# Introduction au machine learning

Mines Fontainebleau of doom

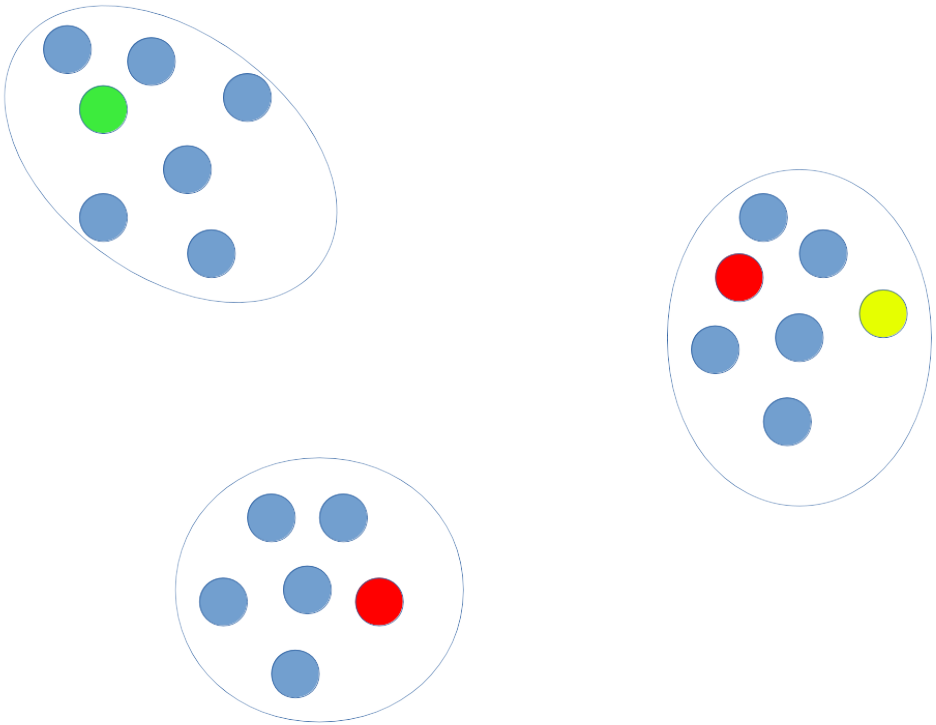
27 janvier 2016

# Définition ?

- Conception et étude d'algorithmes qui peuvent apprendre et faire des prédictions sur des *données*.
- Essentiellement deux types de machine learning :
  - l'apprentissage *non supervisé* (quelle est la structure de mes données ? Est-ce que je peux les regrouper en classes ? Est-ce que je peux simplifier sans perdre (trop) d'information ?)
  - l'apprentissage *supervisé* : étant donné un "grand nombre" de données $X_i$ et leurs étiquettes respectives $Y_i$, est-ce que je peux trouver une fonction $f$ telle que $f(X_i) \approx Y_i$ ? Et est-ce que pour une nouvelle observation $(X, Y)$ j'aurai bien $f(X) \approx Y$ ?
- Il n'est pas interdit de combiner les deux ...

# Apprentissage non supervisé

- L'exemple le plus évident est le *clustering* :



- Notons qu'on regroupe des données qui se ressemblent entre elles, *sans les étiqueter*.

# Apprentissage supervisé

Cas général : on dispose d'une base de données $(X_i)_{i\in[1;N]}$ et des annotations (ou "étiquettes", "labels", "ground truth" ...) correspondantes $Y_i$ ; on cherche une fonction $f$ telle que pour tout i : $f(X_i) \approx Y_i$.
On distingue deux approches *a priori* différentes selon le domaine des valeurs de $Y$ :

- Lorsque $Y$ est un scalaire ($Y$ prend ses valeurs dans $\mathbb{R}$ voire $\mathbb{C}$), on parle de *régression*.
- Lorsque $Y$ prend ses valeurs dans un ensemble non ordonné, par exemple {bleu, rouge, vert} ou {sain, malade}, on parle plutôt de *classification*.

Notons que pour faire de la classification binaire (deux classes) on peut souvent se ramener à un problème de régression et fixer un seuil ; c'est typiquement le cas de la régression logistique.

# Exemples d'apprentissage supervisé

**Classification :**

| X (prédicteurs) | Y |
|---|---|
| données patient (âge, poids, taille ...) | {sain, malade} |
| son | {guitare, piano, violon, ...} |
| image | {photo, dessin, ...} |
| e-mail | {spam, ham} |

**Régression :**

| X (prédicteurs) | Y |
|---|---|
| robot (données capteurs) | angle |
| " | vitesse |
| infos trafic | probabilité d'embouteillage |
| prix au cours des derniers jours | prix le lendemain |

# Fonction d'erreur

Qu'est-ce qu'on entend par $f(X_i) \approx Y_i$ ?

- Typiquement, on se donne une fonction d'erreur $err(f(X_i), Y_i)$
- On cherche alors à minimiser :

$$\frac{1}{N} \sum_{i=1}^{N} err(f(X_i), Y_i)$$

# Fonction d'erreur

Qu'est-ce qu'on entend par $f(X_i) \approx Y_i$ ?

- Typiquement, on se donne une fonction d'erreur $err(f(X_i), Y_i)$
- On cherche alors à minimiser :

$$\frac{1}{N} \sum_{i=1}^{N} err(f(X_i), Y_i)$$

- ... ou pas.
- On cherche en fait à minimiser $\mathbb{E}\left[err(f(X), Y)\right]$

# Formalisation

- On se place dans le cas où les $X_i \in \mathbb{R}^p$ et $Y \in \mathbb{R}$.
- On cherche la fonction $f$ sous la forme $f_\theta(x) = \theta_0 + \theta^T x$

Autrement dit, si on écrit $X_i = (X_i^{(1)}, X_i^{(2)}, ...X_i^{(p)})^T$, on cherche $f_\theta$ de manière à ce que :

$$f_\theta(x) = \theta_0 + \theta_1 x^{(1)} + \theta_2 x^{(2)} + ... + \theta_p x^{(p)}$$

- On choisit de plus $err(f(x), y) = (f(x) - y)^2$

# Solution analytique

Si on veut minimiser l'erreur empirique :

$$\frac{1}{N}\sum_{i=1}^{N}(f_\theta(X_i) - Y_i)^2$$

la solution est donnée par :

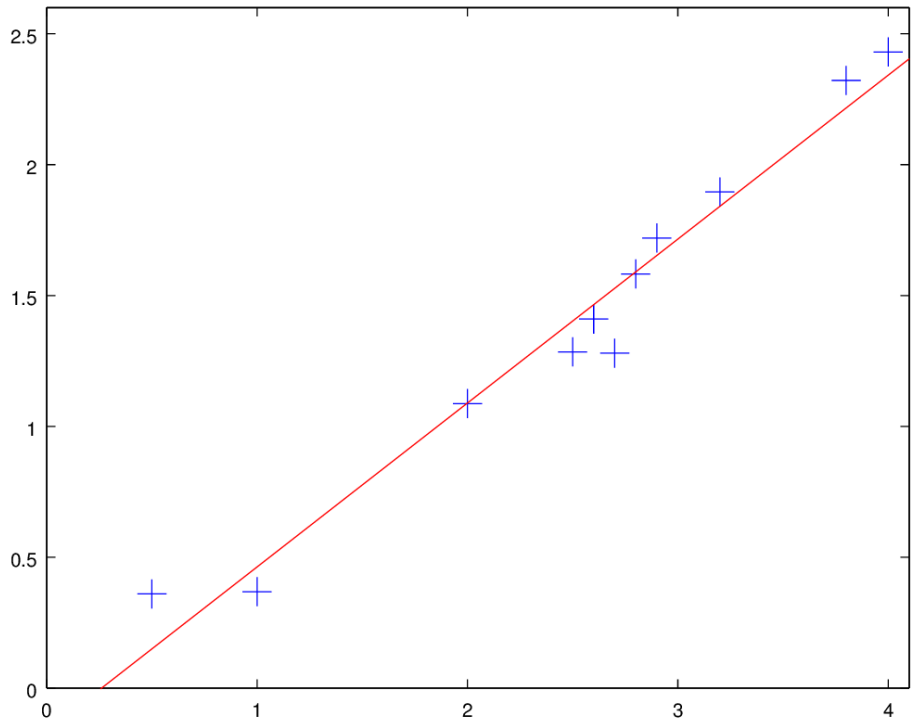
$$\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

où $\mathbf{X}$ est la matrice dont la i-ième ligne est $(1, X_i^{(1)}, X_i^{(2)}...X_i^{(p)})$

... mais est-ce qu'on a intérêt à minimiser l'erreur empirique ?

# Présentation du problème

En dimension 1, on peut visualiser simplement la base de données $(x_i, y_i)$ et le modèle linéaire :




RMS = 0.39729

# Régression polynômiale

- Si on cherche $f$ comme un polynôme de degré (au plus) $d$ :

$$f_d(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + ... + \theta_d x^d$$

# Régression polynômiale

- Si on cherche $f$ comme un polynôme de degré (au plus) $d$ :

$$f_d(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + ... + \theta_d x^d$$

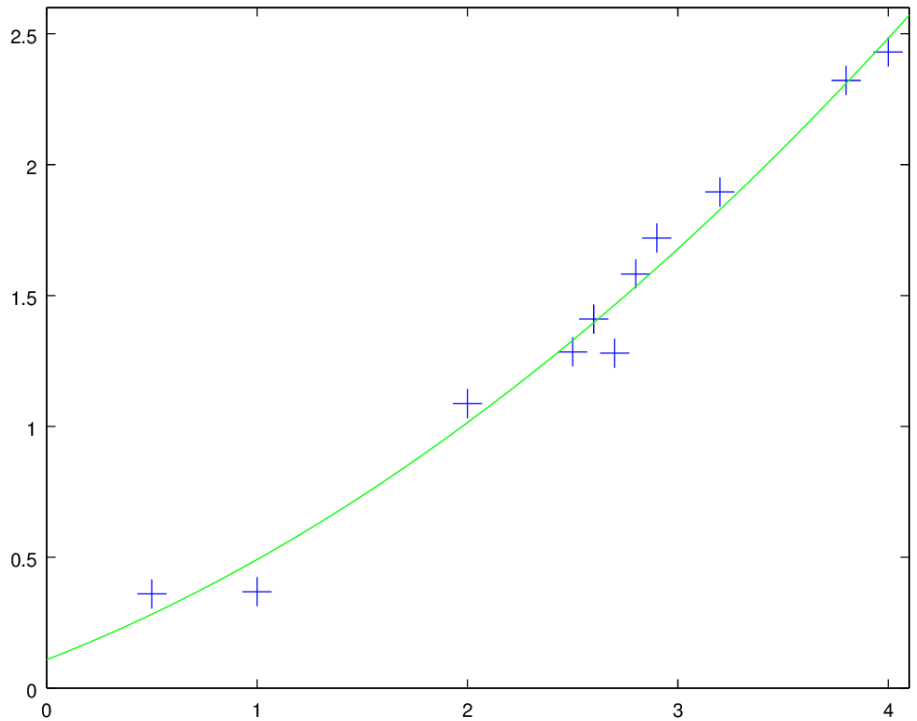
- On peut en fait se ramener au cas de la régression linéaire en dimension $d$ par la transformation :

$$x \to (x, x^2, ..., x^d)$$

- La régression polynômiale peut donc en fait être vue comme une régression linéaire (en dimension supérieure).

# Régression polynômiale

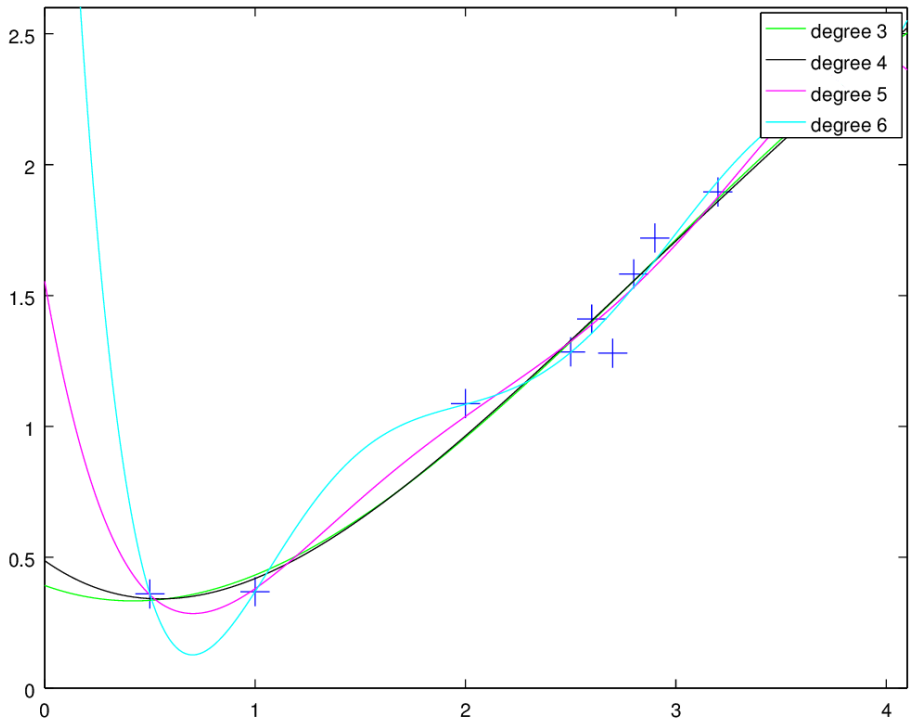
Régression polynômiale d'ordre 2 :




RMS = 0.29227

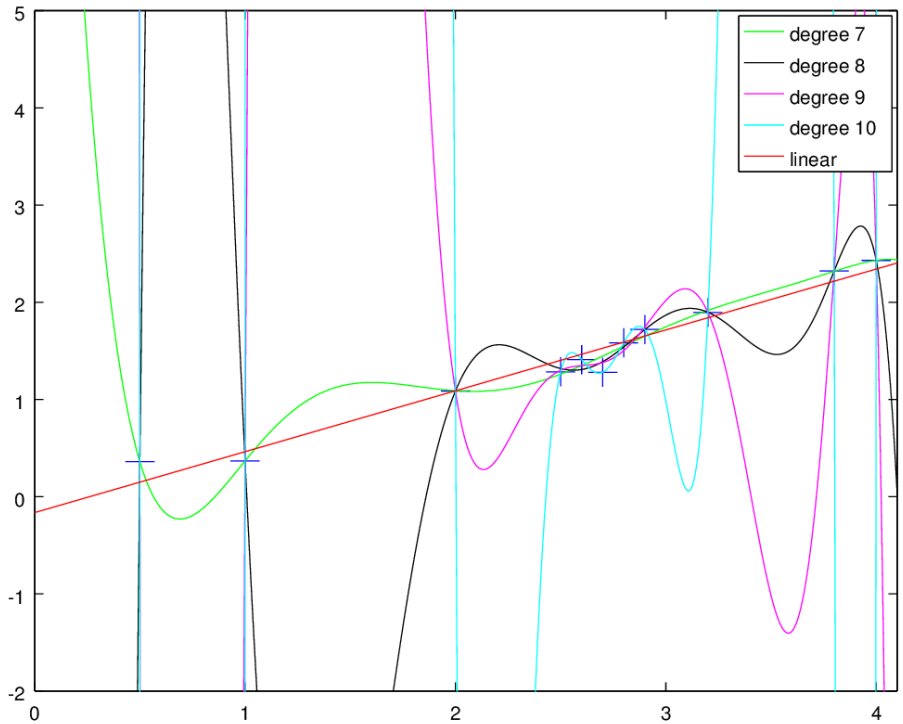
# Régression polynômiale

Régression polynômiale d'ordre supérieur :



# Régression polynômiale

Régression polynômiale d'ordre supérieur :



# Retour sur l'erreur empirique

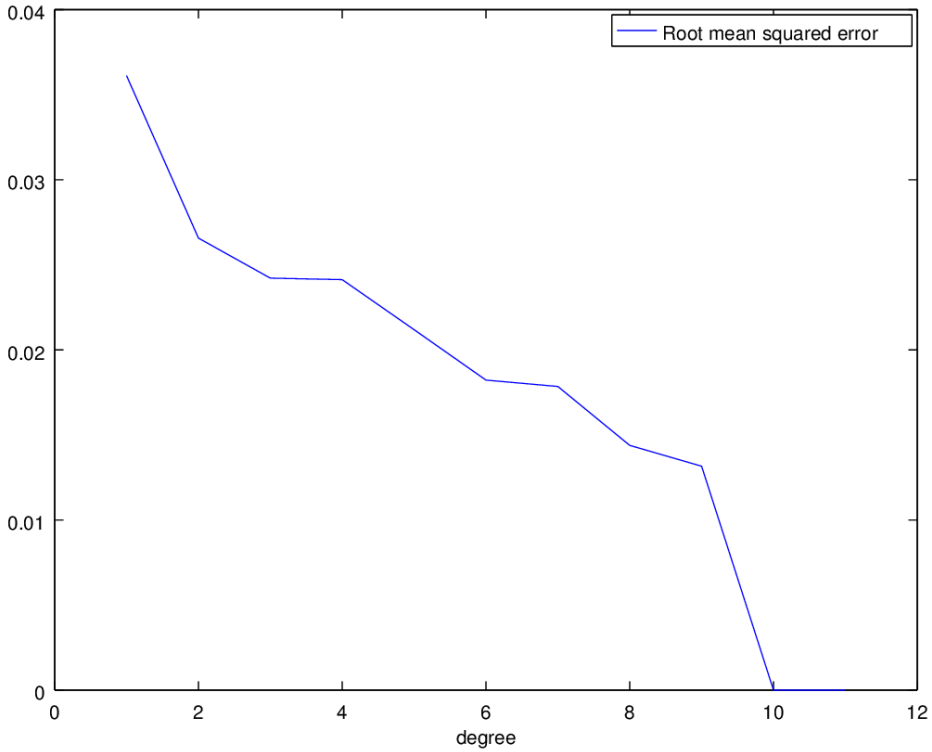



Figure : Erreur moyenne en fonction du degré du polynôme utilisé pour la régression

# Bias/variance tradeoff

Si on suppose qu'il existe une fonction $f_{opt}$ telle que $Y = f_{opt}(X) + \epsilon$ où $\epsilon$ est une variable aléatoire de moyenne nulle et de variance $\sigma^2$, et qu'on note $\hat{f}$ la fonction apprise sur une première base de données, l'erreur attendue sur une *nouvelle* base $\mathbf{X}_{val}$ peut s'écrire :

$$\mathbb{E}\left[(Y - \hat{f}(X))^2\right] = Bias^2 + Var + \sigma^2$$

avec :

- $Bias = \mathbb{E}\left[\hat{f}(X)\right] - f(X)$
- $Var = \mathbb{E}\left[\hat{f}(X) - \mathbb{E}\left[\hat{f}(X)\right]^2\right]$

# Bias/variance tradeoff

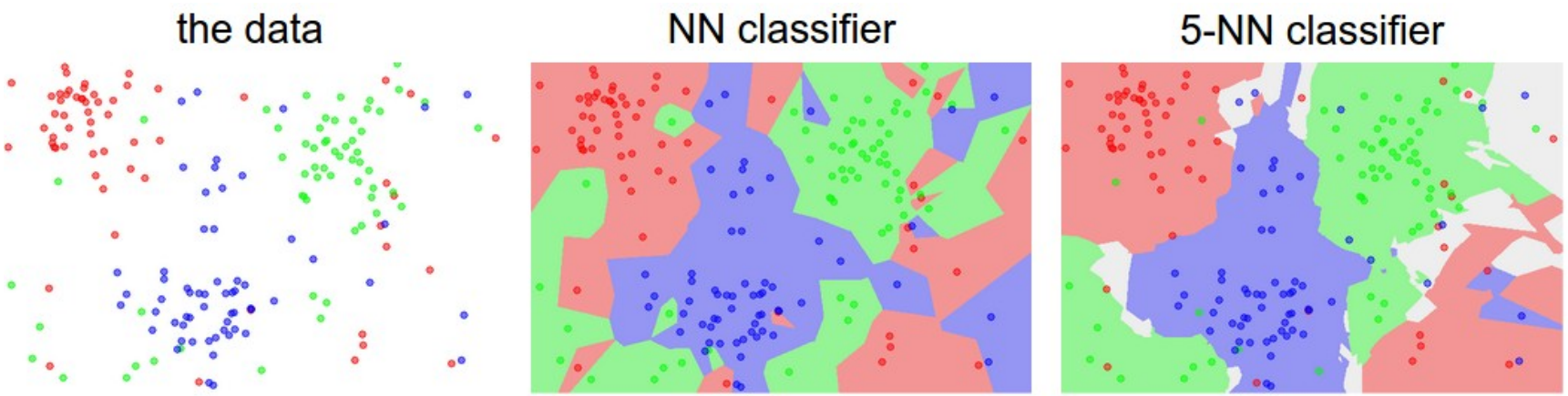
- Le biais *Bias* est d'autant plus petit que le modèle "colle" à la base d'apprentissage (quitte à apprendre du bruit et à prédire n'importe quoi en dehors des points de la base).
- La variance est une sorte de mesure de stabilité du modèle ; elle mesure à quel point deux modèles appris sur des bases similaires (mais distinctes) sont proches en moyenne.

!!! Se méfier donc des articles qui annoncent un taux de bonne classification de 98% ou une erreur moyenne ridiculement petite ; ça ne sert à rien de minimiser le biais si la variance est élevée.
Ce phénomène s'appelle le *surapprentissage* ou *overfit*, et constitue une des erreurs les plus courantes en machine learning.

# Nearest neighbors

1-nearest neighbor :

- 1-NN : on attribue à un point le label de son plus proche voisin.
- k-NN : on attribue le label majoritaire parmi les k plus proches voisins.



# Où est le plus proche voisin ?

- Si les données d'entrée sont réparties uniformément dans le cube $[-1; 1]^p$, où est le point le plus proche de l'origine ?

# Où est le plus proche voisin ?

- Si les données d'entrée sont réparties uniformément dans le cube $[-1;1]^p$, où est le point le plus proche de l'origine ?
- Plus précisément, quelle est la probabilité d'avoir un point à une distance de moins de 1 de l'origine ?

# Où est le plus proche voisin ?

- Si les données d'entrée sont réparties uniformément dans le cube $[-1; 1]^p$, où est le point le plus proche de l'origine ?
- Plus précisément, quelle est la probabilité d'avoir un point à une distance de moins de 1 de l'origine ?
- Réponse : $\frac{V_p}{2^p}$

# Volume de la boule unité

On démontre facilement que :

$$V_{2k} = \frac{\pi^k}{k!}$$

$$V_{2k+1} = \frac{2^{k+1}\pi^k}{(2k+1)!!}$$

# So alone

La probabilité d'avoir un voisin à une distance de moins de 1 est donc :

$$p_{2k} = (\frac{\pi}{4})^k \frac{1}{k!}$$

$$p_{2k+1} = \frac{\pi^k}{2^k(2k+1)!!}$$