**CSI4506: Introduction à l’Intelligence Artificielle**

**Hiver 2012**

**Devoir 3**

**A rendre le 3 Avril 2012**

**I. Planification**

Lire les sections 10.1 et 10.2 du manuel de cours et écrire un résumé (d’environ deux pages) sur la planification classique, basé sur ces sections.

**II. Traitement du Langage Naturel**

Utilisez le langage de programmation de votre choix (peut-être que le Scheme ou le Prolog sera en fait le langage le plus pratique dans ce cas !) pour programmer un système de traitement du langage très simple dont le but sera d’indiquer si une phrase est grammaticalement et sémantiquement correcte.

Il vous faudra :

1. Une grammaire. Choisissez-en une toute simple, telle que :

S 🡪 NP VP

NP 🡪 Art Adj N

VP 🡪 V NP | V

1. Un lexicon. De nouveau, choisissez-en un tout simple du style :

Le : **article**

bébé, poulet, chien, carton, fauteuil : **nom**

rouge, sage, cassé, frit : **adjectif**

mange, casse, boit, saute : **verbe** (indiquer si transitif ou intransitif)

1. Une ontologie. Encore une fois, voici une ontologie toute simple :

 Tout

 / \

 Animé Inanimé

 / \ \

 Humain Animal Objet

 (bébé) (poulet, (carton,

 chien) fauteuil)

Votre système devra prendre des phrases du type :

1. Le bébé casse le carton rouge
2. Le chien sage mange le poulet frit
3. Le poulet saute

Et indiquer que ces phrases sont correctes syntaxiquement et sémantiquement. Par contre, il devra indiquer que les phrases du types :

1. Le bébé carton rouge
2. Le fauteuil mange le poulet
3. Le poulet saute le carton
4. Le carton sage boit

sont incorrectes (indiquez si c’est pour des raisons syntaxiques ou sémantiques).

Vous pouvez vous cantonnez aux exemples donnés ou si vous voulez en faire plus (vous pourrez recevoir des points en bonus !), ajoutez la concordance des genres, des règles de grammaires plus compliquées, etc…

**III. Apprentissage**

Le but de cette partie du devoir est de vous donner une expérience pratique avec l’apprentissage par arbre de décision et réseau neuronaux et de vous faire comparer la construction de systèmes experts à l’apprentissage automatisé. En particulier, je vous pose deux questions :

1. Je vous demande de comparer la performance de C4.5 (J48 dans WEKA – voir ci-dessous) avec celle de perceptrons à couches multiples (MLP) sur la base de données « Mushrooms » [pour continuer notre saga sur les champignons !] de UCI (<http://www.sgi.com/tech/mlc/db/mushroom.all> <http://www.sgi.com/tech/mlc/db/mushroom.names>).
2. Je vous demande, également, de comparer votre expérience dans ce devoir à celle du devoir précédent. Dans le cas de la détection des champignons vénéneux, vaut-il mieux créer un système expert comme vous l’avez fait dans le devoir 2 ? Ou est-il plus simple de créer une base de données et d’utiliser des algorithmes d’apprentissage comme dans ce devoir ? Vous serait-il facile de créer une base de données du style de la base de données de UCI avec les connaissances que vous avez acquises dans le devoir précédent ? Essayez, pour voir, (si vous le pouvez) de créer cette base de données et exécutez J48 dessus, avant de me répondre.

**Conseils pratiques :**

Veuillez noter qu’afin d’utiliser MLP, les valeurs des attributs de la base de données devront être converties en valeur numériques qui devront ensuite être normalisées dans la rangée [0,1]. Vous pouvez normaliser les données de chaque attribut en utilisant la normalisation max-min qui utilise la formule suivante :

*V’ = (V- min)/(max-min)*

avec *V’* correspondant à la nouvelle valeur de l’attribut (entre 0 et 1), *V* correspondant à l’ancienne valeur de l’attribut (entre *min* et *max*), *min* correspondant à la valeur originale minimale de l’attribut et *max* correspondant à la valeur originale maximale de l’attribut. Par exemple, si on considère l’attribut 2,

cap-s urface: fibrous=f, grooves=g, scaly=y, smooth=s

Cet attribute a 4 valeurs qui peuvent être converties en valeurs numeriques. Par exemple, fibrous = 1, grooves = 2, scaly = 3 et smooth = 4. La normalisation max-min convertie ensuite ces valeurs en les valeurs suivantes : fibrous = 0, grooves = .3333, scaly = .6667 et smooth = 1

Afin de vous faciliter la tâche, dans le case de MLP, plutôt que d’utiliser la méthode “10-fold cross-validation”, je vous demande d’utiliser le “2-fold cross-validation”. Veuillez procéder de la sorte: Divisez vos données en deux sous-ensembles : Sous-ensemble A et Sous-ensemble B. Dans le cas des réseaux neuronaux, afin d’établir un nombre approprié H d’unité cachées ainsi qu’un critère d’arrêt, S, veillez diviser le sous-ensemble A en deux sous-ensembles distribués également : Sous-ensembles A.1 et A.2. Utilisez A.1 pour l’entraînement et A.2 pour la validation avec différentes valeurs de H et S.

Notes :

(1) Vous pouvez commencer en assignant au taux d’apprentissage, η, lavaleur

 0.01 et au momentum, α, la valeur 0.9. Ne changez ces valeurs que si c’est

 vraiement nécéssaire (probablement pas).

(2) Veuillez utiliser une manière systématique pour sélectionner un nombre

optimal d’unités cachées et de critère d’arrêt (c'est-à-dire de nombre maximal d’époques)

1. Bien qu’il soit possible d’utiliser Cross-validation directement avec Weka, je vous demande de l’implémenter manuellement dans le cas des réseaux neuronaux car il m’importe que vous utilisiez les ensembles A.1 et A.2 pour la validation des valeurs de H et S. C’est pour cela que j’ai réduit le nombre de « folds » à 2 plutôt que 10.

Une fois que tous les paramètres sont établis (η , α, H et S) [notez que la recherche de paramètres optimaux doit être répétée pour chaque tâche], veuillez entraîner le réseau neuronaux optimal sur toutes les données de l’ensemble A et veuillez le tester sur le sous-ensemble B. Répétez la même série d’expériences en inversant les rôles de A et B et prenez la moyenne des deux résultats.

Utilisez A et B pour vos expériences avec l’arbre de décision également (mais bien entendu, vous n’avez pas besoin de subdivision de l’ensemble d’entraînement ici, sauf si vous voulez expérimenter avec des biais différents ou des options de taillage différentes).

Une fois toutes vos expériences finies, veillez reporter :

1. Les taux d’erreurs obtenues par chaque mode de classification
2. Les paramètre optimaux utilisés pour chaque expérience.

**WEKA, un paquet de logiciel approprié:**

Voici l’adresse où vous pouvez trouver le logiciel Weka ainsi que sa documentation. :

<http://www.cs.waikato.ac.nz/~ml/weka/>

Pour l’utiliser vous devrez mettre vos données sous forma .arff. La documentation arff se trouve à l’adresse suivante :

<http://www.cs.waikato.ac.nz/~ml/weka/arff.html>

Pour utiliser le réseau de neurone vous devez avoir des données de type numérique, ainsi que nous l’avons déjà discuté. Votre fichier arff devrait ressembler à :

 @RELATION sonars

 @ATTRIBUTE attrib1 NUMERIC

 @ATTRIBUTE attrib2 NUMERIC

 @ATTRIBUTE attrib3 NUMERIC

 @ATTRIBUTE attrib4 NUMERIC

 @ATTRIBUTE class {class1,class2}

 @DATA

 .51,.35,.14,.02,class1

 .49,.38,.14,.02,class1

 .47,.32,.13,.02,class2

 .46,.31,.15,.02,class1

 .50,.36,.14,.02,class2

 .54,.39,.17,.04,class2

 .46,.34,.14,0.03,class2

 .50,.34,.15,0.02,class1

…

N’utilisez pas les données numériques pour J48 de manière à pouvoir facilement comprendre l’arbre de décision créé par le système.

L’utilisation de Weka est simple. Tout peut être fait dans l’interface graphique incluse. Lorsque vous lancez Weka vous avez le choix entre quatre modes d’utilisations. Utilisez le mode **explorer**  pour lancer l’interface graphique.

Dans l’interface de l’explorer vous allez utiliser deux onglets. L’onglet « preprocess » et l’onglet « classify ».

L’onglet « preprocess » est utilisé pour ouvrir votre fichier .arff.

Ensuite cliquez sur l’onglet « classify ». (photo ci-dessous) C’est dans cet onglet que vous choisissez l’algorithme, ses paramètres et la méthode d’évaluation. Cliquez sur le bouton « Choose » pour choisir l’algorithme.

L’algorithme neuronal se trouve dans **/classifiers/functions/MultilayerPerceptron**

L’algorithme d’arbre de décision se trouve dans : **/classifiers/trees/J48**

Pour changer les paramètre de l’algorithme cliquez sur le nom de l’algorithme choisit. Vous pouvez ensuite cliquer sur le bouton « more » afin d’avoir des explications sur l’algorithme est ses paramètres.

Choisissez la méthode d’évaluation dans la section « test options » qui se trouve à gauche et cliquez sur le bouton « Start ». Les résultats devraient éventuellement apparaître a droite.

