Condition of joining $L_1[k-1]$ and $L_2[k-1]$ is that it should have (K-2) elements in common and $l_1[k-1] < l_2[k-1]$. For L3, first two element should match. Then

$$L_1[k-1] \bowtie L_2[K-1]$$

The condition simply ensures that no duplicates are generated.

TID	List of item_ID's
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

Scan D for count of each candidate

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

C1

Compare candidate support count with minimum support count

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

L1

Generate C2 candidates from L1

Itemset
{I1,I2}
{I1,I3}
{I1,I4}
{I1,I5}
{I2,I3}
{I2,I4}
{I2,I5}
{I3,I4}
{I3,I5}
{I4,I5}

C2

Scan D for count of each candidate

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

C2

Compare candidate support count with minimum support count

Itemset	sup.count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

L2

Generate C3candidates from L2

Itemset
{I1,I2,I3}
{I1,I2,I5}

C3

Scan D for count of each candidate

Pruning?

Itemset	sup.count
{I1,I2,I3}	2
{I1,I2,I5}	2

C3

Compare candidate support count with minimum support count

Itemset	sup.count
{I1,I2,I3}	2
{I1,I2,I5}	2

L3

Apriori Algorithm

Itemset	sup.count
{I1,I2,I3}	2
{I1,I2,I5}	2

- Prune using Apriori property: all nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?
- Join: C4={I1, I2, I3, I5}, however, subset{I3,I5} is not a member of L2.
- Therefore, C4= ∅, and L3={{I1,I2,I3}, {I1,I2,I5}}, after pruning.

Apriori Algorithm

Find all possible association rules

- ✓ For each frequent itemset G, generate all nonempty subsets of G.
- ✓ For every nonempty subsets s of G, output the rules " $s \Rightarrow (\mathrm{G} - s)$ "
  - Compute confidence for each rule.
  - Prune rules that do not satisfy minConf thresholds
  - Choose the rules with good supports and confidences.

Pruning reduces the number of itemsets to consider

Apriori Algorithm

frequent itemset l={I1,I2,I5}

$$I1 \wedge I2 \Rightarrow I5 confidence = 2/4 = 50\%$$

$$I1 \wedge I5 \Rightarrow I2, confidence = 2/2 = 100\%$$

$$I2 \wedge I5 \Rightarrow I1, confidence = 2/2 = 100\%$$

$$I1 \Rightarrow I2 \wedge I5, confidence = 2/6 = 33\%$$

$$I2 \Rightarrow I1 \wedge I5, confidence = 2/7 = 19\%$$

$$I5 \Rightarrow I1 \wedge I2, confidence = 2/2 = 100\%$$

TID	Items
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

Apriori Algorithm

If {A,B,C,D} is a frequent itemset, candidate rules:

ABC → D, ABD → C, ACD → B, BCD → A, A → BCD, B → ACD,
C → ABD, D → ABC, AB → CD, AC → BD, AD → BC, BC → AD,
BD → AC, CD → AB

If |L| = k, then there are $2^k - 2$ candidate association rules (ignoring L → ∅ and ∅ → L)

Apriori Algorithm

frequent itemset l={I1,I2,I5}

$c(I1\ I2 \rightarrow I5) >= c(I1 \rightarrow I2\ I5)$

L = {A,B,C,D}:

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

confidence of rules generated from the same itemset has an anti-monotone property(反单调性)

Misleading "strong" association rule

- ✓ **Maybe a type of random association rule**

milk⇒male [S=40%,C=40%]

If Minsup is 20%, this is a strong rule.

	male	female	Sum (row)
Buy milk	400	600	1000
Not buy	0	0	0
Sum(col.)	400	600	1000

But, the proportion of male customers is also 40%.

The proportion of male customers who purchase milk **is equal to** the proportion of all male customers.

So, no correlation between LHS and RHS

Misleading “strong” association rule

- ✓ **Negative correlation relationship**

grade(Excellent)⇒breakfast(have) S=60/180
[S=33.3%,C=60%] C=60/100

If Minsup is 20%, this is a strong rule.

	Have breakfast	Not have	Sum (row)
Excellent	60	40	100
not	66	14	80
Sum(col.)	126	54	180

Investigation between grade and having breakfast

S(have)=126/180=70% 后项支持度 70% students have breakfast

But the proportion of excellent student who have breakfast is less than the total proportion(70%).

If the proportion of excellent student who have breakfast is **more than 70%,** the rule contributes to our knowledge.

Misleading “strong” association rule

We should filter out trivial and misleading associations A → B

$$\frac{S(A \wedge B)}{S(A)} - S(B) > 0$$

(60/100)/(100/180)-70%<0

Measure--Lift

A measure of how often the antecedent and consequent occur together than expected by chance.

(Confidence) / [(item B)/ (Entire dataset)]

$$lift = \frac{congfidence(A \rightarrow B)}{support((B)} = \frac{P(B|A)}{P(B)} = \frac{support(A \rightarrow B)}{support(A)support(B)}$$

	lhs 前件		rhs 后件	support	confidence	lift
1	{pot plants}	=>	{whole milk}	0.006914082	0.4000000	1.565460
2	{pasta}	=>	{whole milk}	0.006100661	0.4054054	1.586614
3	{herbs}	=>	{root vegetables}	0.007015760	0.4312500	3.956477
4	{herbs}	=>	{other vegetables}	0.007727504	0.4750000	2.454874
5	{herbs}	=>	{whole milk}	0.007727504	0.4750000	1.858983

Measure--Lift

{Oranges} -> {Apples}(Conf.=75%,Sup.(Apples)=60%)

Lift answers the question: Are customers that buy oranges more likely to buy apples than the average customer?

75% of the customers who bought oranges also bought apples (this is the confidence of the rule). In general, 60% of all of the customers bought apples (support of {Apples}). So the answer is, yes… customers that buy oranges are more likely to buy apples than the average customer.

How much more likely? To figure that out you do .75 / .60 = 1.25. This is gives you the lift value of the rule: 1.25

Measure--Lift

$$lift = \frac{congfidence(A \rightarrow B)}{support((B)} = \frac{P(B|A)}{P(B)} = \frac{support(A \rightarrow B)}{support(A)support(B)} = \frac{P(AB)}{P(A)P(B)}$$

Discuss

Lift value near 1: indicates the rule body and the RHS appear almost as often together as expected, this means that the occurrence of the rule body has almost no effect on the occurrence of the RHS. A并不影响B出现的概率,无实际意义

greater than 1: means they appear together more than expected, positive dependence . 表示推荐（关联）商品的购买率比未推荐前有所提高

less than 1: means they appear less than expected, negative dependence.

Greater lift values indicate stronger association.

P(A ∩ B) = P(A)P(B): A and B are *independent* . *Lift=1*

How many times does B appear when A and B appear together

A和B共同出现是B出现的多少倍数

Measure--Lift

For rules like IF (A and B and C) then (D and E) need to count occurrences of multiple sets

- Support: $P(A \cap B \cap C \cap D \cap E)$
- Confidence: $\frac{\sup port}{P(A \cap B \cap C)}$
- Lift: $\frac{\text{confidence (A and B and C} \Rightarrow \text{D and E)}}{P(D \cap E)}$

Generate all rules and look for high lift, support and confidence.

Summary

Major computational challenges

✓ Huge number of candidates

The main limitation is time required to hold a vast number of candidate sets with much frequent itemsets. it is not an efficient approach for large number of datasets.

A long itemset will contain a combinational number of shorter, frequent sub-itemsets. For example, a frequent itemset of length 100, such as $\{a_1,a_2,\ldots,a_{100}\}$, contains $C_{100}^1 = 100$ Frequent 1-itemsets $a_1,a_2,\ldots,a_{100}$, C_{100}^2 frequent 2-itensets: $(a_1,a_2),(a_1,a_3),\ldots,(a_{99},a_{100}),$ and so on. The total number of frequent itemsets that it contains is thus,

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

✓ Multiple scans of transaction database

Summary

Advantages

① **Easy to interpret:**
The results of association rule mining are easy to understand and interpret, making it possible to explain the patterns found in the data.

② **Can be used in a wide range of applications:**
Association rule mining can be used in a wide range of applications such as retail, finance, and healthcare, which can help to improve decision-making and increase revenue.

Excise

What do you mean by support(M)?

A. Total number of transactions containing M

B. Total Number of transactions not containing M

C. Number of transactions containing M / Total number of transactions

D. Number of transactions not containing M / Total number of transactions

Excise

What happens if the minimum support threshold is set too high?

A. Capture more common itemsets.

B. Decrease the number of candidate rules

C. Miss interesting rare itemsets.

D. Increased computational efficiency.

Excise

What happens to a transaction that does not contain any frequent k-itemsets during future iterations?

A. It cannot contribute to frequent (k+1)-itemsets

B. It is deleted from the database

C. It is flagged for review

D. It contributes to frequent (k+1)-itemsets

Equivalence CLAss Transformation，Eclat: 等价类变换

The Eclat algorithm is defined recursively.

- The initial call uses all the single items with their tidsets.
- In each recursive call, the function IntersectTidsets verifies each itemset-tidset pair with all the others pairs to generate new candidates.
- If the new candidate is frequent, it is added to the set.
- Then, recursively, it finds all the frequent itemsets in the branch.

The algorithm searches in a DFS(深度优先搜索 Depth First Search ）manner to find all the frequent sets.

Equivalence CLAss Transformation，Eclat: 等价类变换

Eclat算法挖掘频繁项集的过程如下：

（1）通过扫描一次数据集，把水平格式的数据转换成垂直格式

（2）项集的支持度计数简单地等于项集的TID集的长度；

（3）从k=1开始，根据先验性质，使用频繁k项集来构造候选(k+1)项集；

（4）通过取频繁k项集的TID集的交，计算对应的（k+1）项集的TID集。

（5）重复该过程，每次k增加1，直到不能再找到频繁项集或候选项集。

Equivalence CLAss Transformation，Eclat: 等价类变换

TID	Items	TID	Items
T100	I1, I2, I5	T600	I2, I3
T200	I2, I4	T700	I1, I3
T300	I2, I3	T800	I1, I2, I3, I5
T400	I1, I2, I4	T900	I1, I2, I3
T500	I1, I3		

项集	TID	项集	TID
I1	{T100,T400,T500,T700,T800,T900}	I4	{T200, T400}
I2	{T100,T200,T300,T400,T600,T800,T900}	I5	{T100, T800}
I3	{T300,T500,T600,T700,T800,T900}		

Equivalence CLAss Transformation，Eclat: 等价类变换

通过取每对频繁项的TID集的交，进行挖掘。设Min_sup=2。总共进行10次交运算，导致8个非空2项集。注意，项集{I1，I4}和{I3，I5}都只包含一个事务，因此它们都不属于频繁2项集的集合。

项集	TID	项集	TID
I1, I2	{T100,T400,T800,T90}	I2, I3	{T300, T600,T800,T900}
I1, I3	{T500, T700,T800,T900}	I2, I4	{T200， T400}
I1, I4	{T400}	I2, I5	{T100,T800}
I1, I5	{T100, T800}	I3, I5	{T800}

项集	TID
I1, I2, I3	{T800, T900}
I1, I2, I5	{T100, T800}