

**Rapport Travaux Pratiques :  
Acquisition de connaissance 2  
- TP 3:  
Génération de règles d'association**

Nicolas Desfeux  
Aurélien Texier

16 mars 2011

**Table des matières**

<b>1</b>	<b>Première génération de règles d'associations</b>	<b>2</b>
<b>2</b>	<b>Étude de la population américaine</b>	<b>5</b>
<b>3</b>	<b>Étude d'articles de presse</b>	<b>9</b>

# 1 Première génération de règles d'associations

**Question 1.1** Dans cette partie, on choisit d'étudier un ensemble de données mettant en relation la météo et le fait de jouer ou non au golf. Pour générer les règles d'association, nous allons utiliser l'algorithme *APriori*.

Voici le résultat fourni par Weka :

```
1  === Run information ===
2
3  Scheme:      weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M
      0.1 -S -1.0 -c -1
4  Relation:    weather.symbolic
5  Instances:   14
6  Attributes:  5
7              outlook
8              temperature
9              humidity
10             windy
11             play
12  === Associator model (full training set) ===
13
14
15  Apriori
16  =====
17
18  Minimum support: 0.15 (2 instances)
19  Minimum metric <confidence>: 0.9
20  Number of cycles performed: 17
21
22  Generated sets of large itemsets:
23
24  Size of set of large itemsets L(1): 12
25
26  Size of set of large itemsets L(2): 47
27
28  Size of set of large itemsets L(3): 39
29
30  Size of set of large itemsets L(4): 6
31
32  Best rules found:
33
34  1. outlook=overcast 4 ==> play=yes 4    conf:(1)
35  2. temperature=cool 4 ==> humidity=normal 4    conf:(1)
36  3. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1)
37  4. outlook=sunny play=no 3 ==> humidity=high 3    conf:(1)
38  5. outlook=sunny humidity=high 3 ==> play=no 3    conf:(1)
39  6. outlook=rainy play=yes 3 ==> windy=FALSE 3    conf:(1)
40  7. outlook=rainy windy=FALSE 3 ==> play=yes 3    conf:(1)
41  8. temperature=cool play=yes 3 ==> humidity=normal 3    conf:(1)
42  9. outlook=sunny temperature=hot 2 ==> humidity=high 2    conf:(1)
43  10. temperature=hot play=no 2 ==> outlook=sunny 2    conf:(1)
```

On voit donc que Weka trouve plusieurs règles d'associations. Les plus significatives (qui sont en accords avec le plus d'exemples) indiquent que lorsque le temps est couvert, la température moyenne, l'humidité normale et le vent nul, le joueur va probablement jouer.

**Question 1.2** Sur le même exemple, on va utiliser différentes mesures pour produire les règles. Voici les règles obtenues :

Listing 1 – "Règles produites avec le lift"

```

1 Best rules found:
2
3 1. temperature=cool 4 ==> humidity=normal 4    conf:(1) < lift:(2)> lev
   : (0.14) [2] conv:(2)
4 2. humidity=normal 7 ==> temperature=cool 4    conf:(0.57) < lift:(2)> lev
   : (0.14) [2] conv:(1.25)
5 3. humidity=high 7 ==> play=no 4    conf:(0.57) < lift:(1.6)> lev:(0.11) [1]
   conv:(1.13)
6 4. play=no 5 ==> humidity=high 4    conf:(0.8) < lift:(1.6)> lev:(0.11) [1]
   conv:(1.25)
7 5. outlook=overcast 4 ==> play=yes 4    conf:(1) < lift:(1.56)> lev:(0.1)
   [1] conv:(1.43)
8 6. play=yes 9 ==> outlook=overcast 4    conf:(0.44) < lift:(1.56)> lev:(0.1)
   [1] conv:(1.07)
9 7. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1) < lift:(1.56)>
   lev:(0.1) [1] conv:(1.43)
10 8. play=yes 9 ==> humidity=normal windy=FALSE 4    conf:(0.44) < lift:(1.56)
   > lev:(0.1) [1] conv:(1.07)
11 9. humidity=normal 7 ==> play=yes 6    conf:(0.86) < lift:(1.33)> lev:(0.11)
   [1] conv:(1.25)
12 10. play=yes 9 ==> humidity=normal 6    conf:(0.67) < lift:(1.33)> lev:(0.11)
   [1] conv:(1.13)

```

Listing 2 – "Règles produites avec le leverage"

```

1 Best rules found:
2
3 1. temperature=cool 4 ==> humidity=normal 4    conf:(1) lift:(2) < lev
   : (0.14) [2]> conv:(2)
4 2. humidity=normal 7 ==> temperature=cool 4    conf:(0.57) lift:(2) < lev
   : (0.14) [2]> conv:(1.25)
5 3. humidity=normal 7 ==> play=yes 6    conf:(0.86) lift:(1.33) < lev:(0.11)
   [1]> conv:(1.25)
6 4. play=yes 9 ==> humidity=normal 6    conf:(0.67) lift:(1.33) < lev:(0.11)
   [1]> conv:(1.13)
7 5. humidity=high 7 ==> play=no 4    conf:(0.57) lift:(1.6) < lev:(0.11) [1]>
   conv:(1.13)
8 6. play=no 5 ==> humidity=high 4    conf:(0.8) lift:(1.6) < lev:(0.11) [1]>
   conv:(1.25)
9 7. outlook=overcast 4 ==> play=yes 4    conf:(1) lift:(1.56) < lev:(0.1)
   [1]> conv:(1.43)
10 8. play=yes 9 ==> outlook=overcast 4    conf:(0.44) lift:(1.56) < lev:(0.1)
   [1]> conv:(1.07)

```

```

11 9. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1) lift:(1.56) <
    lev:(0.1) [1] > conv:(1.43)
12 10. play=yes 9 ==> humidity=normal windy=FALSE 4    conf:(0.44) lift:(1.56) <
    lev:(0.1) [1] > conv:(1.07)

```

### Listing 3 – "Règles produites avec conviction"

```

1 Best rules found:
2
3 1. temperature=cool 4 ==> humidity=normal 4    conf:(1) lift:(2) lev:(0.14)
    [2] < conv:(2)>
4 2. outlook=sunny humidity=high 3 ==> play=no 3    conf:(1) lift:(2.8) lev
    :(0.14) [1] < conv:(1.93)>
5 3. outlook=sunny play=no 3 ==> humidity=high 3    conf:(1) lift:(2) lev
    :(0.11) [1] < conv:(1.5)>
6 4. temperature=cool play=yes 3 ==> humidity=normal 3    conf:(1) lift:(2)
    lev:(0.11) [1] < conv:(1.5)>
7 5. outlook=overcast 4 ==> play=yes 4    conf:(1) lift:(1.56) lev:(0.1) [1] <
    conv:(1.43)>
8 6. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1) lift:(1.56) lev
    :(0.1) [1] < conv:(1.43)>
9 7. play=no 5 ==> outlook=sunny humidity=high 3    conf:(0.6) lift:(2.8) lev
    :(0.14) [1] < conv:(1.31)>
10 8. humidity=high play=no 4 ==> outlook=sunny 3    conf:(0.75) lift:(2.1) lev
    :(0.11) [1] < conv:(1.29)>
11 9. outlook=rainy play=yes 3 ==> windy=FALSE 3    conf:(1) lift:(1.75) lev
    :(0.09) [1] < conv:(1.29)>
12 10. humidity=normal 7 ==> play=yes 6    conf:(0.86) lift:(1.33) lev:(0.11)
    [1] < conv:(1.25)>

```

On constate que les règles produites avec les différentes mesures ne sont pas les mêmes ! En fait, on retrouve globalement les mêmes règles quelque soit les mesures, par contre, elles n'apparaissent pas dans le même ordre. Les différentes mesures donnent entre les règles différent. En effet, les règles produites par un ensemble ne peuvent pas être totalement différentes suivant la mesure que l'on utilise !

**Question 1.3** On vérifie les calculs pour la règle suivante :

$temperature = cool \Rightarrow humidity = normal$

$$confidence(temperature = cool \Rightarrow humidity = normal) = \frac{N_{temp=cool \wedge humidity=normal}}{N_{temp=cool}} = \frac{4}{14} = 1$$

La confidence est juste !

$$lift(temperature = cool \Rightarrow humidity = normal) = N_{\Omega} * \frac{N_{temp=cool \wedge humidity=normal}}{N_{temp=cool} * N_{humidity=normal}} = 14 * \frac{4}{4*7} = 2$$

Le lift est juste !

$leverage(temperature = cool \Rightarrow humidity = normal)$

$$Pr(A, B) - Pr(A) * Pr(B) =$$

$$\frac{N_{temp=cool \wedge humidity=normal}}{N_{\Omega}} - \frac{N_{temp=cool} * N_{humidity=normal}}{N_{\Omega}^2} = \frac{4}{14} - \frac{4*7}{14*14} = 0.138$$

Le leverage est bon (aux arrondis près).

**Question 1.4** La mesure de conviction est une règle assez similaire au lift. Par contre, on s'intéresse ici aux exemples où la partie droite de la règle n'est pas respectée.

## 2 Étude de la population américaine

**Question 2.1** La recherche des attributs pertinents par Weka donne le résultat suivant :

```
1  === Run information ===
2
3  Evaluator:      weka.attributeSelection.CfsSubsetEval
4  Search:         weka.attributeSelection.BestFirst -D 1 -N 5
5  Relation:      adult1
6  Instances:      250
7  Attributes:     15
8                  age
9                  workclass
10                 fnlwgt
11                 education
12                 education-num
13                 marital-status
14                 occupation
15                 relationship
16                 race
17                 sex
18                 capital-gain
19                 capital-loss
20                 hours-per-week
21                 native-country
22                 gain
23 Evaluation mode:  evaluate on all training data
24
25
26
27 === Attribute Selection on all input data ===
28
29 Search Method:
30     Best first.
31     Start set: no attributes
32     Search direction: forward
33     Stale search after 5 node expansions
34     Total number of subsets evaluated: 96
35     Merit of best subset found:      0.236
36
37 Attribute Subset Evaluator (supervised, Class (nominal): 15 gain):
38     CFS Subset Evaluator
39     Including locally predictive attributes
40
41 Selected attributes: 1,5,8,11 : 4
42                     age
43                     education-num
44                     relationship
45                     capital-gain
```

On choisit de garder les 4 attributs choisis par Weka, auxquels on ajoute occupation, race, sex car ils nous semblent intéressants pour étudier leur influence sur les données. On garde également le gain puisque c'est celui que l'on veut expliquer.

**Question 2.4** On applique APriori avec différents paramètres, et on obtient les règles suivantes :

Listing 4 – "Règles produites avec conviction"

```

1 Best rules found:
2
3 1. relationship=_Husband 103 ==> sex=_Male 103    conf:(1) lift:(1.45) lev
:(0.13) [32] < conv:(32.14)>
4 2. relationship=_Husband capital-gain='(-inf -704.5]' 95 ==> sex=_Male 95
   conf:(1) lift:(1.45) lev:(0.12) [29] < conv:(29.64)>
5 3. relationship=_Husband 103 ==> sex=_Male capital-gain='(-inf -704.5]' 95
   conf:(0.92) lift:(1.43) lev:(0.11) [28] < conv:(4.07)>
6 4. education-num='(-inf -9.5]' capital-gain='(-inf -704.5]' 100 ==> gain=_
   <=50K 91    conf:(0.91) lift:(1.19) lev:(0.06) [14] < conv:(2.36)>
7 5. education-num='(-inf -9.5]' 108 ==> gain=_<=50K 96    conf:(0.89) lift
   :(1.16) lev:(0.05) [13] < conv:(1.96)>
8 6. education-num='(-inf -9.5]' 108 ==> capital-gain='(-inf -704.5]' gain=_
   <=50K 91    conf:(0.84) lift:(1.15) lev:(0.05) [11] < conv:(1.61)>
9 7. sex=_Male 172 ==> relationship=_Husband 103    conf:(0.6) lift:(1.45)
   lev:(0.13) [32] < conv:(1.44)>
10 8. gain=_<=50K 191 ==> capital-gain='(-inf -704.5]' 183    conf:(0.96) lift
   :(1.03) lev:(0.02) [4] < conv:(1.44)>
11 9. sex=_Male capital-gain='(-inf -704.5]' 161 ==> relationship=_Husband 95
   conf:(0.59) lift:(1.43) lev:(0.11) [28] < conv:(1.41)>
12 10. sex=_Male 172 ==> relationship=_Husband capital-gain='(-inf -704.5]' 95
    conf:(0.55) lift:(1.45) lev:(0.12) [29] < conv:(1.37)>

```

Listing 5 – "Règles produites avec confiance"

```

1 Best rules found:
2
3 1. relationship=_Husband 103 ==> sex=_Male 103    conf:(1)
4 2. relationship=_Husband capital-gain='(-inf -704.5]' 95 ==> sex=_Male 95
   conf:(1)
5 3. gain=_<=50K 191 ==> capital-gain='(-inf -704.5]' 183    conf:(0.96)
6 4. race=_White gain=_<=50K 153 ==> capital-gain='(-inf -704.5]' 146
   conf:(0.95)
7 5. sex=_Male gain=_<=50K 123 ==> capital-gain='(-inf -704.5]' 117    conf
   :(0.95)
8 6. race=_White sex=_Male gain=_<=50K 102 ==> capital-gain='(-inf -704.5]'
   97    conf:(0.95)
9 7. education-num='(-inf -9.5]' gain=_<=50K 96 ==> capital-gain='(-inf
   -704.5]' 91    conf:(0.95)
10 8. race=_White sex=_Male 141 ==> capital-gain='(-inf -704.5]' 132    conf
   :(0.94)
11 9. sex=_Male 172 ==> capital-gain='(-inf -704.5]' 161    conf:(0.94)
12 10. education-num='(-inf -9.5]' 108 ==> capital-gain='(-inf -704.5]' 100
    conf:(0.93)

```

### Listing 6 – "Règles produites avec le lift"

```

1 Best rules found:
2
3 1. relationship=_Husband 103 ==> sex=_Male 103    conf:(1) < lift:(1.45)>
   lev:(0.13) [32] conv:(32.14)
4 2. sex=_Male 172 ==> relationship=_Husband 103    conf:(0.6) < lift:(1.45)
   > lev:(0.13) [32] conv:(1.44)
5 3. sex=_Male 172 ==> relationship=_Husband    capital-gain='(-inf -704.5]' 95
   conf:(0.55) < lift:(1.45)> lev:(0.12) [29] conv:(1.37)
6 4. relationship=_Husband    capital-gain='(-inf -704.5]' 95 ==> sex=_Male 95
   conf:(1) < lift:(1.45)> lev:(0.12) [29] conv:(29.64)
7 5. relationship=_Husband 103 ==> sex=_Male    capital-gain='(-inf -704.5]' 95
   conf:(0.92) < lift:(1.43)> lev:(0.11) [28] conv:(4.07)
8 6. sex=_Male    capital-gain='(-inf -704.5]' 161 ==> relationship=_Husband 95
   conf:(0.59) < lift:(1.43)> lev:(0.11) [28] conv:(1.41)
9 7. education-num='(-inf -9.5]'    capital-gain='(-inf -704.5]' 100 ==> gain=_
   <=50K 91    conf:(0.91) < lift:(1.19)> lev:(0.06) [14] conv:(2.36)
10 8. gain=_<=50K 191 ==> education-num='(-inf -9.5]'    capital-gain='(-inf
   -704.5]' 91    conf:(0.48) < lift:(1.19)> lev:(0.06) [14] conv:(1.13)
11 9. education-num='(-inf -9.5]' 108 ==> gain=_<=50K 96    conf:(0.89) < lift
   :(1.16)> lev:(0.05) [13] conv:(1.96)
12 10. gain=_<=50K 191 ==> education-num='(-inf -9.5]' 96    conf:(0.5) < lift
   :(1.16)> lev:(0.05) [13] conv:(1.13)

```

### Listing 7 – "Règles produites avec leverage"

```

1
2 Best rules found:
3
4 1. relationship=_Husband 103 ==> sex=_Male 103    conf:(1) lift:(1.45) <
   lev:(0.13) [32]> conv:(32.14)
5 2. sex=_Male 172 ==> relationship=_Husband 103    conf:(0.6) lift:(1.45) <
   lev:(0.13) [32]> conv:(1.44)
6 3. sex=_Male 172 ==> relationship=_Husband    capital-gain='(-inf -704.5]' 95
   conf:(0.55) lift:(1.45) < lev:(0.12) [29]> conv:(1.37)
7 4. relationship=_Husband    capital-gain='(-inf -704.5]' 95 ==> sex=_Male 95
   conf:(1) lift:(1.45) < lev:(0.12) [29]> conv:(29.64)
8 5. relationship=_Husband 103 ==> sex=_Male    capital-gain='(-inf -704.5]' 95
   conf:(0.92) lift:(1.43) < lev:(0.11) [28]> conv:(4.07)
9 6. sex=_Male    capital-gain='(-inf -704.5]' 161 ==> relationship=_Husband 95
   conf:(0.59) lift:(1.43) < lev:(0.11) [28]> conv:(1.41)
10 7. relationship=_Husband 103 ==> race=_White sex=_Male 85    conf:(0.83)
   lift:(1.46) < lev:(0.11) [26]> conv:(2.36)
11 8. race=_White sex=_Male 141 ==> relationship=_Husband 85    conf:(0.6)
   lift:(1.46) < lev:(0.11) [26]> conv:(1.45)
12 9. relationship=_Husband    race=_White 85 ==> sex=_Male 85    conf:(1) lift
   :(1.45) < lev:(0.11) [26]> conv:(26.52)
13 10. sex=_Male 172 ==> relationship=_Husband    race=_White 85    conf:(0.49)
   lift:(1.45) < lev:(0.11) [26]> conv:(1.29)

```

Pour différents jeux de paramètres, on obtient souvent les mêmes règles. Par contre, l'ordre des règles n'est pas forcément le même. Voici quelques exemples de règles produites, que nous allons interpréter :

- Produite par confidence  
`relationship = _Husband 103 ==> sex = _Male 103 conf : ( 1 )`  
 Cette règle indique que le fait d'être un Husband implique que l'on soit un homme. La règle n'a pas forcément d'intérêt, mais est tout à fait logique et vraie pour notre ensemble de données
- Produite par le lift  
`gain = _<=50K 191 ==> education-num= ' (-inf -9.5 ] ' 96 conf : (0.5 ) < lift : (1.16) > lev : (0.05) [13] conv : (1.13)`  
 Le gain est inférieur à 50K implique que le niveau d'éducation est faible. Le gain et le niveau d'éducation sont assez corrélé puisque le lift est supérieur à 1.

**Question 2.5** Voici le résultat que l'on obtient lorsque l'on passe l'option `car` à `true` et la mesure à `confidence` :

```

1
2 === Run information ===
3
4 Scheme:      weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M
               0.1 -S -1.0 -A -c -l
5 Relation:    adult1-weka.filters.unsupervised.attribute.Remove-R2-3,6,12-14-
               weka.filters.unsupervised.attribute.Remove-R2-weka.filters.unsupervised.
               attribute.Discretize-F-B3-M-1.0-Rfirst-last
6 Instances:   250
7 Attributes:  8
8              age
9              education-num
10             occupation
11             relationship
12             race
13             sex
14             capital-gain
15             gain
16 === Associator model (full training set) ===
17
18
19 Apriori
20 =====
21
22 Minimum support: 0.1 (25 instances)
23 Minimum metric <confidence>: 0.9
24 Number of cycles performed: 18
25
26 Generated sets of large itemsets:
27
28 Size of set of large itemsets L(1): 23
29
30 Size of set of large itemsets L(2): 54
31
32 Size of set of large itemsets L(3): 44
33
34 Size of set of large itemsets L(4): 13

```



```

35
36 Best rules found:
37
38 1. age='(-inf -31.5]' education-num='(-inf -9.5]' 38 ==> gain=_<=50K 38
   conf:(1)
39 2. relationship=_Own-child 36 ==> gain=_<=50K 36 conf:(1)
40 3. age='(-inf -31.5]' education-num='(-inf -9.5]' capital-gain='(-inf -704.5]
   ' 36 ==> gain=_<=50K 36 conf:(1)
41 4. relationship=_Own-child capital-gain='(-inf -704.5]' 35 ==> gain=_<=50K
   35 conf:(1)
42 5. age='(-inf -31.5]' education-num='(-inf -9.5]' race=_White 34 ==> gain=_
   <=50K 34 conf:(1)
43 6. age='(-inf -31.5]' relationship=_Own-child 32 ==> gain=_<=50K 32 conf
   :(1)
44 7. age='(-inf -31.5]' education-num='(-inf -9.5]' race=_White capital-gain=
   '(-inf -704.5]' 32 ==> gain=_<=50K 32 conf:(1)
45 8. age='(-inf -31.5]' relationship=_Own-child capital-gain='(-inf -704.5]'
   31 ==> gain=_<=50K 31 conf:(1)
46 9. relationship=_Own-child race=_White 30 ==> gain=_<=50K 30 conf:(1)
47 10. relationship=_Own-child race=_White capital-gain='(-inf -704.5]' 29 ==>
   gain=_<=50K 29 conf:(1)

```

### 3 Étude d'articles de presse

**Question 3.1** L'étape particulière d'Apriori qui joue dans la complexité des calculs est l'étape 1 (trouver tous les itemsets) car 200 mots signifie une complexité en 2 puissance 200 !

**Question 3.2** Pour construire un fichier utilisable par Weka, nous avons travaillé les données à l'aide du script suivant :

Listing 8 – "Méthode"

```

1  #!/usr/bin/perl
2  #
3  # owner Peggy Cellier
4  #
5  # Convertit le fichier articles.txt vers le format de weka
6  # syntax: perl txt2weka.pl articles.10p.txt mots.lst
7  #
8
9  #use strict;
10
11 if ($ARGV[0] eq "") {
12     print "_syntaxe_: _perl_outil_fichierEntree_FichierSortie_\n" ;
13     exit(-1) ;
14 } ;
15 $fichText = $ARGV[0] ; # récupération du fichier texte contenant les phrases
   lemmatisées
16
17
18 if ($ARGV[1] eq "") {

```

```

19         print "_syntaxe:_perl_outil_fichierEntree_FichierSortie_\n" ;
20         exit(-1) ;
21     } ;
22     $fichMots = $ARGV[1] ; # récupération du fichier contenant les mots
        significatifs
23
24     $fichOut=$fichText.".csv";
25
26     open (FICHOUT, ">:encoding(utf8)", $fichOut)
27         || die "\n_impossible_d'ouvrir_le_fichier_nommé_
        $fichOut_\n\n" ;
28
29     #
30     open (FICHTEXT, "<:encoding(iso-8859-15)", $fichText)
31         || die "\n_impossible_d'ouvrir_le_fichier_d'entrée_
        nommé_$fichIn_\n\n" ;
32
33     open (FICHMOTS, "<:encoding(iso-8859-15)", $fichMots)
34         || die "\n_impossible_d'ouvrir_le_fichier_d'entrée_
        nommé_$fichIn2_\n\n" ;
35
36     my @tabMots = ();
37     my $i=0;
38     while($ligne = <FICHMOTS>){
39         chomp($ligne);
40         push(@tabMots,$ligne);
41         if($i==0){
42             print FICHOUT "$ligne";
43             $i=$i+1;
44         } else {
45             print FICHOUT ",_,$ligne";
46         }
47     }
48     print FICHOUT "\n";
49     while($ligne = <FICHTEXT>){
50         my $b=0;
51         foreach my $a (@tabMots){
52             if($b==0){
53                 $virg="";
54                 $b= $b+1;
55             } else {
56                 $virg=",_";
57             }
58             if($ligne=~ ".$a."){
59                 print FICHOUT $virg."true";
60             } else {
61                 print FICHOUT $virg."false";
62             }
63         }
64     }
65     print FICHOUT "\n";
66 }
67 print "Fin_creation\n";

```

```

68 # Rappel :
69 # ($line =~ m/\s$i\s/)
70 # ^ caractère de début de ligne
71 # $ caractère de fin de ligne
72 #
73 #@my @items = ();
74 # push(@items, $line) ; // ajoute un objet
75 # foreach my $i (@items) // parcours du tableau
76
77 close (FICHTEXT);
78 close (FICHMOTS);
79 close (FICHOUT);

```

L'exécution de ce script nous fournit un fichier avec :

- Sur la première ligne : La liste des mots recherchées dans les différents articles.
- Sur les lignes suivantes : La présence de ces différents mots dans les différents articles.

Il faut se représenter ce fichier comme un tableau dans lequel, pour chaque article, on indique la présence ou l'absence des mots que l'on a d'abord identifiés.

Ce fichier, nous pouvons ensuite l'utiliser dans Weka, afin d'en faire l'analyse, et de générer des règles d'associations.

```

1 === Run information ===
2
3 Scheme:          weka.associations.Apriori -N 20 -T 1 -C 1.1 -D 0.05 -U 0.5 -M
   0.05 -S -1.0 -c -l
4 Relation:        tp3
5 Instances:        468
6 Attributes:       200
7                   [list of attributes omitted]
8 === Associator model (full training set) ===
9
10
11 Apriori
12 =====
13
14 Minimum support: 0.15 (70 instances)
15 Minimum metric <lift>: 1.1
16 Number of cycles performed: 7
17
18 Generated sets of large itemsets:
19
20 Size of set of large itemsets L(1): 37
21
22 Size of set of large itemsets L(2): 23
23
24 Size of set of large itemsets L(3): 2
25
26 Best rules found:
27
28 1.  ministre=_true 73 ==>  ministre=_true 73    conf:(1) < lift:(6.41)> lev
    : (0.13) [61] conv:(61.61)

```

29 2. ministre=\_true 73 ==> ministre=\_true 73 conf:(1) < lift:(6.41)> lev  
:(0.13) [61] conv:(61.61)

30 3. élection=\_true 76 ==> élection=\_true 76 conf:(1) < lift:(6.16)> lev  
:(0.14) [63] conv:(63.66)

31 4. élection=\_true 76 ==> élection=\_true 76 conf:(1) < lift:(6.16)> lev  
:(0.14) [63] conv:(63.66)

32 5. campagne=\_true 77 ==> campagne=\_true 77 conf:(1) < lift:(6.08)> lev  
:(0.14) [64] conv:(64.33)

33 6. campagne=\_true 77 ==> campagne=\_true 77 conf:(1) < lift:(6.08)> lev  
:(0.14) [64] conv:(64.33)

34 7. campagne=\_true 77 ==> camp=\_true campagne=\_true 77 conf:(1) < lift  
:(6.08)> lev:(0.14) [64] conv:(64.33)

35 8. campagne=\_true camp=\_true 77 ==> campagne=\_true 77 conf:(1) < lift  
:(6.08)> lev:(0.14) [64] conv:(64.33)

36 9. campagne=\_true 77 ==> campagne=\_true camp=\_true 77 conf:(1) < lift  
:(6.08)> lev:(0.14) [64] conv:(64.33)

37 10. camp=\_true campagne=\_true 77 ==> campagne=\_true 77 conf:(1) < lift  
:(6.08)> lev:(0.14) [64] conv:(64.33)

38 11. campagne=\_true 77 ==> camp=\_true 77 conf:(1) < lift:(4.93)> lev  
:(0.13) [61] conv:(61.37)

39 12. camp=\_true 95 ==> campagne=\_true 77 conf:(0.81) < lift:(4.93)> lev  
:(0.13) [61] conv:(4.18)

40 13. camp=\_true 95 ==> campagne=\_true 77 conf:(0.81) < lift:(4.93)> lev  
:(0.13) [61] conv:(4.18)

41 14. campagne=\_true 77 ==> camp=\_true 77 conf:(1) < lift:(4.93)> lev  
:(0.13) [61] conv:(61.37)

42 15. camp=\_true 95 ==> campagne=\_true campagne=\_true 77 conf:(0.81) <  
lift:(4.93)> lev:(0.13) [61] conv:(4.18)

43 16. campagne=\_true campagne=\_true 77 ==> camp=\_true 77 conf:(1) < lift  
:(4.93)> lev:(0.13) [61] conv:(61.37)

44 17. euro=\_true 136 ==> européen=\_true 71 conf:(0.52) < lift:(3.44)> lev  
:(0.11) [50] conv:(1.75)

45 18. européen=\_true 71 ==> euro=\_true 71 conf:(1) < lift:(3.44)> lev  
:(0.11) [50] conv:(50.37)

46 19. candidat=\_true 91 ==> président=\_true 70 conf:(0.77) < lift:(2.2)>  
lev:(0.08) [38] conv:(2.69)

47 20. président=\_true 164 ==> candidat=\_true 70 conf:(0.43) < lift:(2.2)>  
lev:(0.08) [38] conv:(1.39)

On a choisi après plusieurs essais de choisir le lift comme critère de sélection car il nous paraissait celui qui donnait les meilleurs résultats. Toutes les règles générées ne sont pas fortement intéressantes dans le sens où certaines disent que la présence d'un mot entraîne la présence du même mot (comme par exemple avec "campagne". Cependant, on voit quand même des résultats beaucoup plus intéressants comme par exemple la présence du mot "candidat" qui entraîne la présence du mot "président". Nous avons ensuite essayé de générer 300 règles pour voir ce que l'on obtenait parmi les règles qui n'utilisent pas le même mot, et l'on a pu en découvrir certaines autres très intéressantes également. Les voici :

1 3. chirac=**true** 57 ==> chirac=\_true président=\_true 49 conf:(0.86) <  
lift:(8.21)> lev:(0.09) [43] conv:(5.67)

2 11. présidentiel=\_true 64 ==> présidentiel=\_true président=\_true 64  
conf:(1) < lift:(7.31)> lev:(0.12) [55] conv:(55.25)

3 15. présidentiel=\_true 64 ==> candidat=\_true présidentiel=\_true 48  
 conf:(0.75) < lift:(7.31)> lev:(0.09) [41] conv:(3.38)

4 19. présidentiel=\_true 64 ==> candidat=\_true présidentiel=\_true  
 président=\_true 48 conf:(0.75) < lift:(7.31)> lev:(0.09) [41] conv  
 :(3.38)

5 20. présidentiel=\_true candidat=\_true 48 ==> présidentiel=\_true  
 président=\_true 48 conf:(1) < lift:(7.31)> lev:(0.09) [41] conv  
 :(41.44)

6 21. présidentiel=\_true 64 ==> présidentiel=\_true candidat=\_true  
 président=\_true 48 conf:(0.75) < lift:(7.31)> lev:(0.09) [41] conv  
 :(3.38)

7 22. candidat=\_true présidentiel=\_true 48 ==> présidentiel=\_true  
 président=\_true 48 conf:(1) < lift:(7.31)> lev:(0.09) [41] conv  
 :(41.44)

8 23. présidentiel=\_true président=\_true 64 ==> candidat=\_true  
 présidentiel=\_true 48 conf:(0.75) < lift:(7.31)> lev:(0.09) [41] conv  
 :(3.38)

9 24. présidentiel=\_true candidat=\_true président=\_true 48 ==>  
 présidentiel=\_true 48 conf:(1) < lift:(7.31)> lev:(0.09) [41] conv  
 :(41.44)

10 25. présidentiel=\_true président=\_true 64 ==> présidentiel=\_true  
 candidat=\_true 48 conf:(0.75) < lift:(7.31)> lev:(0.09) [41] conv  
 :(3.38)

11 47. ministre=\_true 73 ==> président=\_true ministre=\_true 48 conf  
 :(0.66) < lift:(6.41)> lev:(0.09) [40] conv:(2.52)

12 50. président=\_true ministre=\_true 48 ==> ministre=\_true 48 conf:(1) <  
 lift:(6.41)> lev:(0.09) [40] conv:(40.51)

13 51. ministre=\_true 73 ==> politique=\_true ministre=\_true 47 conf  
 :(0.64) < lift:(6.41)> lev:(0.08) [39] conv:(2.43)

14 55. candidat=\_true campagne=\_true 55 ==> président=\_true camp=\_true 48  
 conf:(0.87) < lift:(6.28)> lev:(0.09) [40] conv:(5.92)

15 56. président=\_true camp=\_true 65 ==> candidat=\_true campagne=\_true 48  
 conf:(0.74) < lift:(6.28)> lev:(0.09) [40] conv:(3.19)

16 63. élection=\_true 76 ==> président=\_true élection=\_true 61 conf:(0.8)  
 < lift:(6.16)> lev:(0.11) [51] conv:(4.13)

17 268. international=\_true 63 ==> national=\_true 63 conf:(1) < lift:(2.82)  
 > lev:(0.09) [40] conv:(40.65)

18 271. candidat=\_true 91 ==> social=\_true 47 conf:(0.52) < lift:(2.49)>  
 lev:(0.06) [28] conv:(1.6)

19 272. social=\_true 97 ==> candidat=\_true 47 conf:(0.48) < lift:(2.49)>  
 lev:(0.06) [28] conv:(1.53)

20 277. président=\_true 164 ==> candidat=\_true campagne=\_true camp=\_true 48  
 conf:(0.29) < lift:(2.49)> lev:(0.06) [28] conv:(1.24)

21 285. chirac=**true** 57 ==> président=\_true 49 conf:(0.86) < lift:(2.45)> lev  
 :(0.06) [29] conv:(4.11)

22 286. président=\_true 164 ==> chirac=**true** 49 conf:(0.3) < lift:(2.45)> lev  
 :(0.06) [29] conv:(1.24)

23 297. jospin=\_true 59 ==> président=\_true 47 conf:(0.8) < lift:(2.27)>  
 lev:(0.06) [26] conv:(2.95)

24 298. président=\_true 164 ==> jospin=\_true 47 conf:(0.29) < lift:(2.27)>  
 lev:(0.06) [26] conv:(1.21)

Lorsque l'on génère des règles avec le , on obtient des règles différentes, comme par exemple :

```

1 conseil=_true 93 ==> président=_true 62    conf:(0.67) lift:(1.9) lev:(0.06)
  [29] < conv:(1.89)>
2 front=_true 73 ==> national=_true 49    conf:(0.67) lift:(1.89) lev:(0.05)
  [23] < conv:(1.88)>
3 loi=_false art=_false 109 ==> eau=_false 75    conf:(0.69) lift:(1.74) lev
  :(0.07) [31] < conv:(1.88)>
4 national=_true droit=_true 72 ==> politique=_true 47    conf:(0.65) lift
  :(2.01) lev:(0.05) [23] < conv:(1.87)>
5 candidat=_true 91 ==> président=_true camp=_true 50    conf:(0.55) lift
  :(3.96) lev:(0.08) [37] < conv:(1.87)>

```

On constate que certaines absences de mots entraînent l'absence d'autres mots,, ce qui est différent des règles produites par le lift.