Fouille de Données - TP2 : Clustering

M2- 2011-2012

Cette séance a pour objectif de vous permettre d'analyser des données avec des algorithmes de clustering comme K-means, EM, et CobWeb. Vous n'avez peut être pas vu en détail tous ces algorithmes pendant le cours. Vous pouvez donc consulter leur documentation via Weka ou une requête google. Les ressources nécessaires pour ce TP sont disponibles à l'adresse : `http://liris.cnrs.fr/marc.plantevit/ENS/TP/`.
Un compte rendu par (mo | bi | tri)nome [1], au format pdf, devra être envoyé à la fin du TP à `marc.plantevit@univ-lyon1.fr` avec **[M2Pro] TP2 Clustering** en objet. Une seconde version pourra être envoyée avant le 30/11/2011, 23 :59.

Introduction

Nous allons utiliser le paquetage *weka.clusterers* pour l'analyse de données qui ne contiennent pas d'attribut de classe. Ainsi, puisque presque tous les ensembles de données que nous utilisons possèdent un attribut de classe, nous allons ignorer ces attributs (sauf pour les phases d'évaluation). Nous allons utiliser le **Weka Knowledge Explorer**.

Remarque : Si vous souhaitez voir les commandes spécifiques des algorithmes, vous pouvez utiliser le simple CLI (command line options) : `java weka.clusterers.Cobweb -h` pour le clustering conceptuel. Ou encore : `java weka.clusterers.SimpleKMeans -h` pour le k-means clustering.

1 Premiers contacts

Les classes qui implémentent une méthode de clustering dans l'outil Weka, sont regroupées dans le package weka.clusterers. La classe weka.clusterers.Clusterer définit la structure générale commune à toutes les méthodes de clustering. A partir de l'onglet Cluster on peut observer quatre algorithmes implémentés : **SimpleKMeans**, **EM** (expectation-maximization), **CobWeb** et **FarthestFirst**.

Weka affiche le nombre d'exemples assignés à chaque cluster. Weka permet de tester la qualité du modèle sur un jeu de test. C'est la mesure de vraisemblance (log-likelihood) qui est utilisée. Plus la mesure est grande, mieux le modèle caractérise les données. Le test peut être effectué par validation croisée.

La boîte Cluster mode permet de choisir la méthode d'évaluation du modèle extrait.

Use training set : effectue et teste le clustering sur le même jeu de données ;

Supplied test set : teste le clustering sur un jeu de données à spécifier ;

Percentage split : effectue le clustering sur le pourcentage indiqué du jeu de données et teste sur le pourcentage restant ;

Classes to clusters evaluation : teste le clustering relativement à une classe

Pour cet exercice, on considérera les données du fichier **vote.arff**. Ce jeu de données décrit le résultat des votes de chaque représentant au Congrès des Etats-Unis sur les 9 questions clés identifiés par le Congressional Quarterly Almanac.

1.1 Analyse exploratoire

Effectuez une première analyse du jeu de données. Combien d'instances ? Combien d'attributs ? Quels types ? etc.

1. Veillez à conserver les groupes formés initialement.

1.2 K-Means

Effectuez un clustering du jeu de données en utilisant l'algorithme **SimpleKMeans** et en conservant les paramètres par défaut.

Le résultat du clustering est donné avec une instance par cluster représentant le centroïde du cluster.

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 1510.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                                Cluster#
Attribute                                Full Data          0          1
                                             (435)      (214)      (221)
=============================================================================
handicapped-infants                             n          n          y
water-project-cost-sharing                      y          y          n
adoption-of-the-budget-resolution               y          n          y
physician-fee-freeze                            n          y          n
el-salvador-aid                                 y          y          n
religious-groups-in-schools                     y          y          n
anti-satellite-test-ban                         y          n          y
aid-to-nicaraguan-contras                       y          n          y
mx-missile                                      y          n          y
immigration                                     y          y          y
synfuels-corporation-cutback                    n          n          n
education-spending                              n          y          n
superfund-right-to-sue                          y          y          n
crime                                           y          y          n
duty-free-exports                               n          n          y
export-administration-act-south-africa          y          y          y
Class                                  democrat republican   democrat


Clustered Instances

0      214 ( 49%)
1      221 ( 51%)
```

Visualisation :

A partir de la boîte **Result List** (bouton droit, **Visualize cluster assignement**), visualisez la répartition des exemples dans chaque cluster.

Évaluation relativement à une classe

L'option d'évaluation **Classes to clusters evaluation** permet d'assigner une classe à un cluster pendant la phase de test. La classe assignée est la plus fréquente dans le cluster ; une erreur de classement (taux de mal classés) est calculée ainsi que la matrice de confusion. Dans ce cas, l'algorithme ne prend pas en compte la valeur de cet attribut dans le calcul de distance.

- Effectuez un clustering du jeu de données en utilisant la méthode implémentée par SimpleKMeans en conservant les paramètres par défaut avec l'option Classes to clusters evaluation et l'attribut de classe class
- Recommencez en modifiant l'attribut de classe et observer les taux d'erreur
- Conservez les meilleurs résultats et effacer les autres de la liste des résultats. Quel est l'attribut de classe pour lequel l'erreur est la plus faible ?
- Visualisez : A partir de la boîte Result List (bouton droit, Visualize cluster assignement), visualiser la répartition des exemples dans chaque cluster. Les croix représentent les instances classées dans le "bon" cluster et les carrés représentent les instances classées dans le "mauvais" cluster.

Comparaison de modèles

- Lancez l'algorithme SimpleKMeans plusieurs fois avec les valeurs 20, 50, 100, 1000 pour le paramètre random seed avec l'option **Classes to clusters evaluation** et l'attribut de classe **class** et en fixant le nombre de clusters à 2.
- Quel est le meilleur résultat selon le taux d'erreur et la taille de clusters.
- Conservez le meilleur résultat

1.3 EM

La méthode EM (Expectation Maximisation) génère une description probabiliste des clusters en terme de moyenne et écart-type pour les attributs numériques et en terme de nombre pour les attributs nominaux. Chaque cluster est

décrit par sa probabilité a priori et une distribution de probabilité pour chaque attribut. Pour un attribut nominal, est affiché le nombre d'exemples et pour un attribut numérique est affiché les caractéristiques de sa distribution normale. L'option d'évaluation **Classes to clusters evaluation** affiche aussi le **log-likelihood**, (ou log-vraissemblance) assigne une classe au cluster, calcule l'erreur et la matrice de confusion.

- Effectuez un clustering du jeu de données en utilisant la méthode EM avec les paramètres par défaut. Combien de classes sont découvertes ?

Évaluation relativement à une classe

- Effectuez un clustering du jeu de données en utilisant la méthode EM en fixant le nombre de clusters à 2 et avec l'option **Classes to clusters evaluation** et l'attribut de classe **class**.
- Effectuez un clustering du jeu de données en utilisant la méthode EM en fixant le nombre de clusters à 2 et avec l'option **Classes to clusters evaluation** et l'attribut de classe **el-salvador-aid**.
- Comparez les résultats à ceux obtenus à la question 1.2.

1.4 Clustering hiérarchiques avec Cobweb

La méthode Cobweb réalise un clustering hiérarchique où les clusters sont décrits de manière probabiliste. L'algorithme possède deux options importantes : **Cutoff** (par défaut=0.002) et **Acuity** (par défaut=1.0). ? ?

- Lancez un clustering du jeu de données en utilisant la méthode Cobweb avec les paramètres par défaut et avec l'option Classes to clusters evaluation et l'attribut de classe class La fenêtre d'output montre le nombre de noeuds regroupés puis découpés, le nombre de clusters et la structure hiérarchique de clusters. Quel est le nombre de clusters trouvés ?

Évaluation relativement à une classe

- Fixez le paramètre Cutoff à 0.5 de manière à supprimer le découpage (split) de noeuds et donc ramener le nombre de clusters à 2 et avec l'option Classes to clusters evaluation et l'attribut de classe class
- Conservez le paramètre Cutoff à 0.5 avec l'option Classes to clusters evaluation et l'attribut de classe el-salvador-aid
- Comparez les résultats obtenus avec les trois méthodes de clustering.

2 Jeux de données de base pour comparer les techniques de clustering

2.1 Jeu de données « ligne-carré »

En utilisant le jeu de données à 2 attributs x et y pour la représentation des clusters lignes-carrés (lignec1.arff et lignec2.arff),

- Testez l'algorithme des K-moyennes et celui l'Expectation-Maximization.
- Permettent-ils de retrouver les groupes identifiables graphiquement ?
- Visualisez les assignements aux clusters.

2.2 Jeu de données par distribution sur chaque attribut

En utilisant les jeux de données pour la distribution (em2.arff et em3.arff).

- Testez l'algorithme des K-moyennes et l'Expectation maximization.
- Permettent-ils de retrouver les groupes identifiables graphiquement ?
- Visualisez les assignements aux clusters.

2.3 Données Titanic

- Filtrez les données "titanic.arff" ("titanic.txt") pour enlever la prédiction (SURVIVED) des données d'apprentissage.
- Les algorithmes de clustering des K-moyennes et l'Expectation maximization permettent-ils de découper les données en un groupe de survivants et un groupe de non-survivants ?
- Visualisez les assignements aux clusters.

2.4 Iris

- Testez les algorithmes des K-moyennes et l'Expectation maximization, en faisant varier les paramètres sur le jeu de données Iris.

- Ignorez des attributs et tester l'influence sur le résultat.
- Visualisez les assignements aux clusters.

2.5 Bilan

- Testez si possible d'autres jeux de données (de préférence les plus grands i.d. soybean, labour, etc.) et analysez les différents clusters fournis avec SimpleKMeans, EM et Cobweb.
- Finalement, listez les principaux avantages et désavantages de ces algorithmes de clustering utilisés durant la séance.
- Evaluer la qualité d'un clustering est toujours difficile lorsque l'on compare différentes exécutions. D'après ce que vous avez constaté durant cette séance, quels critères pourriez-vous employer pour la qualite ? des clusters.

3 Application

Cet exercice porte sur les jeux de données Clients (Clients.csv) et Immatriculations (Immatriculations.csv) Voici quelques informations sur les attributs

Pour Cilents.csv : ?

- taux : exprime en euros la capacité d'endettement du client (correspond à environ 30% de son salaire)
- 2eme voiture : valeur booléenne indiquant si le client possédait déjà un véhicule principal

Pour Immatriculations.csv :

- puissance : exprimée en chevaux
- longueur : quatre catégories ont été déterminées « courte », « moyenne », « longue » et « très longue »
- prix : en euros

On demande de lancer différentes méthodes de clustering sur ces deux jeux de données et de présenter les solutions les plus pertinentes extraites (on pourra considérer le jeu de données Clients_1.csv dans lequel l'attribut immatriculation a été supprimé). Il s'agira de sélectionner différents attributs des jeux de données initiaux et de lancer différents algorithmes de clustering en faisant varier la valeurs de leurs paramètres

4 Utilisation de weka sans l'interface

Dans cette partie, on va utiliser les classes Java de Weka (voir la doc en ligne).

- Vous avez dû constater que l'algorithme K-Means est sensible à l'initialisation. Écrivez un programme Java qui prend en argument un fichier de données et applique l'algorithme avec différentes graines, et retourne pour chaque valeur l'erreur "SquaredError". Vous pouvez retournez la meilleure segmentation.
- Un autre inconvénient de l'algorithme K-Means est de devoir fixer le nombre de clusters. Écrivez un programme qui prend en argument un fichier de données et qui applique l'algorithme avec différents nombres de clusters. Pour chaque valeur, vous pourrez appliquer l'algorithme avec différentes graines et prendre l'erreur moyenne. Affichez les différentes erreurs moyennes pour chaque valeur de nombre de clusters (k). Testez sur les exemples précédents. Que constatez vous ? Cette approche peut-elle vous aider à trouver le nombre « idéal » de clusters pour le jeu de données considéré ?

Remarque : pour interpréter plus facilement les résultats, on peut visualiser le graphe de la fonction qui associe l'erreur moyenne au nombre de clusters. Pour cela, vous pouvez utiliser **gnuplot**. *Il suffit de créer un fichier contenant les valeurs (une ligne par point de la forme* x, y *et sous gnuplot de l'afficher avec la commande* `plot nomfichier with lines`. *Pour ceux qui sont sous Windows, il vous reste les outils de bureautique habituels.*