Université de Bouira
Département d'Informatique
Module : Fouille de Données
Niveau M1-ISIL

Série TP 2

Motifs fréquents et règles d'association.

I- Travail 1 – Exercice 3 TD

Dans un supermarché, on dispose de la base de transactions suivante :

TID	Items	TID	Items
T1	Lait, Jus, Fromage	T6	Lait, Fromage, Pain, Beurre
T2	Pain, Beurre, Lait	T7	Pain, Beurre, Fromage
T3	Lait, Fromage, Sucre	T8	Jus, Fromage
T4	Pain, Beurre, Sucre	T9	Lait, Fromage, Pain, Beurre
T5	Jus, Sucre, Fromage	T10	Jus, Sucre

1. Transformez cette base en un fichier arff (qui aura la forme d'une table formelle) en utilisant un éditeur de texte de votre choix (ex. Sublime Text). A noter que les attributs sont les items et leurs valeurs possibles sont yes (pour 1) ou no (pour 0).

2. Chargez ce fichier dans l'environnement Weka.
3. Sélectionnez l'algorithme *Apriori* dans l'onglet *Associate* et identifiez ses paramètres.
4. Appliquez l'algorithme Apriori avec un support minimum de 20% et une confiance minimale de 75%.
5. Vérifiez les motifs fréquents obtenu et comparez-les avec ceux obtenus en TD.
6. Donnez les 3 meilleures règles solides obtenues.
7. Modifiez les paramètres afin d'obtenir 10 règles d'association. Sont-elles solides ?
8. Réessayer avec un nouveau jeu de paramètres.

II- Travail 2

Le fichier bank-data.csv contient des données extraites d'un recensement de la population américaine. Le but de ces données est initialement de prédire si quelqu'un gagne plus de 50.000 dollars par an. On va d'abord transformer un peu les données.

1. Chargez ce fichier dans l'environnement Weka. Remarquez que Weka accepte le chargement de fichiers de type autre qu'arff.

Les données comportent souvent des attributs inutiles : numéro de dossier, nom, date de saisie, etc. Il est possible de les supprimer avec Weka en utilisant les filtres.

2. Ici, l'attribut id est une quantité qu'on peut ignorer pour la fouille : supprimez-le avec le filtre *RemoveByName*.

3. Pouvez-vous lancer l'algorithme Apriori ? Pourquoi ?
4. Quels sont les attributs numériques dans ce dataset.
5. Quelles sont les valeurs que peut prendre l'attribut *children* ?
6. Discrétiser les attributs *age*, *income*, et *children* en utilisant le filtre Weka *Discretize* et en forçant le nombre d'intervalles (bins) à 3.
7. Augmentez la lisibilité des valeurs en remplaçant les noms des intervalles générés. (Avec Sublime Text : Sélectionnez la valeur => Find => Replace => Entrez la nouvelle valeur dans le champ Replace With => Replace All.)
8. Sauvegardez le fichier transformé par exemple dans bank-data-transformed.arff.
9. Appliquez l'algorithme Apriori avec les paramètres suivants :
   - ✓ Delta : 0.05
   - ✓ lowerBoundMinSupport : 0.3
   - ✓ metricType : Confidence
   - ✓ minMetric : 0.75
   - ✓ numRules : 20
10. Combien de règles d'association avez-vous obtenu au final ? Pourquoi ?
11. Quel(s) paramètre(s) devez-vous modifier et comment afin d'avoir les 20 règles d'association ?

III- Travail 3 – Utilisation de Weka en tant que java lib

L'objectif de cette partie est de vous familiarisez à l'utilisation de Weka en tant que librairie Java permettant de récupérer les résultats dans un projet Java.

Remarque préliminaire : Une javadoc est fournie avec Weka (dans le dossier *doc*).

1. Commencez par créer un nouveau projet java sous Eclipse.

2. Ajoutez dans le *Java Build Path → librairie* du projet le fichier *weka.jar* (à trouver dans le dossier d'installation de Weka) : Clic droit sur le projet => Properties => Java Build Path => Libraries => Add External JARs => C:/…/weka.jar => Ouvrir => Ok.

3. Créez un nouveau package. Créez une nouvelle classe dans ce package avec une méthode *main.*

4. Modifier la classe comme suit :