Sardar Patel Institute of Technology, Mumbai
Department of Electronics Engineering
B.E. Sem-VII

Experiment: Apriori Algorithm and Association rule mining with WEKA

Name: Vivek Rajput Batch: A UID:2019110046 Branch: ETRX

Objective: **Apply Apriori Algorithm to a given dataset**
Platform : WeKa

Code and output:

Consider the dataset “Groceries” and apply apriori algorithm on it. What are the first 5 rules generated when the min support is 0.001 (0.1%) and min confidence is 0.9 (90%) .

Exercise 1:
The 'database' below has four transactions. What association rules can be found in this set, if the minimum support (i.e coverage) is 60% and the minimum confidence (i.e. accuracy) is 80% ? Trans_id Itemlist
T1 {K, A, D, B}
T2 {D, A C, E, B}
T3 {C, A, B,
E} T4 {B, A,
D} Hint:

Make a tabular and binary representation of the data in order to better see the relationship

between Items. First generate all item sets with minimum support of 60%. Then form rules and

calculate their confidence base on the conditional probability P(B|A) = |B∩A| / |A|. Remember to only take the item sets from the previous phase whose support is 60% or more.

Solution:

Tabular representations :

Transaction	A	B	C	D	E	K
T1	1	1	0	1	0	1
T2	1	1	1	1	1	0
T3	1	1	1	0	1	0
T4	1	1	0	1	0	0

Min support 0.6

Item	Frequency	support
A	4	4/4 = 100%
B	4	4/4 = 100%
C	2	2/4 = 50%
D	3	3/4 = 75%
E	2	2/4 = 50%
K	1	1/4 = 25%

Considering item sets with support ≥ 60%

A = 100%
B = 100%
D = 75%

Considering 2 items at a time

	Freq	support
A B	4	4/4 = 100%
A D	~~3/4~~ 3	3/4 = 75%
B D	3	3/4 = 75%

Considering 3 items at a time

A B D
Freq = 3   support = 3/4 = 75%

Forming rules & finding confidence

A → B	~~P(B∩A)~~ P(B/A) = 4/4 = 100%
B → A	P(A/B) = 4/4 = 100%
A → D	P(D/A) = 3/4 = 75%
D → A	P(A/D) = 3/4 3 = ~~75%~~ 100%
B → D	P(D/B) = 3/4 = 75%
D → B	P(B/D) = 3/3 = 100%
A B → D	P(D/AB) = 3/4 = 75%
D → AB	P(AB/D) = ~~3~~ 3/3 = 100%
AD → B	P(B/AD) = 3/3 = 100%
B → AD	P(AD/B) = 3/4 = 75%
BD → A	P(A/BD) = 3/3 = 100%
A → BD	P(BD/A) = 3/4 = 75%

Considering rules with confidence above 80%

A → B	100%
B → A	100%
D → A	100%
D → B	100%
D → AB	100%
AD → B	100%
DB → A	100%

Above are the final set of rules

Exercise:2

Input file generation and Initial experiments with Weka's association rule discovery.

1. Launch Weka and try to do the calculations you performed manually in the previous exercise. Use
the apriori algorithm for generating the association rules.

The file may be given to Weka in e.g. two different formats. They are called ARFF (attribute-relation file format) and CSV (comma separated
values). Both are given below:

ARFF:

@relation exercise

@attribute exista {TRUE, FALSE}

...

@data

TRUE,TRUE,FALSE,TRUE,FALSE,TRUE

...

...

CSV:

exista,existb,existc,existd,existe,existk

TRUE,TRUE,FALSE,TRUE,FALSE,TRUE

...

...

3

2. Once Data is loaded Click Associate Tab on top of the window.

3. Left click the field of Associator, choose Show Property from the drop down list. The propertywindow of Apriori opens.
4. Weka runs an Apriori-type algorithm to find association rules, but this algorithm is not exactthe same one as we discussed in class.

a. The min. support is not fixed. This algorithm starts with min. support asupperBoundMinSupport (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, orrequired number of rules – numRules (default value 10) have been generated.

c. c. Rules generated are ranked by metricType (default Confidence). Only rules with scorehigher 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 asTrue.

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

Did you succeed? Are the results the same as in your calculations? What kind of file did you use as input?

Yes the results are same as my calculations.

The file used as an input is a CSV file as the dataset.

Dataset:

Relation: grocery

No.	1: Trans_id Nominal	2: exis_A Nominal	3: exis_B Nominal	4: exis_C Nominal	5: exis_D Nominal	6: exis_E Nominal	7: **exis_K** Nominal
1	T1	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
2	T2	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
3	T3	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
4	T4	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE
5							

Output:

Properties:

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.6
metricType	Confidence
minMetric	0.8
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	True

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.6 -S -1.0 -V -c -1
Relation:     grocery
Instances:    5
Attributes:   7
              Trans_id
              exis_A
              exis_B
              exis_C
              exis_D
              exis_E
              exis_K
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.7 (3 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 6

Generated sets of large itemsets:

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

Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 2

Best rules found:

 1. exis_B=TRUE 4 ==> exis_A=TRUE 4    <conf:(1)> lift:(1.25) lev:(0.16) [0] conv:(0.8)
 2. exis_A=TRUE 4 ==> exis_B=TRUE 4    <conf:(1)> lift:(1.25) lev:(0.16) [0] conv:(0.8)
 3. exis_D=TRUE 3 ==> exis_A=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 4. exis_K=FALSE 3 ==> exis_A=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 5. exis_D=TRUE 3 ==> exis_B=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 6. exis_K=FALSE 3 ==> exis_B=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 7. exis_B=TRUE exis_D=TRUE 3 ==> exis_A=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 8. exis_A=TRUE exis_D=TRUE 3 ==> exis_B=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
 9. exis_D=TRUE 3 ==> exis_A=TRUE exis_B=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
10. exis_B=TRUE exis_K=FALSE 3 ==> exis_A=TRUE 3    <conf:(1)> lift:(1.25) lev:(0.12) [0] conv:(0.6)
```

Exercise 3:

Mining Association Rule with WEKA Explorer – Weather dataset

1. To get a feel for how to apply Apriori to prepared data set, start by mining association rulesfrom the weather.nominal.arff data set of Lab One. Note that Apriori algorithm expects data that is purely nominal: If present, numeric attributes must be discretized first.
2. Like in the previous example choose Associate and Click Start button on the left of thewindow, the algorithm begins to run. The output is showing in the right window. 3. 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.

Below rules have been formed with min Support: 0.2 and confidence:0.8

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -V -c -1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.25 (4 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 26

Size of set of large itemsets L(3): 4

Best rules found:
```

```
Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. humidity=normal 7 ==> play=yes 6    <conf:(0.86)> lift:(1.33) lev:(0.11) [1] conv:(1.25)
10. play=no 5 ==> humidity=high 4    <conf:(0.8)> lift:(1.6) lev:(0.11) [1] conv:(1.25)
```

Changing the parameters:

With minSupport:0.2 and confidence:0.5

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -V -c -1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.3 (4 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 9

Size of set of large itemsets L(3): 1

Best rules found:
```

```
Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. humidity=normal 7 ==> play=yes 6    <conf:(0.86)> lift:(1.33) lev:(0.11) [1] conv:(1.25)
 5. play=no 5 ==> humidity=high 4    <conf:(0.8)> lift:(1.6) lev:(0.11) [1] conv:(1.25)
 6. windy=FALSE 8 ==> play=yes 6    <conf:(0.75)> lift:(1.17) lev:(0.06) [0] conv:(0.95)
 7. play=yes 9 ==> humidity=normal 6    <conf:(0.67)> lift:(1.33) lev:(0.11) [1] conv:(1.13)
 8. play=yes 9 ==> windy=FALSE 6    <conf:(0.67)> lift:(1.17) lev:(0.06) [0] conv:(0.96)
 9. temperature=mild 6 ==> humidity=high 4    <conf:(0.67)> lift:(1.33) lev:(0.07) [1] conv:(1)
10. temperature=mild 6 ==> play=yes 4    <conf:(0.67)> lift:(1.04) lev:(0.01) [0] conv:(0.71)
```

The above results show, the rules with confidence level 0.67 have also been included in the best rules. When the temperature is normal and cool the person can play. In the first case when the humidity is high the person does not play. Whereas with 0.67 confidence level the person plays when the humidity is high and temperature is mild.

Exercise 4:

Mining Association Rule with WEKA Explorer – Vote

Now consider a real-world dataset, vote.arff, which gives the votes of 435 U.S. congressmen on 16

key issues gathered in the mid-1980s, and also includes their party affiliation as a binary attribute.

Association-rule mining can also be applied to this data to seek interesting associations.

Load data at the Preprocess tab. Click the Open file button to bring up a standard dialog through which

you can select a file. Choose the vote.arff file. To see the original dataset, click the Edit button, a viewer window opens with dataset loaded. This is a purely nominal dataset with some missing values

(corresponding to abstentions).

Task 1. Run Apriori on this data with default settings. Comment on the rules that are generated. Several of them are quite similar. How are their support and confidence values related?

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -V -c -1
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
=== Associator model (full training set) ===
```

```
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)> lift:(1.63) lev:(0.19) [84] conv:(84.58)
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198    <conf:(1)> lift:(1.63) lev:(0.18) [76] conv:(76
 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210    <conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74)
 4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201    <conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01)
 5. physician-fee-freeze=n 247 ==> Class=democrat 245    <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8)
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18)
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46)
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198    <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85)
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210    <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47)
```

1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219
<conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58)
2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198
==> Class=democrat 198 <conf:(1)> lift:(1.63) lev:(0.18) [76] conv:(76.47)
3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210
<conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74)
4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201
<conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01)
5. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21)
[93] conv:(31.8)
6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197
<conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18)

7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46)

8. adoption-of-the-budget-resolution=y aid-to-Nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198 <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:(14.62)

9. el-Salvador-aid=n aid-to-Nicaraguan-contras=y 204 ==> Class=democrat 197 <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85)

10. aid-to-Nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210
<conf:(0.96)> lift:(1.7) lev:(0.2) [86] Conv:(10.47)

The rules depict that with an above 90% confidence level it can be stated that the person is a Democrat. In the majority cases, the person is a Democrat. When there is adoption-of-the-budget-resolution and physician fees have been frozen and there is no spending on education done the class is a democrat. When there is an aid to Nicaraguan contras and no aid to El Salvador, the class belongs to a democrat. The data shows a high confidence level with respect to the rules with a minimum of 98% confidence that given the following conditions, the class will be a Democrat

Task 2. It is interesting to see that none of the rules in the default output involve Class = republican. Why do you think that is?

The Apriori algorithm looks at the frequency of each instance for the formation of rules. In the dataset given, 267 instances belong to Democrats and 168 instances belong to Republicans. Since there is a bias in the data and the frequency of Democrats is more, the class is more likely to be predicted as a Democrat than a republican.

Exercise 5:

Let’s run Apriori on another real-world dataset.

Load data at Pre-process tab. Click the Open file button to bring up a standard dialog through which
you can select a file. Choose the supermarket.arff file. To see the original dataset, click the Edit button, and a viewer window opens with the dataset loaded.

To do market basket analysis in Weka, each transaction is coded as an instance of which the attributes represent the items in the store. Each attribute has only one value: If a particular transaction does not contain it (i.e., the customer did not buy that item), this is coded as a missing value.
Task 1. Experiment with Apriori and investigate the effect of the various parameters described before. Prepare a brief oral presentation on the main findings of your investigation.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c -1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.4 (1851 instances)
Minimum metric <confidence>: 0.7
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 18

Size of set of large itemsets L(2): 16
```

```
Best rules found:

 1. biscuits=t 2605 ==> bread and cake=t 2083    <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4)
 2. milk-cream=t 2939 ==> bread and cake=t 2337    <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37)
 3. fruit=t 2962 ==> bread and cake=t 2325    <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3)
 4. baking needs=t 2795 ==> bread and cake=t 2191    <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29)
 5. frozen foods=t 2717 ==> bread and cake=t 2129    <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29)
 6. vegetables=t 2961 ==> bread and cake=t 2298    <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25)
 7. juice-sat-cord-ms=t 2463 ==> bread and cake=t 1869    <conf:(0.76)> lift:(1.05) lev:(0.02) [96] conv:(1.16)
 8. vegetables=t 2961 ==> fruit=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41)
 9. fruit=t 2962 ==> vegetables=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41)
10. bread and cake=t 3330 ==> milk-cream=t 2337    <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22)
```

From the above dataset around 217 attributes have been omitted.
With a min support of 0.4 and min confidence of 0.7 the following can be be interpreted:

1. A customer who buys biscuits and milk cream is 80% likely to buy bread and cake.

2. A customer who buys fruit is78% likely to buy bread and cake.
3. The person who has baking needs, frozen foods, and vegetables are 78% likely to buy bread and cake.
4. A customer buying fruits is 75% likely to buy vegetables and vice-versa.
5. A customer purchasing bread and cakes is 70% likely to buy milk cream.
6. In a supermarket, the following foods can be grouped:
    a. Biscuits, Milk cream, baking needs, frozen foods, bread, and cake
    b. Vegetables and fruits can be stacked together.
    c. Juices and bread and cake together
7. The above strategy can give a more customer-friendly service and increase the sales of the supermarket

Inferences:

1. Apriori algorithm takes into account the frequency of an element and the probability that will occur was predicting the rules.
2. The rules come with a certain confidence level depicting with what certainty the rule can be stated.
3. The values of confidence and minimum support depend on the type of data to be predicted.
4. The outcome of a rule depends on the frequency of the class value.
5. If there is an imbalance in the data, the algorithm may give biased or inaccurate results.
6. Weka is a powerful tool to study apriori algorithm in real life scenarios and apply it to the benefit of the user.