Apprentissage Statistique

Organisation

- 15h Cours (5×3)
- 15h Travaux pratiques (5×3): Python.

Intervenants

- Cours : Basarab MATEI
- Travaux pratiques :
  Kaoutar BENLAMINE, Nistor GROZAVU, Parisa RASTIN

Evaluation

- Examen écrit: 60%
- Activité travaux pratiques: 40%

Apprentissage Statistique

Organisation

- 15h Cours (5×3)
- 15h Travaux pratiques (5×3): Python.

Intervenants

- Cours : Basarab MATEI
- Travaux pratiques :
  Kaoutar BENLAMINE, Nistor GROZAVU, Parisa RASTIN

Evaluation

- Examen écrit: 60%
- Activité travaux pratiques: 40%

Apprentissage : qu'est-ce que c'est pour vous ?

Tentatives de définition

Arthur Samuel, 1959

Field of study that gives computers the ability to learn without being explicitly programmed

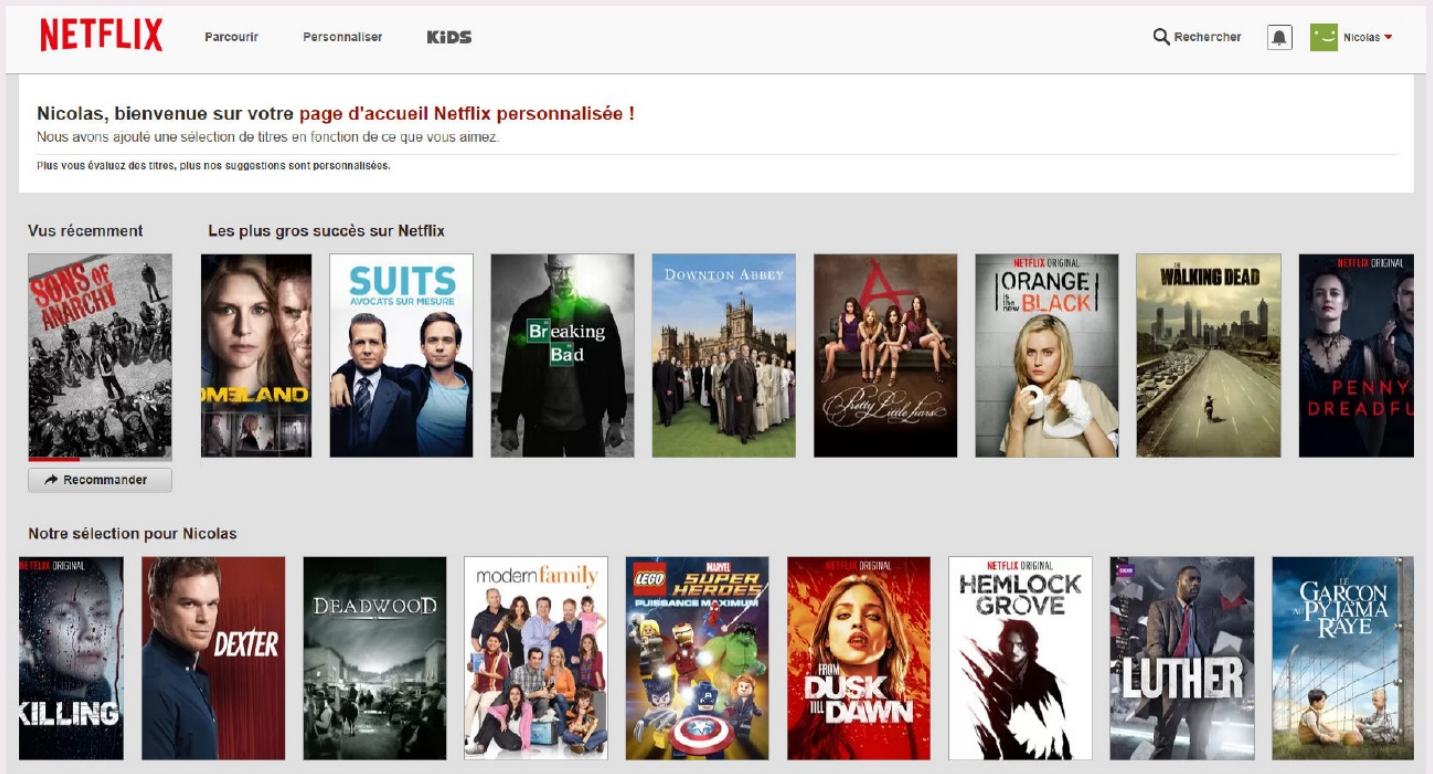
Tom Mitchell, 1998

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

Reconnaissance de chiffres manuscrits

Reconnaissance vocale - analyse du langage

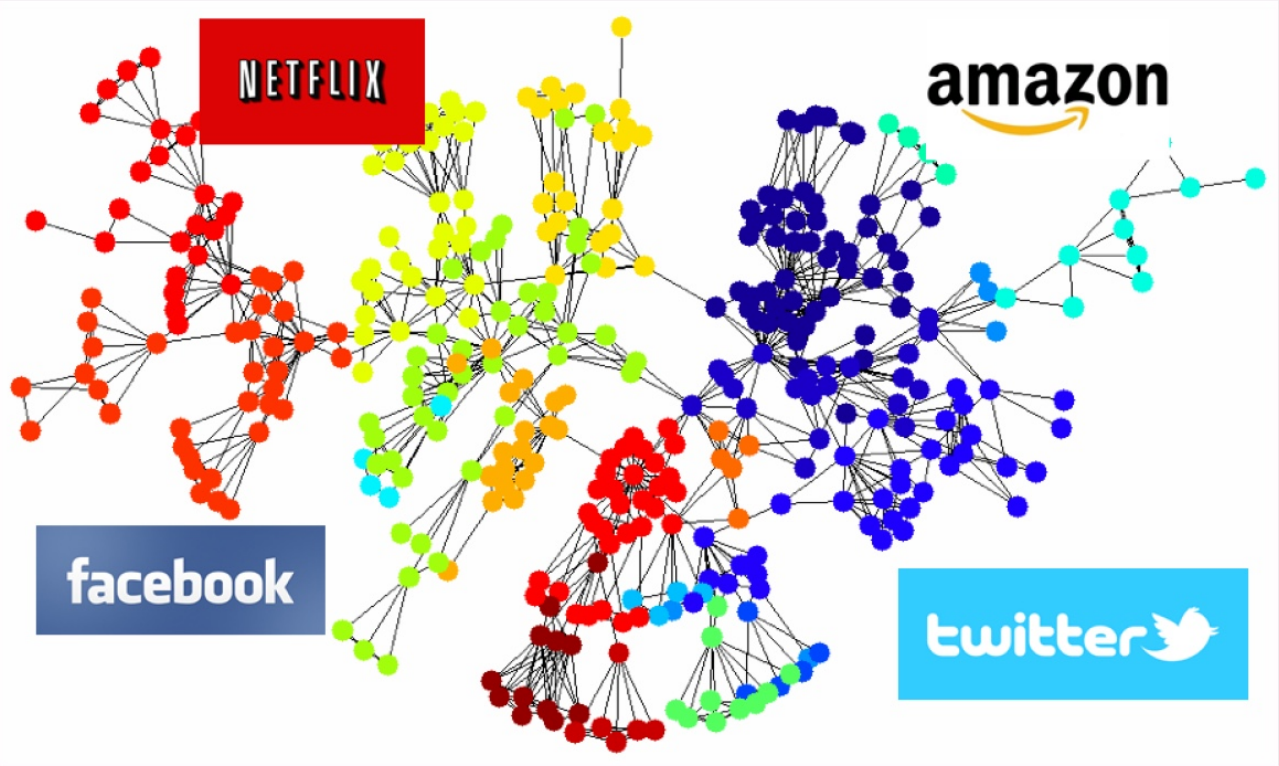
Systèmes de recommandation



Segmentation d'images

Interaction homme-robot

Recherche de communautés dans les réseaux sociaux



Google Car

Et encore ...

- **Robotique**
- Traduction automatique
- Jeux vidéo
- Détection de spam
- Imagerie médicale
- Bio-informatique
- Vidéo-surveillance
- Aide aux personnes
- e-Learning
- Economie
- Publicité
- Big data

Types d'apprentissage

Apprentissage Supervisé

- Apprendre des relations entre entrées et sorties ;
- Un oracle donne des exemples exprimant ces relations ;

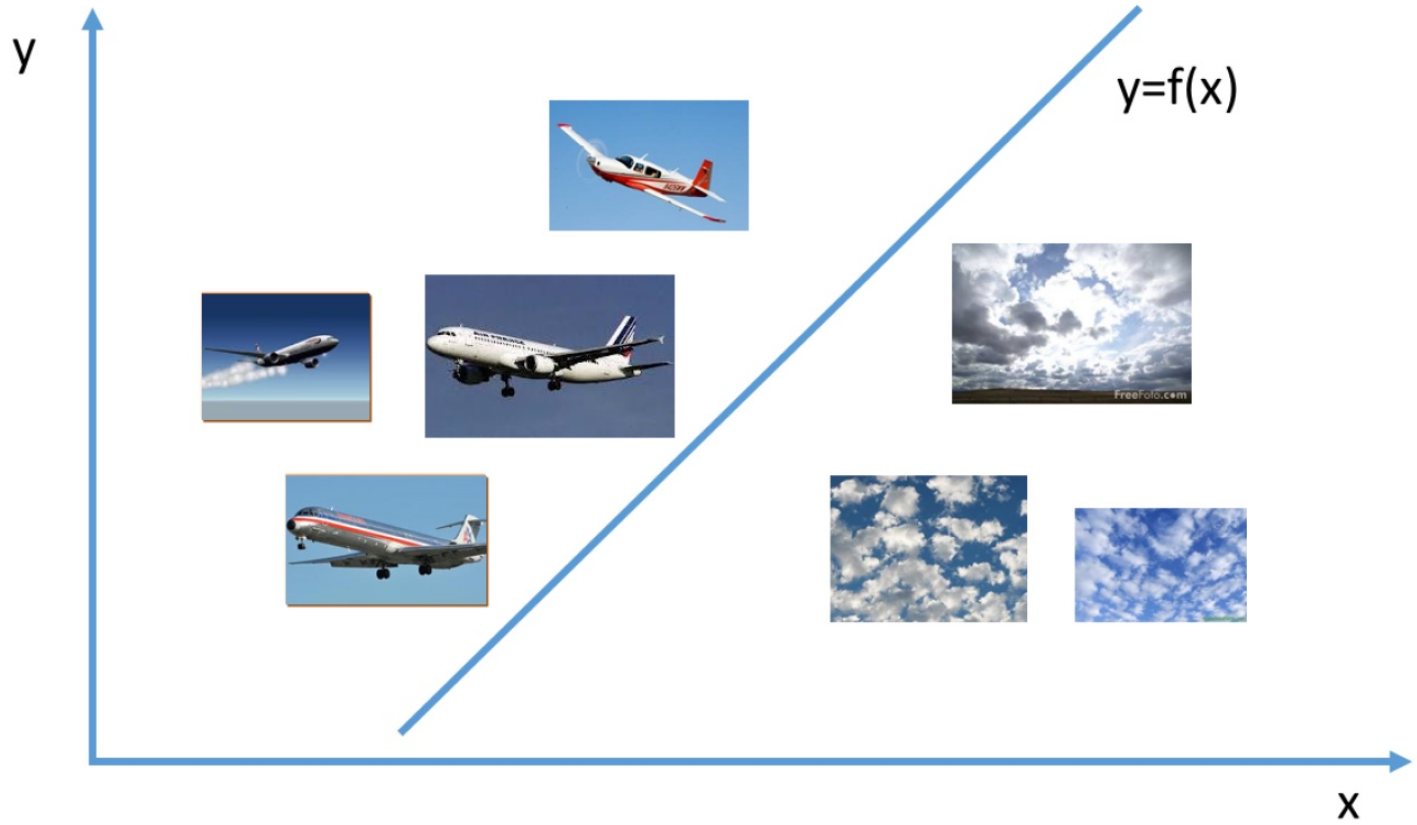
Apprentissage Non-Supervisé

- Apprendre une structure dans un ensemble de données ;
- Pas d'oracle ;

Apprentissage par Renforcement

- Apprendre à se comporter !
- Apprentissage en ligne
- Décisions séquentielles

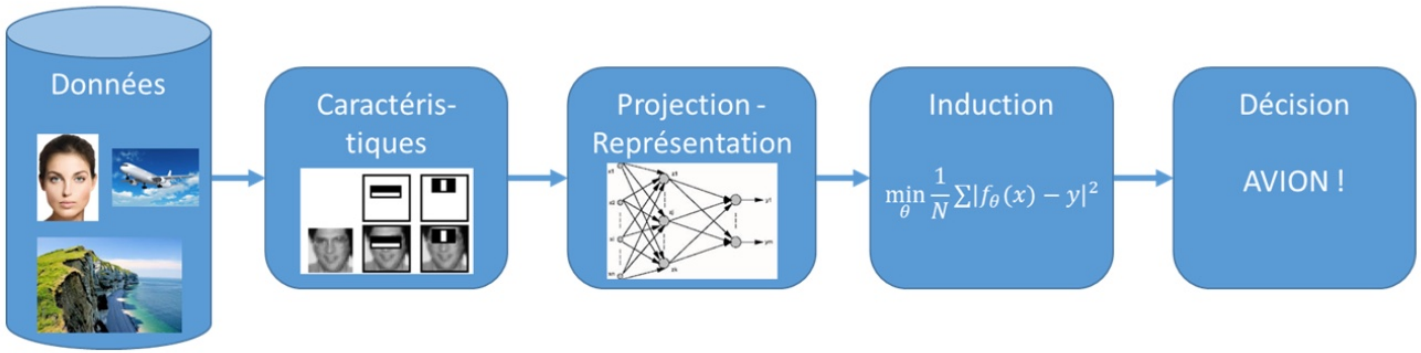
Classification



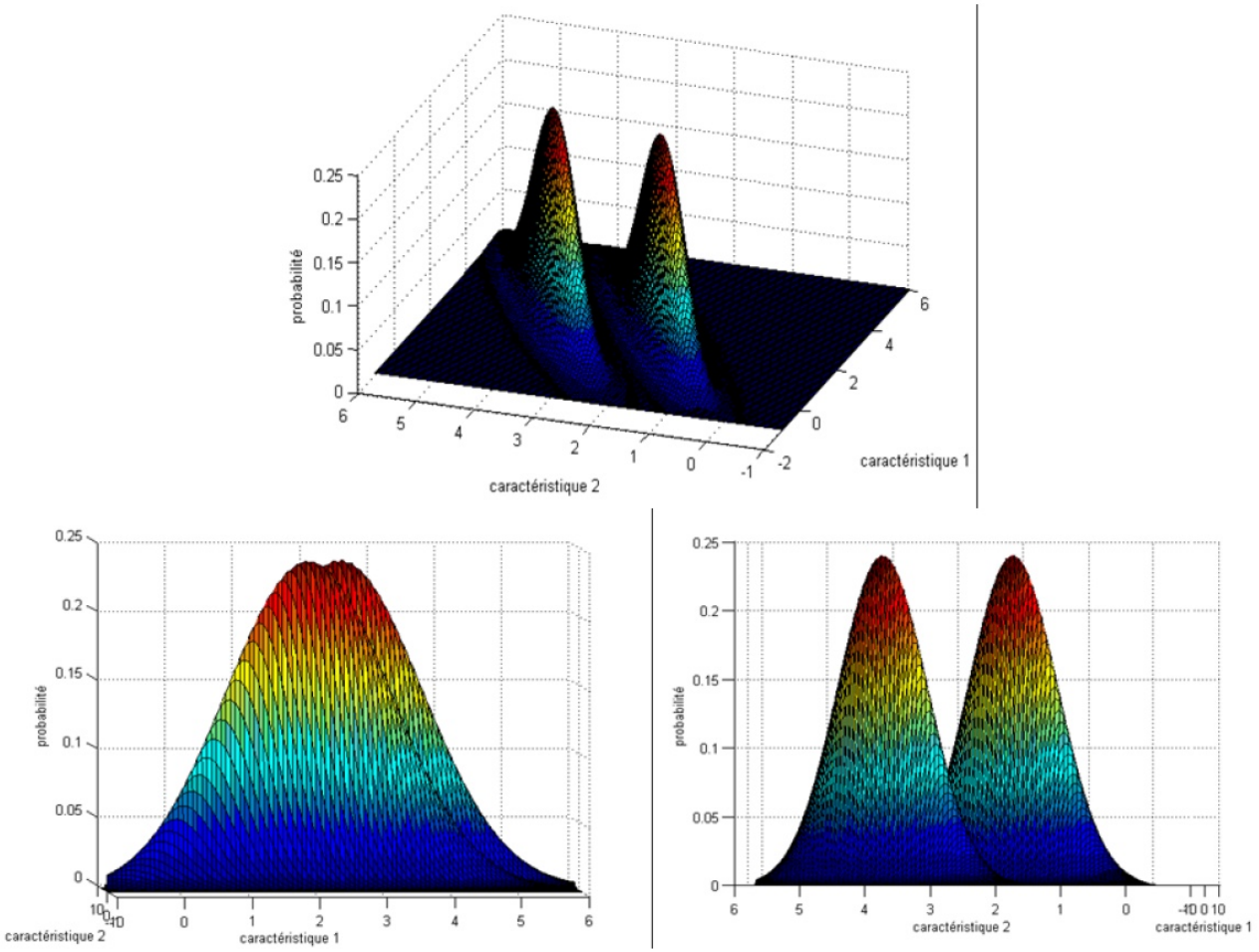
Régression


AS OF 11:33
PSE 4343.07 60.47
FIN 994.51 15.71
HDG 3424.05 54.95
IND 6984.78 110.43
PRO 1500.33 23.42
THE PHILIPPINE STOCK EXCHANGE, INC.
ALL INDEX VALUE
y
5372.83
5167.32
4961.81
09:30
10:00
10:32
11:31
x
y=f(x)
?

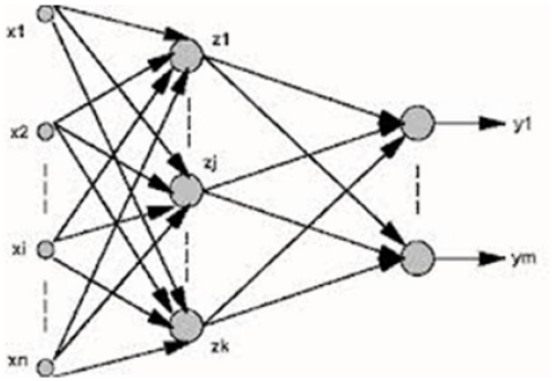
Chaîne de traitement

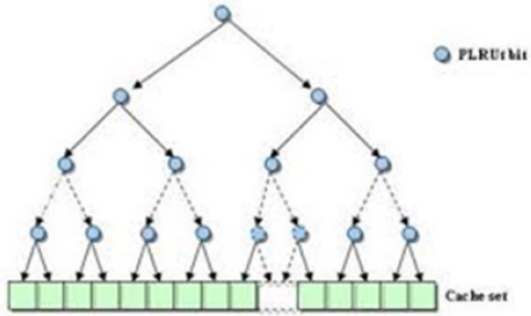


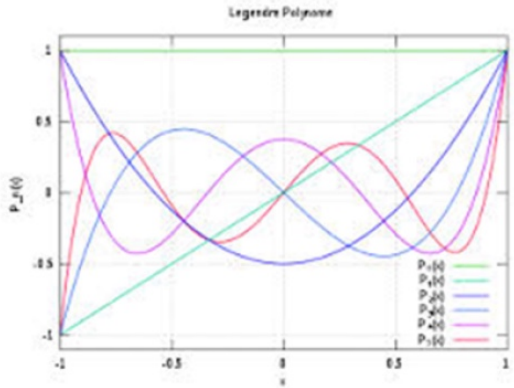
Caractéristiques

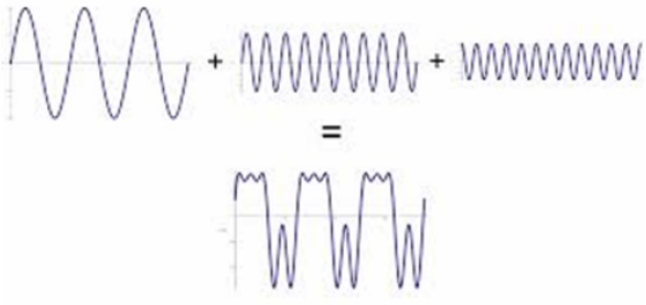


Représentation









Induction

Définition

Etant donné un ensemble $\mathcal{D}$ de données sous forme de couples $\{\mathbf{x}^{(i)}, y^{(i)} = f(x^{(i)})\}$ (avec souvent $\mathbf{x}^{(i)} \in \mathbb{R}^m$ et $y_i \in \mathbb{R}$), il s'agit de trouver une fonction $\hat{f}(x)$ qui permette d'expliquer et de prédire la relation entre des entrées quelconques ($\mathbf{x}$) et les sorties (y) correspondantes.

Méthodes paramétriques

Souvent, on se choisit une représentation paramétrique et on cherche les paramètres grâce aux données le problème devient : Etant donné une famille de fonctions f_θ et des données $\{\mathbf{x}^{(i)}, y^{(i)}\}$, trouver θ tel que $f_\theta(\mathbf{x}) \simeq y$ pour tout $\mathbf{x}$.

Exemple : prédire une position I

Vitesse, position et temps

- Equations du MRU : $p = p_0 + v\Delta t$
- Disons qu'on ne les connaît pas
- On mesure approximativement des positions et des temps

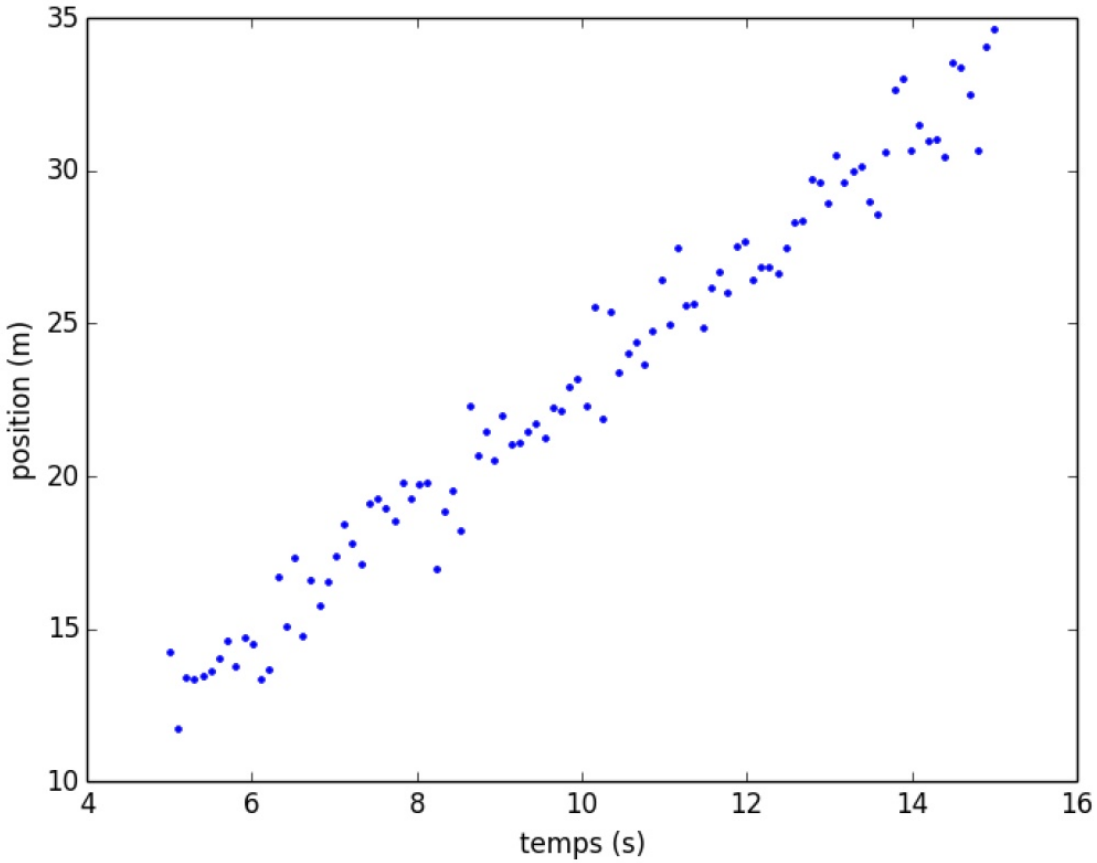
Méthodologie

- Quelles caractéristiques ?
- Quelle famille de fonctions paramétriques ?
- Quelle principe d'induction ?

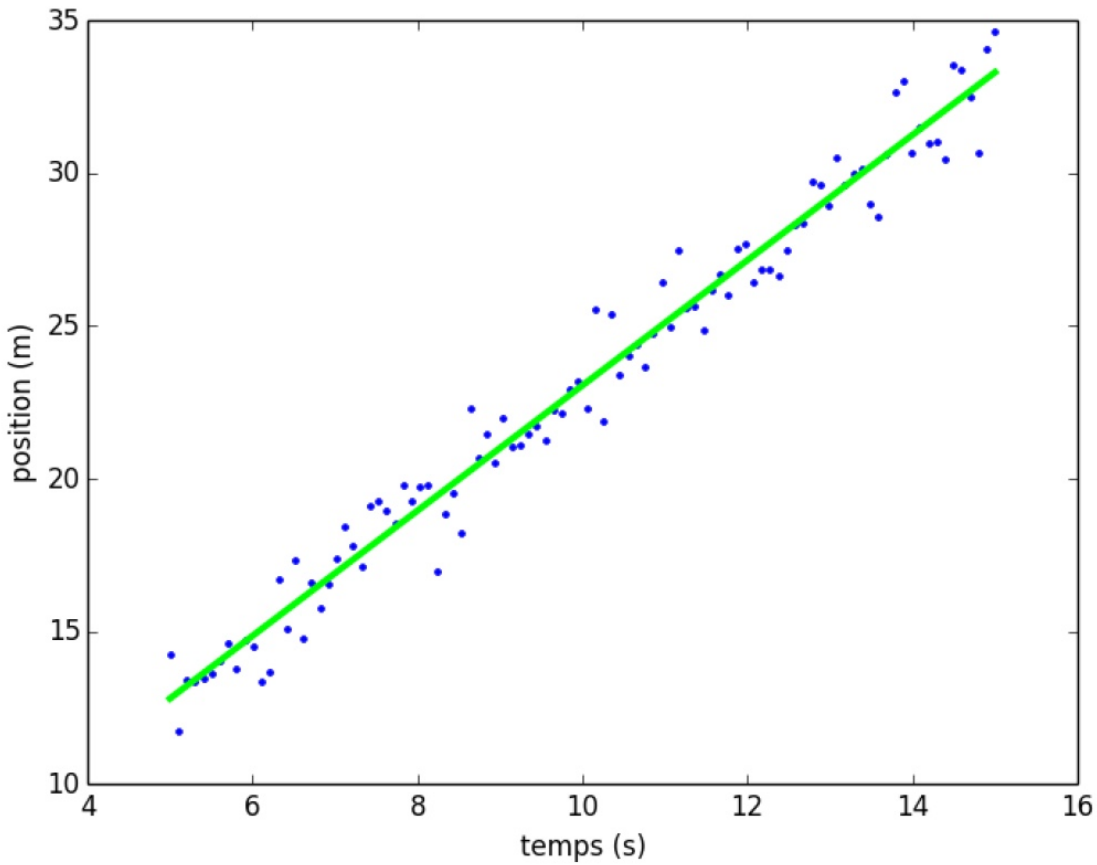
Caractéristiques ?

$$position = f(temps)$$

Exemple : prédire une position II



Exemple : prédire une position III



Exemple : prédire une position V

Induction

Minimisation de :

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$$

Exemple : prédire une position VI

Solution

On dérive et annule :

$$\nabla_\theta \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2 = 0$$

$$\theta = (\sum_i \mathbf{x}^{(i)} \mathbf{x}^{(i)T})^{-1} \sum_i \mathbf{x}^{(i)} y^{(i)}$$

Représentation matricielle

Definitions

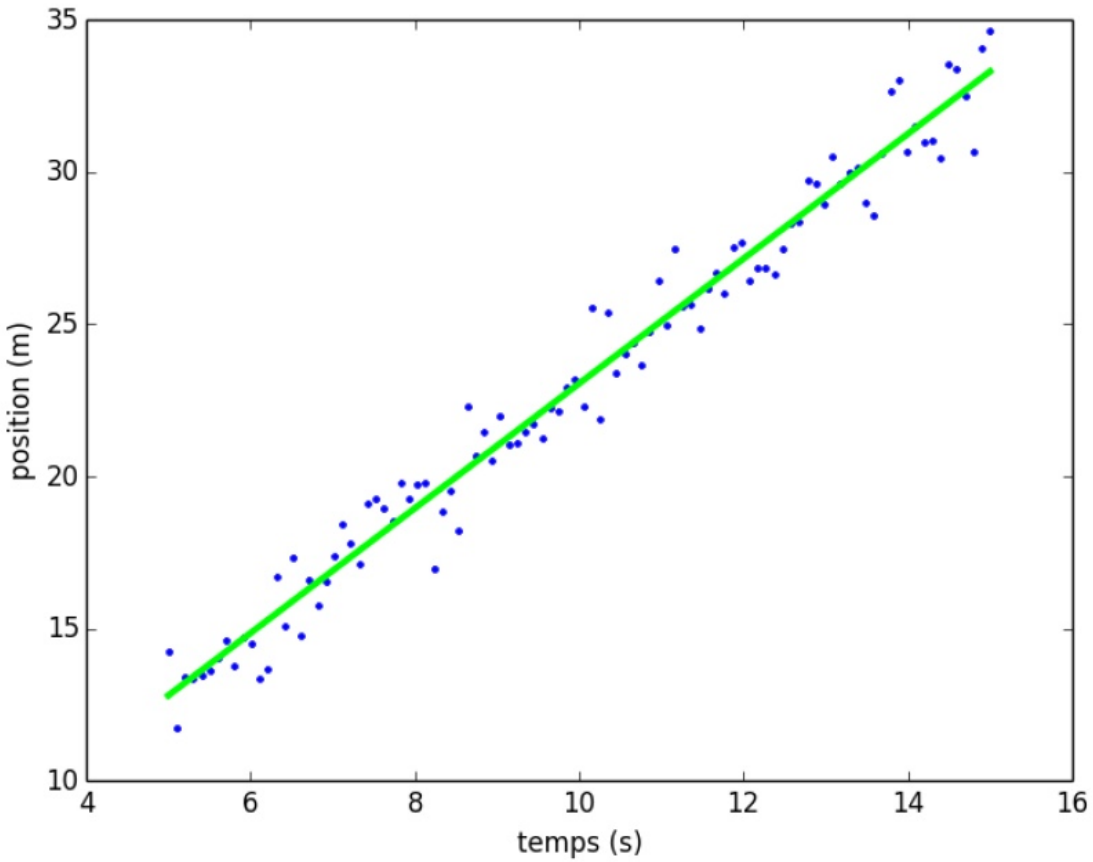
- X est la matrice dont la $i^{\text{ème}}$ colonne est $\mathbf{x}^{(i)}$
- Y est le vecteur dont la $i^{\text{ème}}$ composante est $y^{(i)}$

Solution matricielle

$$J(\theta) = \frac{1}{N}\|(Y - X^T\theta)\|^2$$

$$\theta = (XX^T)^{-1}(XY)$$

Résultat



Pourquoi ça marche ? I

Fonction convexe : définition

- Fonctions "*tournées*" vers le haut.
- Si on trace un segment $[A, B]$ tel que $A = f(a)$ et $B = f(b)$, ce segment est toujours au dessus de $f(x)$ sur l'intervalle $[a, b]$

Pourquoi ça marche ? II

Propriétés

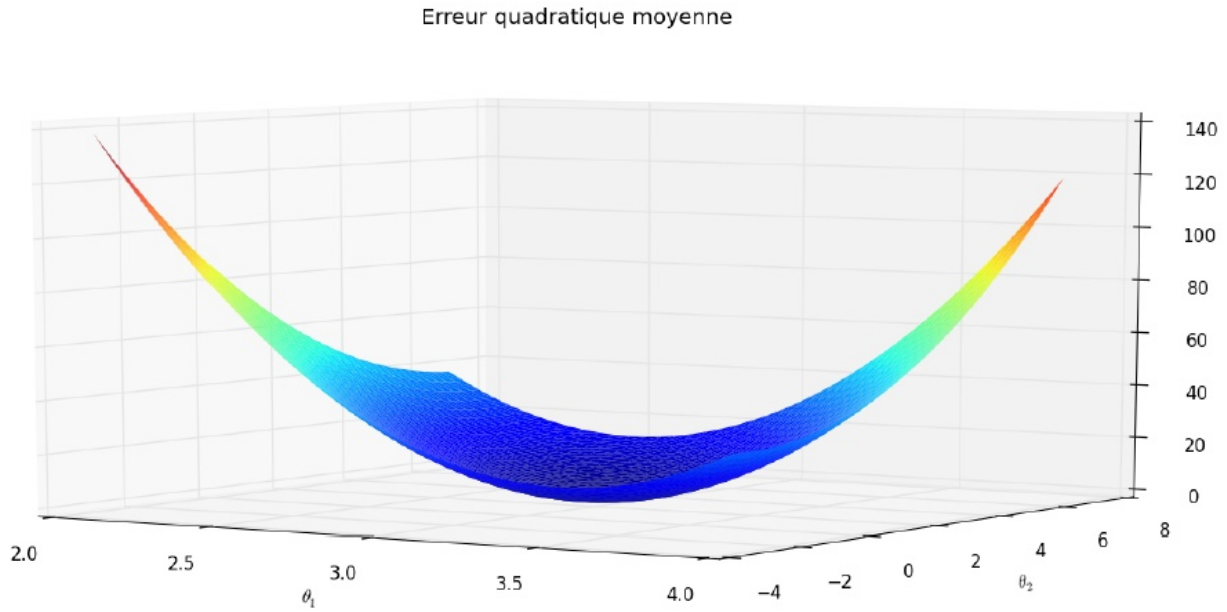
- Les fonctions convexes admettent (au moins) un minimum global
- Il existe une fonction affine qui minore une fonction convexe et elles coïncide au minimum
- les fonctions affines sont convexes
- $x \mapsto x^2$ est une fonction convexe
- La somme de 2 fonctions convexes est une fonction convexe

Pourquoi ça marche ? III

Conséquences

- $y^{(i)} - \sum_m \theta_m x_m^{(i)} = y^{(i)} - \theta^T \mathbf{x}^{(i)}$ est convexe (affine)
- $(y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$ est convexe (carré)
- $\sum_i (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$ est convexe (somme)
- $\frac{1}{N} \sum_i (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$ est convexe (transformation affine)

Pourquoi ça marche ? IV



L'erreur quadratique moyenne dans le cas du TP de la première séance.

Pourquoi ça marche ? V

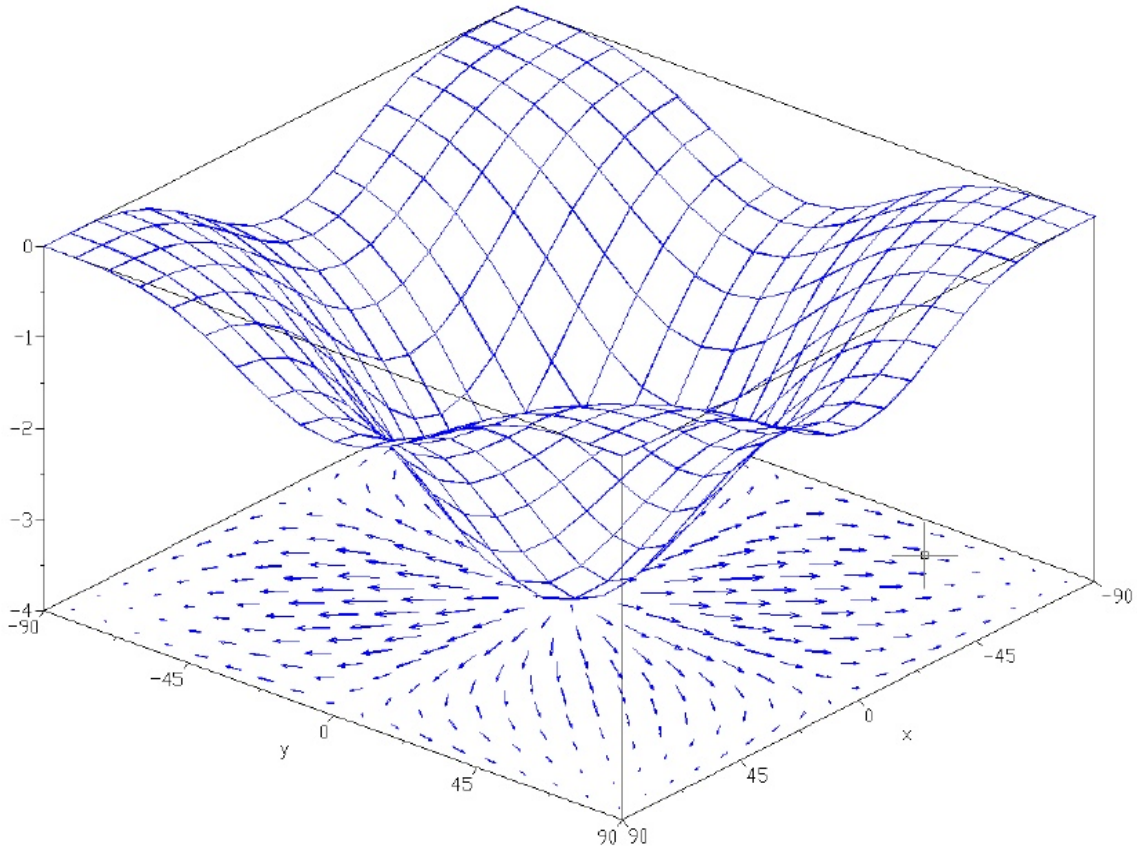
La dérivée

- $\frac{df(x)}{dx} = \lim_{dx \to 0} \frac{f(x) - f(x+dx)}{dx}$
- C'est donc la pente de la fonction en x
- Si la dérivée est nulle on a un extremum
- Dans le cas d'une fonction convexe, c'est un minimum

Le gradient

- Fonction de plusieurs variables (par ex. $f(\theta_1, \theta_2)$)
- $\nabla f(\theta_1, \ldots, \theta_n) = [\frac{\partial f(\theta_1, \ldots, \theta_n)}{\partial \theta_1}, \ldots, \frac{\partial f(\theta_1, \ldots, \theta_n)}{\partial \theta_n}]^T$
- Généralisation de la dérivée
- La fonction est minimum si $\nabla f(\theta_1, \ldots, \theta_n) = \vec{\mathbf{0}}$

Pourquoi ça marche ? VI



NB : La fonction est représentée en 3 mais le gradient est sur un plan 2D !

Pourquoi ça ne marche pas toujours ? I

Il faut inverser une matrice !

- Solution :

$$\theta = \left(\sum_i \mathbf{x}^{(i)} {\mathbf{x}^{(i)}}^T \right)^{-1} \sum_i \mathbf{x}^{(i)} y^{(i)}$$

$$\theta = (XX^T)^{-1}(XY)$$

- Il faut donc calculer $\left(\sum_i \mathbf{x}^{(i)} {\mathbf{x}^{(i)}}^T \right)^{-1}$ ou $(XX^T)^{-1}$

Pourquoi ça ne marche pas toujours ? II

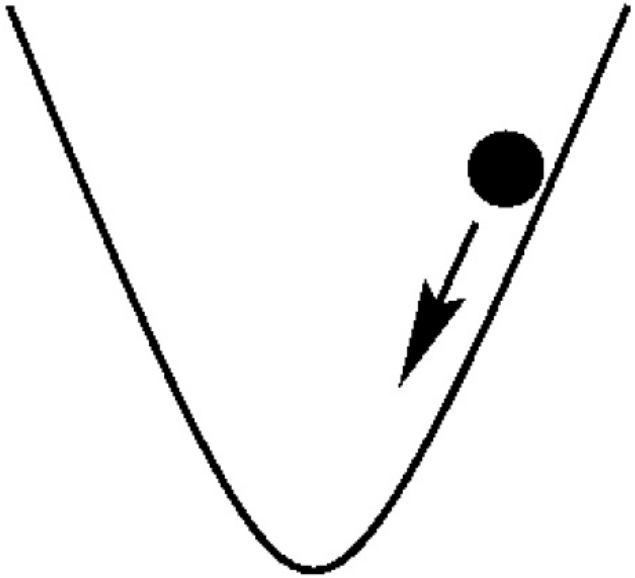
Inverse d'une Matrice

- $A = [n \times n]$ (carrée)
- $AA^{-1} = I$
- $A^{-1} \propto \frac{1}{\det A}$
- $\det A = 0$ si $rang(A) < n$
- A peut donc ne pas être inversible !

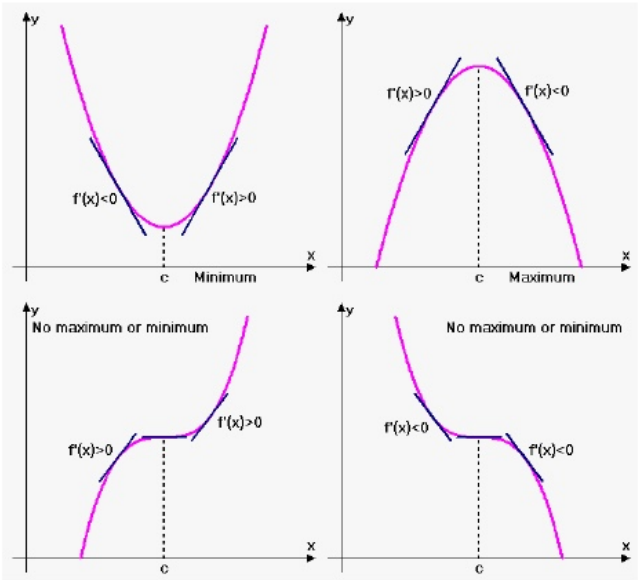
Régularisation ou Pseudo-inverse

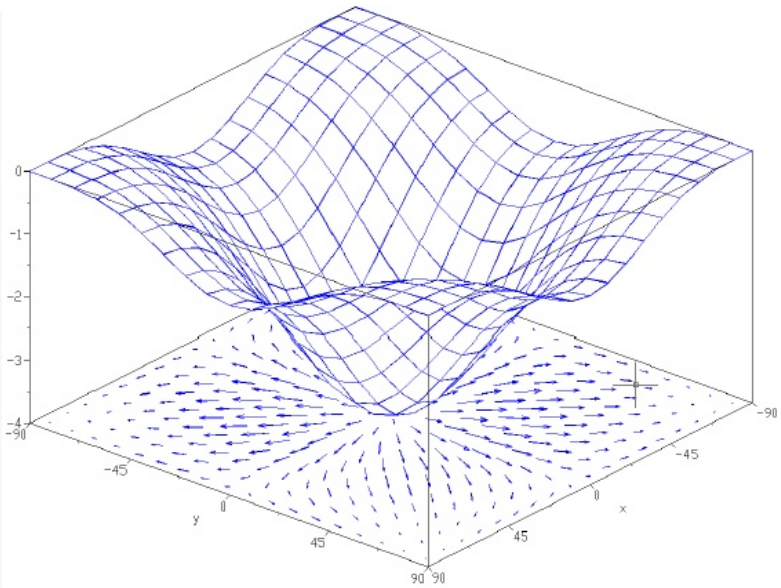
- Régularisation : $A^{+} = (A + \lambda I)^{-1}$
- Moore Pseudo Inverse : $A^{\dagger} = \lim_{\lambda \to 0} A^{T}(AA^{T} + \lambda I)^{-1}$

Utiliser la pente !



La pente - La dérivée - Le gradient





La direction de la pente est donnée par la direction opposée à la dérivée ou au gradient !

Descente de gradient globale (ou Batch)

Algorithme général

- Choisir un point de départ $(\theta_j^0 \forall j)$
- Pour chaque paramètre, répéter jusqu'à convergence

$$\theta_j^{(t+1)} = \theta_j^{(t)} - \alpha_t \frac{\partial J(\theta)}{\partial \theta_j^{(t)}}$$

Régression linéaire

- Choisir un point de départ $(\theta_j^{(0)} \forall j)$
- Pour chaque paramètre, répéter jusqu'à convergence

$$\theta_j^{(t+1)} = \theta_j^{(t)} + \alpha_t \frac{1}{N} \sum_i x_j^{(i)} \left(y^{(i)} - \theta^{(t)\,T} \mathbf{x}^{(i)} \right)$$

Descente de gradient II

Notations matricielles

- Choisir un point de départ ($\theta^{(0)}$)
- Répéter jusqu'à convergence

$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \frac{1}{N} \sum_i x^{(i)} \left(y^{(i)} - {\theta^{(t)}}^T \mathbf{x}^{(i)} \right)$$

$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \frac{1}{N} X \left(Y - X^T \theta^{(t)} \right)$$

NB : α_t est appelé le pas d'apprentissage
NB2 : α_t peut (doit) changer avec t

Le pas d'apprentissage

Intuitions

- Si α_t est grand, on apprend plus vite
- Si α_t trop grand, on oscille si proche de la solution
- α_t doit diminuer avec le temps !

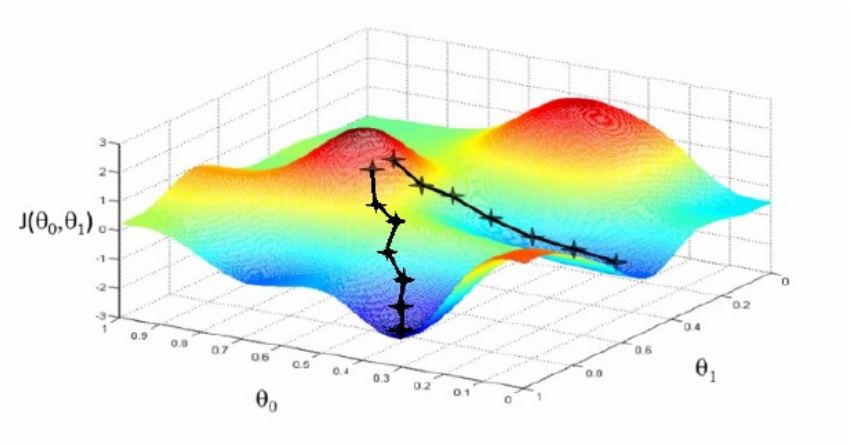
Les contraintes math

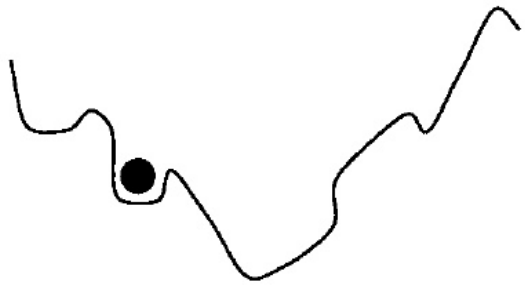
$$\sum_{t=0}^{\infty} \alpha_t = \infty \text{ et } \sum_{t=0}^{\infty} \alpha_t^2 < \infty$$

En pratique, souvent

$$\alpha_t = \frac{A}{B + C * t}$$

Minimum local





Analyse

- Sensibilité aux conditions initiales
- Recuit simulé

Descente de gradient stochastique ou en ligne I

Descente de gradient stochastique ou en ligne II

Intuition

- Utiliser une vision locale du gradient
- Apprendre au fur et à mesure de l'obtention de données
- La règle de mise à jour est une somme, il y a accumulation naturelle.

Avantage

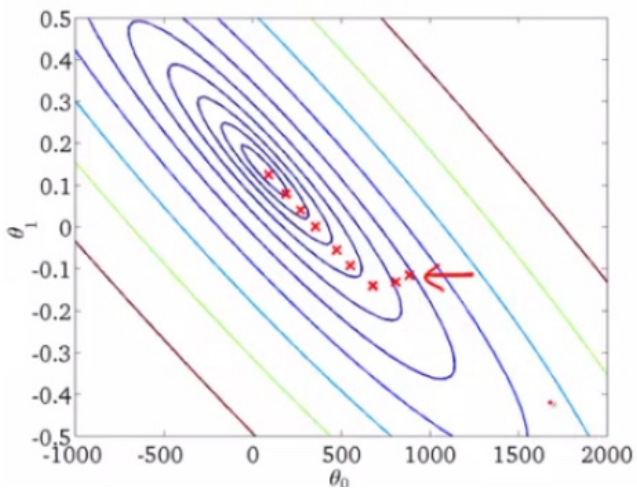
- Il ne faut pas calculer la somme sur toutes les données.
- On apprend dès que les données sont disponibles !
- On peut faire des "mini-batch"

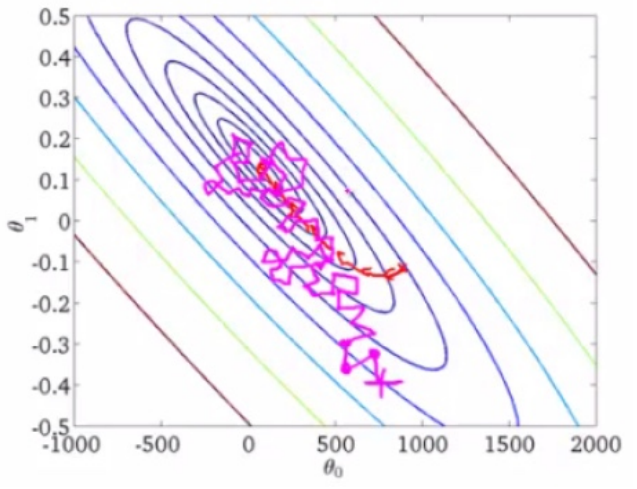
Descente de gradient stochastique ou en ligne III

Algorithme

- Choisir un point de départ $(\theta^{(0)})$
- Faire pour chaque donnée $\{\mathbf{x}^{(i)}, y^{(i)}\}$

$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \left(x^{(i)} \left(y^{(i)} - {\theta^{(t)}}^T \mathbf{x}^{(i)} \right) \right)$$



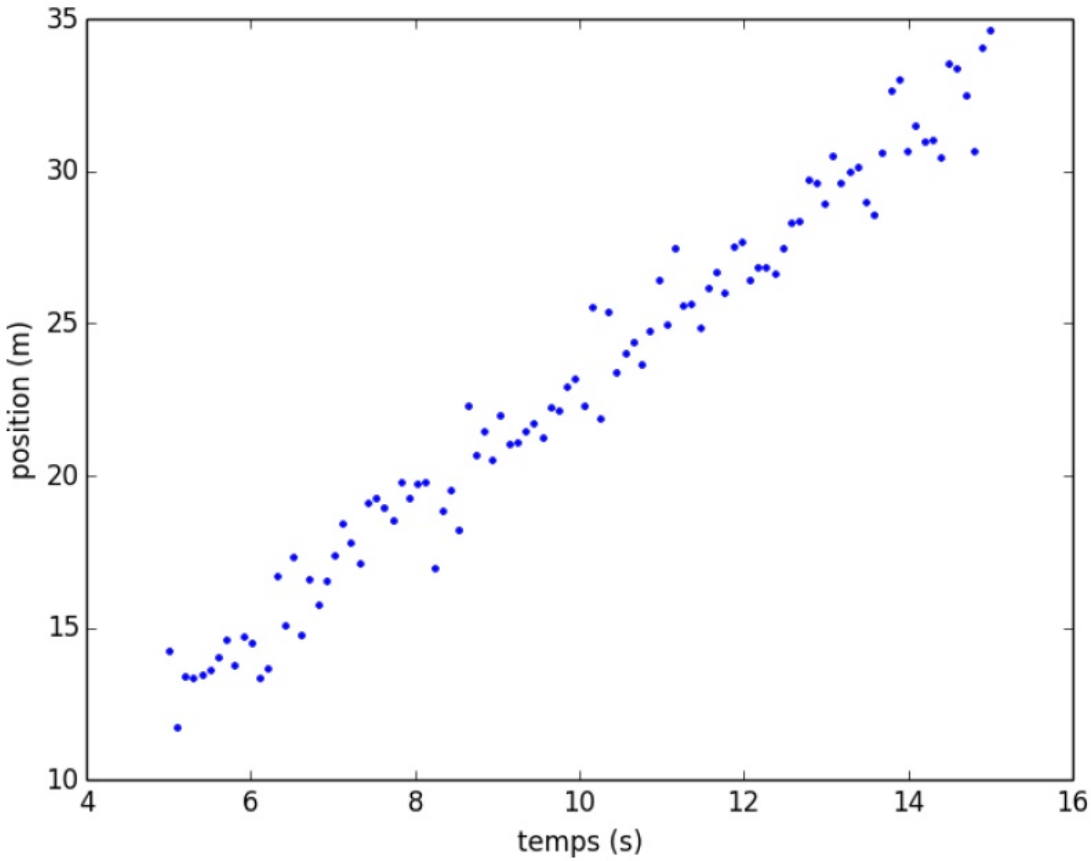


TP

Descente de gradient

- Implémenter les 2 algorithmes
- Analyser et comparer graphiquement leurs vitesses de convergence
- Adapter le pas d'apprentissage pour améliorer la convergence

Rappel cours précédents I



Rappel cours précédents II

Choix de la représentation

- On se choisit une représentation linéaire :

$$y = \sum_m \theta_m x_m = \theta^T \mathbf{x}$$

- On cherche les paramètres θ grâce aux données $\{\mathbf{x}^{(i)}, y^{(i)}\}$.

Induction

On choisit de minimiser l'erreur quadratique moyenne (MSE) :

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2$$

Rappel cours précédents III

Solution

- Pour minimiser, on dérive et on annule :

$$\nabla_\theta \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T \mathbf{x}^{(i)})^2 = 0$$

- La solution est :

$$\theta = (\sum_i \mathbf{x}^{(i)} \mathbf{x}^{(i)T})^{-1} \sum_i \mathbf{x}^{(i)} y^{(i)}$$

Rappel cours précédents IV

Version matricielle

$$J(\theta) = \frac{1}{N}\|(Y - X^T\theta)\|^2$$

$$\theta = (XX^T)^{-1}(XY)$$

Descente de gradient

$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \frac{1}{N} X \left(Y - X^T\theta^{(t)}\right)$$

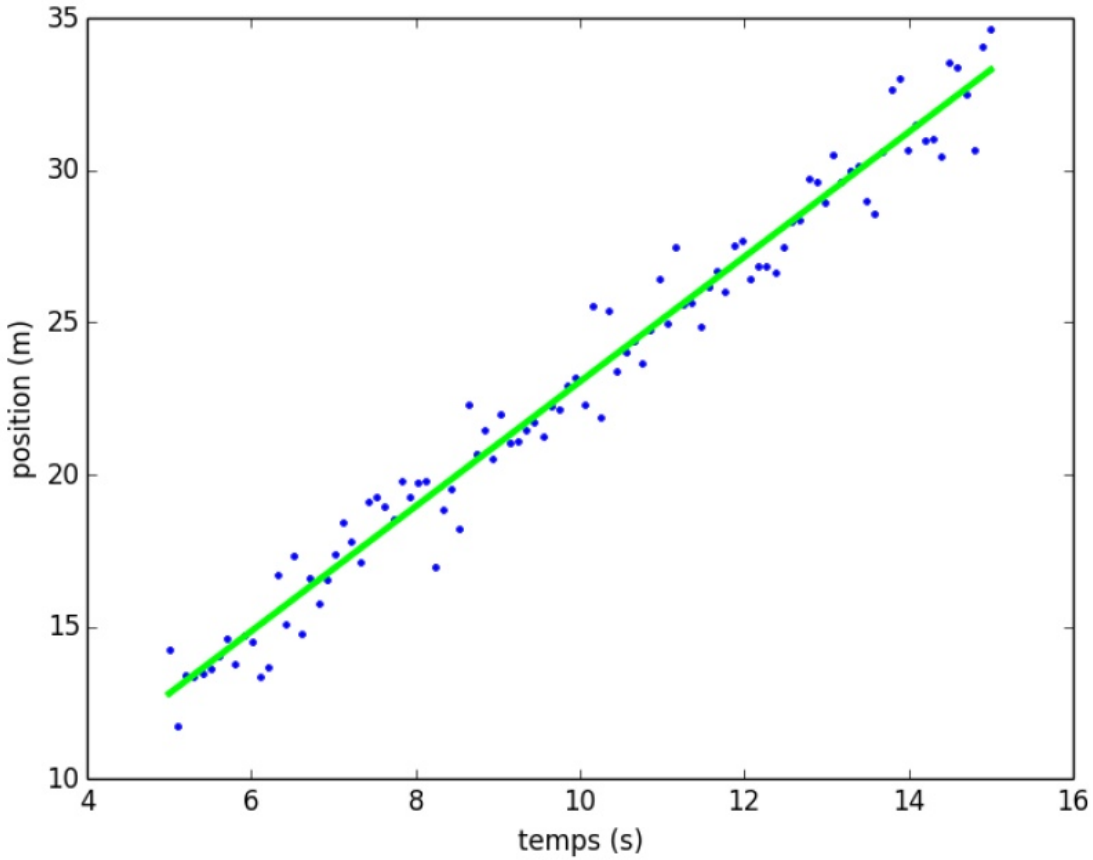
$$\theta^{(t+1)} = \theta^{(t)} + \alpha_t \left(x^{(i)} \left(y^{(i)} - \theta^{(t)^T}\mathbf{x}^{(i)}\right)\right)$$

Pourquoi ça marche ?

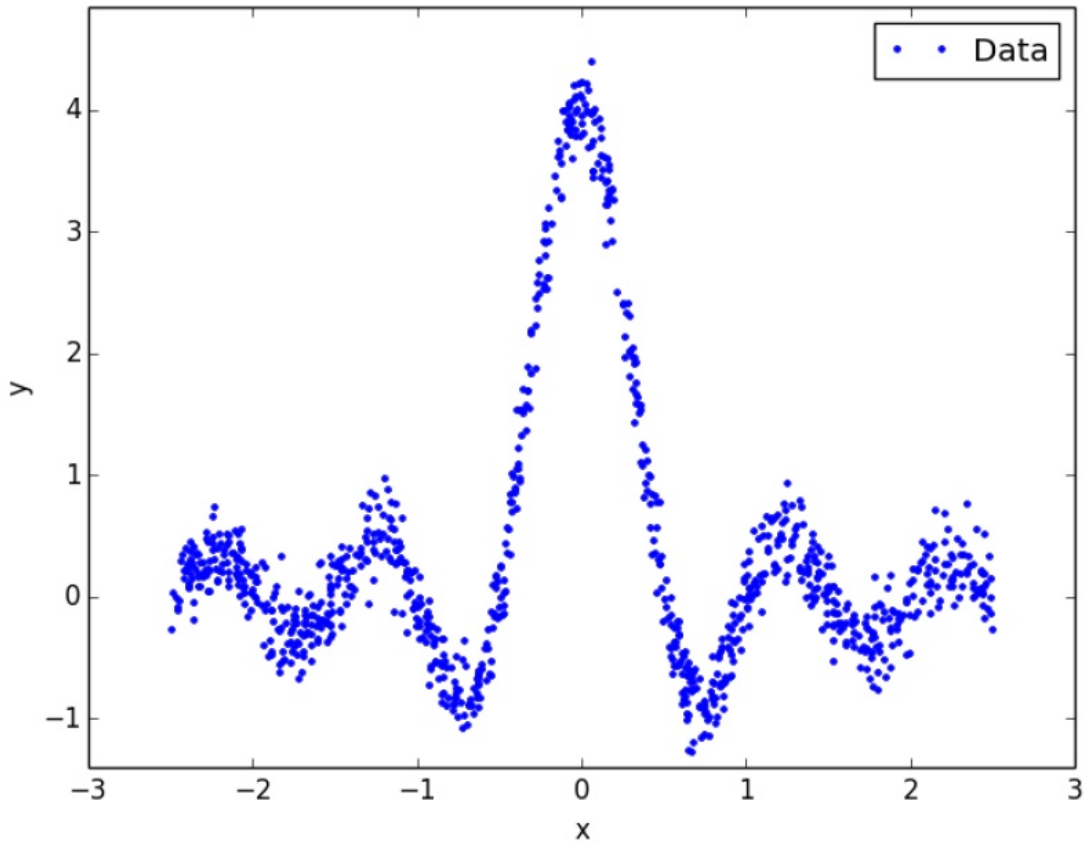
- Fonction de coût convexe et dérivable
- La dérivée donne le minimum

Rappel cours précédents V

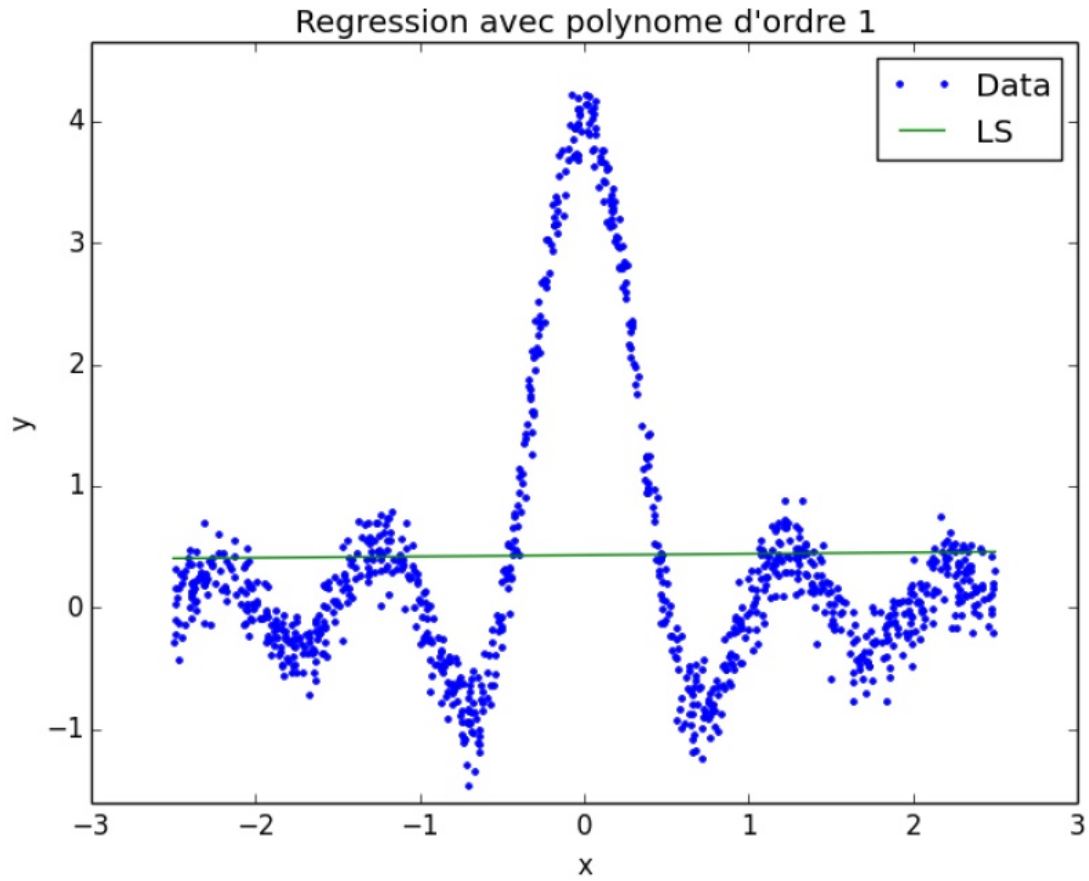
Résultat



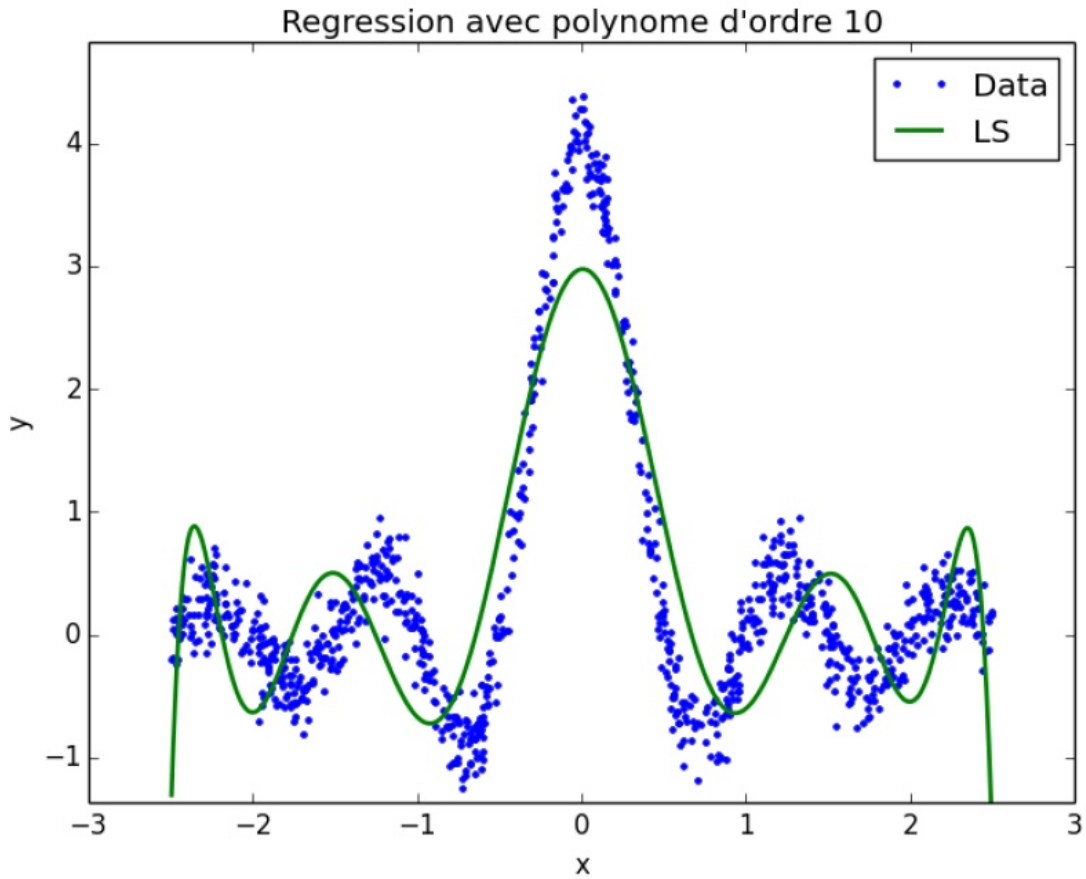
Et si la famille n'est pas évidente I



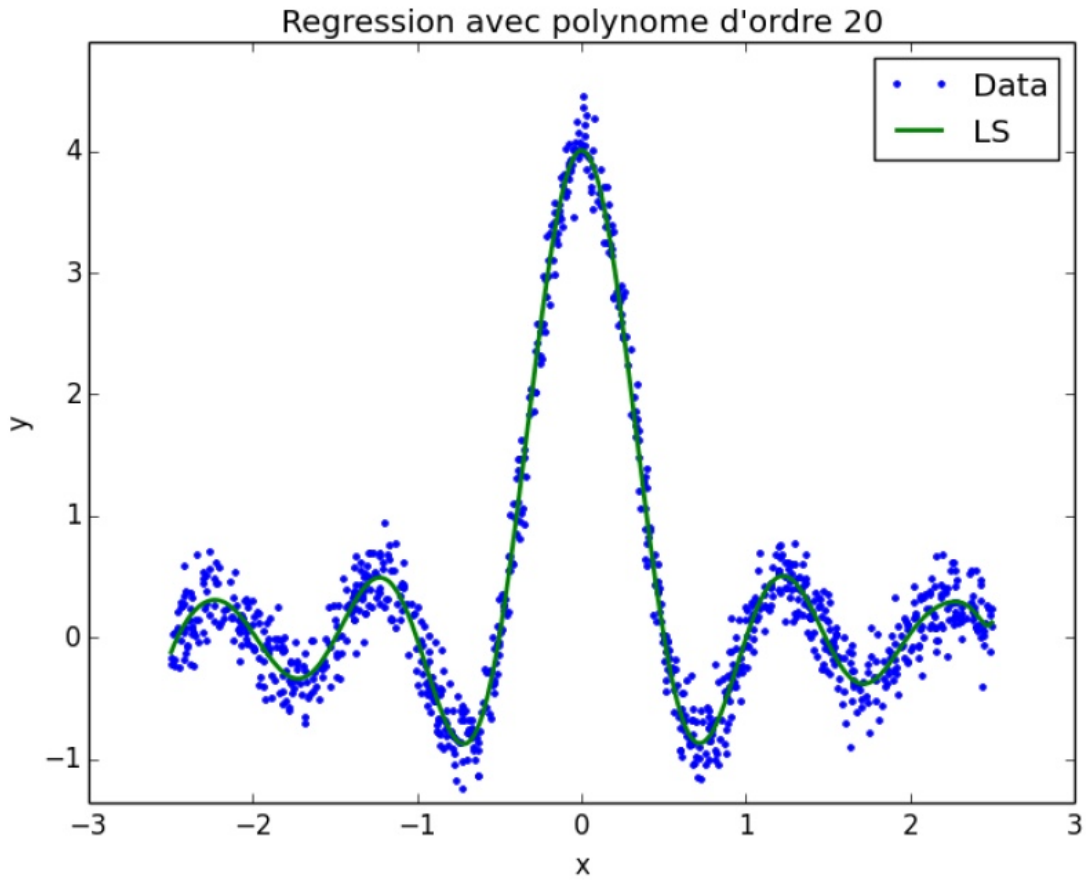
Et si la famille n'est pas évidente II



Et si la famille n'est pas évidente III



Et si la famille n'est pas évidente IV



Questions

Quelle représentation ?

- Quelle forme ?
- Est-ce qu'il suffit de prendre un espace complexe ?

Notion d'espace d'hypothèses

Est-ce qu'avoir plus de données aide ?

- Une fois la forme déterminée, faut-il utiliser toutes les données pour apprendre ?
- Est-ce que la fonction apprise prédit mieux avec beaucoup de données ?

Notion de généralisation

Probabilités **et** Statistique

Mais pourquoi ?

- Quand on ne sait pas, on met des probabilités et des statistiques
- Ici, on ne sait pas quelles données on aura

Probabilités **ou** statistique

- Les deux !
- Description générique = proba
- Calculs en fonction des données = statistique

Définitions I

- Ω **: domaine** ou ensemble des événements possibles.
- $\mathbf{x} \in \Omega$: **événement** (élémentaire).
- $\mathbf{X}$: **variable aléatoire** qui prend ses valeurs dans Ω.
- $\mathbf{P(X = x)}$: **probabilité** que la variable aléatoire X prenne la valeur x c'est à dire que l'événement x se réalise. On note aussi $\mathbf{P(x)}$, **probabilité de l'événement** x.
- $P(X)$: probabilité **marginale** ou **a priori** de la variable X, désigne un ensemble d'équations $P(X = x)$ pour tous les $x \in \Omega$.

Définitions II

Pour 2 variables aléatoires A et B prenant leurs valeurs dans des ensembles Ω_A et Ω_B :

- $P(A \cup B)$ ou $P(A \vee B)$: probabilité de A **ou** B.
- $P(A \cap B), P(A, B)$ ou $P(A \wedge B)$: probabilité de A **et** B appelée **probabilité jointe** de A et B.
- $P(A|B) = \frac{P(A,B)}{P(B)}$: **probabilité conditionnelle** de A sachant B ou **a posteriori** (car disponible après avoir connaissance de B).
- A et B sont **indépendantes** si $P(A|B) = P(A)$ et donc $P(A, B) = P(A) \cdot P(B)$.

Remarque : Les notations impliquant des lettres majuscules désignent toujours un ensemble d'équations.

Définitions III

Un petit exercice :

	Total	Vaccinés (V)
Malades (M)	300	120
Sains (S)	700	480

Calculer : $P(M)$, $P(\bar{V})$, $P(M \cap V)$, $P(M|V)$, $P(M \cup V)$

Vérifier : $P(A|B) = \frac{P(A \cap B)}{P(B)}$

Axiomes de Kolmogorov

Axiome

$$\forall x \in \Omega : P(x) \in [0, 1]$$

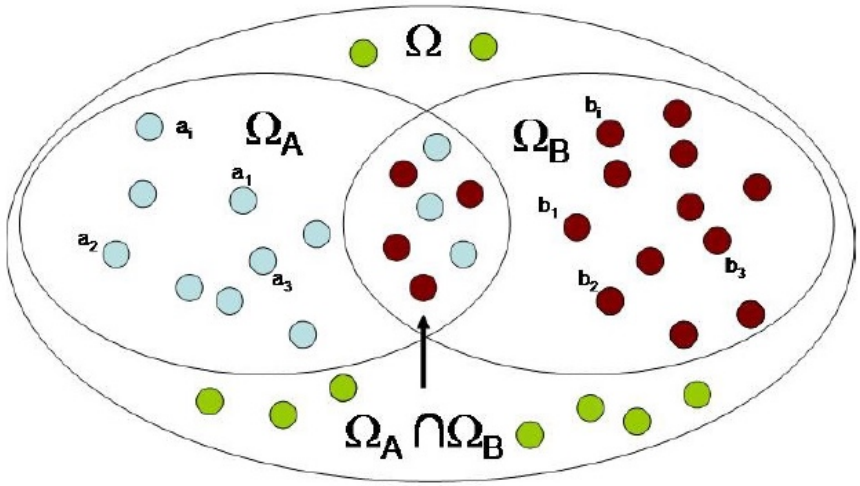
Axiome

$$P(\Omega) = 1 \text{ et } P(\emptyset) = 0$$

Axiome

$$\forall \Omega_i, \Omega_j \in \Omega$$
$$Si\ \Omega_i \cap \Omega_j = \emptyset, \forall i \neq j : P(\cup_i \Omega_i) = \sum_i P(\Omega_i)$$
$$\sigma\text{-additivité}$$

Conséquences



Lemme

$$P(\Omega_A) = P(\cup_i a_i) = \sum_i P(a_i)$$

Lemme

$$P(\Omega_A \cup \Omega_B) = P(\Omega_A) + P(\Omega_B) - P(\Omega_A \cap \Omega_B)$$

Espérance

Définition

$$E[X] = \mu_X = \sum_{x \in \Omega} xP(x)$$

$$E[X] = \mu_X = \int_{\Omega} XdP(X) = \int_{\Omega} XP(X)d(X)$$

Loi faible des grands nombres

Soit (X_n) une suite de n variables aléatoires, i.i.d. (indépendantes identiquement distribuées), de moyenne μ_X, alors

$$\forall \epsilon > 0, \lim_{n \to \infty} P\left(\left| \frac{1}{n} \sum_{i=1}^{n} X_i - \mu_X \right| > \epsilon \right) \to 0$$

Vision statistique de l'apprentissage I

Définition

Etant donné un ensemble $\mathcal{D}$ de N données sous forme de couples $\{x_i, y_i = f(x_i)\}_{0<i<N+1}$ avec $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$ (souvent $\mathcal{X} = \mathbb{R}^m$ et $\mathcal{Y} = \mathbb{R}$), il s'agit de trouver une fonction $f_N(x) \in \mathcal{H}$ qui permette d'expliquer et de prédire la relation entre des entrées quelconques (x) et les sorties (y) correspondantes. $\mathcal{H}$ est appelé l'espace d'hypothèse.

Oracle

L'oracle connaît la relation, il a donné les couples de la base. Il agit selon une loi de probabilité $P(Y|X)$

Vision statistique de l'apprentissage II

Distribution naturelle

Dans la nature, on rencontre les observations x_i selon la distribution $P(X)$. Ainsi, la base de données est constituée de paires tirées selon $P(X, Y) = P(X)P(Y|X)$.

Fonction de perte (*Loss function*)

$\mathcal{L}(f(x), y)$: coût de la décision associée à x par une fonction f quelconque, sachant que la bonne décision était y.
Par exemple :

- Coût quadratique : $(y_i - f(x_i))^2$
- Coût abslou : $|y_i - f(x_i)|$

Vision statistique de l'apprentissage III

Risque Réel

Le risque associé à la fonction de coût est

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}(f(x), y) dP(x, y)$$

C'est donc l'espérance de la fonction de coût.

$$R(f) = E[\mathcal{L}(f(x), y)] = \mu_{\mathcal{L}(f(x),y)}$$

Vision statistique de l'apprentissage IV

Estimation

En pratique, on ne connaît pas la distribution $P(x, y)$. On doit donc se baser sur les données. Grâce à la loi faible des grands nombres, on a :

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)) \to R(f)$$

.

Risque empirique

$$R_N(f) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i))$$

.

Apprendre f_N et estimer $R(f_N)$?

Problème

La loi des grands nombres suppose que les $\mathcal{L}(f(x_i), y_i)$ sont i.i.d.. Si on calcule f_N avec les données, ce n'est plus le cas. En effet, on a fait :

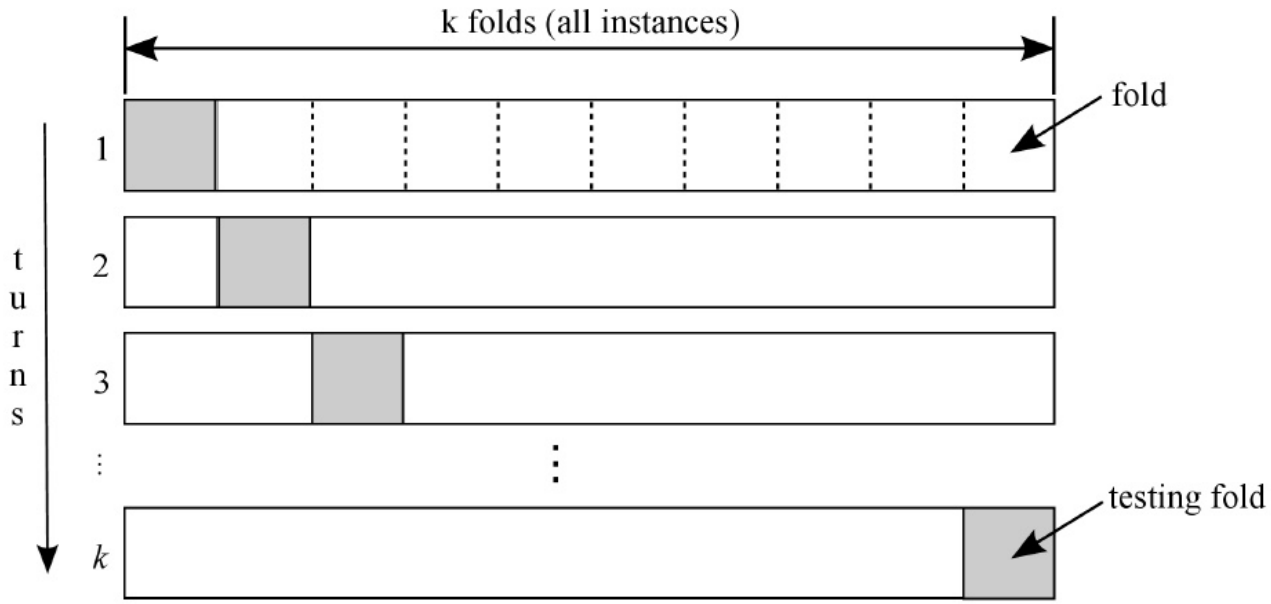
$$f_N = \underset{f \in \mathcal{H}}{\operatorname{argmin}} R_N(f)$$

En particulier ($f_N(x) = \theta_{LS}^T x$) :

$$\theta_{LS} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_i (y_i - \theta^T x_i)^2$$

Et donc, chaque $f_N(x_i)$ est fonction de **tous** les x_i, y_i utilisés.

Validation croisée



$$\hat{R}(f) = \frac{1}{K} \sum_{j=0}^{K-1} \frac{K}{N} \sum_{i=1}^{\frac{N}{K}} \mathcal{L}\left(f_{\frac{N(K-1)}{K}}\left(x_{(j*K/N)+i}, y_{(j*K/N)+i}\right)\right.$$

Apprentissage

Minimisation du risque empirique

- On peut donc estimer le risque $R(f)$ par $R_N(f)$.
- Ca ne veut pas dire que minimiser $R_N(f)$ revient à minimiser $R(f)$

Décomposition du risque

Si R^* est le meilleur risque qu'on puisse obtenir et f_o la meilleure fonction de $\mathcal{H}$ pour approximer f on a :

$$R_N(f) - R^* = \underbrace{R(f_o) - R^*}_{\text{biais}} + \underbrace{R(f_N) - R(f_o)}_{\text{variance}}$$

Compromis Biais-Variance

Biais

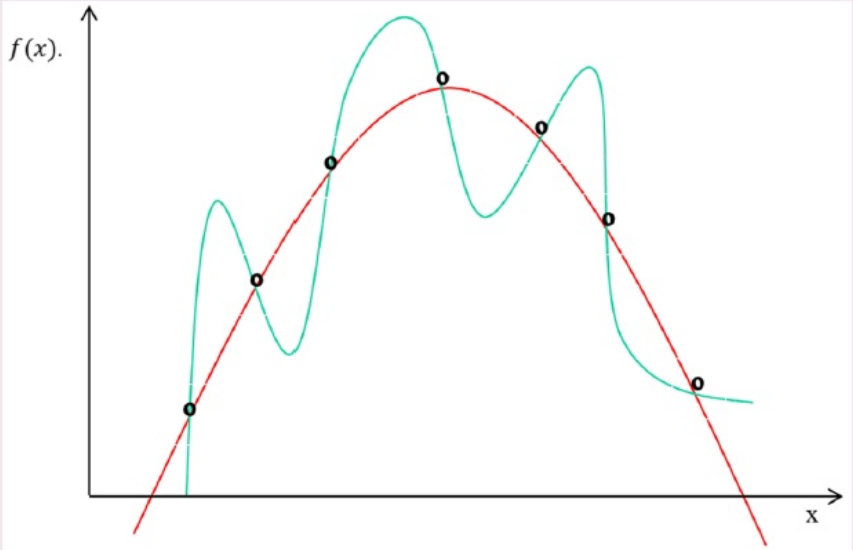
- On ne connaît pas la forme de la fonction f donc, il est difficile de l'estimer
- Il dépend tout de même de la capacité à généraliser des fonctions de $\mathcal{H}$

Variance

- Il est fonction du nombre de données
- Si $\mathcal{H}$ est trop complexe, il faut beaucoup de données
- Si trop de données, on sur-apprend !

Sur-apprentissage

Apprendre par cœur



Généralisation

Il faut apprendre à prédire ce que l'on n'a pas vu ! C'est à dire minimiser le risque réel et pas le risque empirique !

Pistes de solutions

Vapnik : les bornes !

Vladimir Vapnik a démontré que, sous certaines conditions, minimiser le risque empirique mène à minimiser le risque réel. Il calcule des bornes, basées sur les inégalités de concentration.

Validation croisée

Calcul de l'estimation de $R_N(f_N)$ pour différent N et on choisit le N qui minimise.

Régulariser

Ajouter un facteur de régularisation à la fonction de coût pour contraindre l'espace de recherche.

Enrichir l'espace d'hypothèse $\mathcal{H}$

Toujours dans le cas de la régression linéaire :

$$y = \sum_{j=0}^{m} \theta_j \phi_m x = \Theta^T \Phi(x)$$

$$Y = \Phi^T(X)\Theta$$

$$\Phi(X) = \begin{pmatrix} \phi_0(x^{(1)}) & \dots & \phi_0(x^{(N)}) \\ \vdots & \ddots & \vdots \\ \phi_m(x^{(1)}) & \dots & \phi_m(x^{(N)}) \end{pmatrix}$$

Solution :

$$\Theta = \left(\Phi(X)\Phi^T(X)\right)^{-1}(\Phi(X)Y)$$

Exemples

Exemples :

- $\phi_m(x) = x^m$
- $\phi_m(x) = e^{\frac{(x-\mu_m)^2}{2\sigma_m^2}}$