Fiche TP N°2

Classification, Regression et Clustering

Ce TP a pour but de découvrir comment utiliser des algorithmes d'apprentissage automatique supervisés et non supervisés sous WEKA.

1. Selection d'une méthode

A partir de l'onglet *Classify* on peut observer plusieurs algorithmes de classification et de régression qui sont implémentés. La section *Classifier* contient un champ de texte qui donne le nom du classifieur actuellement sélectionné, ainsi que ses options. Un clic sur la zone de texte fait apparaître une boîte de dialogue, comme pour les filtres, que vous pouvez utiliser pour configurer les options du classifieur actuel. Le bouton *Choose* vous permet de choisir un des classifieurs qui sont disponibles dans WEKA. Les méthodes de classfication et de régression dans l'outil WEKA, sont regroupées dans le dans le dossier *weka /classifiers*.

2. Options du test

Le résultat de l'application de la technique choisie sera testé selon les options qui sont définies en cliquant sur la boîte *Test options*. Il existe quatre modes de test :

1. **Use training set :** la technique est évaluée sur sa capacité à prédire la classe des instances sur lesquelles il a été formé ;
2. **Supplied test set :** la technique est évaluée sur la façon dont il prédit la classe d'un ensemble d'instances chargé à partir d'un fichier. Cliquer sur le bouton *Set...* fait apparaître une boîte de dialogue vous permettant de choisir le fichier à tester ;
3. **Cross validation :** le classifieur est évalué par validation croisée, en utilisant le nombre de champs qui sont entrés dans le champ *Folds* ;
4. **Persentage split :** le classifieur est évalué sur la façon dont il prédit un certain pourcentage des données qui sont conservées pour le test. La quantité de données conservées dépend de la valeur saisie dans le champ %.

D'autres options de test peuvent être définies en cliquant sur le bouton *More options.*

3. L'attribut Classe

Les classifieurs dans WEKA sont conçus pour être entraînés à prédire un seul attribut «class », qui est la cible de la prédiction. Par défaut, la classe est considérée comme le dernier attribut des données. Si vous voulez entraîner un classifieur à prédire un attribut différent, cliquez sur la case située sous la boite *Test options* pour faire apparaître une liste déroulante d'attributs à choisir.

4. Entraîner un classifieur

Une fois que le classifieur, les options de test et la classe ont tous été définis, le processus d'apprentissage est lancé en cliquant sur le bouton *Start*. Pendant que le classifieur est occupé à être formé, le petit oiseau se déplace. A l'issue de l'apprentissage, une nouvelle entrée apparaîtra dans la zone Result list, et la zone *Classifier output*, à droite de l'écran, sera remplie de texte décrivant les résultats de l'entraînement et du test.

5. La zone Classifier output

La sortie est divisée en plusieurs sections :

1. **Run information :** Une liste d'informations donnant le nom du dataset, les instances, les attributs et le mode de test qui ont été impliqués dans le processus d'apprentissage.
2. **Classifier model (full training set) :** Une représentation textuelle du modèle de classification qui a été produit sur toutes les données d'entraînement.

Les résultats du mode de test choisi sont décomposés ainsi :

3. **Summary :** Une série de statistiques résumant la précision avec laquelle le modèle a été capable de prédire la vraie classe des instances dans le mode de test choisi.
4. **Detailed Accuracy By Class :** Une décomposition plus détaillée par classe de la précision de prédiction du modèle.
5. **Confusion Matrix :** Montre combien d'instances ont été assignées à chaque classe.

6. Clustering

A partir de l'onglet *Cluster*, on peut observer plusieurs algorithmes implémentés : *EM, FilteredClusterer, HierarchicalClusterer*, *SimpleKMeans*, etc. Les classes qui implémentent les méthodes de clustering dans l'outil WEKA, sont regroupées dans le dossier *weka/clusterers* dans la section *Clusterer*. Un clic sur la zone de texte fait apparaître une boîte de dialogue, comme pour les filtres, que vous pouvez utiliser pour configurer les options de la méthode de clustering actuelle.

Travail à faire

1. Arbre de décision

Les arbres de décision peuvent prendre en charge les problèmes de classification et de régression.

Choisissez l'algorithme de l'arbre de décision :

1. Cliquez sur le bouton "Choisir" et sélectionnez ***REPTree*** dans le dossier *weka /classifiers/Trees*.
2. Cliquez sur le nom de l'algorithme pour revoir les hyper-paramètres de l'algorithme. La profondeur de l'arbre est définie automatiquement, mais une profondeur peut être spécifiée dans l'attribut *maxDepth*.

3. Vous pouvez également choisir de désactiver l'élagage en définissant le paramètre *noPruning* à vrai, bien que cela puisse entraîner une baisse des performances.
4. Le paramètre *minNum* définit le nombre minimum d'instances prises en charge par l'arbre dans un nœud feuille lors de la construction de l'arbre à partir des données d'apprentissage.
5. Cliquez sur "OK" pour fermer la configuration de l'algorithme.
6. Cliquez sur le bouton *Start* pour exécuter l'algorithme sur l'ensemble de données *diabetes.arff*.
7. Avec la configuration par défaut, quelle est la précision de l'algorithme de l'arbre de décision.
8. Un autre algorithme d'arbre de décision plus avancé que vous pouvez utiliser est l'algorithme C4.5, appelé J48 dans WEKA.
9. Vous pouvez examiner la visualisation d'un arbre de décision préparé sur l'ensemble des données d'entraînement en faisant un clic droit sur la *Result list* et en cliquant sur Visualize Tree.

2. K-Nearest Neighbors (kNN)

L'algorithme k-Nearest Neighbors (kNN en abrégé) prend en charge à la fois la classification et la régression.

Choisissez l'algorithme k-Nearest Neighbors :

1. Cliquez sur le bouton *Choose* et sélectionnez ***IBk*** dans le dossier *weka/classifiers/lazy*.
2. Cliquez sur le nom de l'algorithme pour revoir la configuration de l'algorithme.
3. La taille du voisinage est contrôlée par le paramètre k. Par exemple, si k est fixé à 1, les prédictions sont effectuées en utilisant l'instance d'apprentissage la plus similaire à un nouveau modèle donné pour lequel une prédiction est demandée. Les valeurs courantes pour k sont 3, 7, 11 et 21, plus grandes pour les ensembles de données de grande taille. WEKA peut découvrir automatiquement une bonne valeur pour k en utilisant la validation croisée dans l'algorithme en définissant le paramètre *crossValidate* à *True*.
4. Un autre paramètre important est la mesure de distance utilisée. Elle est configurée dans le paramètre *nearestNeighbourSearchAlgorithm* qui contrôle la manière dont les données d'apprentissage sont stockées et recherchées. Par défaut, il s'agit d'un *LinearNNSearch*. En cliquant sur le nom de cet algorithme, vous obtenez une autre fenêtre de configuration dans laquelle vous pouvez choisir un paramètre de *distanceFunction*. Par défaut, *la distance euclidienne* est utilisée pour calculer la distance entre les instances, ce qui est bon pour les données numériques ayant la même échelle. La distance *Manhattan* est à utiliser si vos attributs diffèrent en termes de mesures ou de type.
5. Cliquez sur "OK" pour fermer la configuration de l'algorithme.
6. Cliquez sur le bouton *Start* pour exécuter l'algorithme sur le jeu de données *diabetes*.
7. Quelle est la précision atteinte avec l'algorithme kNN avec la configuration par défaut ?

3. Support Vector Machines (SVM)

Les Support Vector Machines (SVM) ont été développées pour les problèmes de classification binaire, bien que des extensions de la technique aient été faites pour prendre en charge les problèmes de classification et de régression multi-classes.

Choisissez l'algorithme SVM :

1. Cliquez sur le bouton *Choose* et sélectionnez ***SMO*** dans le dossier *weka/classifiers/functions.*
2. Cliquez sur le nom de l'algorithme pour examiner la configuration de l'algorithme. SMO fait référence à l'algorithme d'optimisation efficace spécifique utilisé dans l'implémentation SVM, qui signifie Sequential Minimal Optimization.
3. Le paramètre C, appelé paramètre de complexité dans WEKA, contrôle la flexibilité du processus de traçage de la ligne de séparation des classes. La valeur par défaut est de 1.
4. Un paramètre clé dans SVM est le type de *kernel* à utiliser. La valeur par défaut dans WEKA est *PolyKernel* qui sépare les classes à l'aide d'une ligne courbe ou ondulée. Il est conseillé d'essayer une série de kernel et de valeurs de C différents sur votre problème et de voir ce qui fonctionne le mieux.
5. Cliquez sur "OK" pour fermer la configuration de l'algorithme.
6. Cliquez sur le bouton *Start* pour exécuter l'algorithme sur l'ensemble de données *diabetes.*
7. Avec la configuration par défaut, quelle est la précision atteinte avec l'algorithme SVM ?

4. Régression logistique

La régression logistique est un algorithme de classification binaire. Elle suppose que les variables d'entrée sont numériques. L'algorithme apprend un coefficient pour chaque valeur d'entrée, qui sont combinés linéairement dans une fonction de régression et transformés à l'aide d'une fonction logistique. La régression logistique est une technique rapide et simple, mais elle peut être très efficace pour certains problèmes. La régression logistique ne prend en charge que les problèmes de classification binaire, bien que l'implémentation WEKA ait été adaptée pour prendre en charge les problèmes de classification multi-classes.

Choisissez l'algorithme de régression logistique :

1. Cliquez sur le bouton *Choose* et sélectionnez ***Logistique*** dans le dossier *weka/classifiers/functions.*
2. Cliquez sur le nom de l'algorithme pour revoir la configuration de l'algorithme.
3. L'algorithme peut s'exécuter pendant un nombre fixe d'itérations (*maxIts*), mais par défaut, il s'exécutera jusqu'à ce qu'il soit estimé que l'algorithme a convergé.
4. L'implémentation utilise un estimateur *ridge* qui est un type de régularisation. Cette méthode cherche à simplifier le modèle pendant l'apprentissage en minimisant les coefficients appris par le modèle. Le paramètre *ridge* définit le degré de pression à exercer sur l'algorithme pour réduire la taille des coefficients. Une valeur de 0 désactive cette régularisation.
5. Cliquez sur "OK" pour fermer la configuration de l'algorithme.

6. Cliquez sur le bouton *Start* pour exécuter l'algorithme sur le jeu de données *diabetes*.
7. Avec la configuration par défaut, quelle est la précision atteinte avec la régression logistique ?

5. Naive Bayes

Naive Bayes est un algorithme de classification. Traditionnellement, il suppose que les valeurs d'entrée sont nominales, bien que ses entrées numériques soient prises en charge par l'hypothèse d'une distribution. Il utilise une mise en œuvre simple du théorème de Bayes (d'où le terme naïf) où la probabilité antérieure de chaque classe est calculée à partir des données d'apprentissage et supposée être indépendante les unes des autres. Il s'agit d'une hypothèse irréaliste car nous nous attendons à ce que les variables interagissent et soient dépendantes, bien que cette hypothèse rende les probabilités rapides et faciles à calculer. Même dans le cadre de cette hypothèse irréaliste, les Naive Bayes se sont avérés être un algorithme de classification très efficace. Naive Bayes calcule la probabilité postérieure de chaque classe et prédit la classe dont la probabilité est la plus élevée. En tant que tel, il prend en charge les problèmes de classification binaire et de classification multi-classes.

Choisissez l'algorithme Naive Bayes :

1. Cliquez sur le bouton *Choose* et sélectionnez ***NaiveBayes*** dans le dossier *weka/classifiers/b*ayes.
2. Cliquez sur le nom de l'algorithme pour revoir la configuration de l'algorithme.
3. Par défaut, une distribution gaussienne est supposée pour chaque attribut numérique. Vous pouvez modifier l'algorithme pour utiliser un estimateur à noyau avec l'argument *useKernelEstimator* qui peut mieux correspondre à la distribution réelle des attributs dans votre ensemble de données. Vous pouvez également convertir automatiquement les attributs numériques en attributs nominaux avec le paramètre *useSupervisedDiscretization*.
4. Cliquez sur "OK" pour fermer la configuration de l'algorithme.
5. Cliquez sur le bouton *Start* pour exécuter l'algorithme sur l'ensemble de données *diabetes*.
6. Avec la configuration par défaut, quelle est la précision atteinte avec Naive Bayes.

6. K-Means

Pour le clustering, on considérera le jeu de données *Iris.arff*.

- Effectuez un clustering du jeu de données en utilisant l'algorithme ***SimpleKMeans*** et en conservant les paramètres par défaut. L'option d'évaluation *Classes to clusters evaluation* permet d'assigner une classe à un cluster pendant la phase de test, affiche aussi le log-likelihood, (ou log-vraisemblance), calcule l'erreur et la matrice de confusion.
- 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*.
- Quel est le meilleur résultat selon le taux d'erreur et la taille de clusters.

- Conservez le meilleur résultat.
- À partir de la boîte *Result List* (bouton droit, *Visualize cluster assignement*), visualisez 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.

7. EM

La méthode EM (Expectation Maximisation) génère une description probabiliste des clusters en termes de moyenne et écart-type pour les attributs numériques et en termes 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.

- Effectuez un clustering du jeu de données en utilisant la méthode EM avec les paramètres par défaut. Cliquez sur le bouton *Choose* dans la section *Clusterer* et sélectionnez ***EM*** dans le dossier *weka /clusterers*.
- Dans la fenêtre *Cluster mode,* sélectionner l'option *Classes to clusters evaluation*.
- Cliquez sur le bouton *Start* pour traiter les données. Après un certain temps, les résultats seront présentés à l'écran.
- À partir de l'écran de sortie, qu'observez-vous ?
- Quelle est la précision de l'algorithme EM ?
- Les résultats à ceux obtenus avec la méthode k-means.
- Pour visualiser les clusters, faites un clic droit sur le résultat EM dans la liste des résultats. Sélectionnez *Visualize cluster assignments*.