IFT3335 – TP3

Classification de textes

Ce TP est à réaliser en groupe de 2 personnes (ou seul).
Date de remise : 2 décembre, avant 23 :39.
Vous devez seulement remettre votre rapport de 5 pages.

Ce TP a pour but de vous permettre d'utiliser et de comparer les algorithmes de classification (d'apprentissage) sur des données concrètes. Il existe des outils qui implantent différents algorithmes. Dans ce TP, vous allez utiliser Weka, un package populaire qui implante plusieurs algorithmes, et qui offre une interface conviviale, en plus des outils de prétraitement de données et d'analyse.

Les tâches demandées :

1. Installer Weka sur votre ordinateur (votre ordinateur personnel ou dans votre compte du DIRO).
2. Télécharger les collections de test. Dans ce TP, on utilise 2 collections de test – un sous ensemble de Reuters-21578 et un sous ensemble de OHSUMED.
3. Faire fonctionner Weka sur ces collections, avec les algorithmes demandés (voir plus bas), et explorer librement les autres algorithmes.
4. Écrire un rapport (pas plus de 5 pages) qui discute de votre expérience dans ce TP ainsi que ce que vous avez observé sur les différents algorithmes et les données (notamment des comparaisons entre eux sur la performance et sur le temps d'exécution).

Weka

Weka est un package Open Source très populaire. Il implante différents algorithmes de data-mining et d'apprentissage (Naïve Bayes, Arbre de décision, SVM, réseau de neurones, entre autres). Il offre une interface GUI conviviale pour manipuler et inspecter les données et visualiser les résultats. Ce package peut fonctionner sur les plateformes Linux, Windows et Mac.

Le package (version 3-6-10) est téléchargeable sur ce site : http://www.cs.waikato.ac.nz/ml/weka/

Vous êtes conseillés fortement de lire le tutoriel de Weka et la documentation de Weka.

Il y a aussi une série de cours en ligne par Ian Witten sur Weka (http://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/), et une autre vidéo d'introduction par Brandon Weinberg (http://www.youtube.com/watch?v=IY29uC4uem8). Ces vidéos fournissent une introduction rapide sur Weka.

Les collections de test

Il y a plusieurs collections de test pour la classification de textes. Reuters-21578 est une collection de test très souvent utilisée pour tester des algorithmes de classification (ou catégorisation) de textes. Elle contient des articles de presse (de Reuters), qui sont manuellement classées par des éditeurs. Vous pouvez trouver plus d'information ici : http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection.

La collection contient 135 classes. Pour ce TP, nous allons utiliser seulement 6 classes – 2 grandes classes, 2 moyennes classes et 2 petites classes – pour voir les différences dues à la taille.

- Petites classe : heat.csv, housing.csv
- Moyennes classe : coffee.csv, gold.csv
- Grandes classes : acq.csv, earn.csv

Ces classes sont contenues dans les fichiers suivants :

- reuters-6.zip : contient 6 fichier csv, chacun pour une classe. Ceci est pour faciliter les expériences avec SVM, qui fait la classification binaire.
- reuters-allcat-6.zip : contient les données de toutes les classe, en csv.

La collection OHSUMED (http://ir.ohsu.edu/ohsumed/ohsumed.html) contient 348,566 références médicales. Nous allons aussi utiliser seulement 6 classes :

- Petites classes : Mitosis.csv, Pediatrics.csv
- Moyennes classes : Necrosis.csv, Hyperplasia.csv
- Grandes classes : Pregnancy.csv, Rats.csv

Ces données sont contenus dans :

- ohsumed-6.zip : contient 6 fichiers en csv, chacun pour une classe – ceci est pour faire des expériences avec SVM.
- ohsumed-allcats-6.zip : contient un fichier csv contenant toutes les classes.

Préparatifs

Pour vous familiariser avec Weka, explorer le librement. Explorer au moins les fonctions suivantes :

1. Preprocess

Ceci vous permet de charger les données et faire des prétraitements sur les données, e.g. la sélection des données à traiter, appliquer des filtres pour transformer les données et les attributs, etc.

Pour commencer, ouvrez un ensemble de données existant dans le package (e.g. data/weather.numeric.arff). Vous devez voir un ensemble d'attributs. En cliquant sur chaque attribut, vous pouvez voir la distribution de valeurs dans les données.

2. Classify

Une fois les données chargées, vous pouvez maintenant choisir un algorithme et l'appliquer sur les données. Pour essayer, choisissez (avec choose) une méthode (e.g. tree->J48, qui correspond à l'arbre de décision présenter dans le cours).

Choisissez d'abord « Use training set » dans « Test options » (qui tente d'entraîner un arbre en utilisant tous les exemples d'entraînement), et cliquer sur « Start ». Vous verrez à droite le résultat d'entraînement, avec l'arbre obtenu, ainsi que le résultat de classification sur ce même ensemble de données.

Vous pouvez ensuite choisir Cross-validation, Folds = 4, pour voir l'effet de validation croisée, c'est-à-dire de découper les données en 4 parties, et on fait 4 expériences en

utilisant, à tour de rôle, une partie comme test et les 3 autres parties pour l'entraînement. La validation croisée est à utiliser si vous n'avez pas déjà une collection avec les sous ensembles d'entraînement et de test déjà séparés (c'est le cas pour les sous ensembles de Reuters et OHSUMED qu'on utilise dans ce TP).

3. Vous pouvez maintenant tenter de sélectionner des attributs à utiliser pour la classification dans « Select attributes ». Il y a différentes méthodes pour sélectionner un sous ensemble d'attributs à utiliser dans la classification. Ceci est très utile quand vos données sont très bruitées, avec beaucoup d'attributs qui n'aident pas à la classification. Un nettoyage (une sélection) est très bénéfique dans ce cas. Cette sélection aide aussi à accélérer les traitements.

Pour essayer, choisissez dans « Attribute Evaluator » la méthode InfoGainAttributeEval (la sélection basée sur le gain d'information), et dans « Search Method » la méthode Ranker – qui ordonne les attributs selon leur valeur. En cliquant sur Ranker (une fois c'est choisi), on peut préciser les critères de sélection, par exemple, en fixant un seuil, ou en fixant un nombre d'attributs à garder.

Jouer librement avec les données Reuters incluses dans le package Weka. Notamment, vous devez transformer un texte en un ensemble d'attributs (chaque mot = 1 attribut). Après cette transformation, vous allez pouvoir utiliser les algorithmes de classification. Certaines options vous sont offertes dans cette transformation (avec filter) : filters->unsupervised->StringToWordVector transforme un texte en un ensemble d'attributs mot. De plus, dans ce filtre, il vous serait aussi possible de préciser si le résultat de cette transformation produit un ensemble d'attributs (mots) binaire (présent ou absent) ou avec un poids numérique (fréquence, tf transformé et avec idf).

Une pratique courante dans le domaine de classification de textes et de recherche d'information est de tronquer les mots pour ne garder que les racines. Par exemple, le mot « computer » sera tronqué en « comput ». Ceci a pour but de créer une représentation unique pour une famille de mots semblables (computer, computing, compute, computes, computed). Ce processus est appelé « stemming ». Il y a des méthodes de stemming standard disponibles, dont la méthode de Porter. Le programme correspondant peut être téléchargé à :
http://weka.wikispaces.com/file/view/snowball-20051019.jar/82917267/snowball-20051019.jar

Pour l'intégrer dans le package Weka, vous devez faire :
java -cp /Users/as/Documents/Work/weka-3-6-10/snowball-20051019.jar:weka-3-6-10/weka.jar weka.gui.Main

Lisez le tutoriel et regarder les vidéos pour en apprendre plus. Lisez sur les problèmes de détails dans un autre fichier « TP2-problèmes ».

Vos tâches

Les tâches que vous être demandés à accomplir sur les collections Reuters et OHSUMED sont les suivantes :

1. Charger à tour de rôle chaque collection (Reuters et OHSUMED); segmenter les mots dans le champ « text » ; classer les documents avec Naïve Bayes, en utilisant une validation croisée de 10 folds. Cette méthode constitue la méthode de base.

StringToWordVector

Vous surveillez la performance de cette méthode en notant la précision et le rappel de chaque classe, ainsi que la moyenne sur toutes les classes.

2. Sur la méthode de base, appliquer une sélection d'attributs, avec le gain d'information et de chi-carré, en retenant un nombre fixe (e.g. 1000) d'attributs. Tester encore l'algorithme Naïve Bayes. Essayer avec différents nombres d'attributs pour voir combien d'attributs (environ) vous donne la meilleure performance. Ce test vise à constater si une sélection d'attributs est bénéfique.
3. Sur la méthode de base, appliquer maintenant le stemming (c'est une option disponible du filtre pour convertir un string à un vecteur de mots) pour transformer les mots en leur racine (e.g. *computing* en *comput*). Choisissez snowballStemmer. Refaire les 2 tests ci-dessus. Ceci a pour but de voir l'effet de stemming sur la classification de textes.
4. Choisissez les autres algorithmes de classification : arbre de décision (J48), SVM (SMO – une implantation de SVM) et MultiLayerPerceptron (en choisissant 3 différents nombre de neurones cachés – 5, 10, 30), et répétez les expériences ci-dessus. Ceci vous permet de voir la performance de chaque algorithme de classification sur les collections et sur les différentes classes.
5. Finalement, explorez librement les options et les algorithmes disponibles, et tentez d'améliorer le résultat de classification.

Rapport :

Dans votre rapport, vous devez décrire brièvement les expériences que vous faites, les résultats obtenus, et surtout une analyse sur les résultats. Cette analyse devrait montrer la comparaison entre les différents algorithmes, l'impact de différentes options (stemming, …). Décrivez librement ce que vous observez d'intéressant dans ces expériences.