**PART II: WEKA EXPERIMENTS**

## DATASET 1: SUPERMARKET

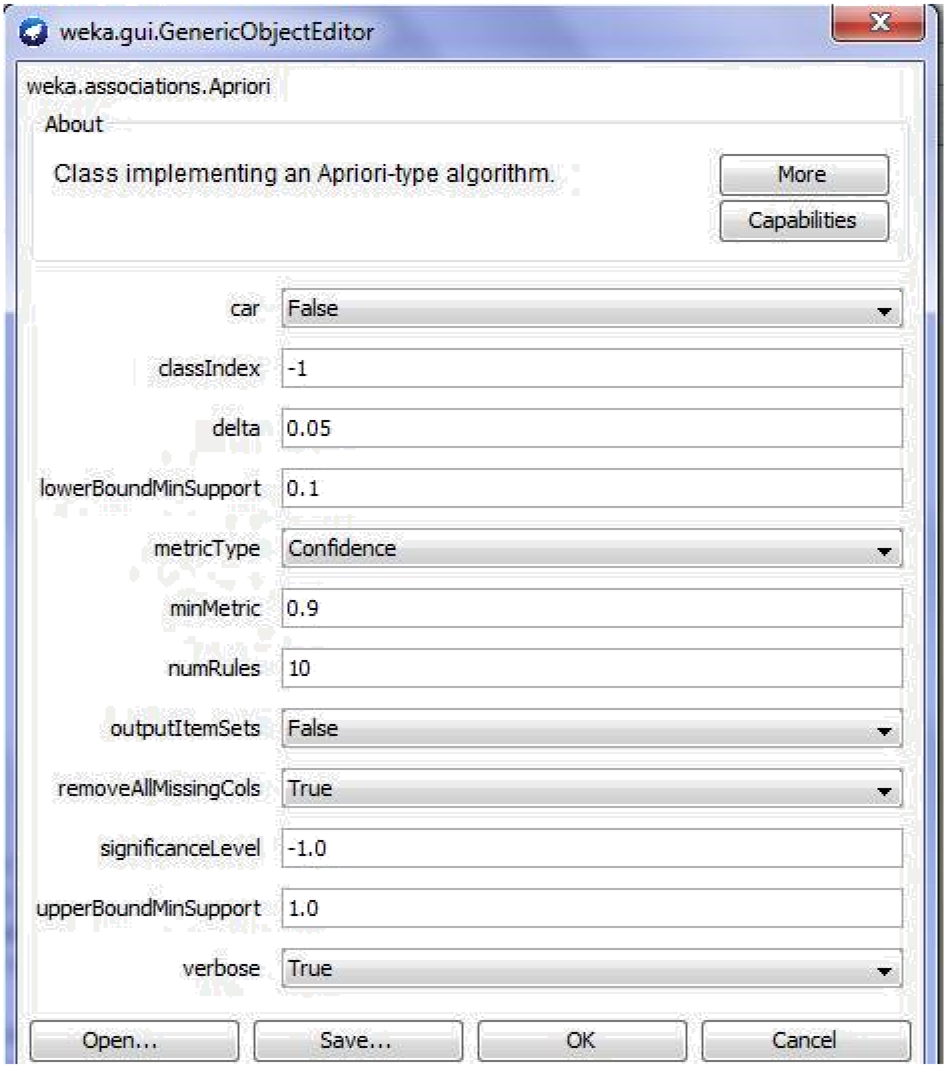
This dataset consists of transactions performed at a supermarket where the rows consist of transactions done by customers and columns denote the departments and product items.

It has 4627 rows and 217 attributes. (out of which one attribute "total" is nominal and remaining ones are unary which contains the value "t" if present otherwise the value is missing). Apriori algorithm works only with categorical values.

In this dataset there are many missing values (>90%) since all the transactions cannot include all the data items.

Frequent items: bread and cake (count=3330), fruit (count=2962), vegetables (count=2961), milk-cream (count=2939).

**Parameters:**




The above are the parameters and their default values.

In this example the metricType is Confidence where the minimum value for confidence is given by the parameter minMetric (in this case it is 0.9 i.e. 90%). Here initially the support value is equal to the upperBoundMinSupport and it decreases with the delta value until either the lowerBoundMinSuppport is achieved or the numRules value is reached.

**car** - This defines the basis of association rule. If set to "True", it would use the specific class attribute for mining and if set to "False", it would use normal association rules.

**classIndex** - This parameter is used if the above *car* parameter is set to "True". It is the index value of the class attribute to be used for mining.

**delta** -- Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated.

**lowerBoundMinSupport** -- Lowest bound for minimum support.

**metricType** -- Set the type of metric by which to rank rules. Confidence is the proportion of the examples covered by the premise that are also covered by the consequence(Class association rules can only be mined using confidence). Lift is confidence divided by the proportion of all examples that are covered by the consequence. This is a measure of the importance of the association that is independent of support. Leverage is the proportion of additional examples covered by both the premise and consequence above those expected if the premise and consequence were independent of each other. The total number of examples that this represents is presented in brackets following the leverage. Conviction is another measure of departure from independence. Conviction is given by P(premise)P(!consequence) / P(premise, !consequence).

**minMetric** -- Minimum metric score. Consider only rules with scores higher than this value.

**numRules** -- Number of rules to find.

**outputItemSets** -- If enabled the itemsets are output as well.

**removeAllMissingCols** -- Remove columns with all missing values.

**significanceLevel** -- Significance level. Significance test (confidence metric only).

**upperBoundMinSupport** -- Upper bound for minimum support. Start iteratively decreasing minimum support from this value.

**verbose** -- If enabled the algorithm will be run in verbose mode.

***Among the above parameters, the important ones are the metricType, minMetric, lowerBoundMinSupport and upperBoundMinSupport.***

In Weka, the support value is first set at the *upperBoundMinSupport* (default = 1.0 i.e. 100% of the total instances) and the *minSupport* value is eventually decreased by the delta value (default=0.05). This procedure is repeated until one of the following condition is achieved:

1. The minSupport value is reached till the *lowerBoundMinSupport*

2. The number of rules generated is equal to *numRules*.

The rules obtained using the default parameters is given as:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723 conf:(0.92)

2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696 conf:(0.92)

3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705 conf:(0.92)

4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746 conf:(0.92)

**ANALYSIS:**

- As seen from the above results, when the biscuits, frozen food and fruits are bought together (788 instances in this case), then it is 90% confident (0.92 confidence) that the customer would also buy the bread and cake (723 instances).
- Also it can be seen that the consequent in all the 4 cases is bread and cake (since it is the most frequent itemset)

**TEST**

**Observation:**

In spite of the support value > 0.15, when the confidence value was decreased, the rules are generated as above.

| Support | Confidence | Rules |
|---|---|---|
| Low sup=0.1 | 0.9 | Upto 27000 rules are generated with highest confidence (i.e. 0.97) and minimum 0.9 confidence within 10 cycles |
| | 0.7 | 10000 rules generated with confidence greater than 0.78 confidence in 9 cycles |
| | 0.5 | All rules generated, however it has low confidence |
| Medium Sup= 0.15 | 0.9 | Less rules generated compared to above support value as the support value is increased itemsets are pruned |
| | 0.7 | Less rules generated however confidence is low |
| | 0.5 | Lowest confidence thus rules not much useful |
| High Sup=0.2 | 0.9 | The rules are not generated for this condition since the support is high enough which would cause |
| | 0.7 | Rules generated however the ones with lower support are discarded |
| | 0.5 | Rules generated however the confidence value is low which is not useful |

| Sr. No. | Metric | minSup | delta | minMetric | num Rules | Number of cycles | Number of attributes | Observation |
|---|---|---|---|---|---|---|---|---|
| 1 | Confidence | 0.15 | 0.1 | 0.9 | 1000 | 10 | 106 | All the rules not generated (only 458 found) since the support value |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | is increased, itemsets having lesser support are removed |
| 2 | Confidence | 0.15 | 0.05 | 0.9 | 1000 | 18 | 106 | All the rules not generated (only 458 found) |
| 3 | Confidence | 0.1 | 0.05 | 0.9 | 10 | 17 | 217 | More computation for more attributes |
| 4 | Confidence | 0.1 | 0.1 | 0.9 | 1000 | 10 | 106 | All rules not generated; only 458 generated as minSup value is reached |
| 5 | Confidence | 0.05 | 0.1 | 0.9 | 10 | 10 | 104 (frequent itemsets removed) | Frequent itemsets removed thus variations are seen in rules. |
| 6 | Confidence | 0.1 | 0.1 | 0.9 | 10 | 10 | 104 (frequent itemsets removed) | No rules generated since for the given support value there weren't any itemsets |
| 4 | Lift | 0.5 | 0.1 | 1.1 | 10 | 8 | 217 | Only 2 rules generated with only 2 itemsets |
| 5 | Lift | 0.2 | 0.1 | 1.1 | 2000 | 8 | 106 | All the rules generated, few of them contain total as antecedent and consequent which might not be interesting. |

**Conclusion:**

- The lower the minimum support and the higher the confidence, we get maximum rules generated and that too with the best confidence value out of which one can choose interesting ones. Also the more the number of rules, we would also have variations in the values of consequents and antecedents.
- If the support is too low then there would be many computations and if the support is too high then there might be few itemsets with lower support which might be expensive items and we miss out on them.
- The delta value if kept 0.05 and if kept 0.1, it doesn't make much significant change in the number of rules and their confidence values. Thus in order to reduce computations, it can be recommended to keep the value high which would in turn reduce the number of cycles.
- If the frequent itemsets are removed then no rules are generated for high confidence (0.9), however if we reduce the value of confidence then the rules are generated even for a higher support (though this is not a desirable case)
- If the frequent itemsets are removed then we can see variations in the rules which would include more itemsets.
- Lift is also a good measure since it considers that case where the consequent has more support itself

## VOTE DATASET

### Using simpleCart decision tree:

=== Run information ===

Scheme:weka.classifiers.trees.SimpleCart -S 1 -M 2.0 -N 5 -C 1.0
Relation: vote
Instances: 435
Attributes: 17 handicapped-
infants water-project-
cost-sharing
adoption-of-the-budget-
resolution physician-fee-freeze el-
salvador-aid religious-groups-in-
schools anti-satellite-test-ban aid-
to-nicaraguan-contras mx-missile

immigration synfuels-
corporation-cutback
education-spending
superfund-right-to-sue crime

duty-free-exports export-
administration-act-south-africa Class

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

CART Decision Tree

```
physician-fee-freeze=(y)
| synfuels-corporation-cutback=(n): republican(141.7/4.0)
| synfuels-corporation-cutback!=(n)
| | mx-missile=(n)
| | | adoption-of-the-budget-resolution=(n): republican(19.28/3.31)
| | | adoption-of-the-budget-resolution!=(n)
| | | | anti-satellite-test-ban=(y): republican(2.2/0.0)
| | | | anti-satellite-test-ban!=(y): democrat(5.01/0.02)
```

| | mx-missile!=(n): democrat(4.99/1.02) physician-
fee-freeze!=(y): democrat(249.66/3.74)

Number of Leaf Nodes: 6

Size of the Tree: 11

Time taken to build model: 0.94 seconds

=== Stratified cross-validation ===
=== Summary ===

| | | |
|---|---|---|
| Correctly Classified Instances | 415 | 95.4023 % |
| Incorrectly Classified Instances | 20 | 4.5977 % |
| Kappa statistic | 0.9034 | |
| Mean absolute error | 0.0817 | |
| Root mean squared error | 0.2003 | |
| Relative absolute error | 17.2189 % | |
| Root relative squared error | 41.1466 % | |
| Total Number of Instances | 435 | |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | Class |
|---|---|---|---|---|---|---|---|
| | 0.955 | 0.048 | 0.97 | 0.955 | 0.962 | 0.967 | democrat |
| | 0.952 | 0.045 | 0.93 | 0.952 | 0.941 | 0.967 | republican |
| Weighted Avg. | 0.954 | 0.047 | 0.954 | 0.954 | 0.954 | 0.967 | |

=== Confusion Matrix ===

```
   a  b  <-- classified as
 255 12 |   a = democrat
   8 160 |  b = republican
```

**Using Association rule Mining:**

```
Apriori
=======

Minimum support: 0.45 (196 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 11

Generated sets of large itemsets:

Size of set of large itemsets L(1): 20

Size of set of large itemsets L(2): 17

Size of set of large itemsets L(3): 6

Size of set of large itemsets L(4): 1

Best rules found:

 1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219    conf:(1)
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198    conf:(1)
 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210    conf:(1)
 4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201    conf:(1)
 5. physician-fee-freeze=n 247 ==> Class=democrat 245    conf:(0.99)
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    conf:(0.99)
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    conf:(0.98)
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198    conf:(0.98)
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    conf:(0.97)
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210    conf:(0.96)
```

**Comparison:**

- **As seen in above run using the Apriori algorithm, the rules are generated which shows the association between different itemsets. ( refer above rules number 7 where el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 conf:(0.98) means that not having el-salvador-aid implies there is 98% confidence that there is aid-to-nicaraguan-contras ).**

  **However in case of SimpleCart, the classification occurs on basis of attribute which is to be split. SimpleCart makes use of binary decision tree.**

- **Rules are generated in parallel in association rule mining i.e. before making a rule two steps are followed: 1. generating frequent itemsets 2. rule generation. The support count of itemsets is considered in order to prune the least frequent itemsets.**

  **Similarly in SimpleCart, before getting to a leaf node, multiple decisions are made at multiple nodes**