Ecole d'ingénieurs
du Canton de Vaud

Tél. : + 41 24 423 21 11
Fax : + 41 24 425 00 50

mailto:info@eivd.ch
http://www.eivd.ch

TRAVAIL DE DIPLÔME 2003

Département Electricité et Informatique
Ecole d'ingénieurs du Canton de Vaud
Route de Cheseaux 1, case postale
CH - 1401 Yverdon - les – Bains

Incorporation d'arbres de décision dans une librairie d'intelligence artificielle (Torch)

Description

Dans bien des domaines, il nous est utile de pouvoir faire de la prédiction. Une des solutions consiste à conserver de l'information sur des cas réels et de s'en servir pour prédire des nouveaux cas. Plusieurs algorithmes d'apprentissage nous permettent de trouver des tendances dans les données recueillies. En utilisant ces algorithmes, il nous est possible de trouver des modèles efficaces de prédiction.

Objectif

Le but de ce projet est d'incorporer et d'adapter un algorithme d'apprentissage (arbre de décision) au sein de la librairie Torch de l'institut IDIAP. Actuellement cette librairie permet de faire de la prédiction au moyen d'autres algorithmes d'apprentissage. L'objectif de Torch étant de contenir les meilleurs algorithmes d'apprentissage, le but de ce projet est de satisfaire partiellement ce désir en implémentant les arbres de décisions.

Qu'est-ce qu'un arbre de décision ?

Un arbre de décision est une structure composée de:

- *Feuilles*, chacune identifiant une classe.
- *Nœud*s chacun spécifiant un test effectué sur un attribut, avec une branche et un sous-arbre pour chaque valeur possible de l'attribut sur lequel on fait le test

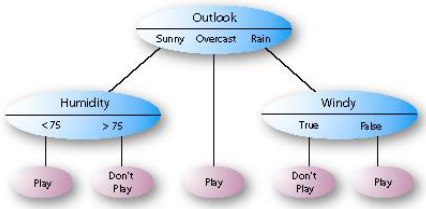
Un arbre de décision peut être utilisé pour classifier un exemple en commençant à la racine de l'arbre, en évaluant le test et en prenant la sortie appropriée. A chaque nœud de décision, la sortie du test est déterminée et l'exemple est dirigé vers la racine du sous-arbre correspondant à cette sortie. Quand ce processus nous amènera finalement à une feuille, la classe de l'exemple sera attribuée (elle sera celle de la feuille).

Un petit cas concret

Pour construire, il nous faut des données sur le domaine dans lequel on souhaite obtenir un modèle de prédiction.

Outlook	Temp	Humidity	Windy	Class
Sunny	75	70	True	Play
Sunny	80	90	True	Don't Play
Sunny	85	85	False	Don't Play
Sunny	72	95	False	Don't Play
Sunny	69	70	False	Play
Overcast	72	90	True	Play
Overcast	83	78	False	Play
Overcast	64	65	True	Play
Overcast	81	75	False	Play
Rain	71	80	True	Don't Play
Rain	65	70	True	Don't Play
Rain	75	80	False	Play
Rain	68	80	False	Play
Rain	70	96	False	Play

Ces données dont on peut voir un exemple ci – contre, sont appelées « *ensemble d'entraînement* ». Afin de construire l'arbre on se réfère à l'algorithme de *Diviser pour Régner* mise au point par *Hunt*. Mais le véritable problème réside dans le fait de trouver l'attribut de Split optimal afin d'obtenir l'arbre le plus simple . Pourquoi ne pas explorer toutes les possibilités d'arbres et sélectionner le meilleur ? Malheureusement, trouver l'arbre de décision le plus petit avec un ensemble d'entraînement est un problème *NP-complet*. Le nombre d'arbres que l'on a examinés, pour un petit ensemble d'entraînement comme le notre est d'environ 4 * 10^6 arbres de décision différents.



Domaine d'utilisation

- Data Mining
- Lutte contre le SPAM
- Banque, Assurance …

Auteur: **Frédéric Etter**
Répondant externe: **Dr. Samy Bengio**
Répondant interne: **Dr. Laura Raileanu**
Sujet proposé par: **IDIAP**

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale