Fouille de Données - TP1 : Weka et règles d'association

M2- 2011-2012

Cette séance pour but de prendre en main **weka***, une plateforme d'algorithmes de data mining écrite en java que nous réutiliserons ainsi que d'expérimenter l'algorithme APriori de génération de règles d'association. Vous pouvez vous mettre en (bi|tri)nôme. Un compte-rendu est demandé à l'issu de la séance (17/11, 12 :00) sous la forme d'un document .pdf (.doc refusé). Vous enverrez ensuite un compte-rendu final (toujours sous format .pdf) ainsi que les sources Java avant le 23/11, 23 :59. Les comptes-rendus doivent être envoyé à marc.plantevit@univ-lyon.fr avec comme objet "[M2TIW] CR TP1". N'oubliez d'indiquer les noms des membres du (mo|bi|tri)nôme dans le corps du message et le compte-rendu.*

1 Présentation et installation de Weka

Weka est un ensemble de classes et d'algorithmes en Java implémentant les principaux algorithmes de data mining. Il est disponible gratuitement 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 :

- Par l'intermédiaire d'une interface utilisateur : 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 y reviendrons sans doute dans un prochain TP.

1. Téléchargez Weka et installez le.
2. Téléchargez l'excellente présentation d'Eibe Frank à `http://liris.cnrs.fr/marc.plantevit/ENS/TP/weka.ppt` et parcourez là (un tutorial sur Weka est également disponible : `http://liris.cnrs.fr/marc.plantevit/ENS/TP/Tutorial.pdf`).

2 Premiers pas

Weka est maintenant installé sur votre compte. Après l'avoir lancé, vous obtenez la fenêtre intitulée Weka GUI Chooser : choisissez l'Explorer. La nouvelle fenêtre qui s'ouvre alors (Weka Knowledge Explorer) présente six onglets :

Preprocess : pour choisir un fichier, inspecter et préparer les données.

Classify : pour choisir, appliquer et tester différents algorithmes de classiffcation : là, il s'agit d'algorithmes de classiffication supervisée qui fera l'objet d'un prochain TP.

Cluster : pour choisir, appliquer et tester les algorithmes de segmentation : nous y consacrerons également un TP.

Associate : pour appliquer l'algorithme de génération de règles d'association. Nous allons l'étudier dans ce TP.

Select Attributes : pour choisir les attributs les plus prometteurs.

Visualize : pour afficher (en deux dimensions) certains attributs en fonctions d'autres.

3 Les données

Les données sont sous un format ARFF -pour Attribute-Relation File Format-. Des exemples de données sont disponibles une fois weka installé [1]. Ouvrez dans un éditeur un de ces fichiers d'exemples et regardez son format. Il est simple et il est facile de convertir des données-par exemple issues d'un tableur- en ARFF. (Il y a même un convertisseur inclus dans Weka du format csv vers le format arff).

Dans l'onglet Preprocess, cliquez sur Open File et ouvrez par exemple le fichier iris.arff : il contient la description de 150 specimens d'iris de trois sortes différentes. Chaque description est composée de quatre attributs numériques

1. Au cas où, `http://liris.cnrs.fr/marc.plantevit/ENS/TP/data/`

(dimensions des sépales et des pétales), et d'un cinquième attribut qui est la classe de cet exemple (i.e. la sorte d'iris à laquelle il appartient). Pour chacun des attributs, vous pouvez obtenir, en cliquant dessus dans la sous-fenêtre Attributes, des statistiques basiques sur la répartition des valeurs pour cet attribut (sous-fenêtre Selected Attribute). On peut appliquer différents filtres aux données ; nous y reviendrons tout à l'heure.

4 Visualisation des données

Pour une première approche des données, passez dans la fenêtre 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). Au départ, le graphique n'est pas très utile, car les axes représentent le numéro de l'exemple.

1. Changez les axes pour mettre la largeur des pétales en abscisse, et la longueur des sépales en ordonnées.
2. Proposez un ensemble de deux règles simples permettant de classer les exemples selon leur genre : quelle erreur commettrez-vous ? Les petits rectangles sur la droite de la fenêtre représentent la distribution des exemples, pour l'attribut correspondant, par rapport à l'attribut (ou la classe) codé par la couleur. En cliquant du bouton gauche sur un de ces rectangles, vous le choisissez comme axe des X, le bouton droit le met sur l'axe des Y.
3. En mettant la classe sur l'axe des X, quels sont à votre avis les attributs qui, pris seuls, permettent le mieux de discriminer les exemples ? Si les points sont trop serrés, le potentiomètre Jitter, qui affiche les points "à peu près" à leur place, vous permet de les visualiser un peu plus séparément : cela peut être utile si beaucoup de points se retrouvent au même endroit du plan.

5 Un premier exemple de règle d'association

1. Lancez Weka, puis l'Explorer. Choisissez le fichier weather.nominal.arff : c'est l'exemple standard du golf (ou du tennis. . .), où tous les attributs ont été discrétisés. Les algorithmes de recherche de règles d'association se trouvent sous l'onglet Associate.
2. Choississez l'algorithme **Apriori**.
3. Vérifiez que tout fonctionne en lançant l'algorithme sans modifier les paramètres du programme.
4. Quelles sont les informations retournées par l'algorithme ?

5.1 Modification des paramètres

En cliquant du bouton droit dans la fenêtre en face 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 classer les règles. Supposons que L désigne la partie gauche de la règle et R la partie droite. Il y en a quatre (L désigne la partie gauche de la règle et R la partie droite) :
- Confidence : la confiance.
- Lift : l'amélioration.
- leverage : proportion d'exemples concernés par les parties gauche et droite de la règle, en plus de ce qui seraient couverts, si les deux parties de la règles étaient indépendantes : $Freq(L \wedge R) - Freq(R)Freq(L)$
- Conviction : similaire à l'amélioration, mais on s'intéresse aux exemples où la partie droite de la règle n'est pas respectée. Le rapport est inversé : $\frac{Freq(L)Freq(\neg R)}{Freq(L \wedge \neg R)}$

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.

1. Sur le fichier weather.nominal.arff, comparer les règles produites selon la mesure choisie.
2. Pour une règle que vous choisirez, vérifiez les calculs de confiance, amélioration, leverage et conviction.
3. Pouvez-vous décrire d'une phrase ce que signifient les notions de leverage et de conviction ?

6 Un deuxième exemple de règles d'association

Le fichier bank-data.cvs 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 :

6.1 Transformation des données

Récupérer le fichier bank-data.csv [2] Revenez à la fenêtre Preprocess.

1. Tout d'abord ouvrez le fichier bank-data.csv : il vous sera proposé d'utiliser un convertisseur : dites-oui ! Weka met à votre disposition des filtres permettant soit de choisir de garder (ou d'écarter) certains exemples, soit de modifier, supprimer, ajouter des attributs. La sous-fenêtre Filters 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 du haut de la sous-fenêtre, puis ajouté à la liste des filtres par la commande Add.
   - On applique les filtres avec la commande Apply Filters.
   - On peut alors remplacer le fichier précedemment chargé par les données transformées, à l'aide du bouton Replace.
   - Ce fichier devient alors le fichier de travail.
   - Le bouton Save sauvegarde ces données transformées dans un fichier.
2. Pouvez vous lancer l'algorithme Aprioir ? Pourquoi ?

6.2 Sélection des attributs

Les données comportent souvent des attributs inutiles : numéro de dossier, nom,date de saisie . . . . Il est possible d'en supprimer 'à la main', à condition de connaître le domaine. On peut aussi lancer un algorithme de data mining, et regarder les attributs qui ont été utilisés : soient ceux-ci sont pertinents, et il est important de les garder, soient ils sont tellement liés à la classe qu'à eux seuls ils emportent la décision (pensez à un attribut qui serait la copie de la classe). Weka a automatisé cette recherche des attributs pertinents dans le filtre AttributeSelectionFilter, qui permet de définir les attributs les plus pertinents selon plusieurs méthodes de recherche (search), en utilisant plusieurs mesures possibles de la pertinence d'un attribut (eval).

1. Ici l'attribut id est une quantité qu'on peut ignorer pour la fouille : supprimez le !

6.3 Discrétisation

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. Le filtre DiscretizeFilter 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.

Ici il y a plusieurs attributs numériques : "children", "income", "age".

1. Discrétiser age et income en utilisant le filtre Weka et en forçant le nombre d'intervalles à 3. Sauver le fichier transformé par exemple dans bank1.arff.
2. L'attibut children est numérique mais ne prend que 4 valeurs : 0,1,2,3 ; pour le discrétiser, on peut soit utiliser le filtre, soit le faire à la main dans le fichier arff.

Remarque : si vous éditez directement le fichier, vous pouvez en profiter pour rendre les données plus lisibles, par exemple en traduisant le nom des attributs, en donnant des noms aux intervalles obtenus par la discrétisation...

2. http ://archive.ics.uci.edu/ml/

6.4 APriori

Sauvez dans bankd.arff le résultat de vos transformations : c'est le fichier qui va servir pour la génération des règles d'association.

1. Appliquez l'algorithme Apriori et tentez d'interpréter les règles produites. Jouez sur les paramètres. Comment se comporte le temps d'exécution en fonction des paramètres ? Quels sont les paramètres les plus "critiques" ?
2. Utilisez l'algorithme Tertius. Que constatez-vous sur la forme des règles ?

7 Promotions de Noël et épicerie de nuit

L'épicerie de nuit de la rue *"remplacez par votre rue favorite"* a décidé à l'approche des fêtes de fin d'année de lancer une vaste opération de promotion. Son patron, fervent adepte des nouvelles technologies et de la fouille de données (ça arrive), vous demande d'utiliser les règles d'associations pour trouver des règles intéressantes pour ses futures promotions. Il va donc réutiliser le bilan d'achats de l'année dernière à la même date :

Achats	Produit 1	Produit 2	Produit 3	Produit 4	Produit 5
Mme Michou	X			X	X
Tonton Gérard	X	X			X
Mme Guénolet					X
Mr Robert			X	X	X
Mr Sar	X	X	X	X	X
Mr causy	X				X
Mme mimi	X			X	X
Mme Fillon		X	X		

TABLE 1 – Table d'achats de l'année 2006-2007

1. Générer un fichier ARFF contenant les données du bilan d'achat
2. Extraire les règles d'associations avec un support de 0.5 puis de 0.1
3. Que pouvez-vous conseiller comme promotion au patron ?

8 Mise en œuvre

Rendez vous sur le site `http://archive.ics.uci.edu/ml/`, choisissez un jeu de données et importer le dans weka afin de le visualiser et extraire des règles d'association.

9 API Weka

Il est possible d'utiliser directement les algorithmes sur des jeux de données en les appelant directement à partir de votre propre code Java.

1. Etudiez l'API de Weka, notamment pour les règles d'Association, puis dans un programme Java, automatisez directement ce que vous avez fait via l'interface graphique en appelant directement les algorithmes nécessaires.
2. Utilisez le code précédent pour étudier le temps d'exécution de l'algorithme Apriori en fonction du seuil de support et du seuil de confiance (vous pouvez générer des graphes à l'aide de gnuplot).