DU Python Big Data

Machine Learning

Partie 1

gilles.michel@sudintralog.com
@unimes.fr

Principe

Mettre en œuvre des algorithmes améliorant **automatiquement** les performances d'un système

Plan

I. Régressions
II. Arbres de décision
III. Forêt aléatoire et apprentissage d'ensemble
IV. SVM (Support Vector Machine)

I. Régressions

- On dispose d'un ensemble de couples (valeur1, valeur2) => nuage de points
- On se fixe un modèle (paramétré) :
  - Affine $ax + b$ (paramètres a et b)
  - Exponentiel a^x (paramètre a)
  - Gaussien $N(m,\sigma)$ (paramètres m et σ)
  - …
- Détermination des valeurs optimales des paramètres du modèle => *training*
- Vérification de la qualité => *test* puis utilisation

Objectif d'une régression

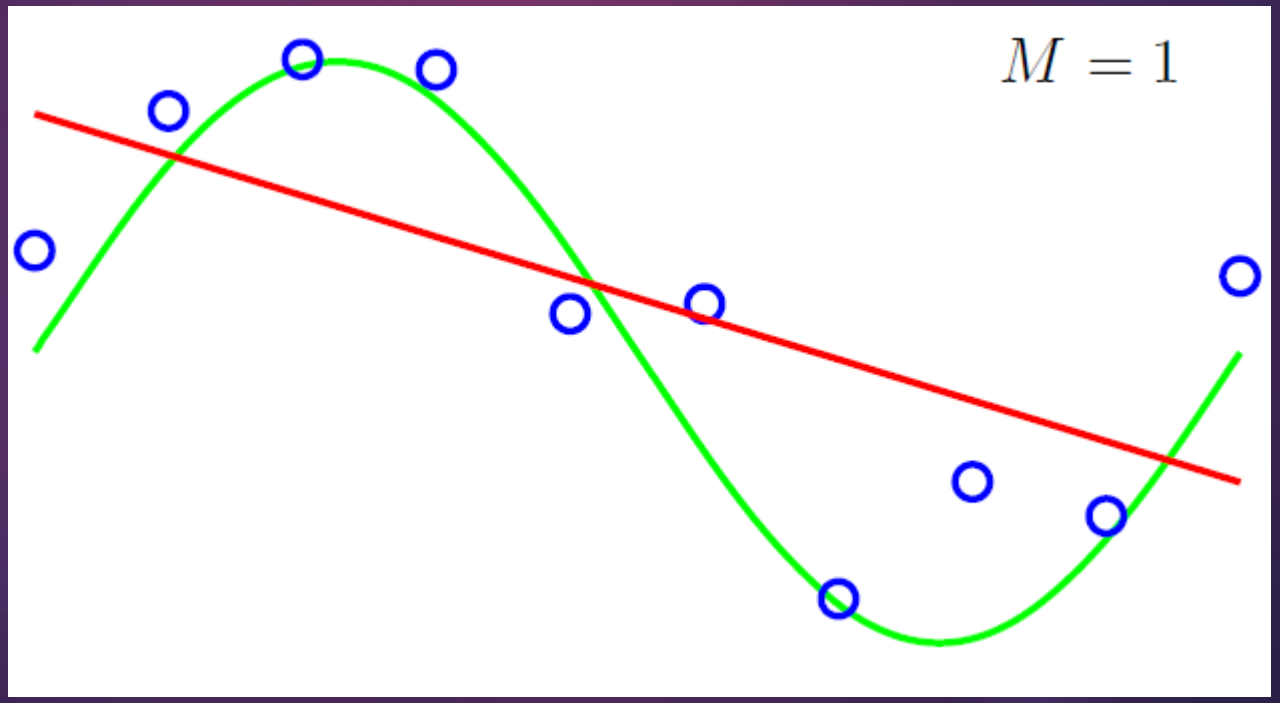
- Interpoler (entre plusieurs points du nuage)
- Extrapoler (prévision)

Fitting

Capacité du modèle à « coller » au nuage

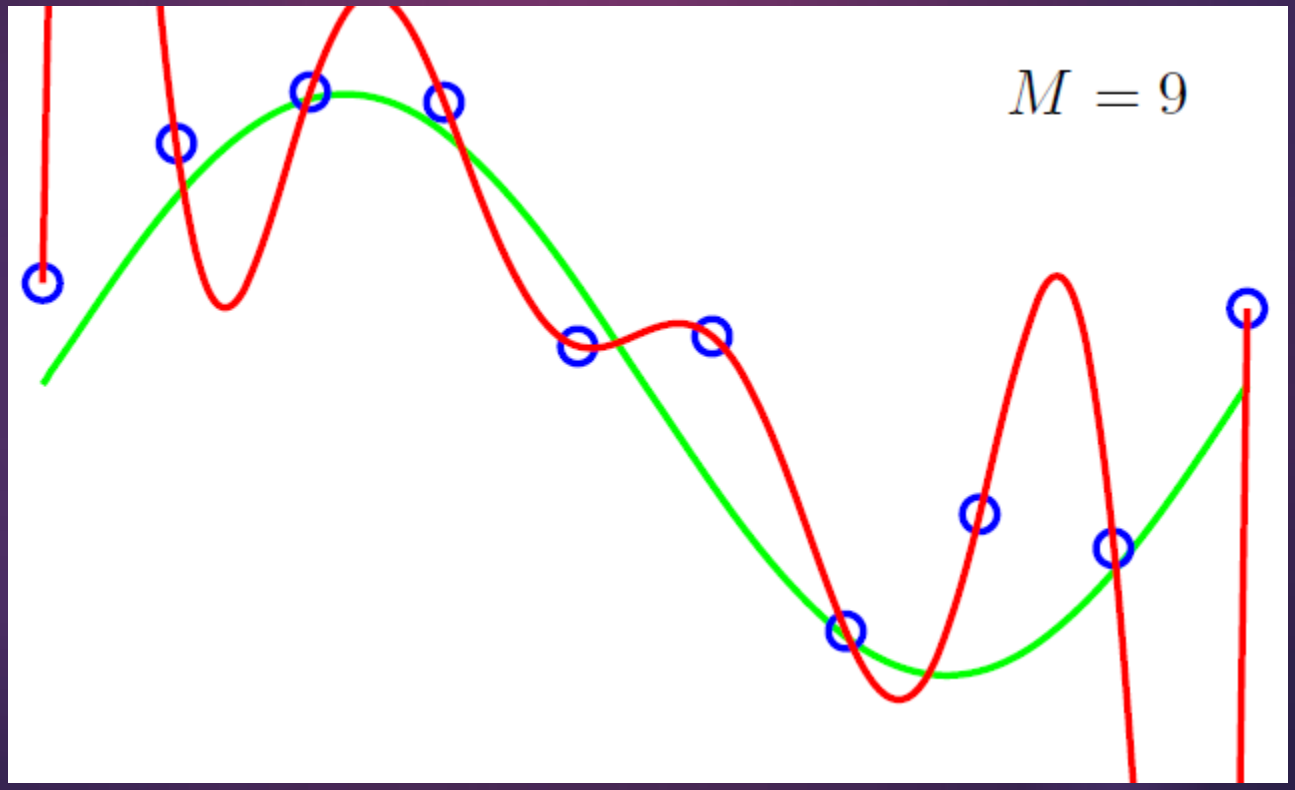
Problèmes de *fitting*

- Underfitting : le modèle est trop éloigné du nuage de points (ex. modèle polynomial de degré M)



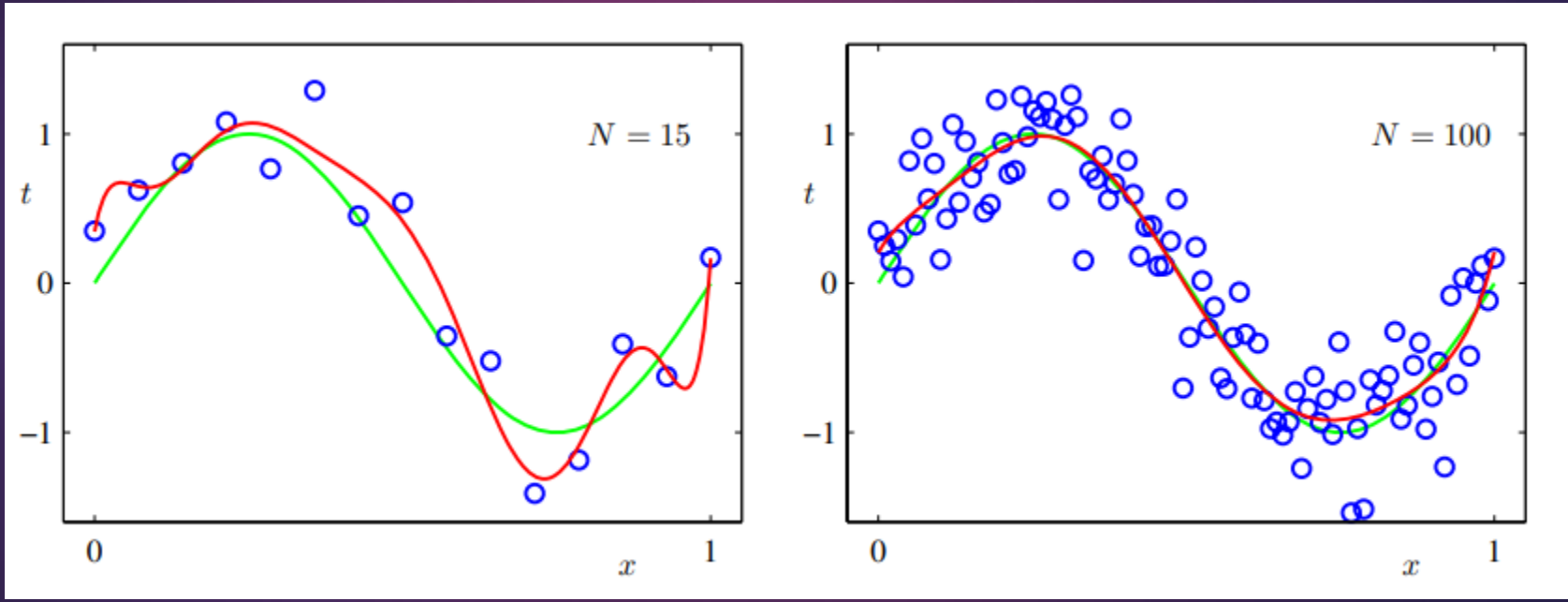
Problèmes de *fitting*

- Overfitting : le modèle colle plus au nuage de points qu'au modèle sous-jacent



Solutions à *l'overfitting*

L'*overfitting* diminue avec la taille des données de l'apprentissage



Solutions à *l'overfitting*

La régularisation : on pénalise certaines données lors de l'apprentissage

Ex. On ajoute à l'erreur calculée à chaque étape, $\lambda\|\vec{w}\|$ où w représente le vecteur des paramètres du modèle.

λ = coef de régularisation

TP1 : régression par Excel (recherche à « tâtons »)

- Déterminer les paramètres du modèle a*exp(b*x) à partir du nuage de points :

(1 ; 12)

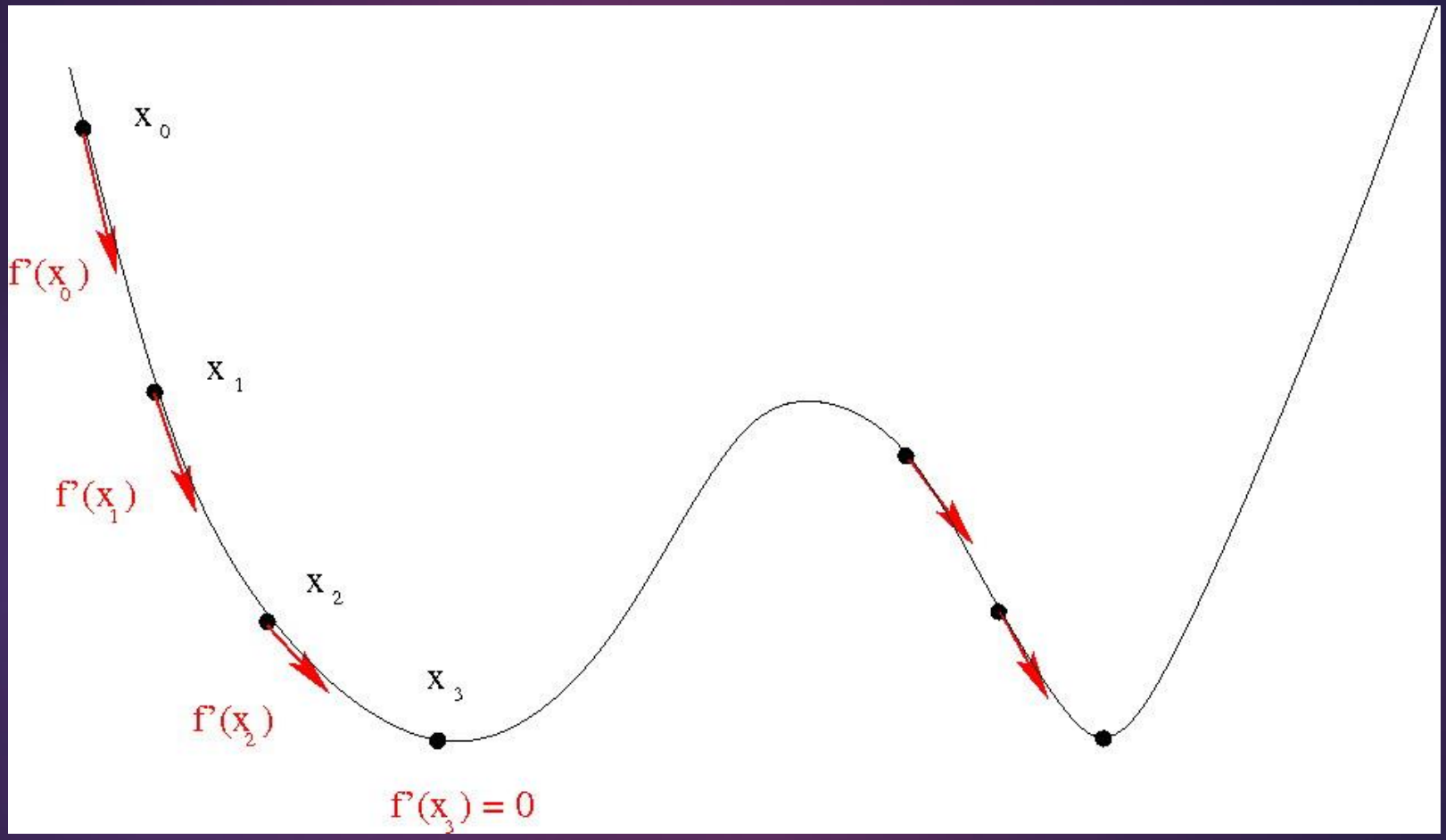
(2 ; 43)

(3 ; 150)

(4 ; 500)

Estimation pour 5 ?

Descente du gradient



Formule d'apprentissage

On définit une suite de vecteurs paramètres *P* par :

$$P = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

$$P_{k+1} = P_k - pas * \nabla J(P_k)$$

Pas (constant ou variable)

Vecteur gradient de *J* (formé des dérivées partielles de *J* par rapport à chaque paramètre (ou poids synaptique) w_i)

TP2 : régression par Python (descente de gradient)

- Déterminer les paramètres du modèle a*exp(b*x) à partir du nuage de points :

(1 ; 12)

(2 ; 43)

(3 ; 150)

(4 ; 500)

Estimation pour 5 ?