1495 UNIVERSITY OF ABERDEEN | CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Association Rule Learning (2)

Data Mining & Visualisation

Lecture 20

2025

Road Map

- Basic concepts
- Apriori algorithm
- Different data formats for mining
- Mining with multiple minimum supports
- Mining class association rules
- Summary

Different data formats for mining

- The data can be in transaction form or table form

Transaction form:

a, b
a, c, d, e
a, d, f

Table form:

Attr1	Attr2	Attr3
a,	b,	d
b,	c,	e

- Table data need to be converted to transaction form for association mining

From a table to a set of transactions

Table form:

Attr1	Attr2	Attr3
a,	b,	d
b,	c,	e

⇒Transaction form:

(Attr1, a), (Attr2, b), (Attr3, d)

(Attr1, b), (Attr2, c), (Attr3, e)

Road Map

- Basic concepts
- Apriori algorithm
- Different data formats for mining
- Mining with multiple minimum supports
- Mining class association rules
- Summary

Problems with the association mining

- Single minsup: It assumes that all items in the data are of the same nature and/or have similar frequencies.
- Not true: In many applications, some items appear very frequently in the data, while others rarely appear.

  E.g., in a supermarket, people buy *food processor* and *cooking pan* much less frequently than they buy *bread* and *milk*.

Rare Item Problem

If the frequencies of items vary a great deal, we will encounter two problems

- If minsup is set too high, those rules that involve rare items will never be found.
- To find rules that involve both frequent and rare items, minsup has to be set very low. This may cause combinatorial explosion because those frequent items will be associated with one another in all possible ways.

Multiple minsups model

The minimum support of a rule is expressed in terms of *minimum item supports* (MIS) of the items that appear in the rule.

Each item can have a minimum item support.

By providing different MIS values for different items, the user effectively expresses different support requirements for different rules.

Minsup of a rule

Let MIS(i) be the MIS value of item i. The *minsup* of a rule R is the lowest MIS value of the items in the rule.

I.e., a rule R: $a_1, a_2, \ldots, a_k \rightarrow a_{k+1}, \ldots, a_r$ satisfies its minimum support if its actual support is $\geq$

$$\min(\text{MIS}(a_1), \text{MIS}(a_2), \ldots, \text{MIS}(a_r)).$$

An Example

Consider the following items:

bread, shoes, clothes

The user-specified MIS values are as follows:

MIS(*bread*) = 2% MIS(*shoes*) = 0.1%

MIS(*clothes*) = 0.2%

The following rule doesn't satisfy its minsup:

clothes $\rightarrow$ *bread* [sup=0.15%,conf =70%]

The following rule satisfies its minsup:

clothes $\rightarrow$ *shoes* [sup=0.15%,conf =70%]

Downward closure property

In the new model, the property no longer holds (?)

E.g., Consider four items 1, 2, 3 and 4 in a database. Their minimum item supports are

MIS(1) = 10%   MIS(2) = 20%

MIS(3) = 5%   MIS(4) = 6%

{1, 2} with support 9% is infrequent, but {1, 2, 3} and {1, 2, 4} could be frequent.

To deal with the problem

- We sort all items in I according to their MIS values (make it a total order).
- The order is used throughout the algorithm in each itemset.
- Each itemset w is of the following form:

  $\{w[1], w[2], \ldots, w[k]\}$, consisting of items,

  $w[1], w[2], \ldots, w[k]$,

  where $\text{MIS}(w[1]) \leq \text{MIS}(w[2]) \leq \ldots \leq \text{MIS}(w[k])$.

The MSapriori algorithm

Algorithm MSapriori(*T*, *MS*)

$M \leftarrow$ sort(I, MS);
$L \leftarrow$ init-pass(M, T);
$F_1 \leftarrow \{\{i\} \mid i \in L, i.count/n \geq \text{MIS}(i)\}$;
for ($k = 2$; $F_{k-1} \neq \emptyset$; k++) **do**
  if k=2 **then**
    $C_k \leftarrow$ level2-candidate-gen(L)
  else $C_k \leftarrow$ MScandidate-gen(F_{k-1});
  end;
  for each transaction $t \in T$ **do**
    for each candidate $c \in C_k$ **do**
      if c is contained in t **then**
        $c.count$++;
      if $c - \{c[1]\}$ is contained in t **then**
        $c.tailCount$++
    end
  end
  $F_k \leftarrow \{c \in C_k \mid c.count/n \geq MIS(c[1])\}$
end
return $F \leftarrow \bigcup_k F_k$;14

First pass over data

- It makes a pass over the data to record the support count of each item.
- It then follows the sorted order to find the first item *i* in *M* that meets MIS(*i*).
  - *i* is inserted into *L*.
  - For each subsequent item *j* in *M* after *i*, if $j.count/n \geq$ MIS(*i*) then *j* is also inserted into *L*, where *j.count* is the support count of *j* and *n* is the total number of transactions in *T*. Why?
- *L* is used by function level2-candidate-gen

First pass over data: example

- Consider the four items 1, 2, 3 and 4 in a data set. Their minimum item supports are:

  MIS(1) = 10%   MIS(2) = 20%

  MIS(3) = 5%   MIS(4) = 6%

- Assume our data set has 100 transactions. The first pass gives us the following support counts:

  {3}.*count* = 6, {4}.*count* = 3,

  {1}.*count* = 9, {2}.*count* = 25.

- **Then** L = {3, 1, 2}, and F_1 = {{3}, {2}}
- Item 4 is not in L because 4.*count*/n < MIS(3) (= 5%),
- {1} is not in F_1 because 1.*count*/n < MIS(1) (= 10%).

Rule Generation

- The following two lines in MSapriori algorithm are important for rule generation, which are not needed for the Apriori algorithm

if $c - \{c[1]\}$ is contained in t **then**

c.tailCount++

- Many rules cannot be generated without them.
- Why?

On multiple minsup rule mining

Multiple minsup model subsumes the single support model.

It is a more realistic model for practical applications.

The model enables us to found rare item rules yet without producing a huge number of meaningless rules with frequent items.

By setting MIS values of some items to 100% (or more), we effectively instruct the algorithms not to generate rules only involving these items.

Road Map

- Basic concepts
- Apriori algorithm
- Different data formats for mining
- Mining with multiple minimum supports
- Mining class association rules
- Summary

Mining class association rules (CAR)

- Normal association rule mining does not have any target.
- It finds all possible rules that exist in data, i.e., any item can appear as a consequent or a condition of a rule.
- However, in some applications, the user is interested in some targets.
  - E.g, the user has a set of text documents from some known topics. He/she wants to find out what words are associated or correlated with each topic.

Problem definition

- Let T be a transaction dataset consisting of n transactions.
- Each transaction is also labeled with a class y.
- Let I be the set of all items in T, Y be the set of all class labels and $I \cap Y = \emptyset$.
- A **class association rule** (**CAR**) is an implication of the form

$$X \rightarrow y, \text{ where } X \subseteq I, \text{ and } y \in Y.$$

- The definitions of **support** and **confidence** are the same as those for normal association rules.

Example

- **A text document data set**

doc 1: Student, Teach, School : Education

doc 2: Student, School : Education

doc 3: Teach, School, City, Game : Education

doc 4: Baseball, Basketball : Sport

doc 5: Basketball, Player, Spectator : Sport

doc 6: Baseball, Coach, Game, Team : Sport

doc 7: Basketball, Team, City, Game : Sport

- Let *minsup* = 20% and *minconf* = 60%. The following are two examples of class association rules:

Student, School $\rightarrow$ Education : [sup= 2/7, conf = 2/2]

game $\rightarrow$ Sport : [sup= 2/7, conf = 2/3]

Road Map

- Basic concepts
- Apriori algorithm
- Different data formats for mining
- Mining with multiple minimum supports
- Mining class association rules
- Summary

Summary

Association rule mining has been extensively studied in the data mining community.

There are many efficient algorithms and model variations.

Other related work includes

- Multi-level or generalized rule mining
- Constrained rule mining
- Incremental rule mining
- Maximal frequent itemset mining
- Numeric association rule mining
- Rule interestingness and visualization
- Parallel algorithms
- …