TP3 - Fouille de Données
Règles d'Association - suite

13 février 2017
Lina Soualmia

1 Préambule

Les réponses aux questions sont à envoyer à fdd.m1.rouen@gmail.com avec comme objet : [FdD] TP3 Gr (2, 3 ou 4) NOM sous la forme d'un fichier (.pdf, .txt, ...). Le fichier `bankd.arff` est également à envoyer. N'oubliez pas d'indiquer le numéro de la question et répondez-y de manière claire. Les fichiers sont à envoyer ce jour au plus tard.

Environnement de travail :

Système WEKA

Vos fichiers `weather.numeric.arff` du TP1 et `weather.discretized.arff` du TP2 pour les sections 2 à 4.

Le fichier `bank-data.cvs` disponible dans les documents partagés de votre liste de diffusion pour la section 5.

2 Algorithme `Tertius`

Les règles d'association extraites avec `Tertius` sont différentes car la partie droite est une disjonction de valeurs au lieu d'une conjonction (avec `Apriori`).

2.1 Question 1

Réalisez une extraction sur votre fichier `weather.nominal.arff` avec l'algorithme `Tertius` en fixant le paramètre `hornClauses` à la valeur `true`.
Identifiez dans le résultat les 3 règles les plus fortes permettant de prédire que l'on va jouer au tennis. Au besoin vous augmenterez le nombre de règles extraites en diminuant la valeur du paramètre `noiseThreshold`.

2.2 Question 2

Identifiez dans le résultat les 3 règles les plus fortes permettant de prédire que l'on ne va pas jouer.

2.3 Question 3

Comparez les règles des questions 1 et 2 avec celles générées par l'algorithme de classification supervisée `Prism` (§TP2 Question 13) et notez vos observations.

3 Algorithme `PredictiveApriori`

Cet algorithme permet de générer des règles de classification par la méthode `Apriori`. Pour cela, vous devez fixer le paramètre `car` à la valeur `true` et indiquer dans le paramètre `classIndex` l'indice de l'attribut de classe (par exemple l'attribut `play`). Laissez la valeur -1 si l'attribut de classe est le dernier. Les règles extraites contiennent alors uniquement une valeur de l'attribut de classe dans la partie droite. La mesure de précision de chaque règle est l'`accuracy` dont la valeur augmente avec la précision de la règle. Le support de l'antécédent et le support de la règle sont également indiqués comme pour l'algorithme `Apriori`.

3.1 Question 4

Appliquez l'algorithme `PredictiveApriori` en fixant le paramètre `car` à la valeur `true` sur le fichier `weather.nominal.arff`. Identifiez dans le résultat les 3 règles les plus précises (`accuracy` maximale) permettant de prédire que l'on va jouer au tennis.

3.2 Question 5

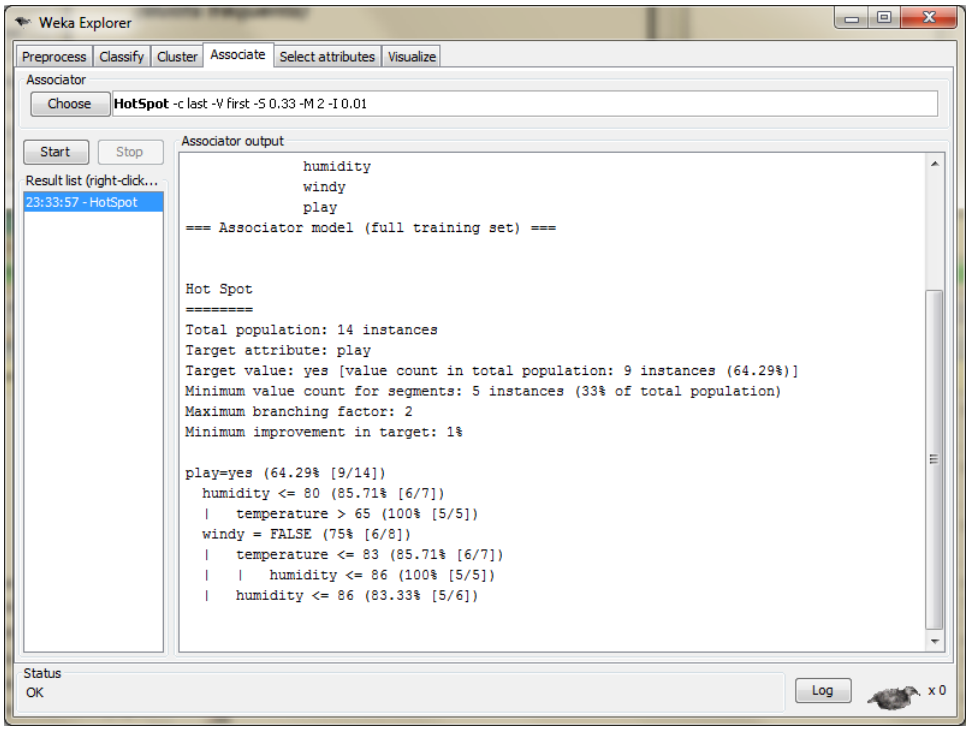
Identifiez dans le résultat les 3 règles les plus précises (`accuracy` maximale) permettant de prédire que l'on ne va pas jouer au tennis.

3.3 Question 6

Comparez les règles des questions 4 et 5 avec celles générées par l'algorithme de classification supervisée `Prism` (§TP2 Question 13) et notez vos observations.

4 Algorithme `HotSpot`

Contrairement aux algorithmes précédents, `HotSpot` est capable de traiter les valeurs numériques. Pour chaque attribut numérique `HotSpot` réalise plusieurs discrétisations et conserve celle permettant d'obtenir les règles extraites les plus précises, c-à-d de confiances maximales. Cet algorithme permet également de générer un ensemble de règles de classification pour une valeur d'un attribut de classe. L'attribut de classe est défini par le paramètre `target` correspondant à l'indice de cet attribut (parmi tous les attributs). La valeur prédite pour cet attribut de classe est définie par le paramètre `targetIndex` correspondant à l'indice de cette valeur (parmi toutes celles possibles).
Le résultat de `HotSpot` est présenté sous forme d'un arbre de décision dont la racine est la valeur prédite pour l'attribut de classe.



Dans l'arbre ci-dessus la valeur prédite est `play=yes` ; elle est représentée par la racine de l'arbre. Chaque nœud de l'arbre correspond à une règle dont l'antécédent est la combinaison des valeurs des attributs en remontant jusqu'à la racine. Par exemple, le nœud en dessous de la racine (`humidity`<=80) définit la règle : `humidity`<=80 →`play=yes` (confiance=85.71% et support =6/7).

4.1 Question 7

Appliquez l'algorithme `HotSpot` sur le fichier non discrétisé `weather.numeric.arff` en mettant la valeur `last` (indice de l'attribut `play` qui est le dernier attribut) pour le paramètre `target` et la valeur `first` (la première valeur de l'attribut `play` correspondant à `yes`) pour le paramètre `targetIndex`. (Si `HotSpot` n'est pas inclus dans votre version de Weka, utilisez Weka 3.6.2 disponible dans les documents partagés de votre liste de diffusion). Indiquez toutes les règles décrites par l'arbre généré en précisant pour chacune sa valeur de confiance et de support.

4.2 Question 8

Appliquez l'algorithme `HotSpot` sur le fichier `weather.discretized.arff` en mettant la valeur `last` (indice de l'attribut `play` qui est le dernier attribut) pour le paramètre `target` et la valeur `last` pour le paramètre `targetIndex` (la seconde valeur de l'attribut `play` correspondant à `no`). Indiquez toutes les règles décrites par l'arbre généré en précisant pour chacune sa valeur de confiance et de support.

5 Associations données bancaires

Le fichier `bank-data.cvs` contient des données extraites d'un recensement de la population américaine. (essayez de comprendre le contenu du fichier avant de commencer le fouille ...)

5.1 Question 9 (transformation des données)

Récupérez le fichier `bank-data.cvs`. Revenez à la fenêtre `Preprocess` et ouvrez le fichier (il vous sera proposé d'utiliser un convertisseur). Weka met à votre disposition des filtres permettant soit de choisir de garder (ou d'écarter) certains exemples, soit de modifier, de supprimer ou d'ajouter des attributs. La sous-fenêtre `Filters` vous permet de manipuler ces filtres. Le fonctionnement général est toujours le même :

- Vous choisissez un ensemble de filtres, chaque filtre, avec ses options, étant choisi dans le menu déroulant du haut de la sous-fenêtre, puis ajouté à la liste des filtres par la commande `Add`.
- Les filtres sont appliqués avec la commande `ApplyFilters`.
- Le fichier précédemment chargé peut être remplacé par celui qui contient les données transformées à l'aide de la fonction `Replace` (le fichier devient alors le fichier de travail) et la fonction `Save` sauvegarde ces données transformées dans un fichier.

Pouvez-vous lancer l'algorithme Apriori ? Pourquoi ?

5.2 Question 10 (sélection des attributs)

Les données comportent souvent des attributs inutiles : numéro de dossier, nom, date de saisie ...etc. Il est possible d'en supprimer à la main, à condition de connaître le domaine. On peut également lancer un algorithme de data mining et regarder les attributs qui ont été utilisés : soit ceux-ci sont pertinents, et il est important de les garder, soit 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`).
Dans le fichier `bank-data.cvs` l'attribut `id` est une information que l'on peut ignorer pour la fouille. Supprimez-le.

5.3 Question 11 (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`. Discrétisez `age` et `income` en utilisant le filtre Weka et en forçant le nombre d'intervalles à 3. Sauvegardez le fichier transformé par exemple dans `bank1.arff`.
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... Sauvez dans `bankd.arff` le résultat de vos transformations (et n'oubliez pas de l'envoyer avec votre compte-rendu de TP).

5.4 Question 11

Appliquez l'algorithme `Apriori` et `FP-Growth` sur le fichier `bankd.arff` et tentez d'interpréter et de comparer les règles produites.

5.5 Question 12

Utilisez l'algorithme `Tertius`. Que constatez-vous sur la forme des règles ?