Université Pierre et Marie Curie
M2 Androide

Modélisation et Simulation Multi-Agents
2019-2020, 1^{er} semestre

MoSiMa - TP n°0

Inférence et Apprentissage

Le mini-projet qui sera utilisé comme support pour les différentes séances est composé de 4 composants principaux : *JmonkeyEngine (jme3)* comme moteur pour l'environnement, *Jade* comme plateforme multi-agent, *Prolog* comme mécanisme d'inférence et *Weka* pour les algorithmes d'apprentissage.

Ce document a pour but de vous permettre de vous familiariser avec Prolog et Weka. Si vous connaissez déja Prolog, passez directement à Weka. L'ensemble des éléments abordés ici ne doit pas vous prendre plus d'1h.

1 Prise en main de prolog

L'architecture cognitive que nous allons mettre en œuvre s'appuie sur un moteur d'inférence supportant le chaînage avant et arrière (Prolog). Il nous sera nécessaire de pouvoir appeler Prolog depuis Java et inversement. Il existe de nombreuses librairies offrant cette fonctionnalité. La plus performante mais également la plus compliquée à installer est la combinaison *swi-prolog/jpl*. Celle-ci est (théoriquement) installée sur les machines de la PPTI.

- Un tutoriel de démarrage pour l'exécution de requêtes prolog simples par chaînage avant est disponible ici : `http://www.swi-prolog.org/pldoc/man?section=quickstart`
- Lisez ce tutoriel de révision (chapitre 1 uniquement) : `http://www.learnprolognow.org/lpnpage.php?pageid=online`

2 Prise en main de Weka

WEKA [1] et MOA [2] sont deux librairies d'apprentissage automatique développées en Java par l'université de Waikato en Nouvelle Zélande. Nous utiliserons Weka 3.8 et, dans un premier temps, l'un des algorithmes d'apprentissage supervisé élémentaire qu'elle fourni, J48, pour l'apprentissage d'arbres de décisions.

Les tutoriels Weka sont disponibles ici :

- Vidéos :`https://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/`
- Livres : `https://www.cs.waikato.ac.nz/~ml/weka/book.html`

1. Disponible ici : `https://www.cs.waikato.ac.nz/~ml/weka/downloading.html`
2. Spécialisée dans les flux de données : `https://moa.cms.waikato.ac.nz/`

1. Créez un répertoire projet java et téléchargez puis dézipez le jar de weka dans celui-ci.
2. Dans un terminal, lancez Weka en mode GUI avec la commande *java -jar weka.jar &*
3. Ouvrez l'explorer puis faites *open file-data* pour ouvrir le fichier *weather.nominal.arff* qui se situe dans le répertoire "data" de Weka. Un histogramme doit maintenant apparaitre. Cette base, présentée sur la figure 1 comprend 14 entrées, chacune d'elle présente la prévision météo, la température, le taux d'humidité, le vent, et indique pour chacune d'elle si les enfants ont joués ou non. L'objectif est de construire un arbre de décision permettant de prédire en fonction de ces données météo si les enfants vont aller jouer.
4. Sélectionnez l'onglet *Classify*, cliquez sur *Choose* et choisissez le classifieur J48 (répertoire *tree*).
5. Cliquez sur *Start*. J48 est exécuté sur la base d'apprentissage. L'arbre obtenu est visible en mode texte dans le haut de la fenetre, et en mode graphique en faisant bouton-droit sur l'objet visible dans *result list* puis *Visualize tree* (voir figure 2)

Si les principes inhérents à l'apprentissage supervisé et aux arbres de décision sont momentanément oubliés, référez vous aux rappels de cours distribués avec le TP.

1. Seuls 50% des exemples sont bien classés. Pourquoi ?
2. Essayez d'autres algorithmes de classification. Font-ils mieux ?
3. Changez la base de donnée pour travailler sur diabetes.arff. J48 est-il toujours le plus mauvais ? Que ce passe t'il si vous changez la taille des bases d'apprentissage et de test ?
4. Exécutez J48 sur la base initiale depuis votre code JAVA en passant par l'API présentée ici : `https://waikato.github.io/weka-wiki/use_weka_in_your_java_code/`. Passer par l'API sera nécessaire dès la semaine prochaine.

```
@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

FIGURE 1 – Contenu de la base weather.nominal

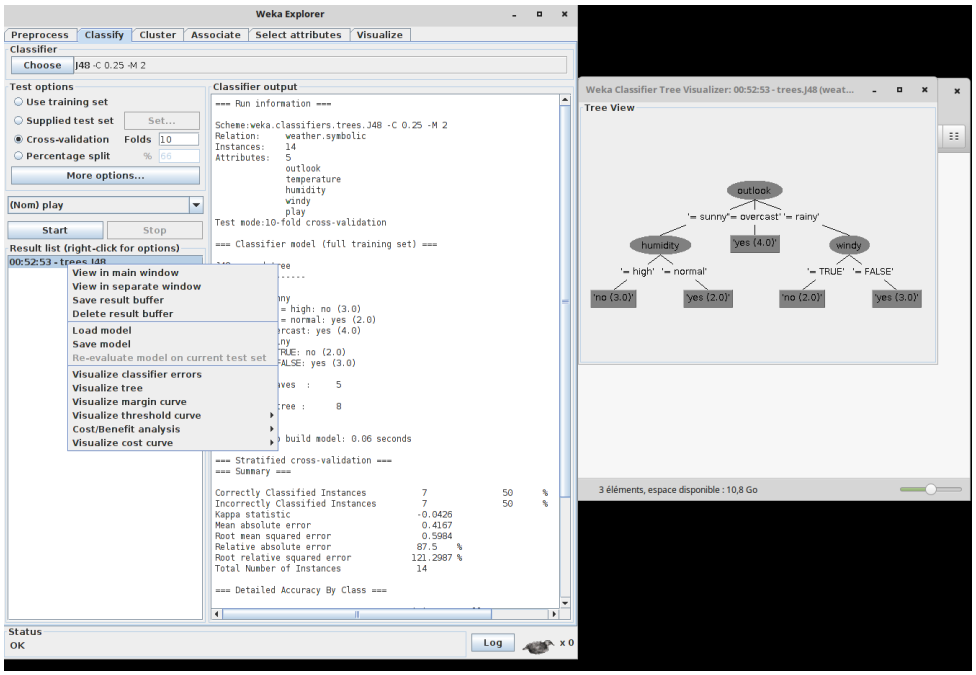

FIGURE 2 – Affichage de l'arbre de décision obtenu