Machine learning : un tour d'horizon

Quentin de Laroussilhe

April 18, 2014

Introduction

Apprentissage

Mécanismes conduisant à l'acquisition de nouveaux savoirs et de nouvelles compétences.

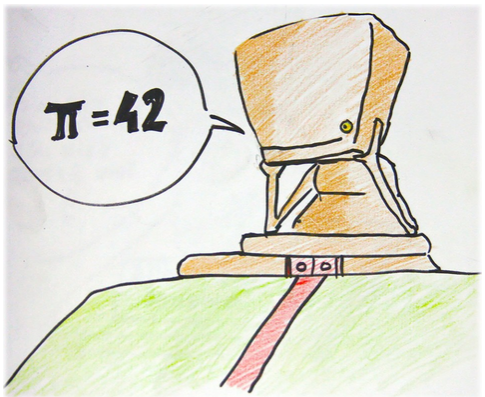
Vecteurs d'apprentissage

- Répétition, mémorisation
- Inférence
- Abstraction
- Intuition
- ...

Apprentissage automatique

Un algorithme d'apprentissage permet d'entrainer un modèle destiné à approcher les solutions d'une instance d'un problème.

Ils permettent de résoudre des problèmes sur de larges volumes de données qu'il serait impossible de traiter avec des moyens algorithmiques traditionnels.

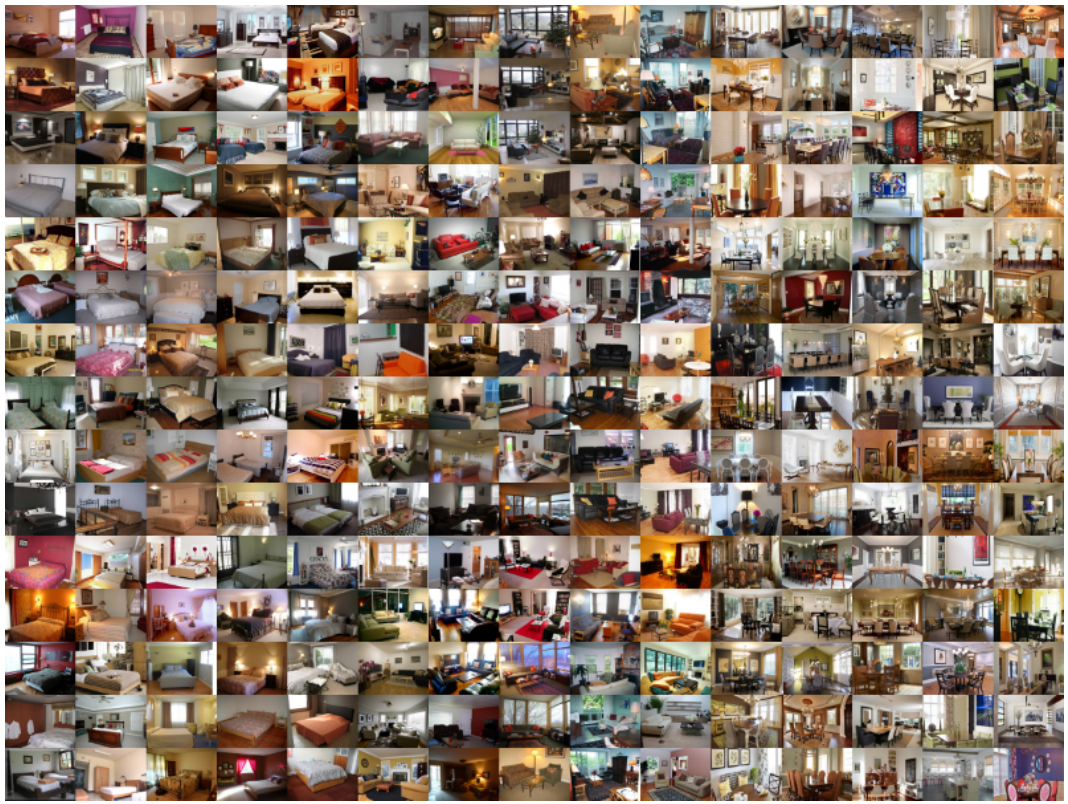


Quelques applications

- Reconnaissance de caractères, traitement image et parole
- Trading automatisé
- Moteurs de recherche / suggestion
- Business intelligence
- Robotique

Concepts généraux

Dataset



Feature extraction

L'extraction de feature a pour but de caractériser les données d'apprentissage.

- Réduction de la dimension du problème
- Définies par un exepert ou un algorithme

Exemples

- Individu : {poids, age, taille, ...}
- Programme informatique : {AST, report d'une test suite}
- Caractère : {valeur des pixels du caractere redimensionné}
- Cours boursier : {open, close, high, low}

Apprentissage supervisé

Type d'apprentissage visant à apprendre à partir d'un ensemble d'exemples étiquetés à étiqueter des éléments inconnus.

Autrement formulé, à partir d'un set d'input et d'un set d'output, on cherche à trouver la relation liant input et output.

Apprentissage supervisé : classification

Un problème de classification consiste à trouver une relation entre un input et une classe, parmis un set fini de classes possibles.

Exemples

- Détection du visage, classes $= \{$présence, absence$\}$
- OCR, classes $= \{$caracteres identifiables$\}$

Apprentissage supervisé : régression

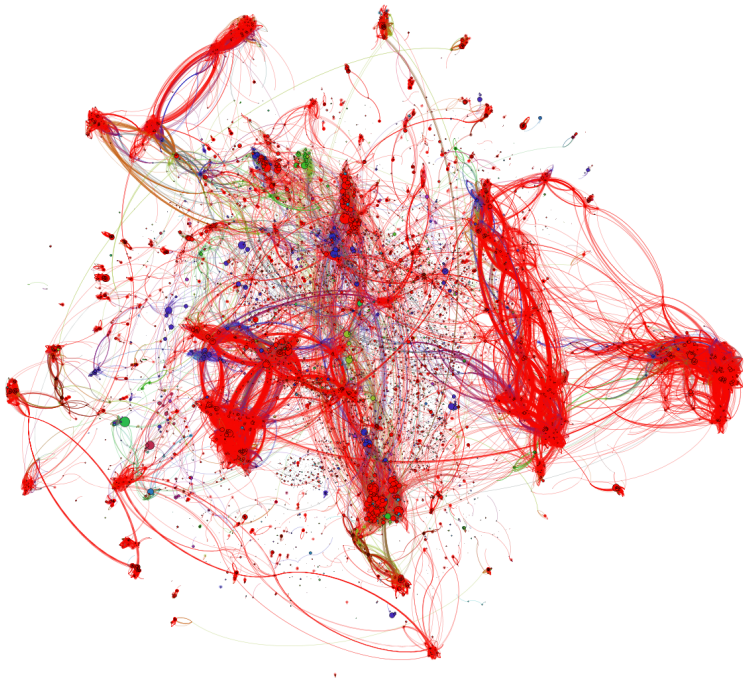
Un problème de regression consiste à trouver la relation liant les variables d'un dataset à une variable de prediction.

Exemples

- Prévision de la fréquentation des routes
- Prévision de la charge d'un serveur
- Estimation de la consommation d'electricité d'un menage

Apprentissage non-supervisé

Extraire des patterns (structure sous-jacente) dans un training set non étiqueté. On observe un set de données pour trouver les relations cachées qui s'y trouvent.



Syntastic and functional variability of code submission in a ML course, Stanford University

Apprentissage par renforcement

Apprendre un comportement (policy) optimal au sein d'un environnement en se basant sur un feedback régulier de la série d'actions effectuée.

- Béhaviorisme (Pavlov, Watson, Skinner)
- Black-box
- Contingence de renforcement : {state, action, reward}

Apprentissage par renforcement : policy search

Exploration de l'espace des comportements de façon a maximiser le nombre de récompenses reçues.

- Utilisation de metaheuristiques
- Problème d'optimisation complexe
- Optimums locaux, No Free Lunch...

Tradeoff biais variance

Il est impossible d'effectuer un apprentissage sans poser un apprioris (un biais) sur les données que l'on étudie.

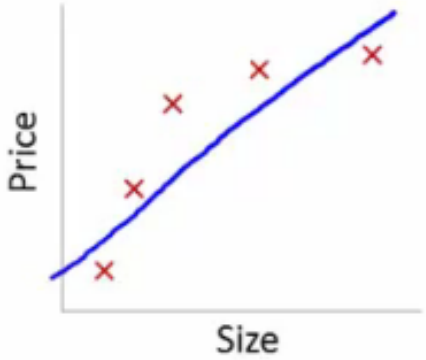
Définitions

- Biais : à quel point le modèle est "flexible"
- Variance : à quel point le résultat d'apprentissage va varier suite à d'une faible modification du training set

Tradeoff biais variance

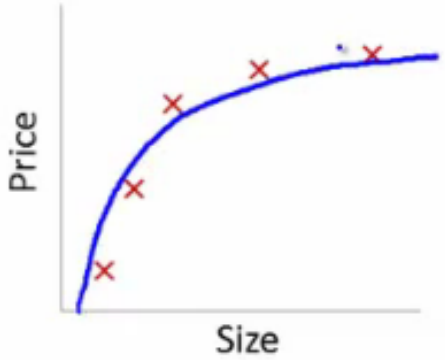
- Plus le biais est faible, plus la variance augmente, inversement
- Une variance forte implique une faible tolérance au bruit, une généralisation moins importante
- Un biais trop fort empêche de trouver un modèle qui va fitter le training set

Tradeoff biais variance



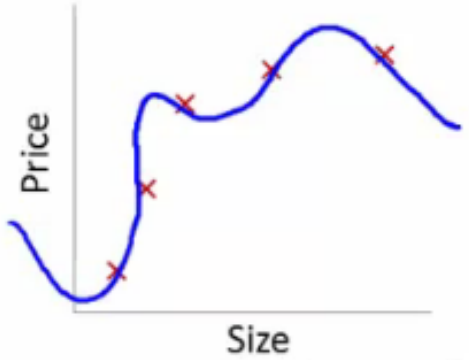
$\theta_0 + \theta_1 x$

High bias
(underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

Choisir un modèle d'apprentissage

Le modèle va imposer un biais sur l'apprentissage.

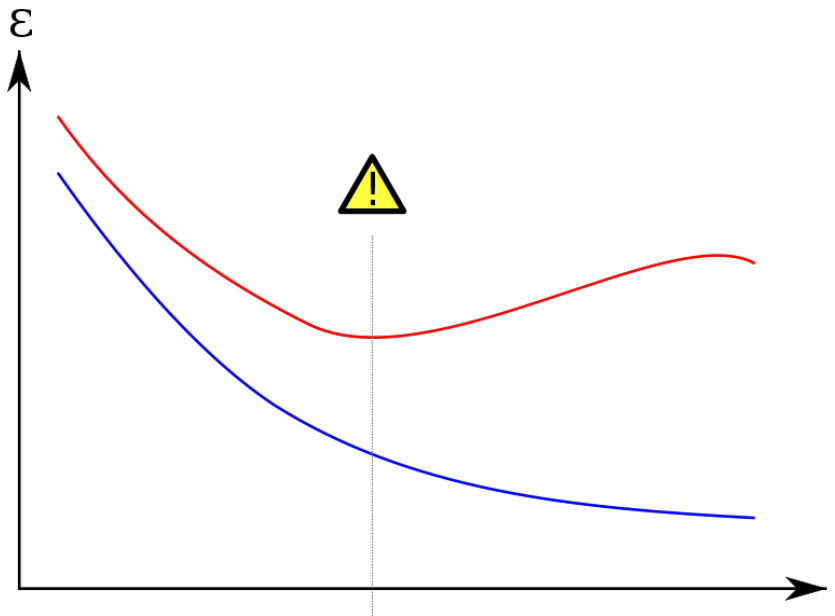
Comment choisir ?

- Etude ou hypothese de la répartition dataset
- Comparaison de plusieurs modèles
- Meta-learning

Valider un modèle

Il est nécessaire de tester le modèle une fois entrainé. On ne peut tester le modèle sur les données servant à l'entrainement (une interpolation polynomiale de degré n peut approximer exactement n-1 exemples).

Cross validation :



Modèles de regression

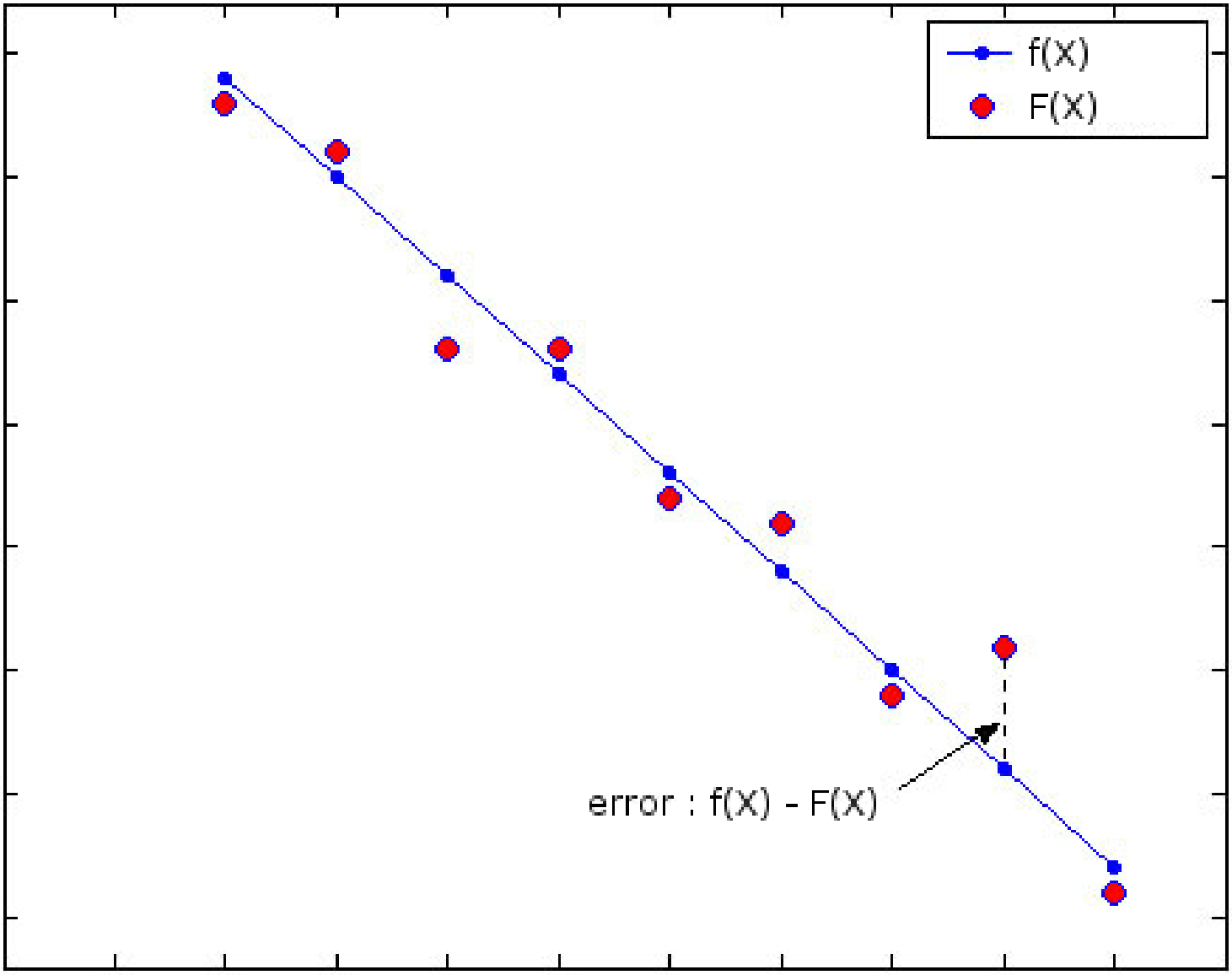
Notations

Training set

$$T = \begin{bmatrix} 1, T_{1,1}, ..., T_{1,k} \\ 1, T_{2,1}, ..., T_{2,k} \\ ... \\ 1, T_{n,1}, ..., T_{n,k} \end{bmatrix}$$

- $T_i \in \mathbb{R}^k$ est le i^{eme} vecteur d'exemple
- k features
- n exemples

Régression linéaire



Régression linéaire

Relation linéaire à ajuster par pondération

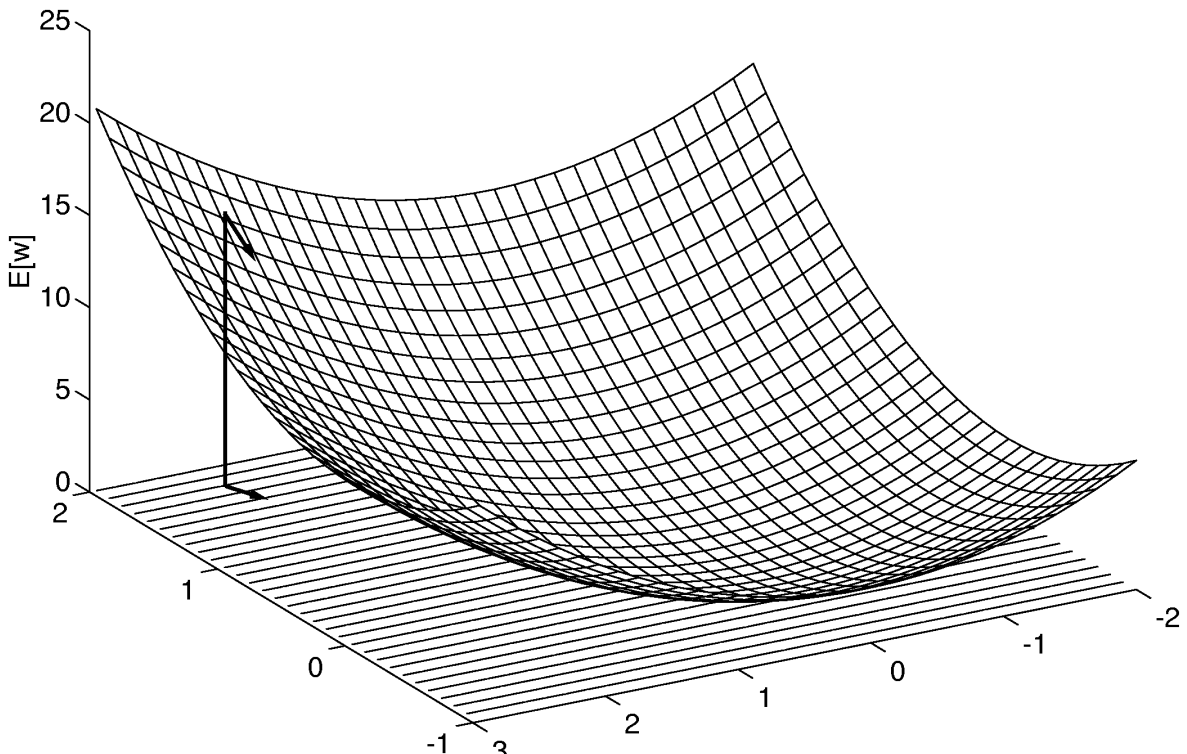
On va chercher à modéliser une relation inconnue $F : \mathbb{R} \mapsto \mathbb{R}$ par une relation linéaire de la forme :

$$f : X \to \sum_{i=0}^{k} (X_i W_i)$$

Régression linéaire : batch gradient descent

$$Erreur = \epsilon = \frac{1}{2}\sum_{i=1}^{n}(f(T_i) - F(T_i))^2$$

$$\frac{\partial \epsilon}{\partial W} = \left[\frac{\partial \epsilon}{\partial W_0}, ..., \frac{\partial \epsilon}{\partial W_k}\right]$$



Régression linéaire : batch gradient descent

$$Erreur = \epsilon = \frac{1}{2}\sum_{i=1}^{n}(f(T_i) - F(T_i))^2$$

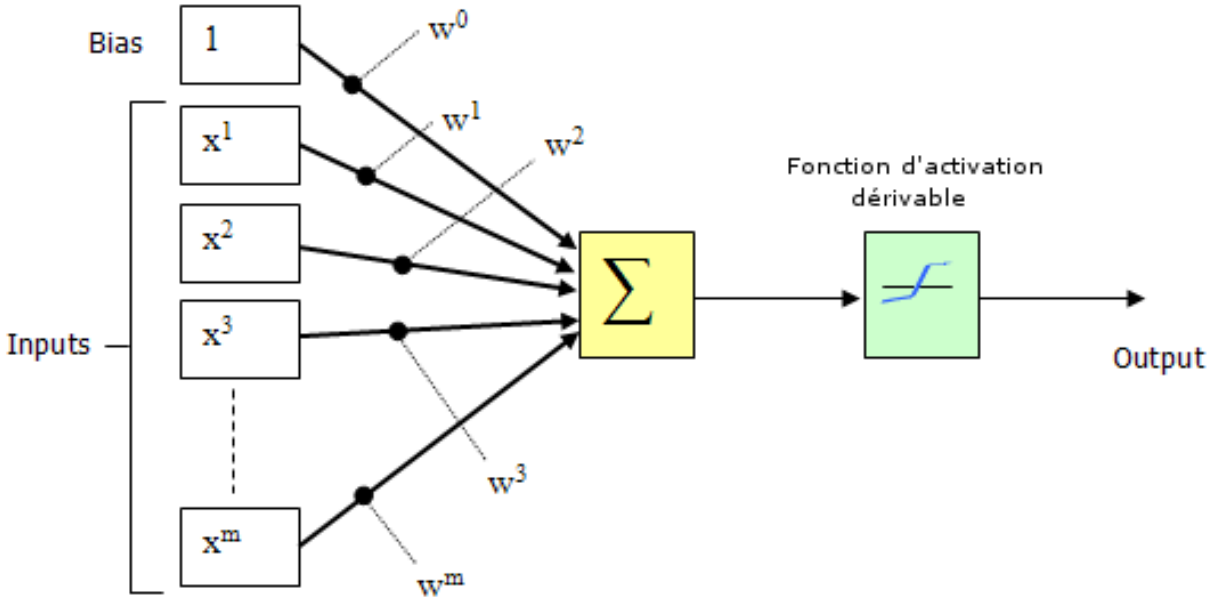
Expression du gradient

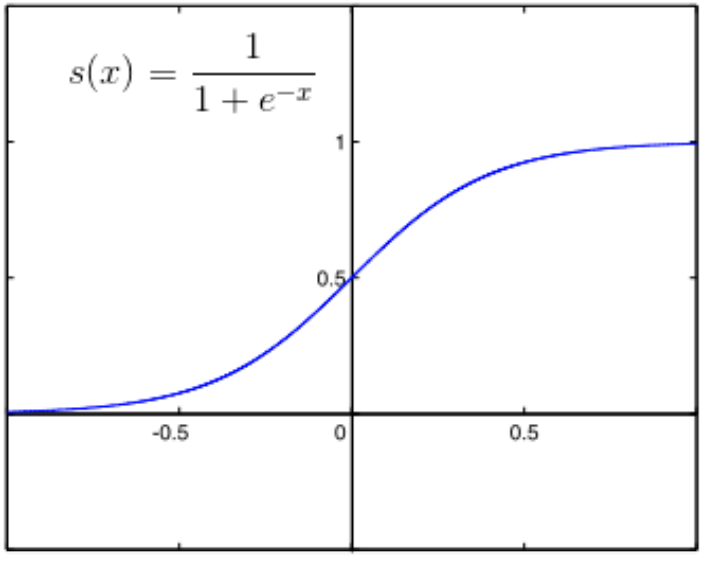
$$\frac{\partial \epsilon}{\partial W_j} = \sum_{i=1}^{n} T_{i,j}(f(T_i) - F(T_i))$$

Suivre la pente de l'erreur

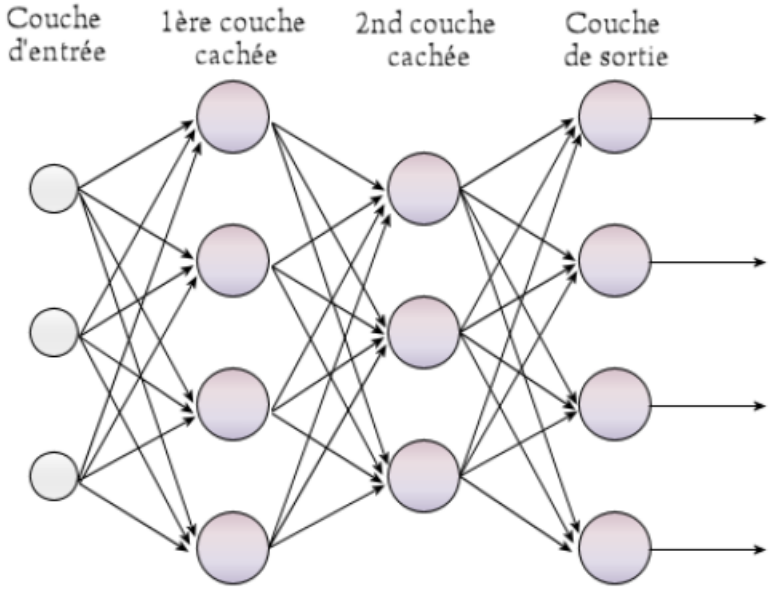
$$W_j := W_j - \alpha \frac{\partial \epsilon}{\partial W_j}$$

Perceptron

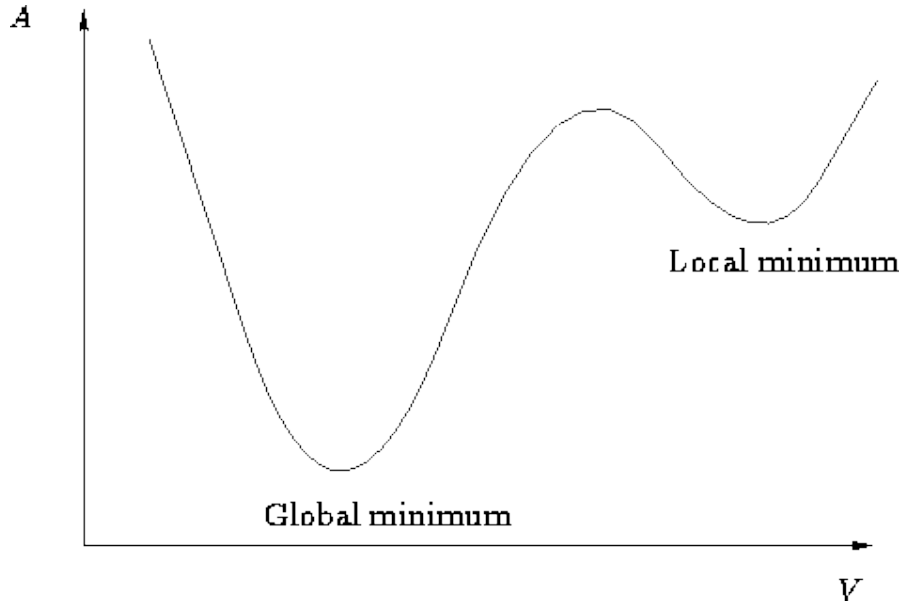

Bias
1
w^0
x^1
w^1
w^2
x^2
Fonction d'activation
dérivable
Inputs
x^3
Σ
Output
w^3
x^m
w^m


$s(x) = \frac{1}{1+e^{-x}}$
1
0.5
-0.5
0
0.5

Réseaux de neurones (MLP)



Optimums locaux



Modèles de classification

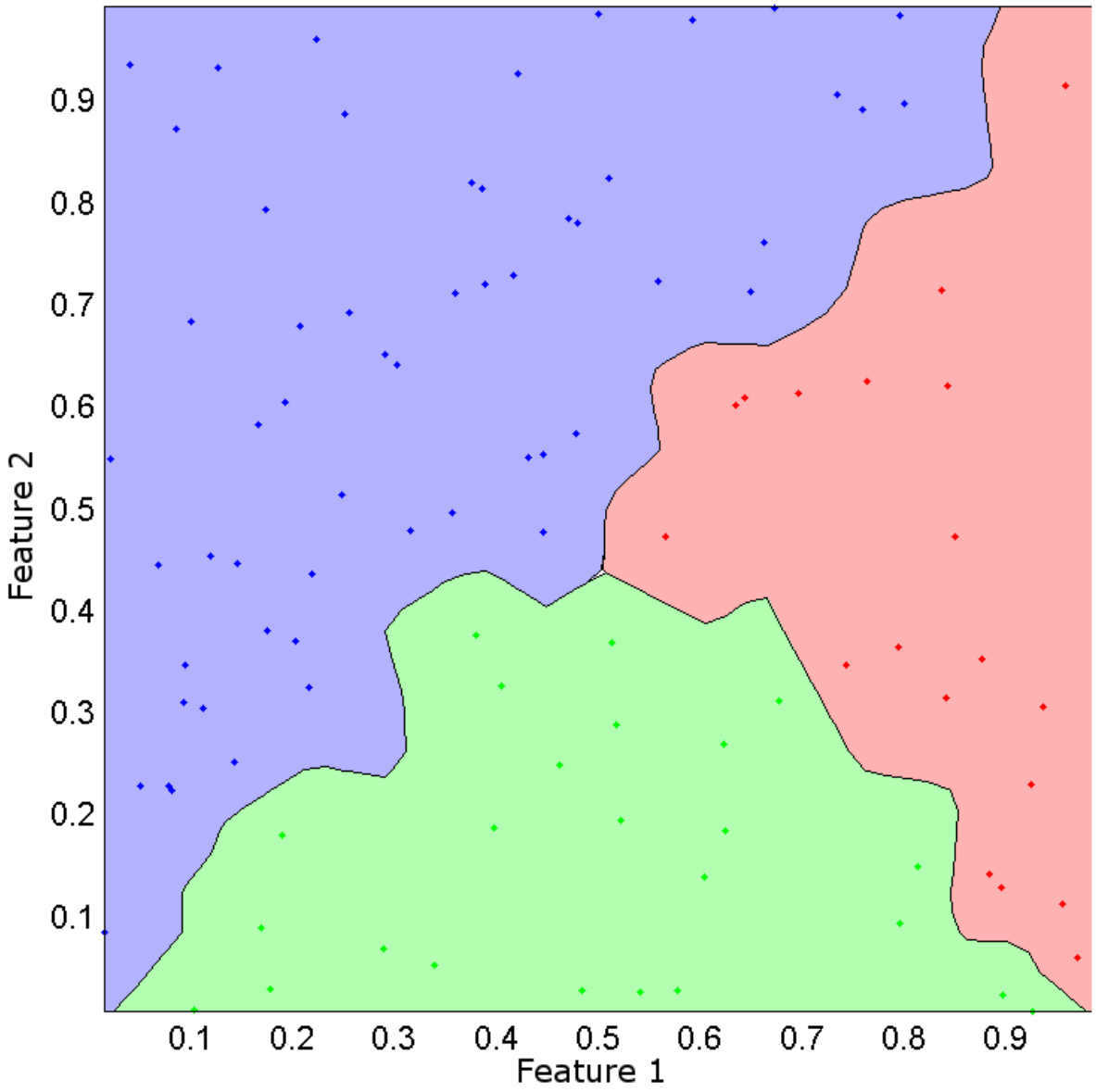
K-nn

k-nearest neighbors

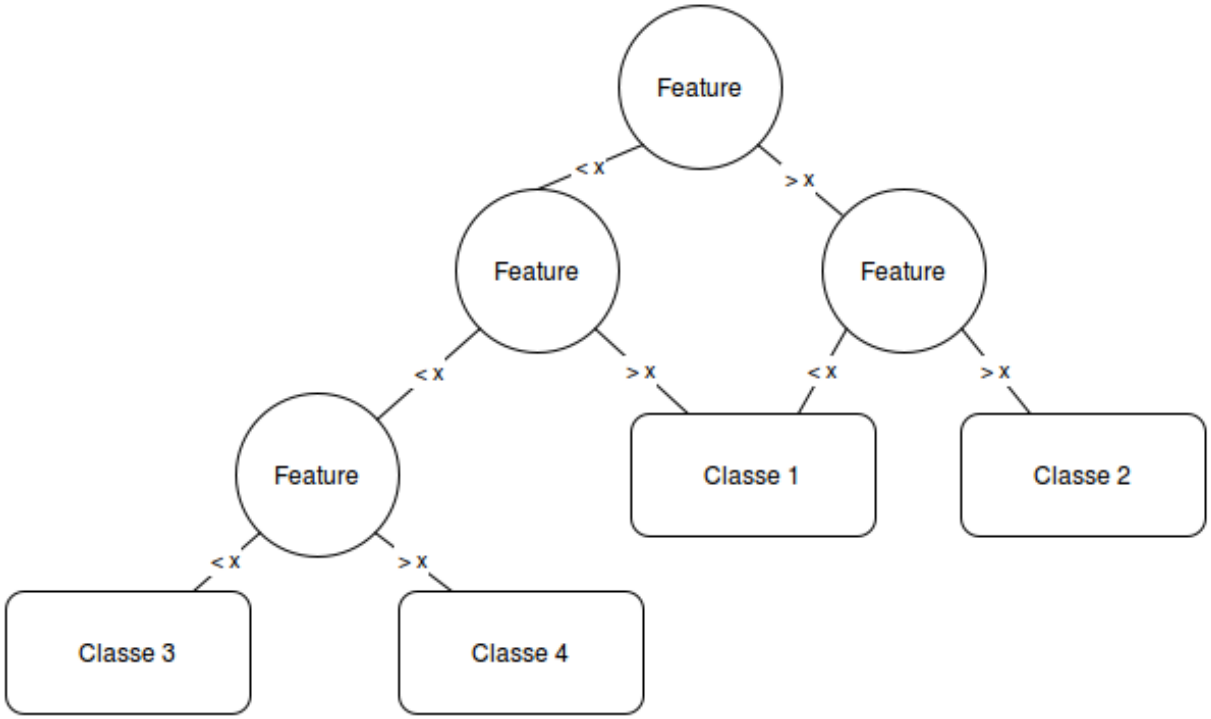
Points clefs

- Apprentissage supervisé
- Algorithme de classification
- Décision en fonction de la classe des k plus proches exemples du training set
- Vote à la majorité
- Couteux sur le calcul de la distance entre les points

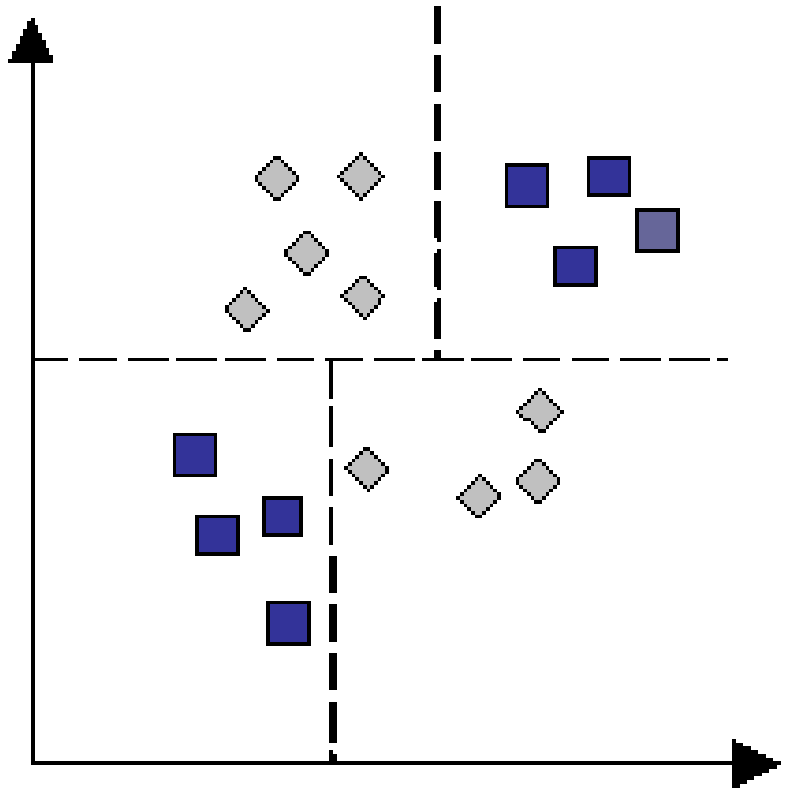
K-nn


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Feature 2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Feature 1

Decision tree



Decision tree



Decision tree : ID3

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Decision tree : ID3

- Inductive Dichotomiser 3
- Classifie selon des features booléennes ou parmis un set fini de classe
- Biais : Construction greedy de l'arbre basé sur une mesure d'entropie

Algorithme

1. Calcul du gain d'information pour chaque split possible
2. Choix de la feature présentant le gain d'information le plus fort pour splitter
3. Récursion jusqu'à ce que toutes les features aient été selectionné ou qu'il n'y ait plus d'exemples
4. Classe choisie en feuille : vote à la majorité

Decision tree : ID3

Entropie

$$H(S) = -\sum_{x \in X} p(x) log_2 p(x)$$

- X : L'ensemble des classes possibles
- p(x) : Ratio des elements de S qui sont de classe x

Gain d'information

$$G(F_k) = H(S) - \sum_{t \in T} p(t) H(T)$$

- T : Différents subsets après découpage selon F_k
- p(t) : Ratio d'elements de S qui sont de classe x

Random forest

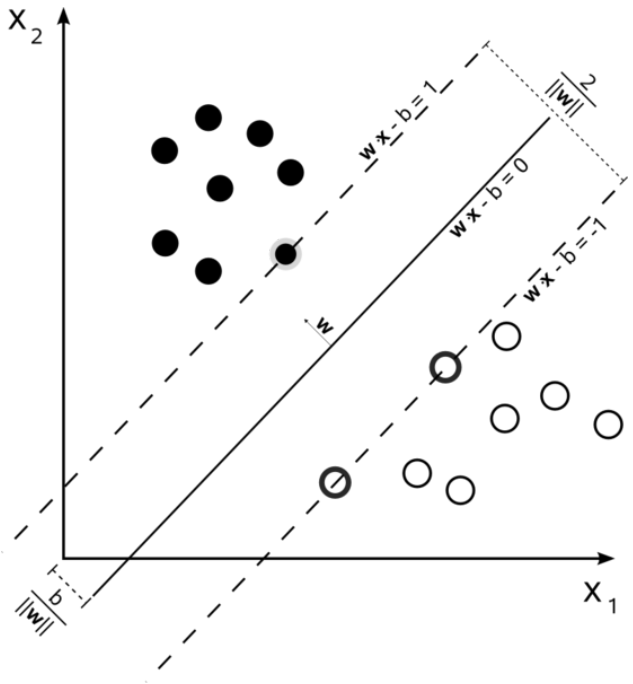
Tree bagging

Si l'on possède n arbres de decisions, on peut prendre la décision finale selon un vote à la majorité.

Random forest

- On construit n arbres de decisions sur un subset d'exemples du training-set.
- Chaque arbre est construit selon un subset de features aléatoire

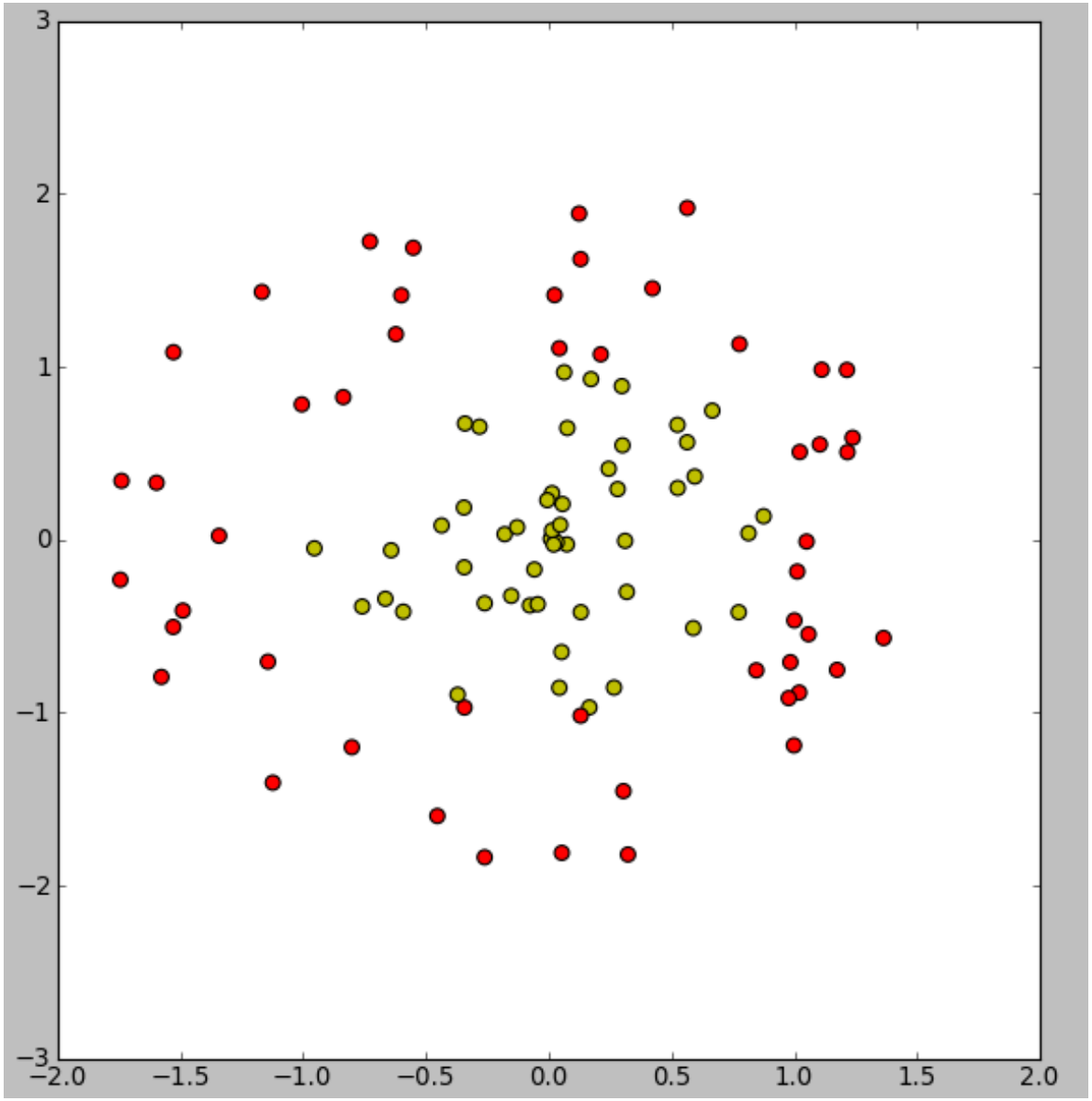
SVM



Objectif

Trouver un hyperplan séparateur à marge maximale de deux classes d'exemples.

Kernel Trick



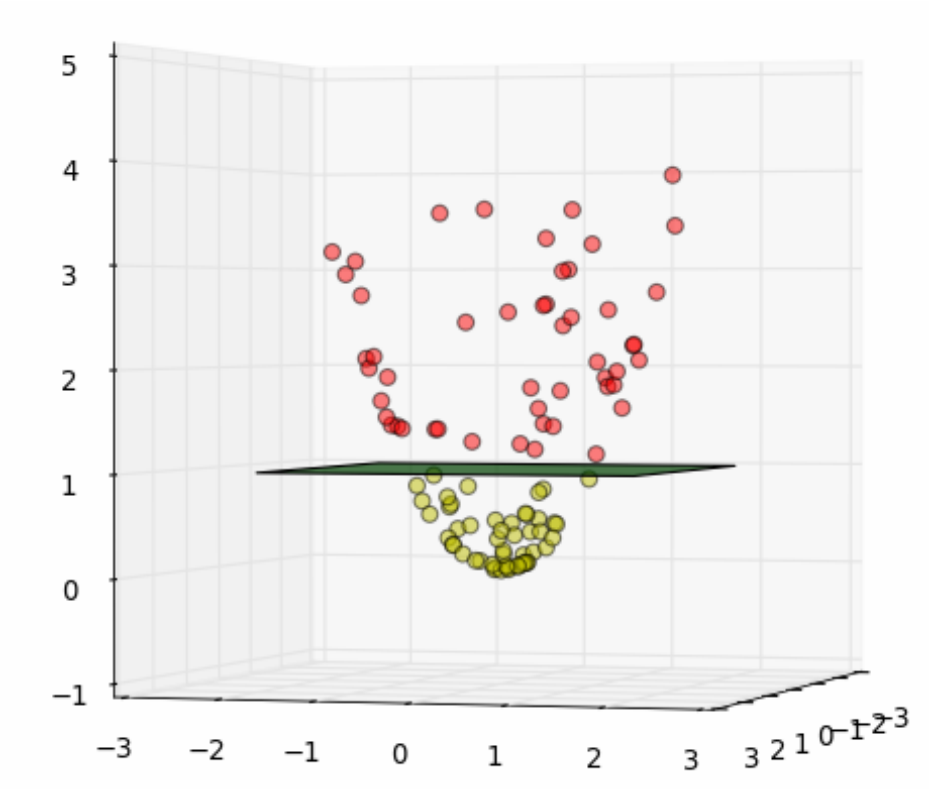
Kernel Trick

- On ne peut pas résoudre le problème avec un classifieur linéaire
- On projete les données dans un espace de dimension supérieure où il existe un hyperplan séparateur

Transformation vers l'espace de redescription

$\varphi : S \rightarrow V$ où V est appelé espace de redescription.

Kernel Trick



$$\varphi : (x, y) \mapsto (x, y, x^2 + y^2)$$

Kernel Trick

On peut exprimer le produit scalaire dans l'espace de redescription directement depuis l'espace d'origine grace à une fonction appelée kernel qui vérifie la condition de Mercer.

$K(x, y) = \langle \varphi(x), \varphi(y) \rangle_V$

Le calcul du produit scalaire ne coute plus O(dim(V)) ce qui pouvait mener à des calculs trop complexes.

Exemple de kernel

Kernel polynomial : $K(x, y) = (\langle x, y \rangle + k)^d$

Clustering et réduction dimensionnelle

Clustering

Il peut-être intéressant de partitionner les données en plusieurs clusters partageant des caractéristiques communes.

Intérêts

- Segmentation
- Extraction de connaissance
- Réduction du nombre d'exemples

Réduction dimensionnelle

Quand le nombre de features est important :

- Le modèle sera complexe à entrainer
- Il est probable que certaines features soient corrélées
- Certaines features peuvent s'avérer inutiles pour résoudre le problème

Dans ce cas, il est possible d'effectuer une réduction dimensionnelle avant l'algorithme d'apprentissage supervisé.

K-moyenne

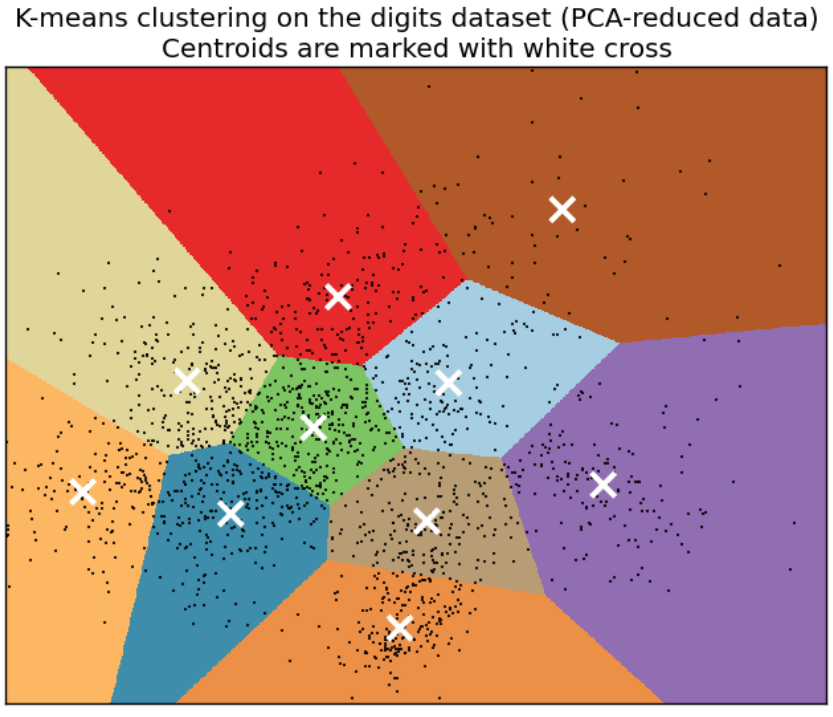
Points clefs

- Algorithme de clustering
- Basé sur la distance entre les points
- NP-difficile

Algorithme

1. Placer k barycentres dans l'espace
2. Assigner chaque point au barycentre le plus proche pour former k clusters
3. Recalculer les k barycentres de chaque cluster
4. Tant que les barycentres bougent, goto 2 !

K-moyenne : partition de Voronoi



Principal component analysis

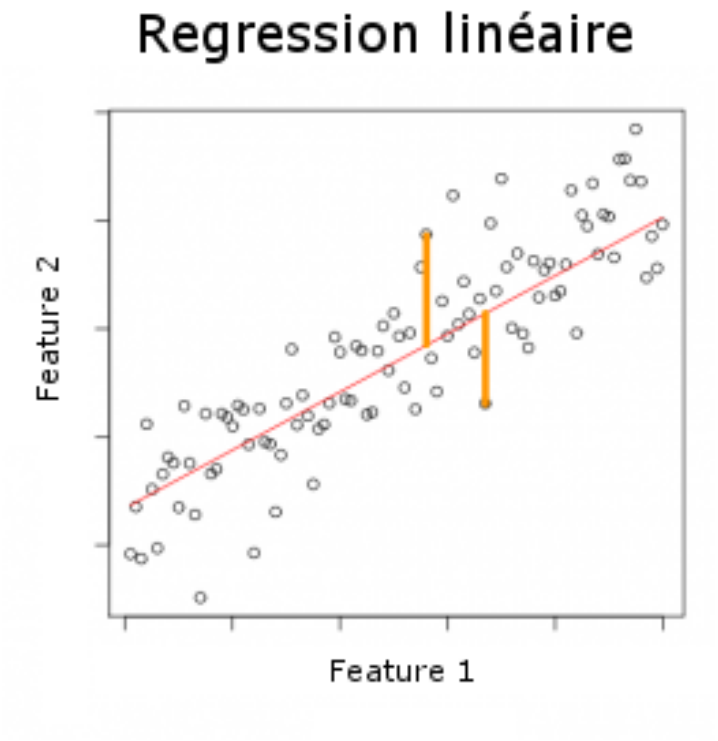
Objectif

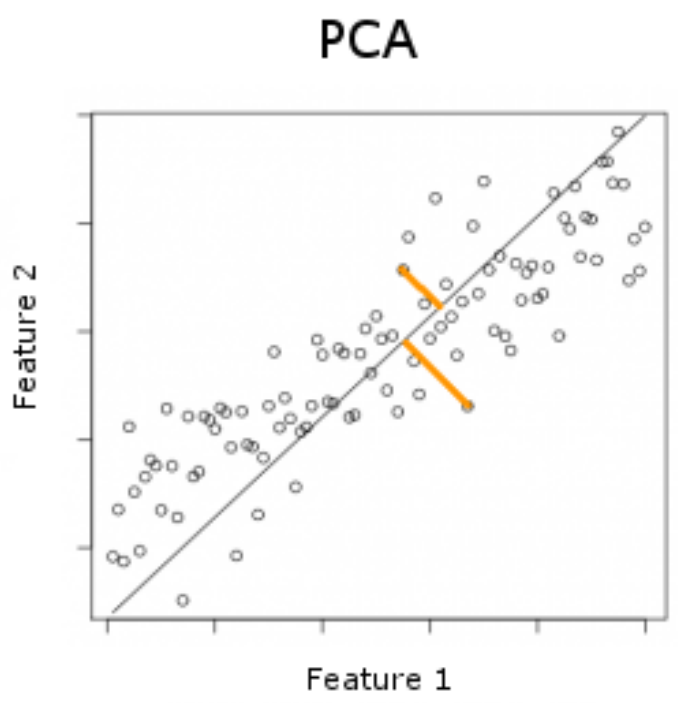
Décoreler ou débruiter des features en explicitant une transformation linéaire pour les projeter dans un espace plus représentatif.

Methode

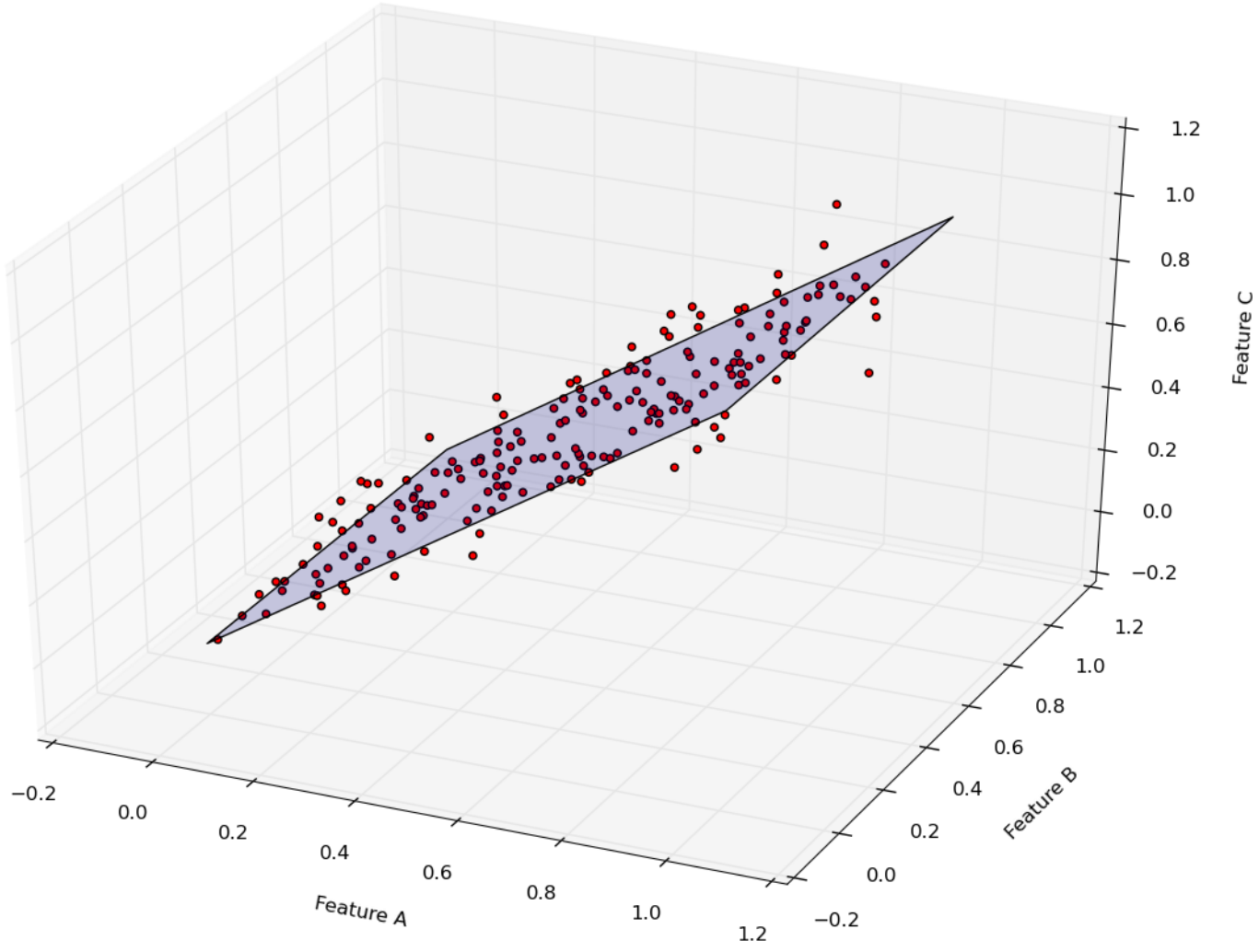
- Trouver une base orthogonale sur laquelle projeter les données de façon à maximiser la variance sur chaque axe.
- Trouver une base orthogonale sur laquelle projeter les données de façon à minimiser la distance entre les points et leurs projetés.

Principal component analysis





Principal component analysis



Principal component analysis

Covariance

Soient X et Y des vecteurs contenant chacun n valeurs d'une feature :

$$cov(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (X_i - moy(X))(Y_i - moy(Y))$$

Matrice de covariance

$$\Sigma = \begin{bmatrix} cov(X_1, X_1) & \dots & cov(X_1, X_k) \\ \dots & \dots & \dots \\ cov(X_1, X_k) & \dots & cov(X_k, X_k) \end{bmatrix}$$

Principal component analysis

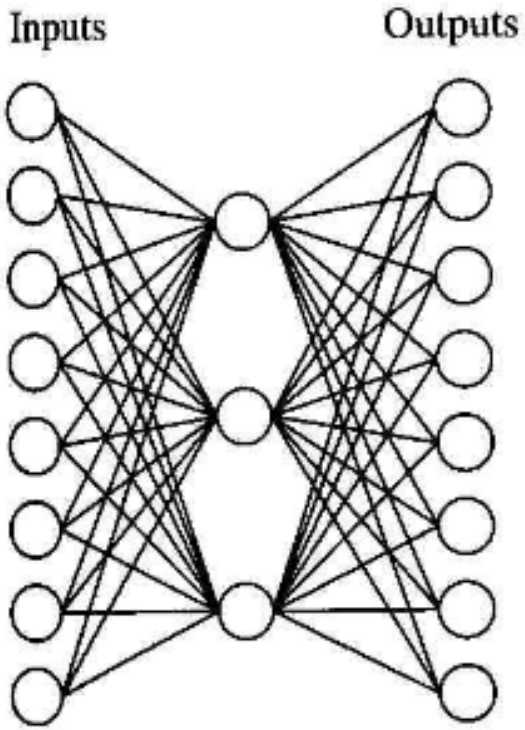
Réduction vers un espace de dimension k

1. Recentrer le centre du nuage de point à l'origine
2. Calcul de la matrice de covariance
3. Calcul des vecteurs propres et les valeurs propres de la matrice
4. Trier les vecteurs propres en fonction des valeurs propres
5. Construire la matrice B contenant k vec. propres (un par ligne)
6. Projeter les points via la tansformation $p(X) = BX$

Deep-learning

- Préentrainement non supervisé de réseaux de neurones multicouches
- Puis apprentissage supervisé du problème à résoudre en utilisant l'abstraction fournie par le préentrainement.
- Bons résultats dans la reconnaissance d'image (convolutionary NN)

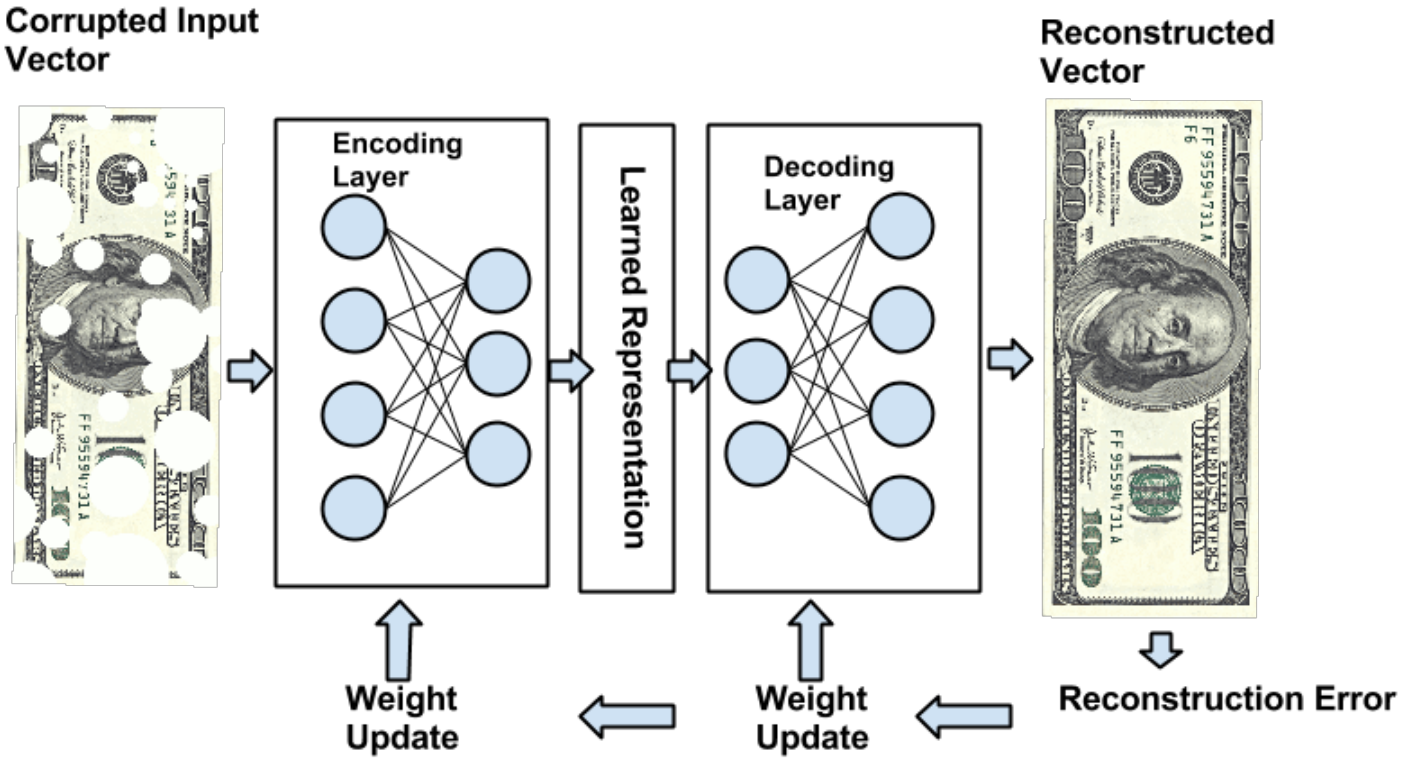
Autoencoders



Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.15	.99	.99	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.01	.11	.88	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

Denoising autoencoder

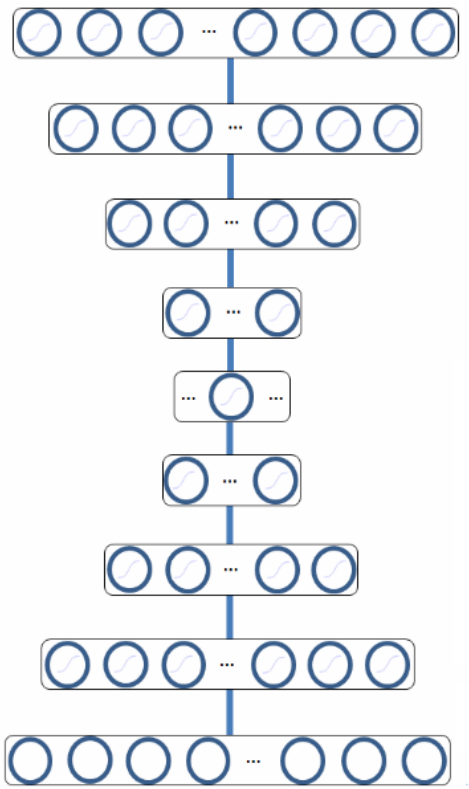
Globalement le même principe que l'autoencoder, mis à part que l'on rajoute du bruit dans les données et on demande à l'autoencoder de reconstruire les données sans le bruit.



Article : mrlabs.org/?p=62

Stacked denoising autoencoder

Entrainement successif de denoising autoencoders, en réduisant petit à petit le nombre de dimensions au fur et à mesure des couches.



Conclusion

- Python
  - scipy (numpy, matplotlib, iPython)
  - matplotlib (Dataviz, animations)
  - theano (Deep learning, transparent GPU computing)
  - iPython notebook (atelier de calcul scientifique webbrowser)
- C/C++
  - Boost uBlas (Basic linear algebra library)
- Matlab

Questions

Des questions ?

Contact

Mail : q.delaroussilhe@underflow.fr

Blog : http://www.underflow.fr