

Livrables du projet

Les étudiants devront approcher leur problème en utilisant plusieurs des méthodes vues dans le cours et/ou demandées comme tâche additionnelle. Les différentes approches au problème sont les suivantes:

1. Contexte et état de l'art du problème
 - 1.a. Analyser le contexte des données:
 - domaine d'activité ou d'origine des données
 - phénomène ou processus à l'origine des données
 - contexte de mesure/obtention des données (entrées principalement)
 - contexte d'obtention de la valeur de sortie (annotation, mesure, binarisation/catégorisation, etc.)
 - 1.b. Comprendre/définir la question à laquelle on cherche à répondre
 - classification pure versus décision diagnostique ou autre?
 - importance relative de l'explication et de la performance
 - 1.c. Comprendre les antécédents de modélisation
 - existe-il des articles présentant des résultats sur ce problème (si non: discuter avec l'expert ou la personne de référence (prof, assistant))
 - voici quelques possibles questions à se poser:
 - quelles sont les performances atteignables?
 - quelles sont les approches utilisées/ proposées?
 - existe-t-il des solutions à base de logique floue?
 - 1.d. "Apprivoiser" et analyser les données
 - Utiliser des méthodes de statistique descriptive classique pour comprendre les données: boîtes à moustaches, corrélations, etc.
 - Faire quelques graphiques illustrant les distributions des données
 - autres méthodes à choix (recherche et exploration de méthodes existant sur Python)?
2. Solution "experte"
 - 2.a. Implémenter une version "experte" du classificateur en utilisant le code Python
 - une première solution sera fournie par le professeur (et/ou l'assistant)
 - (optionnel: le groupe peut proposer une solution propre au groupe basée sur leurs observations des données)
 - 2.b. Analyser le système
 - analyser les performances globales du système: pourcentage de classification, erreur numérique, sensibilité/spécificité, autres?
 - analyser la "connaissance" contenue dans le système: règles, fonctions d'appartenance,
 - analyser quelques cas
 - 2.c. Amélioration des performances du classificateur par "tuning" manuel d'un ou plusieurs paramètres:
 - Adaptation des fonctions d'appartenance
 - Ajout/élimination de variables à certaines règles et/ou
 - Ajout/élimination de règles complètes

- 2.d. Discussion sur l'interprétabilité de la solution
- 3. Modélisation par co-évolution (FUGE)
 - 3.a. FUGE sur une instance simple de training/validation
 - Séparer les données en training/validation sets
 - Définir une combinaison de critères de performance (fitness)
 - Définir et/ou chercher de façon itérative ou séquentielle les paramètres de recherche (évolution) de l'algorithme (via script)
 - utiliser des paramètres du système flou fournis par les enseignants
 - Sélectionner quelques classificateurs et valider leurs performances sur le set de test
 - 3.b. Analyse du compromis (trade-off) entre performance et taille
 - Explorer les paramètres structuraux du classificateur flou (règles, variables, fonctions d'appartenance?)
 - Analyser les chiffres et les graphiques de performance versus taille pour tirer des conclusions
 - Sélectionner deux classificateurs: LE meilleur en termes numériques et LE meilleur en termes d'équilibre performance / interprétabilité
 - les analyser, les comparer et conclure
 - 3.c. Analyse de la tolérance à l'incertitude
 - données manquantes
 - bruit / variabilité des entrées
- 4. Une des approches suivantes devra être développée comme une contribution additionnelle des étudiants (au choix)
 - 4.a. Apprentissage semi-supervisé (ci-dessous une approche possible, mais pas la seule alternative)
 - Extraction de "règles" par clustering flou (fuzzy c-means, adaptation de k-means, ou autre)
 - Affectation des sorties (conséquents) par statistiques (p.ex. simple majorité, décomposition en sub-clusters (alpha-cut) ou approche hiérarchique)
 - Application de critères d'Interprétabilité (dès l'extraction de règles, amélioration par post-traitement)
 - 4.b. Modélisation supervisé par apprentissage (neuro-fuzzy ou similaire) quelques alternatives sont:
 - ANFIS + amélioration de l'interprétabilité
 - NEFCLASS/NEFCON + amélioration de la performance
 - ANN + extraction de règles floues interprétables
 - 4.c. Implémentation de logique floue de type 2
 - 4.d. Une autre méthode trouvée dans la littérature et implémentée et testée par l'équipe.