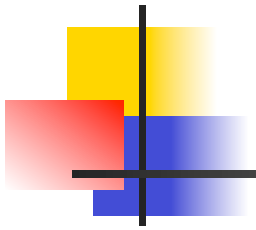
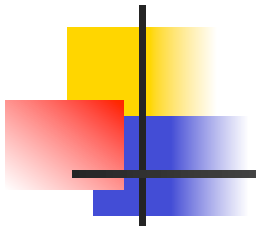


WEKA
Waikato Environment for Knowledge Analysis

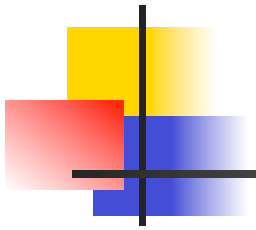
Association Rules

Prof. Pietro Ducange



Introduction

- The WEKA implementation of the APRIORI and FP-growth algorithms will be presented
- An example of association rules mining will be discussed
- A number of exercises will be proposed

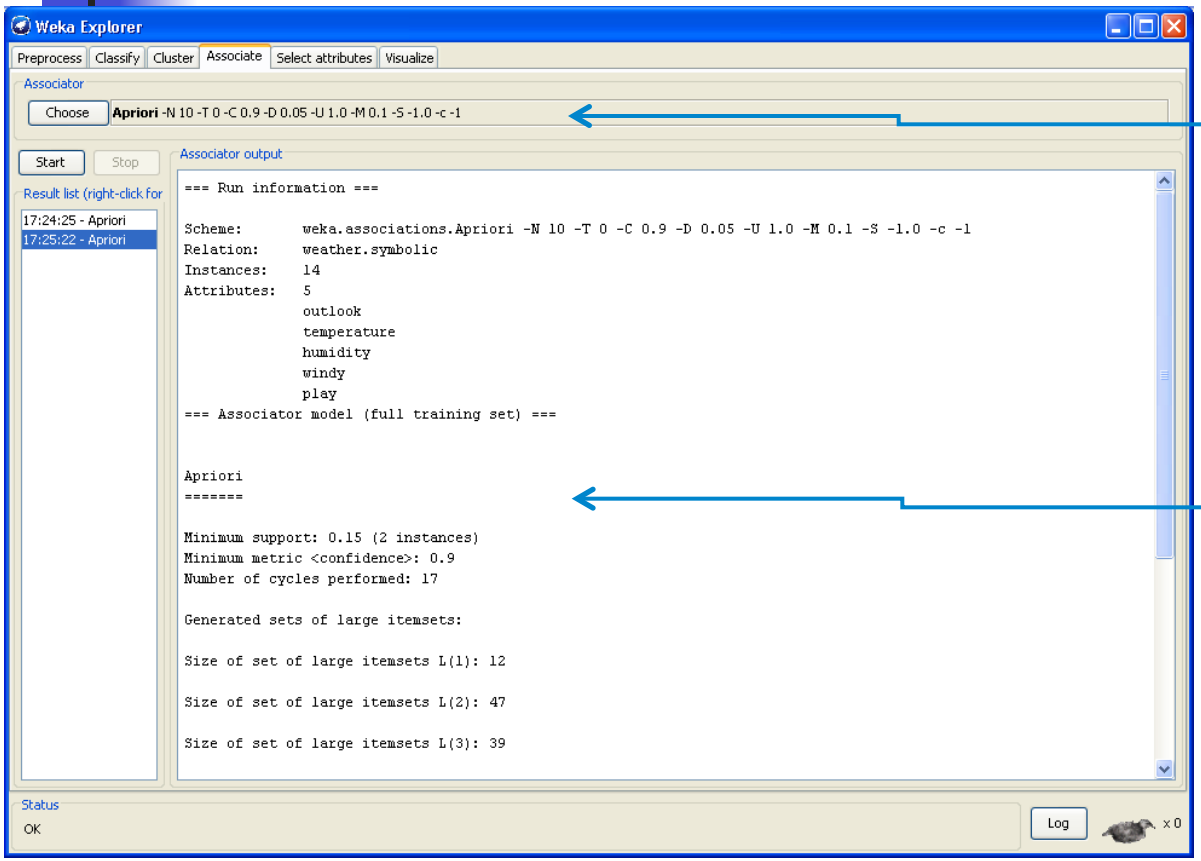


APRIORI

Method:

- Initially, scan DB once to get frequent 1-itemset
- Generate length (k+1) candidate itemsets from length k frequent itemsets
- Test the candidates against DB
- Terminate when no frequent or candidate set can be generated

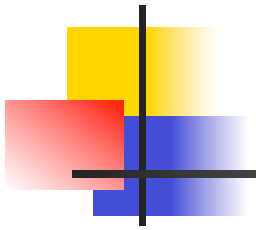
Association-rule



Name of the currently selected association-rule learner and its options

Output of the algorithm

java weka.associations.Apriori –t data\weather.nominal.arff



In Weka, by default, Apriori tries to generate ten rules.

It begins with a minimum support of 100% of the data items and decreases this in steps of 5%.

It terminates when there are at least ten rules with the required minimum confidence, or when the support has reached a lower bound of 10%.

APRIORI Parameters

```
General options:

-t <training file>
        The name of the training file.
-g <name of graph file>
        Outputs the graph representation (if supported) of the associator to a file.

Options specific to Apriori:

-N <required number of rules output>
        The required number of rules. (default = 10)
-T <0=confidence | 1=lift | 2=leverage | 3=Conviction>
        The metric type by which to rank rules. (default = confidence)
-C <minimum metric score of a rule>
        The minimum confidence of a rule. (default = 0.9)
-D <delta for minimum support>
        The delta by which the minimum support is decreased in
        each iteration. (default = 0.05)
-U <upper bound for minimum support>
        Upper bound for minimum support. (default = 1.0)
-M <lower bound for minimum support>
        The lower bound for the minimum support. (default = 0.1)
-S <significance level>
        If used, rules are tested for significance at
        the given level. Slower. (default = no significance testing)
-I
        If set the itemsets found are also output. (default = no)
-R
        Remove columns that contain all missing values (default = no)
-V
        Report progress iteratively. (default = no)
-A
        If set class association rules are mined. (default = no)
-c <the class index>
        The class index. (default = last)
```

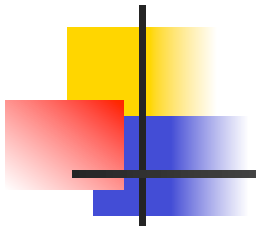
Associator output (default parameters)

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:
Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 47
Size of set of large itemsets L(3): 39
Size of set of large itemsets L(4): 6

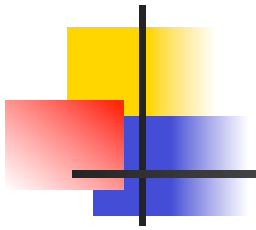
Support of the rule

Best rules found:
1. outlook=overcast 4 ==> play=yes 4 conf:(1)
2. temperature=cool 4 ==> humidity=normal 4 conf:(1)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 conf:(1)
4. outlook=sunny play=no 3 ==> humidity=high 3 conf:(1)
5. outlook=sunny humidity=high 3 ==> play=no 3 conf:(1)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 conf:(1)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 conf:(1)
8. temperature=cool play=yes 3 ==> humidity=normal 3 conf:(1)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 conf:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 conf:(1)



Exercise (i)

- Transform the stock.txt dataset into the .arff format and save the dataset
- Discretize the dataset by using 5 bins and save the dataset
- Generate the set of association rules by using the APRIORI algorithm with default parameters
- Calculate the average confidence and support

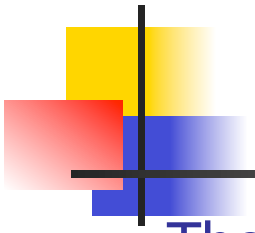


Exercise (ii)

- Repeat the previous exercise changing the APRIORI parameters as follows:

1) Set the maximum number of rules to 1000
2) Set the minimum confidence to 0.75
3) Set the minimum support to 0.2
4) Set the minimum confidence to 0.50

FP-growth

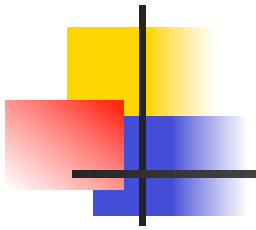


The FPGrowth Approach (J. Han, J. Pei, and Y. Yin, SIGMOD' 00) allows frequent itemset discovery without candidate itemset generation. Two step approach:

Step 1: Build a compact data structure called the FP-tree
Built using 2 passes over the data-set.

Step 2: Extracts frequent itemsets directly from the FP-tree

Exercise (i)



- Load the weather.nominal dataset
- Apply the FP-growth algorithm with default parameters.