

# DIC9150 Concepts fondamentaux de l'informatique cognitive

## Arbres de Décision

Roger Villemaire

Département d'informatique  
UQAM

le 3 octobre 2023



© 2016-2023 Roger Villemaire, villemaire.roger@uqam.ca  
Creative Commons Paternité - Pas d'Utilisation Commerciale - Pas de Modification 3.0 non transcrit.

# Plan

1 Apprentissage Machine

2 Arbres de décision

# Connaissance

- Jusqu'à maintenant la connaissance présente dans nos solutions logicielles était donnée de façon explicite.
- Il n'est néanmoins pas toujours possible ou même souhaitable de donner une description si explicite
  - car elle pourrait être très complexe à décrire (travail considérable, risque d'erreur),
  - ou encore c'est qu'on ne la connaît pas !
- L'*apprentissage machine* (Machine Learning) vise à développer des méthodes pour que la machine puisse, par elle-même, construire cette description.

# Exemples

- Par exemple :
  - Comment démonter un moteur ?
  - Comment choisir un moyen de transport ?
  - Comment identifier un fruit ?
  - Autoriser un crédit ?
  - Comment reconnaître une photo d'une maison ?
- Il peut exister une “meilleure” façon de faire cette tâche, mais elle peut être difficile à décrire, même par un expert.
- On peut aussi vouloir déterminer comment une certaine population effectue la tâche, en pratique.
- Il se peut aussi qu'on n'ait aucune base, ni aucune théorie pour appréhender ce genre de tâche, du moins de façon efficace.

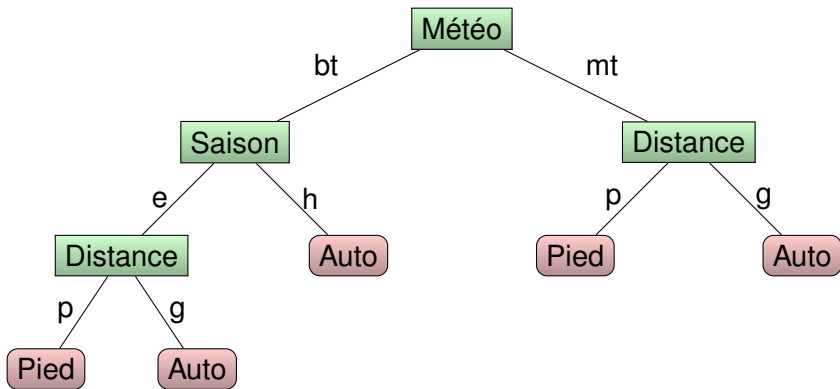
# Apprentissage Machine (Machine Learning)

- La connaissance peut toujours être représentée par une fonction, au sens mathématique du terme.
- L'apprentissage machine consiste à déterminer, par un algorithme, la représentation d'une fonction.
- On cherche donc, pour une fonction  $f$  à déterminer une bonne approximation  $h$ , qu'on appelle une *hypothèse*.
- On ne connaît pas, a priori, la fonction  $f$ , on utilisera donc  $h$  à la place.

# Types d'Apprentissage Machine

- Apprentissage supervisé
  - l'algorithme reçoit en entrée un certain nombre de valeurs correctes  $f(a)$ ,  $f(b)$ ,  $f(c)$ , etc.
- Apprentissage par renforcement
  - l'algorithme, lors de l'usage de l'hypothèse pour une valeur précise  $h(a)$ , ne reçoit qu'une indication d'adéquation de  $h(a)$  (récompense reward),
- Apprentissage non-supervisé
  - l'algorithme ne reçoit aucune indication au sujet des valeurs correctes.

# Arbre de décision



# Apprentissage supervisé pour construire un Arbre de décision

- On reçoit en entrée un certain nombre de décisions correcte, par exemple
  - (Météo=bt,Saison=e,Distance=g,Décision=Auto)
- L'algorithme construit un arbre de décision qui pourra être utilisé pour trancher dans des cas futurs.



# L'algorithme C4.5<sup>1</sup>

- Choisir le paramètre le plus “crucial” et décomposer les données d’entraînement selon les valeurs de ce paramètre.
  - crucial = principalement celui qui offre le meilleur *gain d’information*, selon la théorie de Shannon,
  - avec une compensation pour ne pas favoriser indûment les attributs qui branchent énormément.
- Recommencer la même procédure pour construire les sous-arbres.
- Arrêter à un certain niveau et prendre la décision majoritaire.

---

1. Quinlan, J. Ross ; C4.5 : Programs for Machine Learning ;  
Morgan Kaufmann ; 1993

# Méthodologie

- De façon générale la méthodologie en apprentissage supervisé est :
  - d'obtenir un *jeu de données* (étiquetées),
  - le diviser en ensemble d'*entraînement* et de *test*,
  - faire l'apprentissage sur le jeu d'entraînement et tester la qualité des réponses sur l'ensemble de test.
- Les tests ne sont pas biaisés car l'ensemble de test est indépendant de l'ensemble d'entraînement.
- L'objectif est de *généraliser* et non de performer sur l'ensemble d'entraînement.

# Conclusion

- Les arbres de décision offrent une manière concise de représenter un certain type de connaissance.
- L'apprentissage supervisé permet de construire un arbre, même quand la connaissance est changeante ou difficile à décrire manuellement.
- En fait, la construction de l'arbre par l'algorithme d'apprentissage permet d'extraire une connaissance qui est implicite dans les données d'entraînement. Ceci permet d'étendre la portée de cet ensemble au delà des cas spécifiques qu'il contient. De plus, l'arbre lui-même peut être analysé pour déterminer une justification aux décisions prises.