

Correction de l'examen - UP3

partie Clustering, Classification, Règles d'Association

le 12 décembre 2023

Les documents papier sont autorisés,..... L'ordre de résolution des sujets n'est pas imposé, bien au contraire.

La phrase ci-dessus n'était pas innocente. Ordre conseillé : 2 avant 1.

1. **Transactions d'achat** 8pt : 3 pt pour dérouler l'algorithme de votre choix, 2pt pour obtenir la liste correcte et complète des itemsets fréquents, 3pt pour les RA

La base suivante traduit une série de transactions de type "panier".

Id	Transaction
T100	C, A, F, E
T200	C, A, F, E, I, N
T300	C, I, E, L
T400	L, I, A, N, E
T500	E, L, A, I, T
T600	L, A, T, E

Soient les limites du support (*min_support*) à 49% et de la confiance (*min_confiance*) à 80%.

- Déroulez un algorithme vu en cours pour calculer tous les itemsets fréquents par rapport à *min_support* (*Attention : la limite est donnée en pourcentage !*)
- A partir du résultat calculé au point précédent calculer toutes les règles d'association de type $X \rightarrow Z$ qui sont au-delà de *min_confiance* (X et Z sont des items).

On déroule Apriori, Apriori-TID ou ECLAT et on le fait soit sur les transactions complètes, soit sur les transactions dont on a ignoré l'item *E*.

On obtient 15 itemsets fréquents qui sont les suivants :

```
  items      support
[1] {C}        0.5
[2] {L}        0.66
[3] {I}        0.66
[4] {A}        0.833
[5] {E}         1
[6] {C, E}     0.5
[7] {I, L}     0.5
[8] {A, L}     0.5
[9] {E, L}     0.66
[10] {A, I}    0.5
[11] {E, I}    0.66
[12] {A, E}    0.83
[13] {E, I, L} 0.50
[14] {A, E, L} 0.50
[15] {A, E, I} 0.50
```

Pour les RA on en obtient 5 qui ne sont pas trop intéressantes (LIFT à 1), résultat conforme au point suivant car elles contiennent E :

lhs	rhs	support	confidence	coverage	lift	count
[1] {C}	=> {E}	0.5000000	1.0000000	0.5000000	1	3
[2] {L}	=> {E}	0.6666667	1.0000000	0.6666667	1	4
[3] {I}	=> {E}	0.6666667	1.0000000	0.6666667	1	4
[4] {A}	=> {E}	0.8333333	1.0000000	0.8333333	1	5
[5] {E}	=> {A}	0.8333333	0.8333333	1.0000000	1	5

Pour les générer on avait deux possibilités :

- soit on prenait tous les 7 itemsets de 2 éléments
- soit on prenait uniquement les 2-itemsets sans E et on ne trouvait pas de RA ayant une bonne confiance, puis on intégrait E avec le 1-itemsets fréquents pour obtenir les RA ci-dessus selon les observations du point 2.

2. **Item trop fréquent** 3pt : 1pt effet avec et sans F dans un itemset quelconque , 1pt l'effet de F dans les RA, à gauche et à droite, 1pt pour tirer les bonnes conclusions

Dans une base de transactions l'item F apparaît dans toutes les transactions. Quel est son impact dans le calcul des itemsets fréquents et dans les règles d'association ? Selon l'énoncé $support(F) = 1$. Si X est un itemset sans F , alors $support(X) = support(X, F)$.

On peut faire les calculs de support en ignorant complètement F et en rajoutant après F aux itemset fréquents calculé.

Soit $X \rightarrow Y$ une RA qui ne contient pas de F , il est évident que :

$$support(X \rightarrow Y) = support(X, F \rightarrow Y) = support(X \rightarrow Y, F)$$

$$conf(X, F \rightarrow Y) = conf(X \rightarrow Y), \text{idem pour le LIFT}$$

Reste à analyser $F \rightarrow X$ et $X \rightarrow F$, en considérant que $support(X) > min_support$.

$$conf(F \rightarrow X) = \frac{support(X, F)}{support(F)} = support(X)$$

$$LIFT(F \rightarrow X) = \frac{support(X, F)}{support(F) support(X)} = 1$$

Sous l'hypothèse que $support(X) > min_confidence$; la RA $F \rightarrow X$ est une RA à prendre en compte, mais son LIFT indique qu'elle n'est pas significative.

$$conf(X \rightarrow F) = \frac{support(X, F)}{support(X)} = 1$$

$$LIFT(X \rightarrow F) = \frac{support(X, F)}{support(F) support(X)} = 1$$

Donc $X \rightarrow F$ est une RA pour tout X fréquent, mais elle n'est pas significative vu son LIFT.

3. **Drôles de fonctions** 4 pt : 2 pt à chaque item - 0,5pt la bonne réponse et 1,5pt la justification.

Soit $de : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ la fonction de distance euclidienne.

- (a) Un ensemble de données dans \mathbb{R}^2 est donné en coordonnées polaires (r, ϕ) . On considère $dp : (\mathbb{R}^+ \times [0, 2\pi])^2 \rightarrow \mathbb{R}_+$ comme

$$dp(x, y) = de((r_x, \phi_x), (r_y, \phi_y))$$

Est-ce que cette fonction dp peut être utilisée dans un algorithme de clustering ?

Oui, dp est une distance car elle a propriété de la définitivité ($dp(x, y)$ induit $x = y$), de symétrie et d'inégalité triangulaire.

- (b) Un ensemble de données dans \mathbb{R}^2 est donné en coordonnées cartésiennes (x, y) . Soient $r1$ et $r2$ deux points fixes du plan. Pour chaque point, on défini $f : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ comme :

$$f(x) = \min(de(x, r1), de(x, r2))$$

et pour une paire de points $df : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ comme :

$$df(x, y) = |f(x) - f(y)|$$

Même question : est-ce que cette fonction df peut être considérée comme une fonction distance ou dissimilarité ? Non, car dp n'a pas la propriété de définitivité : existent x et y $df(x, y) = 0$ sans que $x = y$. Il suffit de prendre deux points sur le médiatrice du segment donnée par les deux repaires, $r1$ et $r2$, et placer un point x d'un coté et un point y symétriquement de l'autre coté du centre du segment. De toute évidence $f(x) = f(y)$ ce qui induit $df(x, y) = 0$ et pourtant les points sont différents avec des coordonnées cartésiennes différentes.

4. **mammal.dentition** 2 pt : 1pt pour l'élément dans l'ensemble de test, 1pt pour le cas de l'ensemble d'apprentissage

Dans le TP sur le dataset **mammal.dentition** on devait mettre en place une classification avec k-NN. Une consigne indiquait de ne pas considérer les classes avec une seule observation. Pourquoi ?

Comme on travaillait uniquement avec ce petit dataset, si laisse une classe avec un seul élément dans l'ensemble d'apprentissage il risque de faiblir le score du choix parmi k voisins. Si l'élément de la classe isolée est dans l'ensemble de test, le résultat est forcément faux.

5. **code R et résultats** 3 pt : 1 pt pour l'explication du code, 1pt pour la variable **er**, 1pt pour l'explication

On exécute le code suivant dans la console R :

```
n <- nrow(donnees) ## donnees est un datafram avec 3 colonnes

I <- sample(1:n, (2*n)/3) # indices
J <- setdiff(1:n, I) # autres indices

cl <- donnees[I,3] # la classe à apprendre

dlnr <- donnees[I,1:2] #
dtest <- donnees[J,1:2] #

library (class)

mknn3 <- knn(dlnr, dtest, cl, k=3)
t <- table(mknn3, donnees[J,3])
er <- (t[1,2] + t[2,1])/sum(t)
```

Que fait ce code ?

Pour une première exécution du code on obtient $er = 0.1174$ et pour une seconde exécution $er = 0.08823$. Est-ce normal ?

Le code découpe de manière aléatoire l'ensemble initial en données d'apprentissage (2/3) et ensemble de test (1/3).

Ensuite un applique k-NN. La variable `er` calcule le taux d'erreur. Le code laisse supposer qu'il y a deux classes à prédire.

Le taux d'erreur change entre deux exécutions car le découpage initial change.