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

Série TP 1

Prise en main du logiciel Weka et extraction de motifs fréquents.

I- Présentation et installation de Weka

Weka (Waikato Environment for Knowledge Analysis) est un ensemble de classes et d'algorithmes en Java implémentant les principaux algorithmes de data mining. Il est disponible gratuitement et en open source à l'adresse www.cs.waikato.ac.nz/ml/weka, dans des versions pour Unix et Windows. Ce logiciel est développé en parallèle avec un livre : Data Mining par I. Witten et E. Frank (éditions Morgan Kaufmann).

Weka peut s'utiliser de plusieurs façons :

- Via une interface graphique GUI: permettant de charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité. C'est la méthode utilisée dans ce TP.
- Sur la ligne de commande.
- Par l'utilisation des classes fournies à l'intérieur de programmes Java : toutes les classes sont documentées dans les règles de l'art. Nous la verrons dans un prochain TP.

II- Premiers pas avec Weka

1. Téléchargez Weka et installez-le.
2. Lancez le logiciel Weka et identifiez ses composants.
3. Vous obtenez la fenêtre intitulée Weka GUI Chooser : choisissez l'Explorer.
4. Identifiez les six onglets du Weka Knowledge Explorer, qui sont :
   - *Preprocess* : pour choisir un fichier, inspecter et préparer les données.
   - *Classify* : pour choisir, appliquer et tester différents algorithmes de classification : là, il s'agit d'algorithmes de classification supervisée.
   - *Cluster* : pour choisir, appliquer et tester les algorithmes de segmentation.
   - ***Associate*** : pour appliquer l'algorithme de génération de règles d'association.
   - *Select Attributes* : pour choisir les attributs les plus prometteurs.
   - *Visualize* : pour afficher (en deux dimensions) certains attributs en fonctions d'autres.

III- Format de données dans Weka

Concepts de base

- Un tableau des données ou une collection d'exemples (*dataset*).
- Chaque ligne de ce tableau représente une observation qui est décrite par un vecteur (*instance*).
- Chaque colonne représente une variable (*attribute*) qui peut être quantitative (numeric), qualitative (nominal) ou textuelle (string).

WEKA utilise (entre autres) le format de fichier *ARFF* - Attribute Relation File Format - pour enregistrer les données. Un fichier *arff* est composé d'une liste d'exemples définis par leurs valeurs d'attributs.

Un fichier *arff* comprend toujours trois types d'informations : un nom pour la base de données, des attributs, et des données. Exemple :

```
@RELATION iris
@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
```

Entête

Données

La chaîne de caractères *@RELATION* permet de donner un nom à la base de données. Par exemple, dans le cas du fichier *iris.arff*, le nom donné est *iris*.

La chaîne de caractères *@ATTRIBUTE* permet de définir un attribut. Un attribut peut être de 4 types :

- réel (NUMERIC ou REAL)
- Nominal ({valeurs-possible}) : par exemple :
  @ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica} signifie que l'attribut *class* peut avoir comme valeur soit *Iris-setosa*, soit *Iris-versicolor* ou soit *Iris-virginica*.
- Chaîne de caractère (STRING)
- Date (date [<date-format>])

1. Des exemples de datasets sont disponibles une fois Weka installé. Ces exemples se trouvent dans le répertoire *data* d'installation de Weka. Copiez ce répertoire *data* et collez-le dans un autre emplacement (le D:/ par exemple).
2. Ouvrez dans un éditeur (Sublime Text par exemple) un de ces fichiers d'exemples et vérifiez son format.
3. A noter que Weka propose un éditeur de fichier *arff* (*tools→arffViewer*) permettant de visualiser les fichiers *arff* sous la forme d'un tableau, et éventuellement de les modifier.
4. A noter que d'autres formats peuvent être importés dans Weka : CSV, BDD SQL, etc.

IV- Analyse des données avec Weka Explorer

1. Depuis l'*Explorer* Weka, cliquez sur le bouton *Open file* de l'onglet *Preprocess*. Choisissez le fichier de données *iris.arff et* ouvrez-le. Un certain nombre d'informations vont alors apparaître dans *Weka Explorer*.
2. Combien y a-t-il d'instances dans le dataset ?
3. À quoi correspond la zone *Selected attribute* à droite juste sous le bouton *Apply* ?

4. Vérifiez ce qui se produit lorsque vous cliquez sur les différents attributs. À quoi correspondent les valeurs *Name*, *Type*, *Missing*, *Distinct*, et *Unique* ?
5. Quels sont les attributs servant à décrire les instances ? Pour chacun des attributs, quel est son type et ses valeurs possibles ?

V- Visualisation des données avec Weka Explorer

La visualisation de données peut permettre de se faire une idée de l'organisation de celles-ci.

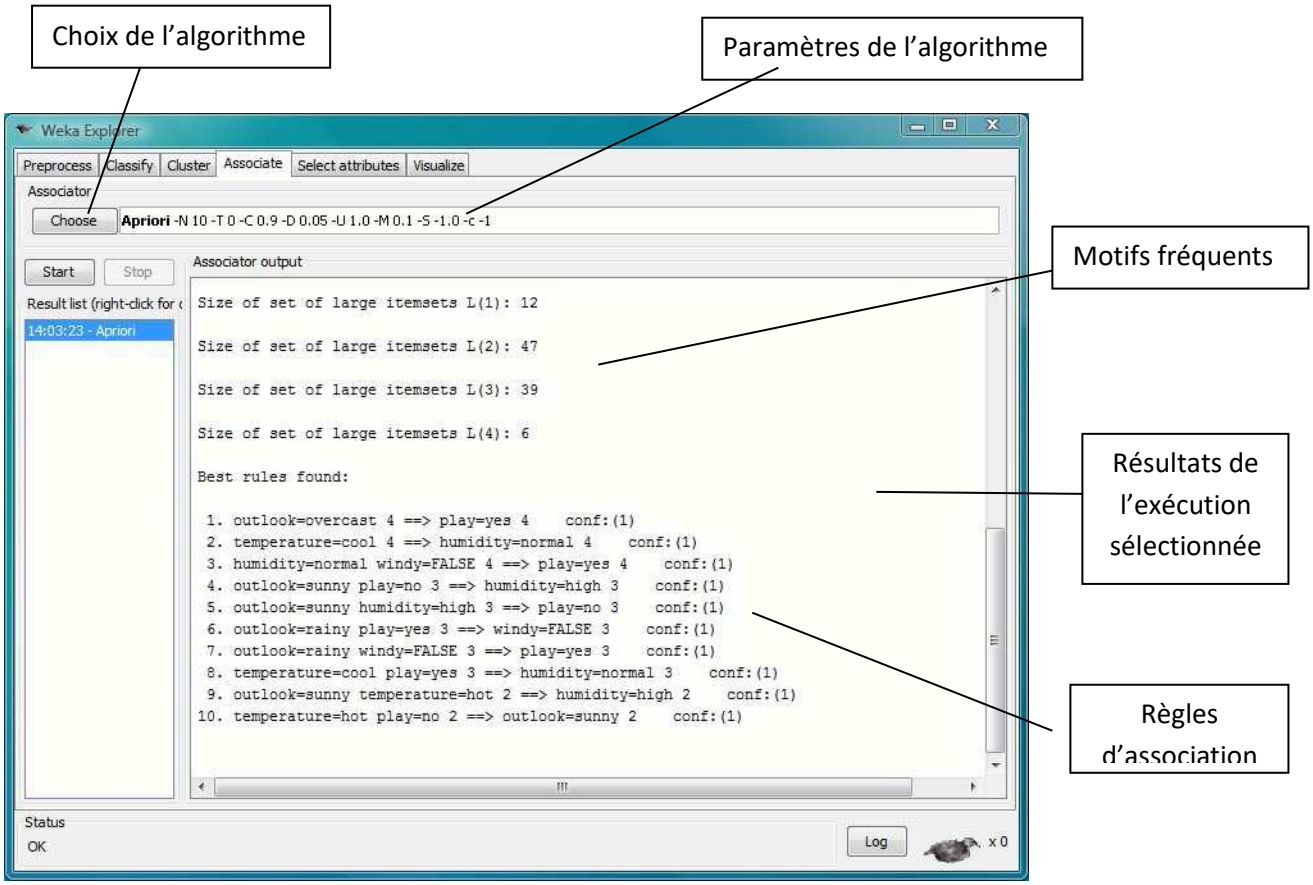
1. Passez dans l'onglet *Visualize*.

Vous y voyez un ensemble de 25 graphiques (que vous pouvez ouvrir en cliquant dessus), qui représentent chacun une vue sur l'ensemble d'exemples selon deux dimensions possibles, la couleur des points étant leur classe. Sur le graphique, chaque point représente un exemple : on peut obtenir le descriptif de cet exemple en cliquant dessus. La couleur d'un point correspond à sa classe (détaillé dans la sous-fenêtre Class colour).

2. Changez les axes pour mettre la longueur des sépales (sepallength) en abscisse (X), et la classe des iris (class) en ordonnées (Y). Quelle différence constatez-vous entre les iris-setosa et les iris-virginica ?
3. Que constatez- vous sur l'interaction entre sepallength (X) et petallength (Y) ?
4. Qu'elle est l'utilité du Jitter ?

VI- Premier exemple sur les règles d'association

L'extraction de motifs fréquents et de règles d'association est accessible par l'onglet ***Associate***. Les algorithmes implantés sont *Apriori*, HotSpot, predictiveApriori et Tertius.



1. Depuis l'onglet *Preprocess*, chargez le dataset weather.nominal.arff dans Weka.
2. Depuis l'onglet *Associate*, cliquez sur *Choose* et choisissez l'algorithme *Apriori*.
3. Exécutez-le, sans modifier ses paramètres par défaut, en cliquant sur *Start*.
4. Quelles sont les informations retournées par l'algorithme ?
5. Identifiez dans le résultat les trois règles les plus fortes (confiance et support maximaux) permettant de prédire que l'on va jouer au tennis, c'est à dire les règles contenant play=yes dans la partie droite.

1.	=> play=yes conf : ( )
2.	=> play=yes conf : ( )
3.	=> play=yes conf : ( )

Algorithme Apriori - Modification des paramètres

Le fonctionnement de l'algorithme Apriori dans Weka diffère légèrement de celui vu en cours. D'abord, l'utilisateur définit le nombre de règles *numRules* qu'il souhaite obtenir et la valeur de départ du seuil minsupport *upperBoundMinSupport*. Ensuite, cette implémentation recherche les règles en diminuant successivement minsupport avec un pas *delta* défini jusqu'à ce que :

- ✓ soit le nombre de règles demandé est atteint ;
- ✓ soit la borne inférieure définie pour minsupport *lowerBoundMinSupport* est atteinte.

En cliquant du bouton droit dans le champ à coté du bouton Choose, on a accès aux paramètres de l'algorithme. Le bouton More détaille chacune de ces options :

delta: fait décroître le support minimal de ce facteur, jusqu'à ce que soit le nombre de règles demandées a été trouvé, soit on a atteint la valeur minimale du support lowerBoundMinSupport.
lowerBoundMinSupport: valeur minimale du support (minsup en cours). Le support part d'une valeur initiale, et décroît conformément à delta.
metricType : la mesure qui permet de mesurer la précision des règles. (ex. confiance)
minMetric : la valeur minimale de la mesure en dessous de laquelle on ne recherchera plus de règle.
numRules : Le nombre de règles que l'algorithme doit produire.
removeAllMissingCols : enlève les colonnes dont toutes le valeurs sont manquantes.
significanceLevel : test statistique
upperBoundMinSupport : valeur initiale du support.
outputItemSets : pour afficher ou non les motifs fréquents.

1. Sur le fichier weather.nominal.arff, jouer sur les différents paramètres puis comparer les règles produites selon les mesures choisies. Comment se comporte le temps d'exécution en fonction des paramètres ?
2. Mettez *outputItemSets* à *true*. Identifiez dans le résultat quatre 2-itemsets fréquents.

VII- Un deuxième exemple sur les règles d'association

1. Chargez le jeu de données weather.numeric.arff dans Weka. Quelles sont les différences avec le précédent dataset weather.nominal.arff ?

2. Essayez d'exécuter l'algorithme Apriori avec ses paramètres par défaut. Que constatez-vous ? Pourquoi ?

Application impossible de ces algorithmes car le jeu contient des valeurs numériques.

Certains algorithmes ont besoin d'attributs discrets pour fonctionner, d'autres n'acceptent que des attributs continus (réseaux de neurones, plus proches voisins). D'autres encore acceptent indifféremment des attributs des deux types. Weka dispose de filtres pour discrétiser des valeurs continues.

Par conséquent, les attributs numériques Temperature et Humidity doivent être discrétisés afin d'appliquer l'extraction de règles d'association.

Transformation des données – Les Filtres

Weka met à votre disposition des **filtres** permettant soit de choisir de garder (ou d'écarter) certaines **instances**, soit de modifier, supprimer, discrétiser, ajouter des **attributs**. Le cadre *Filters* dans l'onglet *Preprocess* vous permet de manipuler les filtres. Le fonctionnement général est toujours le même :

- Vous choisissez un ensemble de filtres, chaque filtre, avec ces options, étant choisi dans le menu déroulant en cliquant sur *Choose*.
- On applique les filtres avec le bouton *Apply*.
- Le bouton *Save* sauvegarde ces données transformées dans un fichier.
- Le bouton *Undo* permet de revenir en arrière afin d'annuler les effets du filtre appliqué.

Discrétisation

Weka dispose de nombreux filtres pour discrétiser des valeurs continues.

Le filtre *Discretize* permet de rendre discret un attribut continu et ceci de plusieurs façons :

- ✓ En partageant l'intervalle des valeurs possibles de l'attribut en intervalles de taille égale.
- ✓ En le partageant en intervalles contenant le même nombre d'éléments.
- ✓ En fixant manuellement le nombre d'intervalles (bins).
- ✓ En laissant le programme trouver le nombre idéal de sous intervalles.

Le filtre *PKIDiscretize* transforme les attributs numériques en attributs qualitatifs. Pour chaque attribut créé, les modalités correspondent à des intervalles de valeurs de même fréquence (même nombre d'instances pour chaque modalité).

3. Allez sur l'onglet *Preprocess* et définissez un filtre *PKIDiscretize* pour discrétiser les attributs Temperature et Humidity :
   - Filter => Choose => weka => filters => unsupervised => attribute => PKIDiscretize => close.
   - Sélectionnez l'attribut temperature => Apply. Refaire avec l'attribut humidity.

4. Visualisez les valeurs de Temperature qui doivent être discrétisées selon les modalités suivantes :
   - ✓ '(inf-70.5]' : valeurs inférieures ou égales à 70,5.
   - ✓ '(70.5–77.5]' : valeurs entre 70,5 et 77,5 incluse.

- ✓ ‘(77.5–inf[’ : valeurs supérieures à 77,5.

Et les valeurs de Humidity doivent être discrétisées selon les modalités suivantes :

- ✓ ‘(inf-77.5]’ : valeurs inférieures ou égales à 77,5.
- ✓ ‘(77.5-88]’ : valeurs entre 77,5 et 88 incluse.
- ✓ ‘(88-inf]’ : valeurs supérieures à 88.

5. Sauvegardez le résultat de transformation dans un fichier "weather.nominal.dicretized.arff".
6. Pouvez-vous lancer l’algorithme Apriori après cette discrétisation ?

7. Afin d'augmenter la lisibilité des valeurs, ouvrez ce fichier dans un éditeur de texte et remplacez les noms des intervalles générés comme indiqué dans le tableau ci-dessous. (Avec Sublime Text : Sélectionnez la valeur => Find => Replace => Entrez la nouvelle valeur dans le champ Replace With => Replace All.)

Temperature		Humidity	
Valeur	Remplacée par	Valeur	Remplacée par
‘\‘(inf-70.5]\’’	cool	‘\‘(inf-77.5]\’’	low
‘\‘(70.5–77.5]\’’	temperate	‘\‘(77.5-88]\’’	medium
‘\‘(77.5–inf[\’’	hot	‘\‘(88-inf]\’’	high

8. Sauvegardez ensuite le fichier et chargez-le dans Weka pour vérifier.