Acquisition de Connaissances 2
4INFO

Recherche de règles d'association

Ce TP est encadré sur deux séances (au total 4h). Il est à réaliser par groupe de 5 à 8 étudiants.

1 Implémentation de l'extraction de règles d'association

Vous devrez implémenter *from scratch* un algorithme de recherche de règles d'association. Vous avez le choix en ce qui concerne le langage utilisé (Java, C, C++, python, caml, ...) et en ce qui concerne l'algorithme utilisé. En cours, nous avons vu Apriori mais vous pouvez implémenter un autre algorithme.

Deux options sont possibles :

- Option 1 : Votre implémentation devra permettre de paramétrer les indices statistiques utilisés pour mesurer la pertinence des règles.
- Option 2 : Votre implémentation devra permettre de choisir parmi plusieurs indices statistiques proposés (au moins ceux vus en cours) ainsi que le type d'itemsets calculés.

1.1 Option 1 : Indices statistiques

En ce qui concerne le paramétrage des indices statistiques, nous avons vu en cours différents indices (par ex : confiance, lift). Pour une règle donnée $A \rightarrow B$, ces mesures sont définies à partir de 4 valeurs :

- $T(A)$: le nombre de transactions qui ont A dans leur description ;
- $T(B)$: le nombre de transactions qui ont B dans leur description ;
- $T(A \cap B)$: le nombre de transactions qui ont A et B dans leur description ;
- $T(\emptyset)$: le nombre total de transactions.

On vous demande de faire en sorte que votre algorithme puisse permettre de choisir une mesure quelconque définie à partir de ces 4 valeurs. Par exemple il faut pouvoir rentrer une nouvelle mesure qui n'est pas bien connue mais qui peut se définir à partir de ces 4 valeurs. Comment ce paramétrage modifie-t-il votre algorithme ?

1.2 Option 2 : Itemsets

En ce qui concerne le type d'itemsets, on vous demande, pour votre implémentation, que le choix du type d'itemsets à utiliser soit paramétrable et que l'on puisse choisir entre trois types : les itemsets fréquents, les itemsets maximaux fréquents et les itemsets fermés fréquents (voir la définition donnée ci-dessous pour ce dernier type d'itemsets). Dans votre rapport vous indiquerez comment vous gérez les itemsets fermés et maximaux et les différences observées (par ex. : temps de calcul, taille des résultats) entre l'utilisation de ces trois types d'itemsets.

Un itemset fermé fréquent, I, est un itemset fréquent tel qu'il n'existe pas d'autre itemset fréquent tel que I soit inclus dedans et que leurs supports soient égaux. Par exemple, si nous considérons les itemsets avec les supports suivants : $support(\{A, B\}) = 3$, $support(\{A, B, C\}) = 3$, $support(\{A\}) = 5$. Les itemsets $\{A, B, C\}$ et $\{A\}$ sont fermés mais pas $\{A, B\}$ qui est inlus dans $\{A, B, C\}$ qui a le même support.

1.3 Comparaison avec Weka

Comparez votre implémentation avec l'outil Weka, notamment en terme de temps d'exécution et de taille de jeux de données traités.

2 Données

Vous trouverez deux jeux de données sur lesquels appliquer votre algorithme sur le Moodle. Vous devrez commenter les résultats obtenus avec votre algorithme sur ces jeux de données.

Documents à rendre

Vous devez rendre les documents suivants :

- code source,
- tests,
- compte rendu (entre 10 et 15 pages) en pdf non compressé.

En ce qui concerne le compte rendu, il doit contenir :
- le qui (vous) ;
- le quoi (quel problème attaquez vous, pourquoi, dans quel but) ;
- le comment (quels sont vos choix et pourquoi, quels ont été les problèmes rencontrés et quelles sont les solutions trouvées) ;
- les réponses aux éventuelles questions qui sont posées ;
- la conclusion.

Vous devez présenter lors d'une soutenance de 15 min (+ 10 min de questions) les résultats obtenus.