Notes de cours
OPT6 : Apprentissage avancé

Adrien Pavao

Novembre 2017

1 Introduction

L'apprentissage automatique (machine learning) s'appuie sur des bases théoriques. Ce domaine évolue vite et on y trouve des théories et techniques avancées.

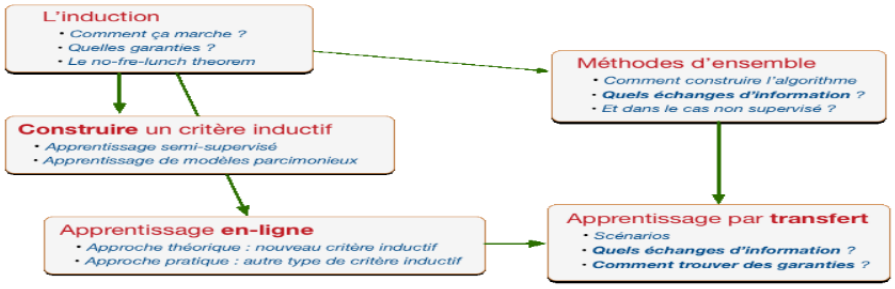

Figure 1: Les points qui seront abordés dans ce cours.

1.1 Quelques notions

- Vladimir Vapnik a posé les bases théoriques de l'apprentissage statistique[1].
- Apprentissage PAC : Probably Approximately Correct[2].
- **No-free-lunch theorem :** Il n'existe pas de méthode d'apprentissage qui soit meilleure qui les autres. Même performance que l'aléatoire sur l'ensemble des problèmes possibles. Si une méthode est efficace sur un problème elle sera mauvaise sur un autre. L'intégrale sur tous les problèmes est toujours la même, quelque soit la méthode. Idem pour les mesures (de distance par exemple). Les méthodes sont adaptées à des **classes** de problèmes. **Tous les algorithmes inductifs se valent.**

[1] https://en.wikipedia.org/wiki/Statistical_learning_theory
[2] https://en.wikipedia.org/wiki/Probably_approximately_correct_learning

- **Adversarial learning :** L'intersection entre le machine learning et la sécurité informatique.
- **Régularisation :** Un processus consistant à ajouter de l'information à un problème pour éviter le surapprentissage. Cette information prend généralement la forme d'une pénalité envers la complexité du modèle. On peut relier cette méthode au principe du **rasoir d'Occam**. D'un point de vue bayésien, l'utilisation de la régularisation revient à imposer une distribution a priori sur les paramètres du modèle.

  Une méthode généralement utilisée est de pénaliser les valeurs extrêmes des paramètres, qui correspondent souvent à un surapprentissage. Pour cela, on va utiliser une norme sur ces paramètres, que l'on va ajouter à la fonction qu'on cherche à minimiser. Les normes les plus couramment employées pour cela sont L_1 et L_2.
- Différentes normes permettent de calculer la taille totale d'un vecteur ou d'une matrice dans un espace vectoriel. Quelques exemples courants :

  Norme L_0 :
  $$||x||_0 = \sqrt[0]{\sum_i x_i^0}$$
  Norme L_1 :
  $$||x||_1 = \sum_i |x_i|$$
  Norme L_2 :
  $$||x||_2 = \sqrt{\sum_i x_i^2}$$
- **Distance de Manhattan :** La distance entre deux points lorsque les déplacements sont faits selon un réseau ou un quadrillage.

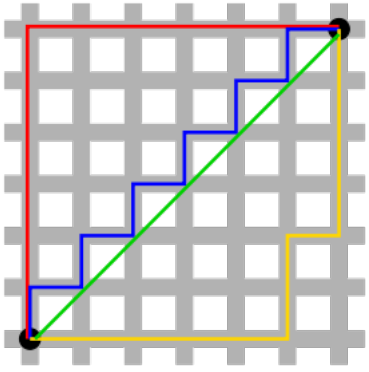

Figure 2: Distance de Manhattan (chemins rouge, jaune et bleu) contre distance euclidienne en vert.

Le nom de cette distance fait sans doute référence à l'architecture en quadrillage du quartier de Manhattan, à New York.

- Référence d'un livre intéressant : *Hofstadter - "Godel, Escher, Bach".*
- **Complexité de Kolmogorov :** Une fonction permettant de quantifier la taille du plus petit algorithme nécessaire pour engendrer un nombre ou une suite quelconque de caractères. Cette quantité peut être vue comme une évaluation d'une forme de complexité de cette suite de caractères.
- Effets de séquences : Une séquence d'information influe sur le résultat de l'induction (à préciser).
- L'algorithme d'apprentissage du perceptron converge si les données sont linéairement séparables. Cette convergence est en nombre fini d'étapes et ce nombre est indépendant du nombres d'exemples, de la distribution des exemples, et quasiment pas de la dimension de l'espace d'entrée. Il dépend de la marge de séparation des nuages de points et du diamètre de la boule (à préciser).
- **L'espace des versions :** Toutes les hypothèses correctes (qui ne font pas d'erreurs) d'après les données d'apprentissage.

2 Induction

On déduit à partir d'un échantillon.

1. Espace d'hypothèse H.
2. Critere inductif $S \times h \in H \rightarrow$ Score $\in \mathbb{R}$. Par exemple minimiser le taux d'erreur.
3. Methode d'exploration de H.

On ne dispose pas de théorie bien établie de l'induction (exemple Fibonacci).

2.1 Différents types d'apprentissage

- **Descriptif :** Non supervisé. On cherche des régularités dans les données. On ne souhaite pas extrapoler, on ne s'interesse qu'à l'échantillon de données dont on dispose. La "matière noire" de l'apprentissage.
- **Prédictif :** Supervisé. L'échantillon de données sert à apprendre une hypothèse sur les données pour prédire ensuite sur de nouvelles données. On cherche des corrélations entre les données.
- **Prescriptif :** On cherche des **causalités**. Il s'agit d'une tâche difficile.

Les frontières entre ces types d'apprentissage peuvent être floues, on peut par exemple commencer par une description des données pour les comprendre avant de faire un modèle prédictif.

2.2 Différentes méthodes d'apprentissage

- Méthodes d'ensemble (clustering).
- Supervisé.
- L'induction :
  - Apprentissage semi supervisé
  - Apprentissage de modèles parcimonieux (peu de paramètres).
- Apprentissage en ligne (online learning) : Les données arrivent en cours de route.
- **Apprentissage par transfert :** Cela vise à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles. On peut le voir comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes.

2.3 Types de biais

L'induction necessite d'être biaisé, d'avoir une preference pour certaines hypothèses, de ne pas connaitre toutes les fonctions possibles. Un biais classique et très humain est la préférence des hypothèses les plus simples (rasoir d'Ockham).

- **Biais de représentation (déclaratif) :** On ne peut représenter qu'un petit nombre de fonctions possibles. Le langage dans lequel on exprime le problème ne me permet pas de tout représenter.
- **Biais de recherche (procédural) :** La procédure de recherche avantage certaines hypothèses. Par exemple on reste proche de notre point initial de recherche dans l'espace des fonctions possibles.

Souvent un peu des deux.

2.4 Risques

On distingue deux types de risques :

- **Risque réel** (ce que l'on veut minimiser). La performance à venir si j'utilise l'hypothèse h. Cela prend la forme d'une espérance de coût d'usage de h. Le risque réel (loss function) :

$$R(h) = \int_X^Y l(h(x), y)p(x, y)\mathrm{d}x\mathrm{d}y$$

La meilleure hypothèse est $k\star = ArgminR(h)$ avec $h \in H$. En d'autres termes, on cherche l'hypothèse qui minimise le risque réel. l est la métrique d'évaluation et p la loi de probabilité hypothétique ayant générée la distribution des données.

- **Risque empirique**. Puisque l'on cherche la loi de probabilité, on ne la connait pas et on ne peut donc pas s'en servir pour calculer le risque, en pratique. On calcul donc le risque empirique :

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} l(h(x_i), y_i)$$

  C'est donc la moyenne des distances entre les prédictions avec h et les valeurs réelles des données. $\hat{h}$ est l'hypothèse minimisant le risque empirique.

On cherche les liens entre toutes ces notions.

Pour Vladimir Vapnik

Plutôt que de minimiser le risque empirique :

- Pour une hypothèse h : $P_{S \sim P_{XY}}(\hat{R}(h) = 0 \wedge R(h) > \epsilon)$

On ne veux pas d'une hypothèse qui paraisse bonne mais qui soit en réalité mauvaise. Le principe de minimisation du risque empirique n'est sain que s'il y a des contraintes sur l'espace des hypothèses.

(formule et principe de consistance universelle ?)

3 Transduction

On ne s'interesse qu'à un point. Le nombre de données requises pour tirer une conclusion n'est plus influencé par le nombre de dimension. On parle de **transductive learning**.

- Transduction : Cas d'induction limite (une question).
- Analogie : Cas de transduction limite (un exemple).

4 Analyse du perceptron

1. On suppose que l'ensemble des hypothèses H est fini.
2. **Cas réalisable :** $\exists h \in H tq R(h) = 0$ ($\neq 0 \rightarrow$ cas non-réalisable.)
3. S est supposé tiré *i.i.d.*