

Intelligence Artificielle et Jeux



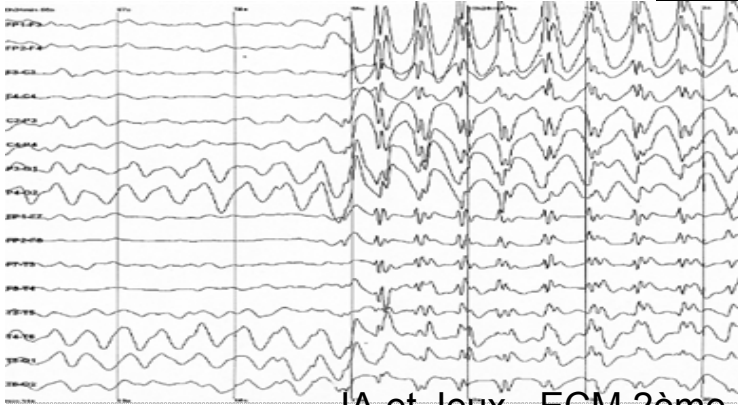
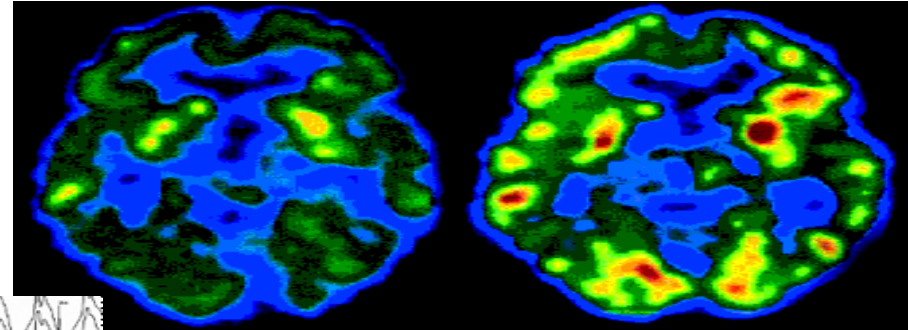
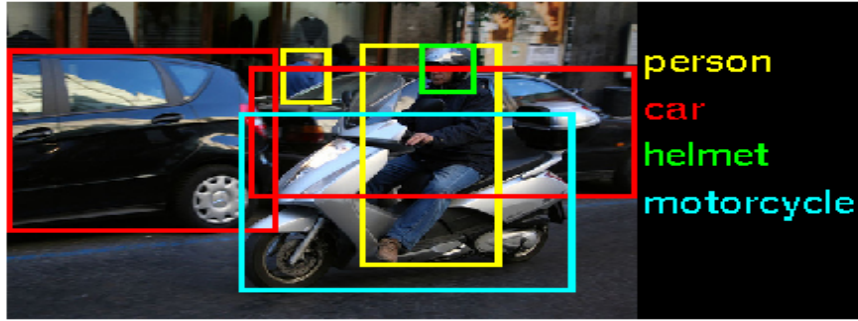
Thierry Artières

LE MACHINE LEARNING

Programmation traditionnelle

- Texte rédigé dans un langage informatique permettant par la réalisation successive d'opérations élémentaires de réaliser une tâche complexe
- Phases importantes
 1. **Phase d'analyse (fonctionnelle) ou de conception**
 2. Phase de réalisation ou de programmation
 3. Phase de livraison

Phase d'analyse ?



Le machine learning : un paradigme de programmation

Quel algorithme?



**A group of people
shopping at an
outdoor market.**

**There are many
vegetables at the
fruit stand.**

[Vinyals et al., 2014]

Autres applications



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."

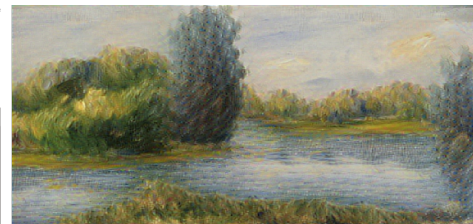


from his travels it might have been
from his travels it might have been
from his travels it might have been
from his travels it might have been
from his travels it might have been
from his travels it might have been

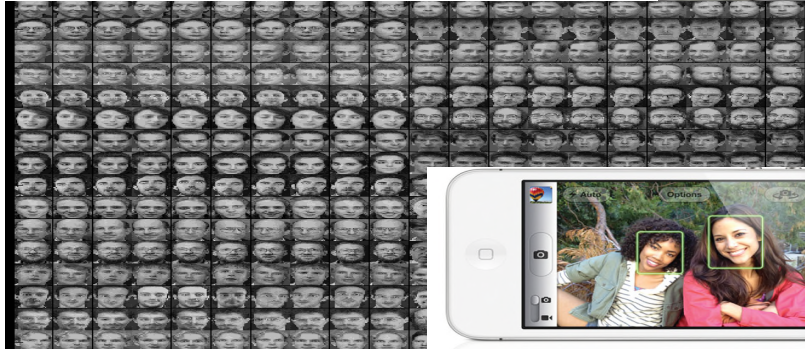
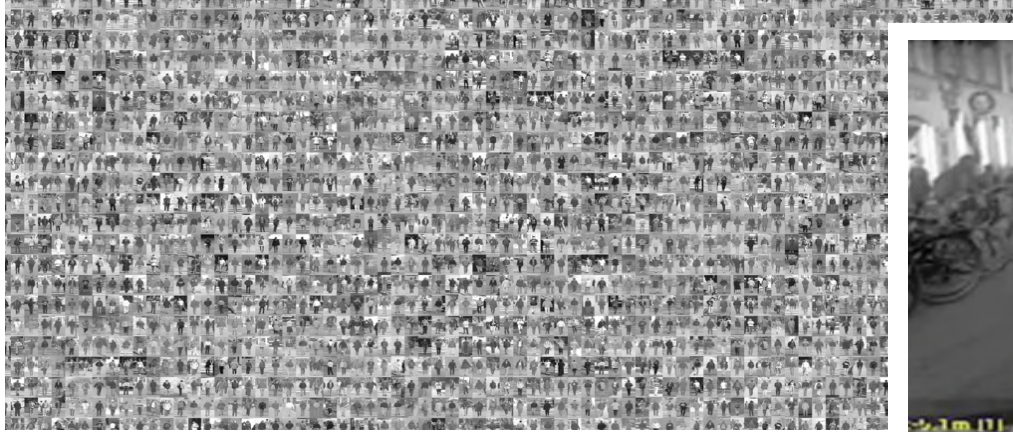
more of national temperament
more of national temperament
more of national temperament
more of national temperament
more of national temperament
more of national temperament



Synthesized Image



Apprentissage à partir d'exemples



L'Apprentissage Automatique

- Programme qui réalise une tâche sans qu'on ait une quelconque idée de l'algorithme

⇒ On apprend une fonction $y = f(x)$...

... avec un ensemble d'exemples (entrée , sortie)

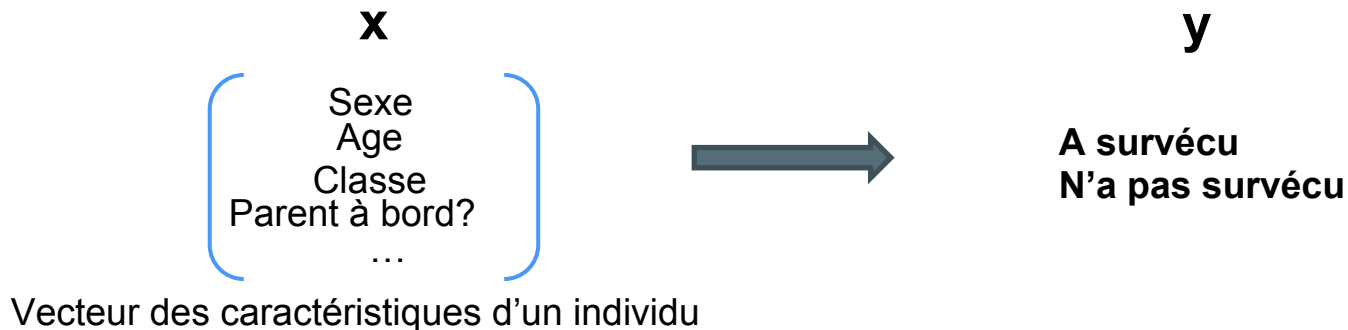


L'Apprentissage Automatique

- Programme qui réalise une tâche sans qu'on ait une quelconque idée de l'algorithme

⇒ On apprend une fonction $y = f(x)$...

... avec un ensemble d'exemples (entrée , sortie)



L'Apprentissage Automatique

- Programme qui réalise une tâche sans qu'on ait une quelconque idée de l'algorithme

⇒ On apprend une fonction $y = f(x)$...

... avec un ensemble d'exemples (entrée , sortie)

⇒ **Machine Learning = concevoir des algorithmes pour apprendre f**

L'Apprentissage Automatique

- Adapté
 - Pour des tâches de décision, de prédiction
 - Pour lesquelles on ne connaît pas d'algorithme
 - Pour lesquelles on ne veut pas passer par une étape de formalisation
- Une difficulté majeure
 - On met au point un système sur des données d'apprentissage
 - **On utilise le système sur d'autres données**

Tâches classiques du ML : ce que l'on ne sait pas programmer !

Tâches de perception : audition, vision

Langage naturel : classification, résumés, traduction

Prédiction : bourse, consommation énergie, maintenance prédictive, churn...

Recommandation à la Amazon, Netflix

Web advertising

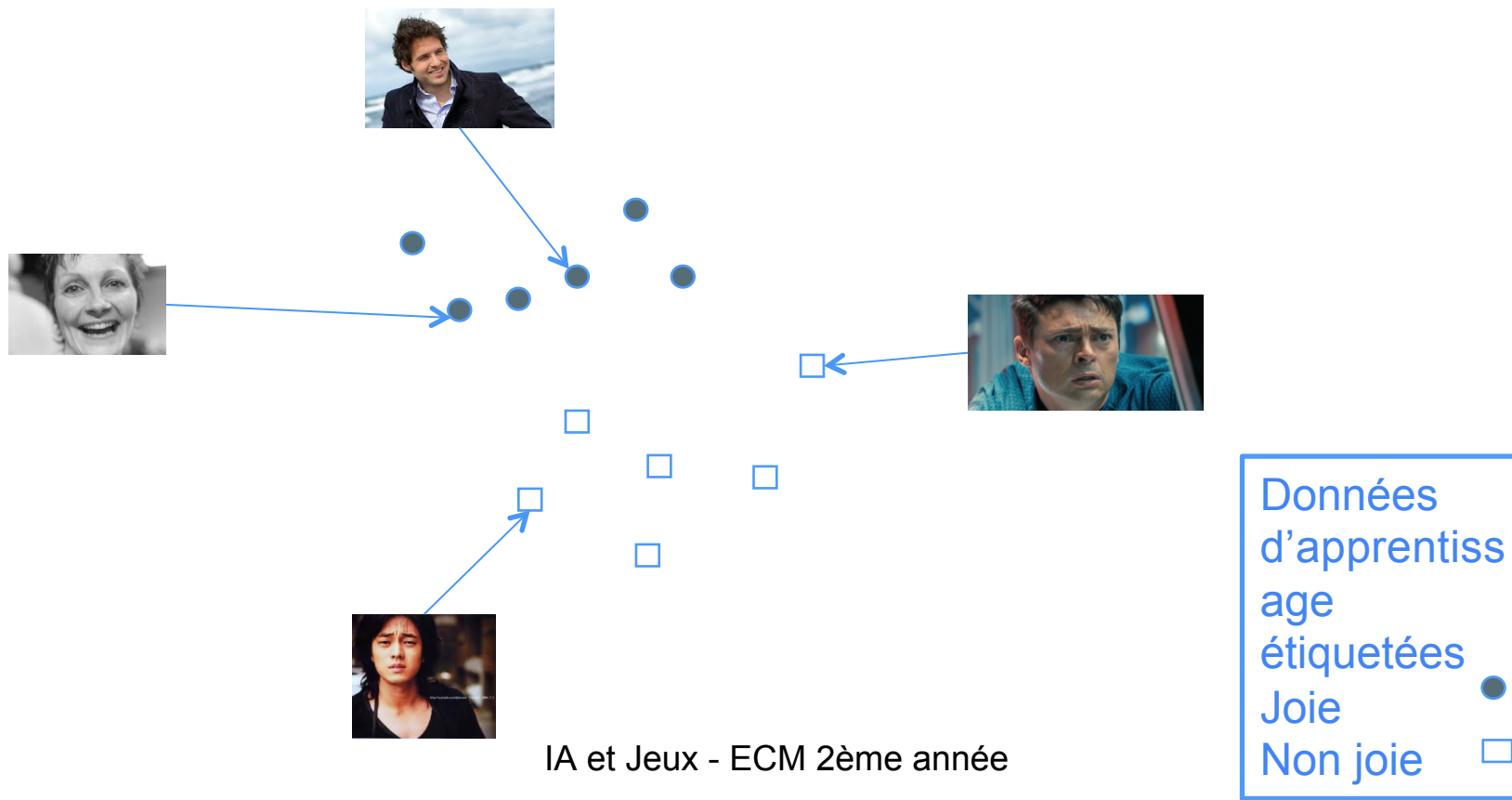
Raisonnement

Planification

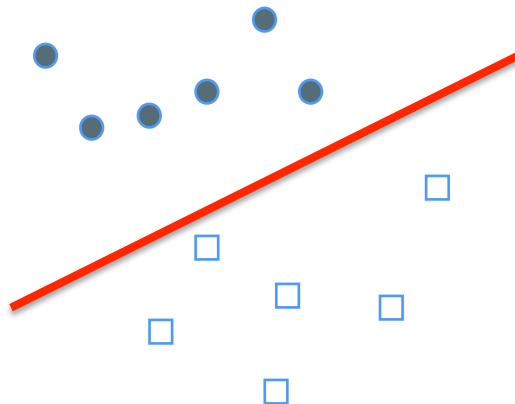
Robotique ...

LE PROBLÈME DE LA GÉNÉRALISATION

Classification d'images



Discrimination en 2D



Le modèle
(séparateur linéaire)
appris

L'apprentissage supervisé



Apprentissage supervisé

On cherche à **apprendre** une fonction : $y = f(x)$

- Où x = forme observée

Discret, réel, vecteur réel, etc

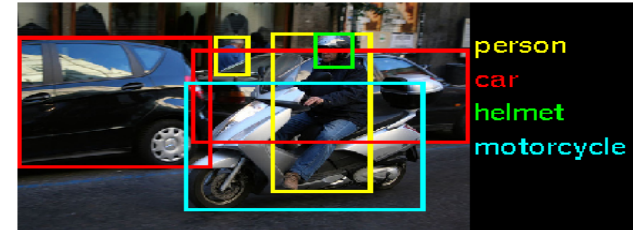
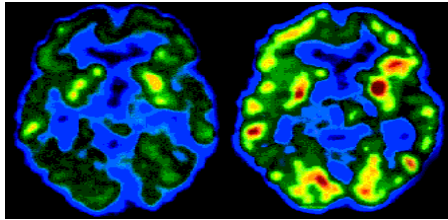
- y = sortie associée

Réel ou vecteur de réels (régression)

Variable catégorielle (classification)

- À partir d'un échantillon fini (i.i.d.) d'exemples $\{(x^i, y^i), i = 1..N\}$
 - Les observations sont des réalisations d'une variable aléatoire de loi inconnue
- En général on approxime f en cherchant dans une famille paramétrique $\{f_w, w \in \mathcal{W}\}$ celle qui se rapproche le plus de f (i.e. le w tel que f_w se rapproche le plus de f)

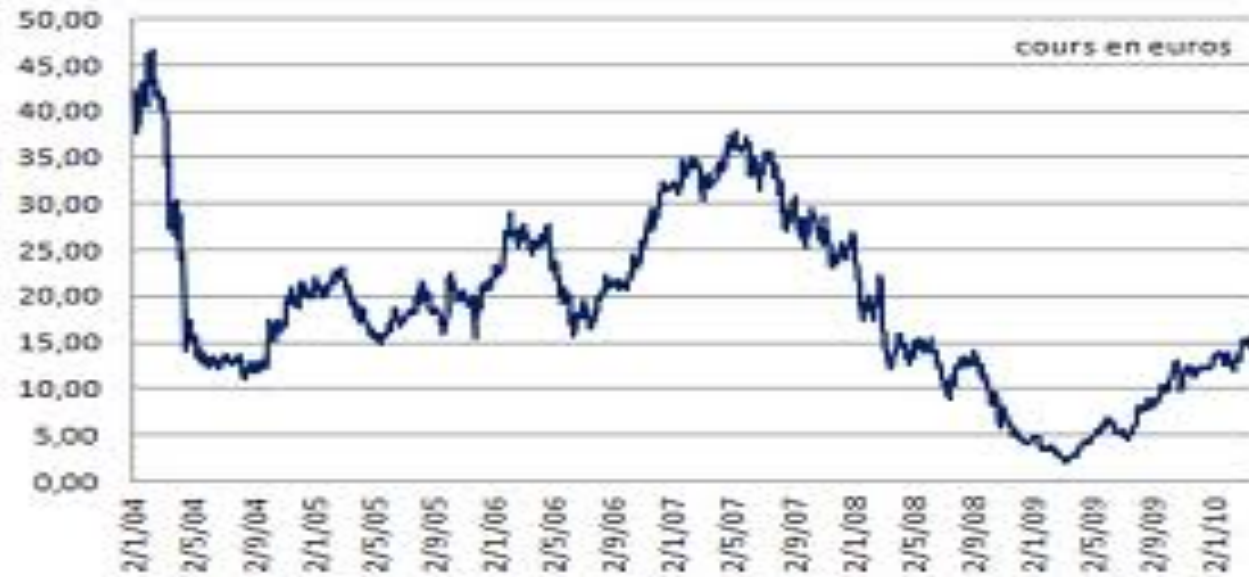
Discrimination



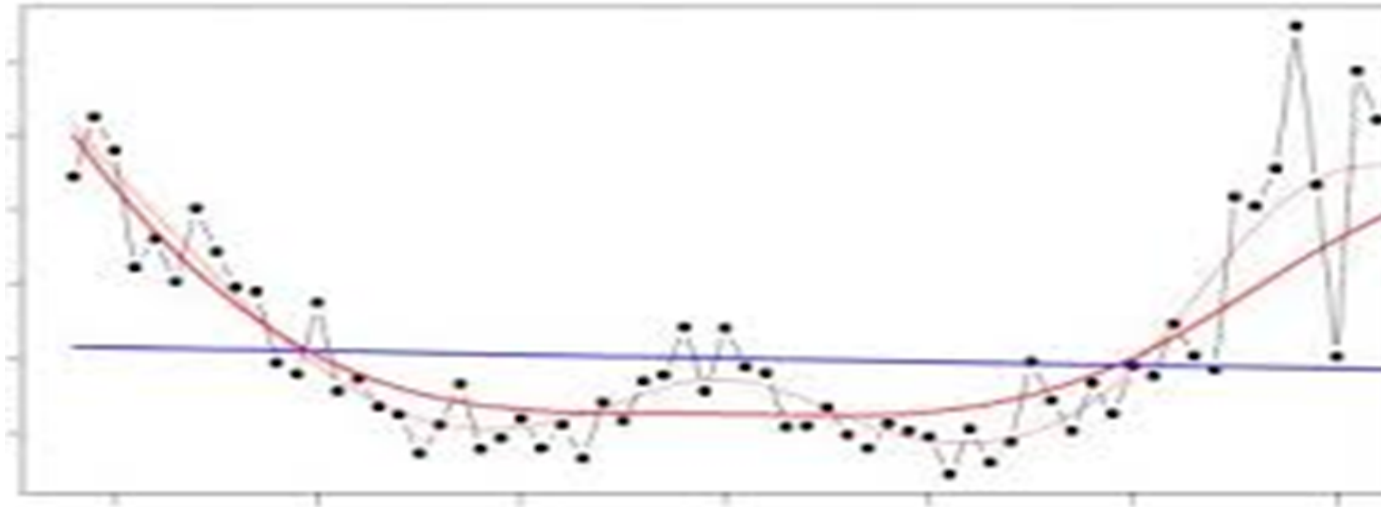
Titanic challenge



Prédiction



Régression



Une difficulté majeure

- On met au point un système sur des données
- **On utilise le système sur d'autres données**
 - **Notion de capacité de généralisation**

UN PREMIER CLASSIFIEUR SIMPLE : LES K PLUS PROCHES VOISINS

- Principe
 - Pour classer un exemple x
 - On détermine dans la base d'apprentissage les K exemples qui sont les plus proches de x
 - On regarde parmi ces K voisins la classe majoritaire
 - On reconnaît x de cette classe
- Intérêt
 - Modèle performant asymptotiquement
- Points faibles
 - Modèle performant asymptotiquement
 - Ce n'est pas un modèle à proprement parler
 - Pas de synthèse des données d'apprentissage

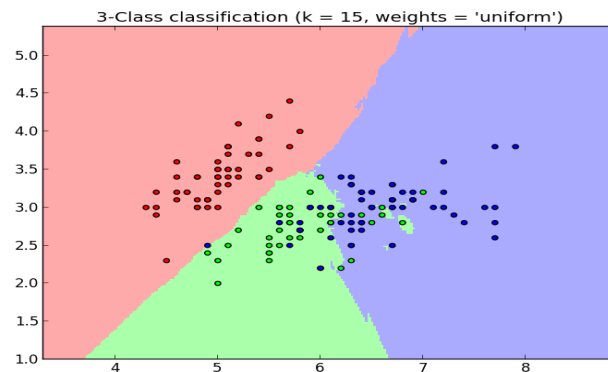
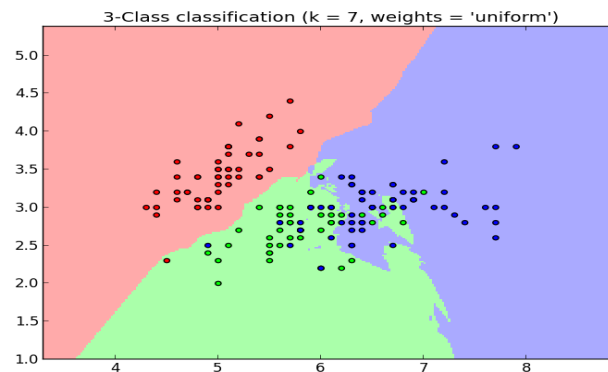
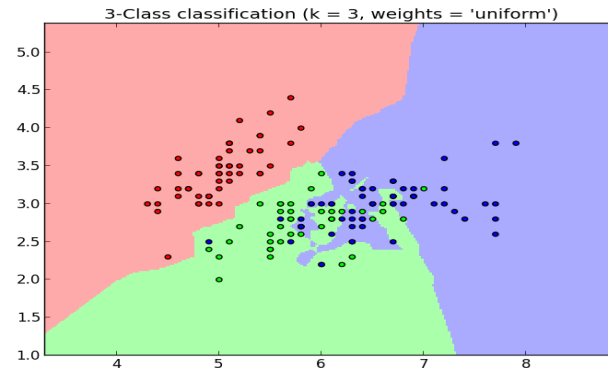
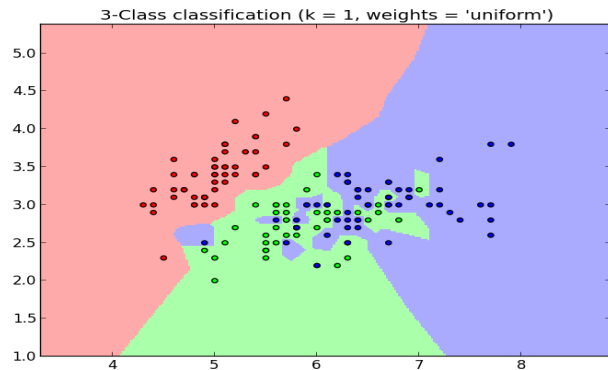
Les K ppv comme estimateurs de densité

- Pour un ensemble de données $D=\{x^i\}$, on fixe k

$$\Rightarrow \hat{p}(x) \equiv \frac{k}{NV} \approx p(x)$$

- Pour un point x quelconque, on considère une hypersphère centrée en x , on la fait grandir jusqu'à ce qu'elle englobe k points, soit V son volume

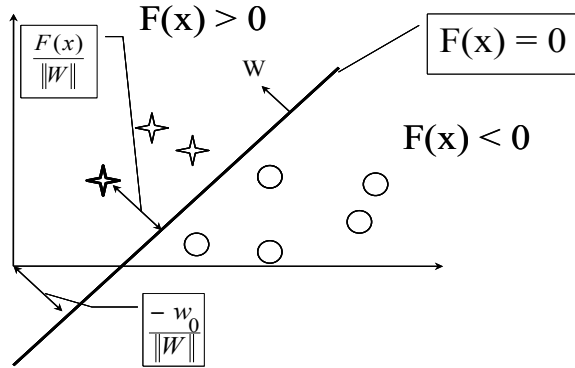
$$k \propto N^{1/2}$$



Nombre de voisins

UN PREMIER MODÈLE DE CLASSIFICATION SUPERVISÉE : LE PERCEPTRON [ROSENBLATT 1970]

Fonction discriminante linéaire



$$F(x) = \langle w, x \rangle = w^t x$$

- Avec vecteur forme étendu
 $x = (1, x_1, \dots, x_p)$ et $w = (w_0, w_1, \dots, w_p)$
- Surface de décision: hyperplan
 $F(x) = 0$

- Si x est un vecteur forme étendu

– Cas à 2 classes

$$F(x) = w^t x = \sum_{i=1}^p w_i x_i \quad \left\{ \begin{array}{l} > 0 \Rightarrow x \in C_1 \\ < 0 \Rightarrow x \in C_2 \end{array} \right.$$

Algorithme d'apprentissage

- Entrées :

Ensemble d'apprentissage $D = \{(x^i, y^i), i = 1..N\}$

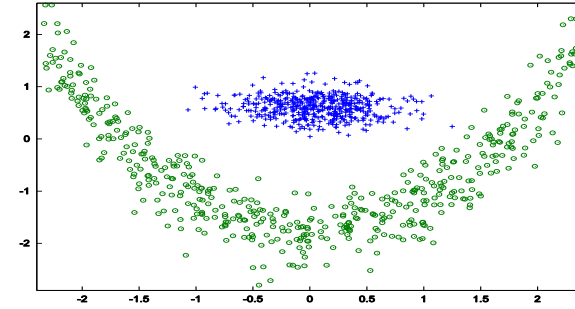
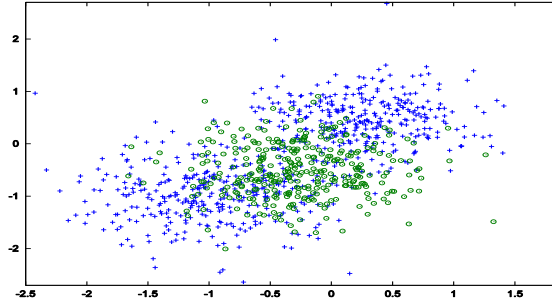
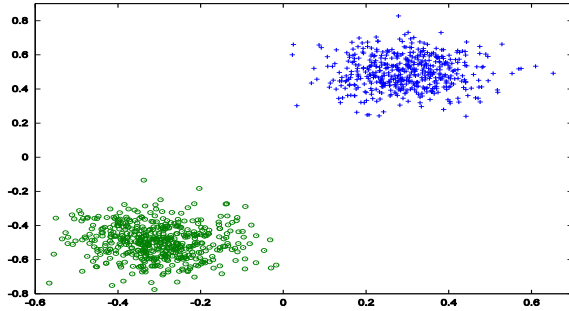
- Choisir un w initial
- Itérer jusqu'à [critère d'arrêt satisfait]
 - Choisir au hasard un exemple $i \in \{1, \dots, N\}$
 - Si l'exemple i est bien classé $\Leftrightarrow y^i \times \langle w, x^i \rangle > 0$
 - Alors : Ne rien faire
 - Sinon : Modifier w suivant
$$w = w + y^i x^i$$

Arrêt si

- Aucune erreur en apprentissage
- Nb Itérations maximal atteint

Critère d'apprentissage : $\sum_{i=1}^N 1_{y^i \times F(x^i) < 0}$

INTERET ET LIMITES DU PERCEPTRON



Si les données d'apprentissage ne sont pas linéairement séparables

- Pas de convergence
- Pas de solution optimale

Comportement de l'apprentissage

- Séquence chaotique de solutions durant l'apprentissage

[http://scikit-learn.org/stable/modules/generated/
sklearn.linear_model.LogisticRegression.html](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

LA RÉGRESSION LOGISTIQUE

Régression logistique

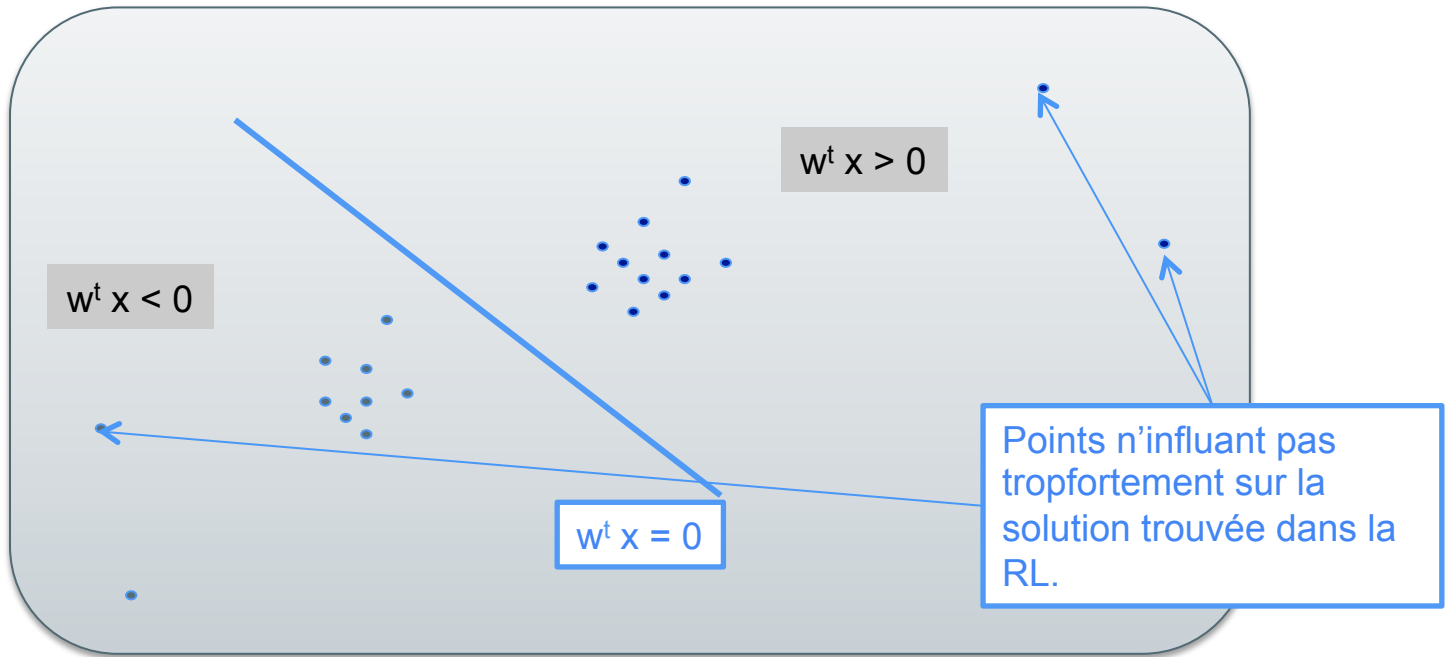
- Modèle discriminant pour la classification
- Version à 2 classes

$$p(C = 1|x) = \frac{1}{1 + e^{-w^t x}}$$

- Apprentissage par maximisation de la vraisemblance

$$\hat{w} = \operatorname{argmax}_w \prod_{i=1}^N P(C_i|x^i)$$

- Mieux adapté à la classification que des classifieurs linéaires appris par minimisation d'un critère d'erreur quadratique



Maximisation de la vraisemblance

- Vraisemblance

$$L = \log \left(\prod_{i=1}^N P(C_i | x^i) \right) = \sum_{i=1}^N \log P(C_i | x^i)$$

- Avec :

$$P(C_i | x^i) = P(C_i = 1 | x^i)^{C_i} [1 - P(C_i = 1 | x^i)]^{1-C_i}$$

- Donc :

$$L = \sum_{i=1}^N C_i \times \log P(C_i = 1 | x^i) + (1 - C_i) \times \log [1 - P(C_i = 1 | x^i)]$$

- **Optimisation par descente de gradient.**

En bref

- Classifieur linéaire
- Meilleur comportement que le perceptron pour les points loin de la frontière de décision
- Interprétation probabiliste des sorties
- Possibilité de modifier la fonction de coût pour intégrer des critères additionnels

FORMALISATION DE L'APPRENTISSAGE COMME UN PROBLÈME D'OPTIMISATION

- Base d'apprentissage

$$\{(x^i, y^i), i = 1..N\}$$

- Critère d'apprentissage

$$R_{emp}(w) = \sum_{i=1}^N l(f_w(x^i), y^i)$$

- Fonction de perte : loss

- Erreur quadratique
- 0/1 loss
- ...

Risque réel et empirique

Le risque empirique

$$R_{emp}(w) = \sum_{i=1}^N l(f_w(x^i), y^i)$$

est une estimation du risque réel

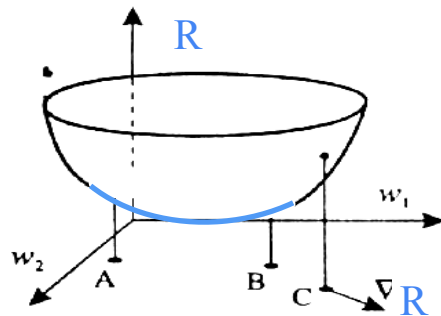
$$R(w) = \int_{(x,y)} l(f_w(x), y) p(x, y) dx dy$$

L'apprentissage comme un problème d'optimisation

- Trouver le meilleur modèle sur la base d'apprentissage

$$R_{emp}(w) = \sum_{i=1}^N l(f_w(x^i), y^i)$$

= Optimisation d'une fonction de w



- Trouver le meilleur modèle est un peu différent

ESTIMATION DE LA PERFORMANCE ET SELECTION DE MODELE

IA et Jeux - ECM 2ème année

Objectifs

- Quel est le meilleur modèle de régression pour un jeu de données fixé ?
 - Structure du modèle
 - Paramètres du modèle
 - Nécessité de bien estimer la performance
 - Pour une structure de modèle donnée
 - Pour le jeu de données disponible
- ⇒ Découpage Train / Validation / Test
- ⇒ Multiples découpages Train / Test (Cross Validation) pour :
- ⇒ estimation de la variance des résultats
 - ⇒ Exploiter toutes les données en test
- ⇒ Grid Search pour le réglage de ce qui ne s'optimise pas numériquement

Train / Validation / Test

- Découpage des données en 3 parties
 - Training Set : pour l'optimisation des paramètres du modèle
 - Validation Set : pour le choix du meilleur modèle si on en teste plusieurs
 - Test Set : pour l'estimation de l'erreur en généralisation du meilleur modèle
- Mais
 - Limite la taille des données d'apprentissage : on apprend moins bien
 - Limite la taille des données d'évaluation : on évalue moins bien