Devoir: Weka Knowledge Flow

Introduction

En plus de l'Explorer, l'environnement de data mining Weka offre un autre outil qui permet de faire l'analyse de données sous forme de flot: c'est le Knowledge Flow. La plupart des fonctionnalités de l'Explorer sont accessible via le Knowledge Flow mais pas toutes. De même le Knowledge Flow a des fonctionnalités qui ne sont pas disponible dans l'Explorer. Par exemple le Knowledge Flow est capable traiter les données en batch et en Streaming tandis que l'Explorer ne fait que du batch.

Le Knowledge flow permet de faire du data mining en disposant des outils (chargement de données, filtrage, évaluation, algorithme d'apprentissage, visualisation, etc), de les configurer en définissant leurs différents paramètres et de les connecter (définir la communication entre les différents outils). Une fois le diagramme de flot terminé il suffit de cliquer sur un bouton pour lancer le processus. On peut visualiser le résultat le résultat de chaque outil après que le processus soit terminé.

Exemple 1

Cet exemple utilise le dataset titanic[1] et consiste à prédire si une personne survit ou pas étant donnée un ensemble d'information sur la personne comme le sexe, l'age, etc. Une fois le modèle chargé avec un ArffLoader et la classe à prédire désignés avec un ClassAssigner, un filtre RemoveByName est utilisé pour supprimer les attributs comme le nom, le ticket, etc. L'outils CrossValidationFoldMaker est utilisé sur le résultat du filtre pour spécifier que la validation croisée sera utilisée. On utilise ensuite deux méthodes d'apprentissage supervisé: Un RandomForest et un NaiveBayes. Les performances des deux modèles sont évaluées avec un ClassifierPerformanceEvaluator. Un ModelPerformanceChart et un TextViewer sont enfin utilisés pour afficher la courbe ROC et visualiser différentes métriques (Rappel, précion, TP rate, FP rate, etc). Il est ainsi facile de choisir le meilleur des deux modèles.

Exemple 2

Ce second exemple effectue un apprentissage incrémentale(streaming) sur le dataset segment-challenge.arff (Image Segmentation) disponible dans Weka. Etant donnée sept classes d'images il est question de prédire de quelle classe est chacune des images. Comme dans l'exemple 1, un ArffLoader est utilisé pour charger les données et un ClassAssigner permet de sélectionner la classe à prédire. Seulement, l'outil de chargement est configuré pour charger les données instance après instance. Les données étant disponible de façon incrémentale, un modèle incrémentale est utilisé: NaiveBayesUpdatable. L'évaluation se fait aussi de façon incrémentale avec un IncrementalClassifierEvaluator. Enfin un StripChart permet de voir l'évolution des performances du modèle au fur et à mesure que les instances sont disponible.

Les fichiers .kf de chacun de ces exemples sont disponible sur github[2]. Ils peuvent être téléchargés et être ouvert par Weka.

[1] Dataset Titanic: https://www.kaggle.com/frankl1miky/titanicdmmlcsv
[2] Fichiers .kf: https://github.com/frankl1/TechnologieDuDecisionl-M2-ISIMA/tree/master/Devoir%201