PROJ631 – Projet algorithmique

Titre : Arbres de décision – de Id3 à C4.5

Descriptif général

Un arbre de décision est un outil d'aide à la décision représentant par un arbre des ensembles de choix conduisant aux décisions. Les différentes décisions possibles sont situées aux extrémités des branches (les « feuilles » de l'arbre), et sont atteintes en fonction de décisions prises à chaque étape. L'arbre de décision est un outil utilisé dans des domaines variés tels que la sécurité, la fouille de données, la médecine, etc. Il a l'avantage d'être lisible et rapide à exécuter. Il s'agit de plus d'une représentation calculable automatiquement par des algorithmes d'apprentissage supervisé, comme par exemple l'algorithme Id3.

L'objectif du projet est de comprendre et d'implémenter l'algorithme Id3 puis de l'enrichir à travers différentes extensions proposées dans l'algorithme C4.5.

Descriptif détaillé

L'algorithme Id3 repose sur une construction itérative de l'arbre de décision basée sur le contenu informationnel des données. Celles-ci sont regroupées dans une table dont chaque ligne constitue un exemple décrit par différents attributs. Un exemple est généralement noté ($\mathbf{x}$, y) où y est la valeur de l'attribut que l'on souhaite prédire, appelé classe ou décision et $\mathbf{x} = x_{A_1}, ..., x_{A_m}$, les valeurs des m autres attributs de nom A_i. L'apprentissage d'un arbre de décision se fait sur un ensemble d'instances $S = \{(\mathbf{x}, y)\}$ appelé ensemble d'apprentissage. Le qualificatif « supervisé » associé à l'apprentissage signifie que la valeur de l'attribut cible à prédire, i.e. y, est connue pour les exemples de S. Ceux-ci ont été étiquetés ou labélisés pour que l'apprentissage puisse être réalisé.

Une fois appris, un arbre de décision constitue un modèle qui pourra être exploité pour prédire la classe d'un exemple dont on ne connaît que la description $\mathbf{x}$, c'est-à-dire la valeur des m attributs qui vont permettre de parcourir l'arbre jusqu'à atteindre la feuille qui contient la prédiction de classe.

Le principe de l'algorithme Id3 qui réalise la construction de l'arbre à partir de S est le suivant :

1. Sélectionner le « meilleur » attribut A_i pour faire croître l'arbre :
   - Le « meilleur » attribut est celui qui possède le gain d'information le plus élevé pour S.
   - Un nouveau nœud $n(A_i)$ est ajouté à l'arbre et des branches sont créées pour chaque valeur possible de l'attribut A_i, i.e. $v_{i_1}, v_{i_2}, \ldots, v_{i_K}$.
2. Partitionner l'ensemble d'apprentissage « local » S en K sous-ensembles $S_1, S_2, \ldots, S_K$, chaque sous-ensemble regroupant les données satisfaisant au test $A_i = v_{i_k}$, k= 1, ..., K.
3. Le processus de construction continue (étapes **1** et **2**) jusqu'à satisfaire une des conditions d'arrêt suivantes :
   - L'ensemble d'exemples associés au nœud courant est vide.

- Tous les exemples d'apprentissage associés au nœud courant ont la même valeur de classe, auquel cas une feuille avec cette valeur de classe est retournée.
- Tous les attributs ont été utilisés sur la branche en cours de développement, auquel cas une feuille est retournée avec la classe majoritaire parmi les exemples associés au nœud courant.

Cet algorithme est décrit en détails par son inventeur Ross Quinlan dans l'article de référence ci-joint où l'on trouvera défini le critère de gain d'information exploité pour la sélection du « meilleur » attribut.

Basé sur ID3, l'algorithme C4.5 également conçu par Ross Quinlan présente les extensions suivantes :

- Adaptation de la fonction de gain d'information;
- Gestion des attributs à valeurs continues lors de la construction de l'arbre;
- Post-élagage de l'arbre;
- Gestion des attributs à valeurs manquantes;

Travail à réaliser

Votre programme devra réaliser la **phase d'apprentissage** d'un arbre Id3 à partir d'un ensemble de données d'apprentissage $\mathbf{S_{app}}$:

1. Lecture de S_{app}
   - nom des différents attributs,
   - liste des valeurs possibles pour chaque attribut,
   - données.
2. Choix de l'attribut cible et construction de l'arbre Id3 avec S_{app} selon l'algorithme détaillé ci-dessus.
3. Détermination de la matrice de confusion en apprentissage $\mathbf{M_{app}}$.
   Cette matrice est de taille c x c pour un problème à c classes. La valeur de $M_{app}[i][j]$ est le nombre d'exemples de S_{app} de classe i affectés à la classe j par l'arbre Id3 appris avec S_{app}.

puis réaliser la **phase de prédiction** pour un nouvel ensemble de données $\mathbf{S_{pred}}$:

4. Lecture de S_{pred}
5. Prédiction de l'attribut cible pour les exemples de S_{pred} par parcours de l'arbre Id3 appris avec l'ensemble d'apprentissage S_{app}.
6. Détermination de la matrice de confusion en prédiction $\mathbf{M_{pred}}$ lorsque les données de S_{pred} le permettent.
   Cette matrice M_{pred} est de nature similaire à M_{app} mais la valeur de $M_{pred}[i][j]$ est maintenant le nombre d'exemples de S_{pred} de classe i affectés à la classe j par l'arbre Id3 précédemment appris avec S_{app}.

Une fois réalisées les étapes **1** et **6**, vous pourrez proposer une alternative à Id3 en construisant un arbre C4.5. L'étape 2 permettra alors un choix entre un arbre Id3 et un arbre C4.5 avec différentes options de configuration selon que vous avez implémenté les **extensions** suggérées par **C4.5**, c'est-à-dire :

7. Adaptation de la fonction de gain d'information
   Etude et implémentation de l'indicateur de Gain Ratio et de l'indice de Gini
8. Gestion des attributs à valeurs continues lors de la construction de l'arbre
   Dans la construction de l'arbre, le test d'égalité $A_i = v_{ik}$ ne pouvant être réalisé pour l'infinité de valeurs que peut prendre un attribut à valeur réelle est remplacé par un unique test $A_i < v_i$ où v_i est une valeur réelle unique à déterminer parmi une liste de valeurs pouvant potentiellement apporter un gain d'information.
9. Post-élagage de l'arbre
   L'arbre est intégralement développé avant de procéder à son élagage pour lui assurer une meilleure capacité de généralisation.
10. Gestion des attributs à valeurs manquantes
    Lorsque des valeurs d'attribut sont manquantes dans l'ensemble d'apprentissage, il s'agit de fixer une stratégie de remplacement de la valeur manquante.

Les extensions **7** à **10** sont déjà évoquées dans l'article de référence de Id3 qui vous permettra de les mettre en œuvre. Elles sont également présentes dans certains codes accessibles en ligne notamment celui de Ross Quinlan.

Données fournies

L'archive fournie contient des fichiers de données au format csv (Comma Separated Values). C'est un format de texte simple qui permet l'import et l'export de fichiers dans des tableurs tels qu'Excel ou des bases de données telles que MySQL.

Les fichiers **golf.csv** et **golf_bis.csv** correspondent aux données utilisées à titre d'exemple dans l'article de référence.

Les fichiers **soybean_app.csv** (données d'apprentissage) et **soybean_pred.csv** (données de prédiction) correspondent à un problème de classification multi-classe (19 classes) dans lequel les exemples sont décrits par 35 attributs catégoriels. Les valeurs ? correspondent à des valeurs d'attribut manquantes. Une description des données est accessible à l'adresse https://archive.ics.uci.edu.

Article de référence

J.R. Quinlan, "Induction of Decision Trees", Machine Learning, vol. 1, pp. 81-106, 1986.

Site Web de l'auteur Ross Quinlan : https://www.rulequest.com/Personal/

Sur le plagiat

Le plagiat est une forme de fraude définie dans la charte **anti-plagiat** adoptée par l'Université Savoie Mont Blanc - https://dsi.univ-smb.fr/profil/pers/charte-anti-plagiat-2014.pdf - pouvant mener à des sanctions disciplinaires. Pour lutter contre ce phénomène, l'établissement s'est doté d'un outil de détection du plagiat permettant d'évaluer le degré d'authenticité d'un document.

En particulier, dans ce module il n'est pas admissible

- de présenter un code trouvé sur internet et/ou copié d'un autre projet sans le mentionner explicitement,
- de présenter un code non compris.