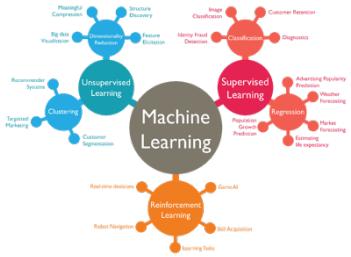

Etude d'un algorithme de Machine Learning

L'équipe de recherche PETRUS s'intéresse aux algorithmes de machine learning (ML) pour différentes raisons : (1) dans le cadre du projet PREVADOM (assistance de personnes âgées à domicile) pour détecter les activités des personnes (via des capteurs IOT) et signaler d'éventuels problèmes, comme par exemple, une diminution de la fréquence des douches, des repas ou sur la durée des nuits ; (2) dans le cadre de deux thèses sur la distribution d'algorithmes ML dans un réseau opportuniste d'une part et dans un réseau pair à pair (P2P) d'autre part.

L'objectif de ce TER est de réaliser, avec **plusieurs groupes de deux ou trois étudiants (au plus 10 groupes),** l'étude d'algorithmes de ML (un par groupe) afin d'améliorer collectivement nos connaissances (et celles des participants) sur ces algorithmes. Plus précisément, il s'agit pour chaque groupe de :

(1) Choisir un algorithme de ML (parmi la liste proposée) qu'on appelle ci-après ALGO ;
(2) Utiliser scikit-learn (voir ci-dessous) pour tester ALGO sur le jeu de données fourni avec scikit-learn ;
(3) Faire un deuxième test avec un jeu de données qui vous sera fourni ou que vous proposerez ;
(4) Chercher à faire fonctionner ALGO en apprentissage semi-supervisé (voir ci-dessous) ;
(5) Tenter de faire fonctionner ALGO en distribué (voir ci-dessous).

Les encadrants (Luc Bouganim (luc.bouganim@inria.fr) et Ludovic Javet (ludovic.javet@inria.fr)) vous fourniront un ensemble de ressources (liens WEB, jeu de données, pré-sélection d'algorithmes, vidéo explicatives) qui permettront d'accélérer la prise en main et l'avancement du projet. **Une réunion aura lieu courant février** (dans les premières semaines du TER) pour répondre aux questions de chaque groupe. Il vous sera demandé de rédiger **un court rapport** indiquant la démarche que vous avez suivi et décrivant le jeu de données, les algorithmes utilisés et la mise en œuvre pour tester l'algorithme choisi, **ainsi que le code** de scikit-learn que vous aurez modifié. **Une soutenance** de l'ensemble des groupes (20 mn par groupe) sera faite en fin de TER afin que chaque groupe puisse bénéficier des avancées des autres groupes. La liste précise des algorithmes à étudier sera fournie dès que nous aurons une idée du nombre de groupes intéressés par ce sujet. **La difficulté pour réaliser les 5 étapes ci-dessus dépend de l'algorithme choisi (voire, du jeu de données). La note du TER prendra donc bien évidemment en compte ces aspects** (ainsi, un groupe qui travaillerait sur un algorithme difficile et qui n'arriverait qu'à faire les 3 premières étapes ne serait pas pénalisé, si les difficultés sont justifiées).

Algorithmes d'apprentissage : Nous considérerons des algorithmes comme la classification naïve bayésienne, les SVM, les arbres de décisions, les régressions linéaires, les réseaux de neurones, etc…

Scikit-learn : Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs. Elle propose de nombreuses bibliothèques d'algorithmes à implémenter, clé en main. Cf. https://fr.wikipedia.org/wiki/Scikit-learn et https://scikit-learn.org/

Apprentissage supervisé : consiste à apprendre une fonction de prédiction à **partir d'exemples annotés**, au contraire de l'apprentissage non supervisé.

Apprentissage semi-supervisé : Techniques d'apprentissage automatique qui utilisent un ensemble de données annotées et non annotées afin d'améliorer significativement la qualité de l'apprentissage. Elles sont utilisées car l'annotation de données est une tâche complexe, couteuse, voire impossible dans certains cas (comme par exemple pour des personnes âgées). **Dans le cadre du projet**, on divisera le jeu de données en trois parties : (1) LAB ; (2) non-LAB et (3) TEST. Les annotations (labels) seront supprimées de la partie non-LAB mais conservées pour LAB et TEST. LAB et non-LAB seront utilisées pour l'apprentissage semi-supervisé et la partie TEST permettra de vérifier la qualité du modèle obtenu. Ainsi, en variant la proportion du jeu de données LAB/non-LAB, on pourra observer l'influence de la quantité d'annotations sur la qualité du modèle obtenu.

Apprentissage distribué : L'idée est de (1) découper le jeu de données en n partitions, (2) de faire tourner ALGO sur chaque partition séparément pour obtenir n sous-modèles et (3) finalement de proposer un moyen de combiner ces sous modèles (par exemple, en moyennant les paramètres de chaque sous modèle). Suivant l'algorithme de ML, cette distribution peut s'avérer simple, complexe, voire impossible à réaliser. Dans les deux derniers cas, vous tenterez d'expliquer pourquoi cela est complexe ou impossible.