Association Rules

The Apriori Algorithm

Andrew Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527

Tel: 319 - 335 5934 Fax: 319 - 335 5669
andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Association rules Introduction

- Mining for associations among items in a large database of sales transaction is an important database mining function.
- For example, the information that a customer who purchases a keyboard also tends to buy a mouse at the same time is represented in association rule below: *Keyboard ⇒ Mouse [support = 6%, confidence = 70%]*

Association Rules

- Based on the types of values, the association rules can be classified into two categories: Boolean Association Rules and Quantitative Association Rules
- ***Boolean Association Rule***: Keyboard ⇒ Mouse [support = 6%, confidence = 70%]
- ***Quantitative Association Rule***: (Age = 26 … 30) ⇒ (Cars =1, 2) [Support 3%, confidence = 36%]

Minimum Support Threshold

- The support of an association pattern is the percentage of task-relevant data transactions for which the pattern is true.

IF A⇒ B

$$support\ (A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{total\_\#\_of\_tuples}$$

Minimum Confidence Threshold

- Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern.

IF A ⇒ B

$$confidence(A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_containing\_A}$$

Itemset

- A set of items is referred to as itemset.
- An itemset containing *k* items is called ***k*-itemset**.
- An itemset can also be seen as a conjunction of items (or a predicate)

Frequent Itemsets

- Suppose *min_sup* is the minimum support threshold.
- An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to *min_sup*.
- If an itemset satisfies minimum support, then it is a frequent itemset.

Strong Rules

- Rules that satisfy both a minimum support threshold and a minimum confidence threshold are called strong.

Association Rule Mining

- Find all frequent itemsets
- Generate strong association rules from the frequent itemsets

Apriori Algorithm (1)

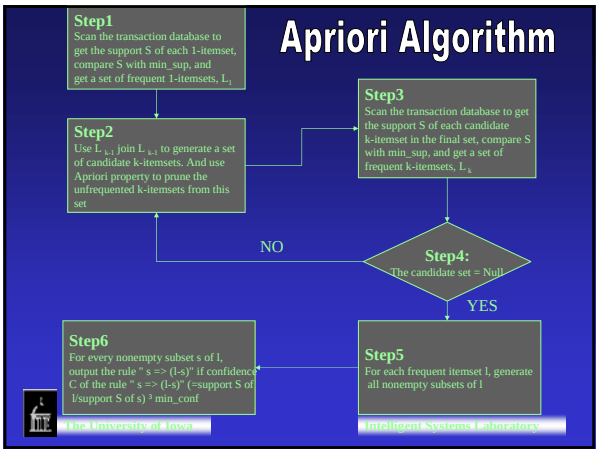
- Apriori algorithm is an influential algorithm for mining *frequent itemsets* for Boolean association rules.

Apriori Algorithm (2)

- Uses a ***Level-wise search***, where ***k*-itemsets** (An itemset that contains *k* items is a ***k-itemset*)** are used to explore (*k*+1)-itemsets, to mine frequent itemsets from transactional database for Boolean association rules.
- First, the set of frequent 1-itemsets is found. This set is denoted L1. L1 is used to find L2, the set of frequent 2-itemsets, which is used to fine L3, and so on, until no more frequent *k*-itemsets can be found.

Association rule mining process

- Find all ***frequent itemsets*:**
  - Each support **S** of these frequent itemsets will at least equal to a pre-determined min_sup (An *itemset* is a subset of items in I, like A)
- Generate ***strong association rules*** from the frequent itemsets:
  - These rules must be the frequent itemsets and must satisfy min_sup and min_conf.



Apriori Property

- Reducing the search space to avoid finding of each L_k requires one full scan of the database
- If an itemset *I* does not satisfy the minimum support threshold, *min_sup*, the *I* is not frequent, that is, P (*I*) < *min_sup*.
- If an item A is added to the itemset *I*, then the resulting itemset (i.e., I∪A) cannot occur more frequently than *I*. Therefore, I ∪A is not frequent either, that is, P (I ∪A) < *min_sup*.

The University of Iowa Intelligent Systems Laboratory

Example 1

Apriori Algorithm

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

1-Itemsets	Sup-count
I1	6
I2	7
I3	6
I4	2
I5	2

$$support\ (A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{total\_\#\_of\_tuples}$$

The University of Iowa Intelligent Systems Laboratory

Example 1

Apriori Algorithm

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

1-Itemsets	Sup-count
1	6
2	7
3	6
4	2
5	2

Join

2-Itemsets	Sup-count
1, 2	4
1, 3	4
1, 4	1
1, 5	2
2, 3	4
2, 4	2
2, 5	2
3, 4	0
3, 5	1
4, 5	0

Min support = 2

Prune

Frequent 2-Itemsets	Sup-count
1, 2	4
1, 3	4
1, 5	2
2, 3	4
2, 4	2
2, 5	2

Join

Frequent 3-Itemsets	Sup-count
1, 2, 3	2
1, 2, 5	2

The University of Iowa Intelligent Systems Laboratory

Example 2

Problem data

An example with a transactional data D contents a list of 5 transactions in a supermarket.

TID	List of items (item_IDs)
1	Beer(I1), Diaper(I2), Baby Powder(I3), Bread(I4), Umbrella(I5)
2	Diaper(I2), Baby Powder(I3)
3	Beer(I1), Diaper(I2), Milk(I6)
4	Diaper(I2), Beer(I1), Detergent(I7)
5	Beer(I1), Milk(I6), Coca Cola (I8)

The University of Iowa Intelligent Systems Laboratory

Solution Procedure

Step 1 ***min_sup*** **= 40% (2/5)**

C1 → L1

Item_ID	Item	Support
I1	Beer	4/5
I2	Diaper	4/5
I3	Baby powder	2/5
~~I4~~	~~Bread~~	~~1/5~~
~~I5~~	~~Umbrella~~	~~1/5~~
I6	Milk	2/5
~~I7~~	~~Detergent~~	~~1/5~~
~~I8~~	~~Coca-cola~~	~~1/5~~

Item_ID	Item	Support
I1	Beer	4/5
I2	Diaper	4/5
I3	Baby	2/5
I6	Milk	2/5

$$support\ (A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{total\_\#\_of\_tuples}$$

The University of Iowa Intelligent Systems Laboratory

Solution Procedure

Step 2

C2

Item_ID	Item	Support
{I1, I2}	Beer, Diaper	3/5
~~{I1, I3}~~	~~Beer, Baby powder~~	~~1/5~~
{I1, I6}	Beer, Milk	2/5
{I2, I3}	Diaper, Baby powder	2/5
~~{I2, I6}~~	~~Diaper, Milk~~	~~1/5~~
~~{I3, I6}~~	~~Baby powder, Milk~~	~~0~~

Step 3

L2

Item_ID	Item	Support
{I1, I2}	Beer, Diaper	3/5
{I1, I6}	Beer, Milk	2/5
{I2, I3}	Diaper, Baby powder	2/5

The University of Iowa — Intelligent Systems Laboratory

Solution Procedure

Step 4: L2 is not Null, so repeat Step2

Item_ID	Item
{I1, I2, I3}	Beer, Diaper, Baby powder
{I1, I2, I6}	Beer, Diaper, Milk
{I1, I3, I6}	Beer, Baby powder, Milk
{I2, I3, I6}	Diaper, Baby powder, Milk

C3 =Null

The University of Iowa — Intelligent Systems Laboratory

Solution Procedure

Step 5

***min_sup*=40% *min_conf*=70%**

Item_ID	Item	Support(A B)	Support A	Confidence
I1 I2	Beer Diaper	60%	80%	75%
~~I1 I6~~	~~Beer Milk~~	~~40%~~	~~80%~~	~~50%~~
~~I2 I3~~	~~Diaper Baby powder~~	~~40%~~	~~80%~~	~~50%~~
I2 I1	Diaper Beer	60%	80%	75%
I6 I1	Milk Beer	40%	40%	100%
I3 I2	Baby powder Diaper	40%	40%	100%

$$support\ (A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{total\_\#\_of\_tuples}$$

$$confidence(A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_containing\_A}$$

The University of Iowa — Intelligent Systems Laboratory

Solution Procedure

TID	List of items (item_IDs)
1	Beer(I1), Diaper(I2), Baby Powder(I3), Bread(I4), Umbrella(I5)
2	Diaper(I2), Baby Powder(I3)
3	Beer(I1), Diaper(I2), Milk(I6)
4	Diaper(I2), Beer(I1), Detergent(I7)
5	Beer(I1), Milk(I6), Coca Cola (I8)

Item_ID	Item	Support(A B)	Support A	Confidence
I1 I2	Beer Diaper	60%	80%	75%
~~I1 I6~~	~~Beer Milk~~	~~40%~~	~~80%~~	~~50%~~
~~I2 I3~~	~~Diaper Baby powder~~	~~40%~~	~~80%~~	~~50%~~
I2 I1	Diaper Beer	60%	80%	75%
I6 I1	Milk Beer	40%	40%	100%
I3 I2	Baby powder Diaper	40%	40%	100%

The University of Iowa — Intelligent Systems Laboratory

Solution Procedure

Step 6

***min_sup* = 40% *min_conf* = 70%**

Strong rules		Support	Confidence
I1=> I2	Beer=> Diaper	60%	75%
I2=> I1	Diaper=> Beer	60%	75%
I6 => I1	Milk=> Beer	40%	100%
I3 => I2	Baby powder=> Diaper	40%	100%

The University of Iowa — Intelligent Systems Laboratory

Results

- Some rules are believable, like Baby powder ⇒ Diaper.
- Some rules need additional analysis, like Milk ⇒ Beer.
- Some rules are unbelievable, like Diaper ⇒ Beer.

Note this example could contain unreal results because its small data.

The University of Iowa — Intelligent Systems Laboratory

Reference

- J. Han, M. Kamber (2001), *Data Mining*, Morgan Kaufmann Publishers, San Francisco, CA.