**Data Warehousing & Data Mining**

Laboratory # 13

**1/3/2019**

**UET TAXILA**

**Engr. Sidra Shafi**



**Apriori Algorithm**

|  |  |  |
| --- | --- | --- |
| **Statement Purpose:** |  |  |
|  |  |

The objective of this lab is to implement Apriori Algorithm to generate Association Rules.

**Resources required:**

1. A desktop computer

2. WEKA 3-8-3 (Use stable Version)

**Apriori Algorithm**

**Theoretical Aspects:**

In data mining, association rule learning is a popular and well researched method for discovering

interesting relations between variables in large databases. Based on the concept of strong rules, Agrawal introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule {onion,potatoes}=>{burger} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy burger. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics.

In computer science and data mining, Apriori is a classic algorithm for learning association rules.

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing).

**Definition:**

Following the original definition by Agrawal the problem of association rule mining is defined as: Let **I = {i1, i2, ..., in}** be a set of *n* binary attributes called *items*. Let **D = {t1, t2, ..., tn}** be a set of transactions called the *database*. Each transaction in **D** has a unique transaction ID and contains a subset of the items in **I**. A *rule* is defined as an implication of the form **X**\_**Y** where **X, Y** ⊆ **I** and **X∩ Y = ∅.** The sets of items (for short *itemsets*) **X** and **Y** are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule respectively.

**Apriori algorithm**

**General Process**

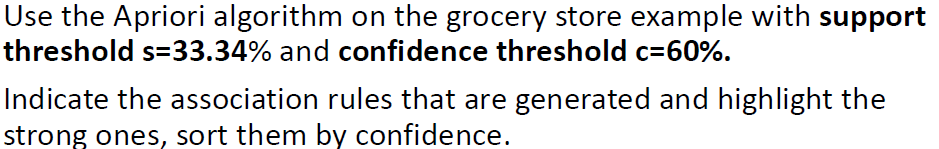
Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all *frequent itemsets* in a database.

2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

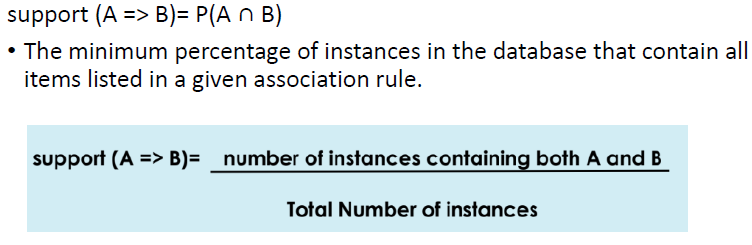
While the second step is straight forward, the first step needs more attention.

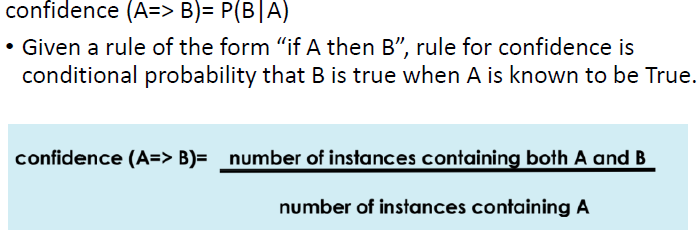
Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). **The set of possible itemsets is the power set over *I* and has size 2*n* − 1 (excluding the empty set which is not a valid itemset).** Although the size of the powerset grows exponentially in the number of items *n* in *I*, efficient search is possible using the ***downward-closure property*** of support (also called *anti-monotonicity*) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent.

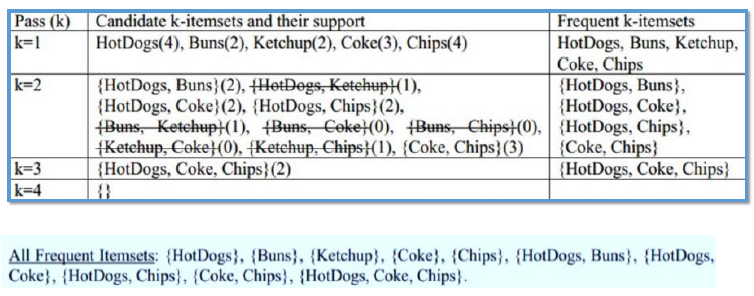
**Question:**

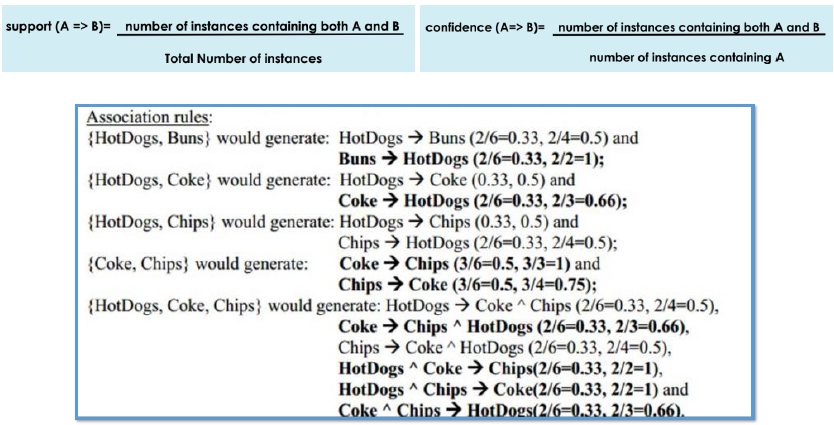


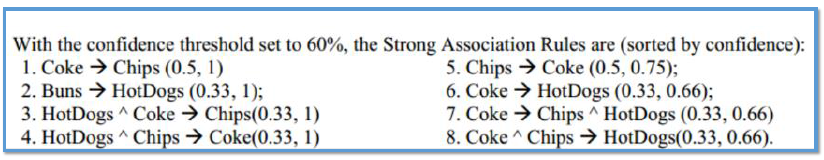
**Rule for Support:**



**Rule for Confidence:**

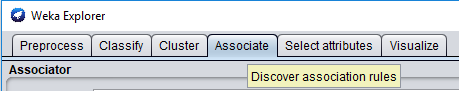
**Total number of instances=6**

****

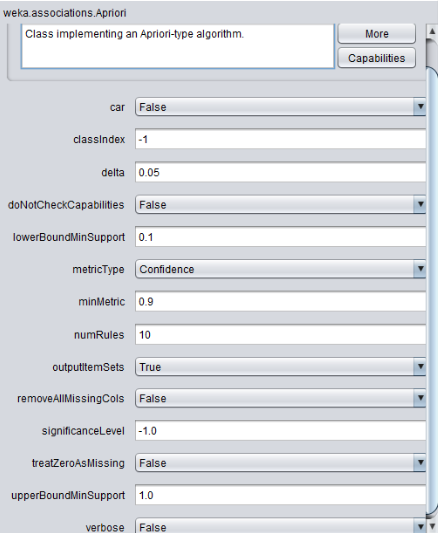
****

**Practice Exercise:**

1. Open Explorer tab in Weka.
2. Load Weather.nominal Dataset. Note that Apriori algorithm expects data that is purely nominal: If present, numeric attributes must be discretized first.
3. Click Associate Tab on top of the window.



1. Open Generic Object Editor. The property window of Apriori opens:



1. Weka runs an Apriori-type algorithm to find association rules, but this algorithm is not exact the same one as discussed in class.

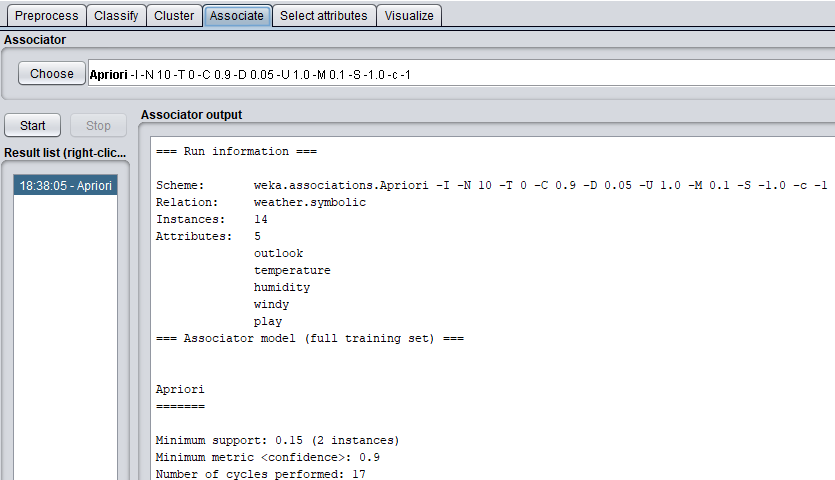
a. The min. support is not fixed. This algorithm starts with min. support as ***upperBoundMinSupport*** (default 1.0 = 100%), iteratively decrease it by ***delta*** (default 0.05 = 5%). Note that *upperBoundMinSupport* is decreased by delta *before* the basic *Apriori* algorithm is run for the first time.

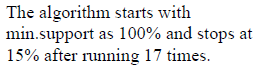
b. The algorithm stops when ***lowerBoundMinSupport*** (default 0.1 = 10%) is reached or required number of rules – ***numRules*** (default value 10) have been generated.

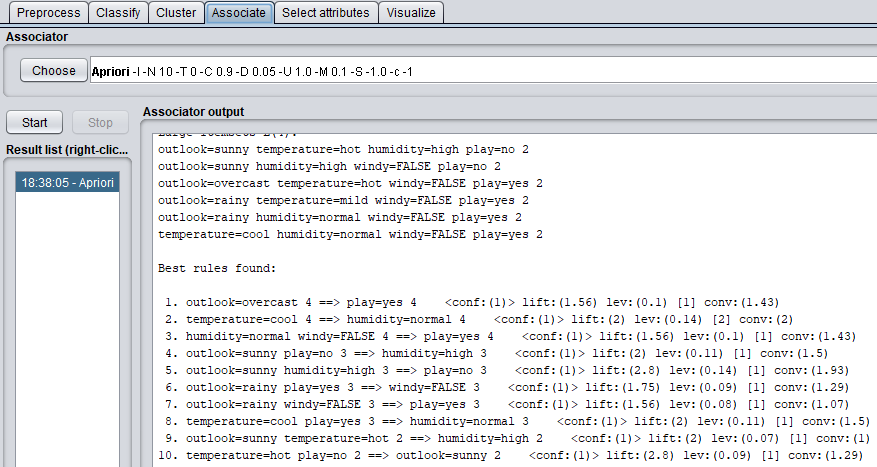
c. Rules generated are ranked by ***metricType*** (default *Confidence*). Only rules with score higher than ***minMetric*** (default 0.9 for *Confidence*) are considered and delivered as the output.

d. If you choose to show the all frequent itemsets found, ***outputItemSets*** should be set as *True*.

1. Click **Start** button on the left of the window, the algorithm begins to run. The output is showing in the right window.

****

****

****

1. You could re-run Apriori algorithm by selecting different parameters, such as

*lowerBoundMinSupport*, *minMetric* (min. confidence level), and different evaluation metric (confidence vs. lift), and so on.

**Example 2: Run Apriori on Contact Lenses Dataset and Examine the Output.**

**Description of dataset.**

% 5. Number of Instances: 24

%

% 6. Number of Attributes: 4 (all nominal)

%

% 7. Attribute Information:

% -- 3 Classes

% 1 : the patient should be fitted with hard contact lenses,

% 2 : the patient should be fitted with soft contact lenses,

% 1 : the patient should not be fitted with contact lenses.

%

% 1. age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic

% 2. spectacle prescription: (1) myope, (2) hypermetrope

% 3. astigmatic: (1) no, (2) yes

% 4. tear production rate: (1) reduced, (2) normal

%

% 8. Number of Missing Attribute Values: 0

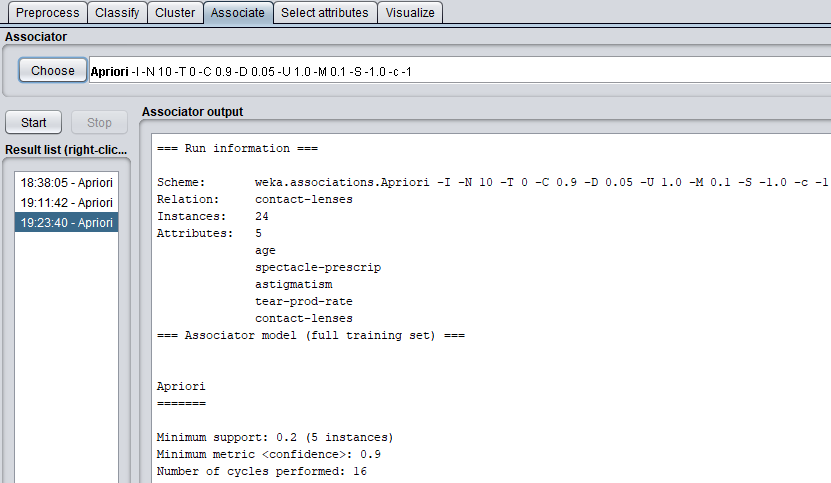
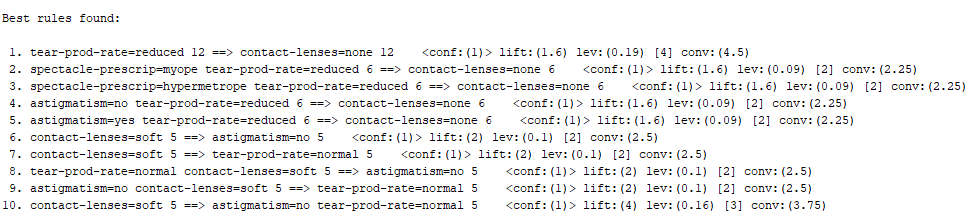
%

% 9. Class Distribution:

% 1. hard contact lenses: 4

% 2. soft contact lenses: 5

% 3. no contact lenses: 15

****

**Lab Tasks Marks: 10**

**Q1. Given a transaction database: Marks: 5**

{bread milk butter beer}

{bread butter water jam beer}

{beer diapers bread butter jam}

{butter milk juice}

{diapers beer juice water}

1. **For minimal support 0.6 find all frequent itemsets. 2**
2. **For minimal confidence 0.7 find association rules of the form item1 -> {item2, item3}. 2**
3. **How many certain association rules can you find? (with confidence 1.0). 1**

**Q2. Run Apriori on dataset named Association Rules\_lab13.csv given with lab manual and examine the output. Marks: 5**

**Instructions:**

1. **Click edit to check the dataset first.**
2. **Using the Attributes Filter in WEKA preprocess panel, remove Transaction ID Attribute. 1**
3. **Convert All Attributes to nominal. 1**
4. **Apply APRIORI algorithm on the dataset. 2**
5. **Interpret APRIORI output. 1**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*