

IFT 615 – Intelligence Artificielle

Hiver 2022

Enjeux liés au développement d'applications d'apprentissage automatique

Professeur: Froduald Kabanza

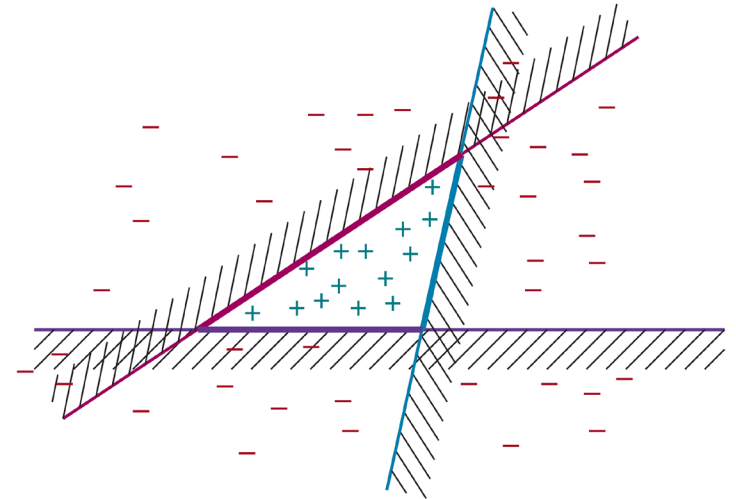
Assistants: D'Jeff Nkashama & Jean-Charles Verdier

Sujets couverts par cette leçon

- Apprentissage ensembliste
- Optimisation d'hyperparamètres
- Explicabilité des modèles
- L'IA et l'éthique

Apprentissage ensembliste (ensemble learning)

- L'apprentissage ensembliste consiste à choisir un ensemble de modèles h_1, \dots, h_n et de combiner leurs prédictions en faisant une moyenne, en faisant voter les modèles ou en utilisant une autre agrégation.
- On fait ça enfin de:
 - ◆ réduire le biais ou
 - ◆ réduire la variance
- Illustration d'un ensemble de 3 classifieurs linéaires. Il classe positivement un exemple si et seulement si l'exemple est quantifié positivement par les chacun des classifieurs.



Apprentissage ensembliste (ensemble learning)

- **Bagging** consiste à
 - ◆ Générer K ensembles d'entraînement par échantillonnage avec remplacement dans le jeu de données.
 - ◆ Pour chaque ensemble k des K ensembles, entraîner un modèle h_k . Tous les h_k utilisent le même type d'algorithme.
 - ◆
$$h(x) = \frac{1}{K} \sum_{i=1}^K h_i(x)$$
- A tendance à réduire le biais lorsqu'on a un petit jeu de données d'entraînement.
- *Random Forest* est en quelque sorte une application de *bagging* aux arbres de décision à part qu'on varie aléatoirement le choix aléatoire des attributs plutôt que le choix d'ensemble d'entraînement.
 - ◆ À chaque nœud, on échantillonne un sous-ensemble d'attribue, et on choisi un attribut de cet ensemble l'attribut avec le plus de gain d'information.

Apprentissage ensembliste (ensemble learning)

- **Stacking** consiste à
 - ◆ Entraîner K modèles différents (e.g., réseau de neurone + arbre de décision) sur le même jeu de données
 - ◆ $h(x) = \frac{1}{K} \sum_{i=1}^K h_i(x)$

Apprentissage ensembliste (ensemble learning)

- **Boosting** consiste à
 - ◆ À chaque exemple du jeu de données, on associe un poids.
 - ◆ Au départ, les poids sont les mêmes pour chaque exemple.
 - ◆ Pour i de 1 à K modèles :
 - » Générer un modèle h_i
 - » Mettre à jour les poids des exemples pour les biaiser en faveur des exemples mal classifiés
 - ◆ Chaque modèle reçoit un poids en fonction de sa performance.
 - ◆ $h(x) = \frac{1}{K} \sum_{i=1}^K z_i h_i(x)$, z_i étant le poids du modèle h_i
- *ADABOOST* est un algorithme très utilisé, basé sur cette technique. Il est utilisé le plus souvent avec les arbres de décision.



Optimisation d'hyper-paramètres

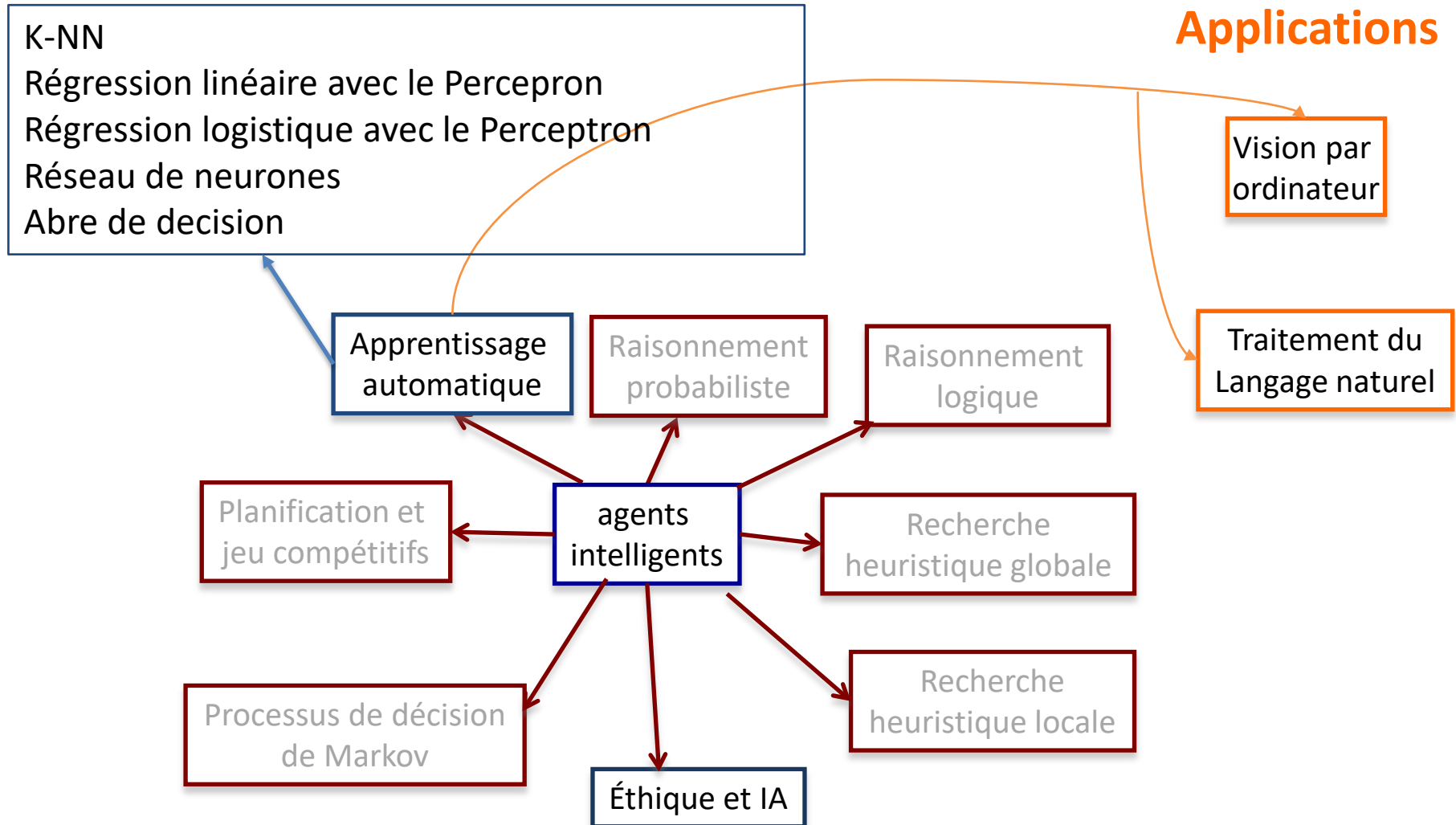
- Nous avons vu que les algorithmes d'apprentissage ont plusieurs hyperparamètres.
- Le choix des hyperparamètres relève en quelque sorte d'essai et erreur.
- C'est un problème d'optimisation en soi : recherche d'une configuration optimale des hyperparamètres.
- La fonction objective n'étant pas différentiable, on ne peut pas utiliser la descente du gradient.
 - ◆ Nous verrons des algorithmes appropriés plus tard dans le cours (recherche locale)

L'IA et l'éthique

- Dangers
 - ◆ Militarisation
 - ◆ Persuasion
 - ◆ Surveillance
 - ◆ Impact de l'automatisation sur l'emploi
 - ◆ Sécurité des applications
 - ◆ Exploitation de l'IA par les menaces en cybersécurité
 - ◆ Iniquité
- Explicabilité

Sujets couverts par le cours

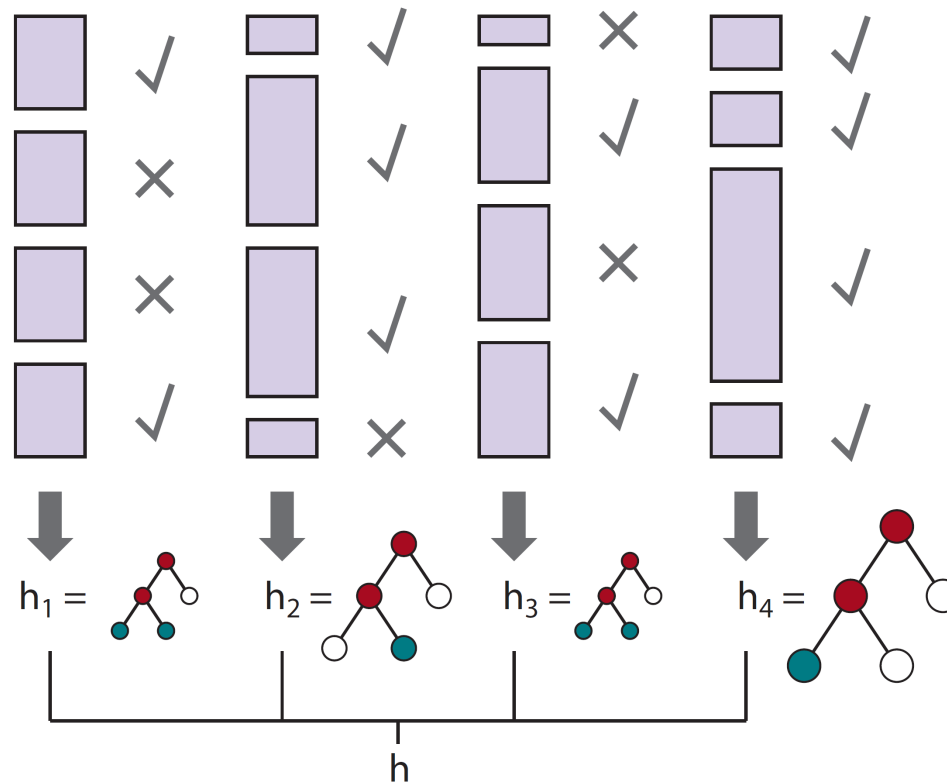
Concepts et algorithmes



Vous devriez être capable de...

- Expliquer le concept d'apprentissage ensembliste et de façon plus spécifique les techniques de *bagging*, *stacking* et *boosting*.
- Expliquer le problème d'optimisation d'hyperparamètres et pourquoi on ne peut pas le résoudre avec une descente du gradient.
- Expliquer l'enjeu d'explicabilité des modèles.
- Être conscient qu'il y a des enjeux d'éthique en IA

Illustration de *Boosting*



La taille de l'exemple illustre son poids relative.
De même, la taille de l'arbre illustre son poids relatif