Basics of Machine Learning

SD 210 - P3

Lecture 2 - Linear classifiers

Florence d'Alché-Buc

Contact: florence.dalche@telecom-paristech.fr,
2A Filière SD, Télécom ParisTech,Université of Paris-Saclay, France

Table of contents

Outline

Statistical learning: a methodology

- Three main problems to be solved :
  - **Representation problem**: determine in which representation space the data will be encoded and determine which family of mathematical functions will be used
  - **Optimization problem (focus of the course)**: formulate the learning problem as an optimization problem, develop an optimization algorithm
  - **Evaluation problem**: provide a performance estimate

Statistical learning for supervised classification

Two main family of approaches:

1. Discriminant approaches : just find a classifier which does not estimate the Bayes classifier
2. Generative probabilistic approaches that are built to model $h(x) = \hat{P}(Y = 1|x)$ using $\hat{p}(x|Y = 1)$, $\hat{p}(x|Y = -1)$ and prior probabilities.

Outline

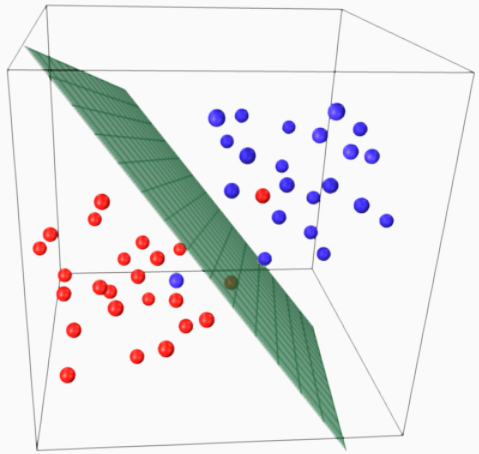
Classifieur linéaire

Définition

Supposons $\mathbf{x} \in \mathbb{R}^p$

$f(\mathbf{x}) = \text{signe}(h(x)) = \text{signe}(\mathbf{w}^T\mathbf{x} + w_0)$

L'équation : $\mathbf{w}^T\mathbf{x} + w_0 = 0$ définit un hyperplan dans l'espace euclidien $\mathbb{R}^p$



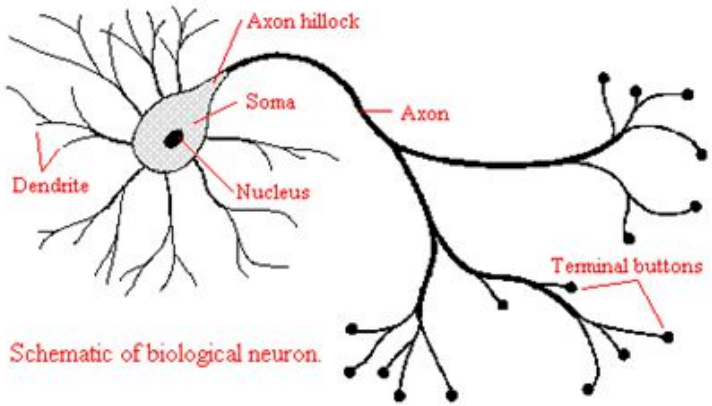
How to learn a linear classifier ?

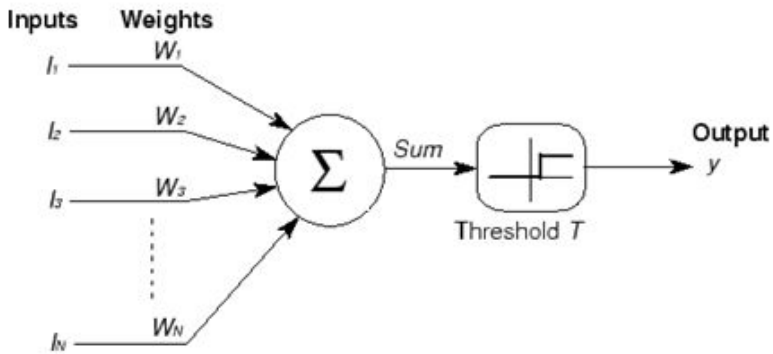
- Model: perceptron or formal neuron (Rosenblatt 1957, 1959)
- Learning algorithm: formerly, perceptron rule, then (stochastic) gradient descent algorithm for perceptron

A linear classifier: the formal neuron and perceptron

- First model proposed by McCullogh and Pitts (physiologists) in 1943 to model the activity of a neuron
- Input signals represented by a vector $\mathbf{x}$ is processed by a neuron whose weighted synapses are linked to the input
- The neuron computes a weighted sum of the components of the signal
- Rosenblatt proposed a learning rule in 1959

Le neurone

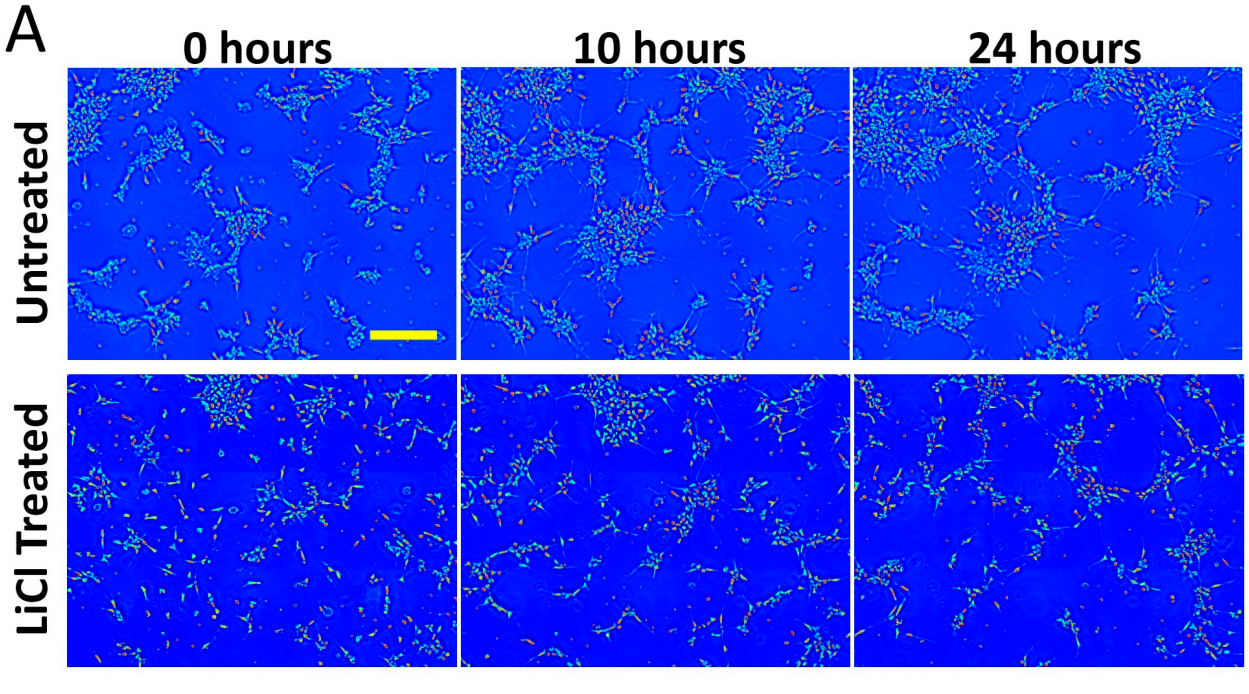




Neurone biologique

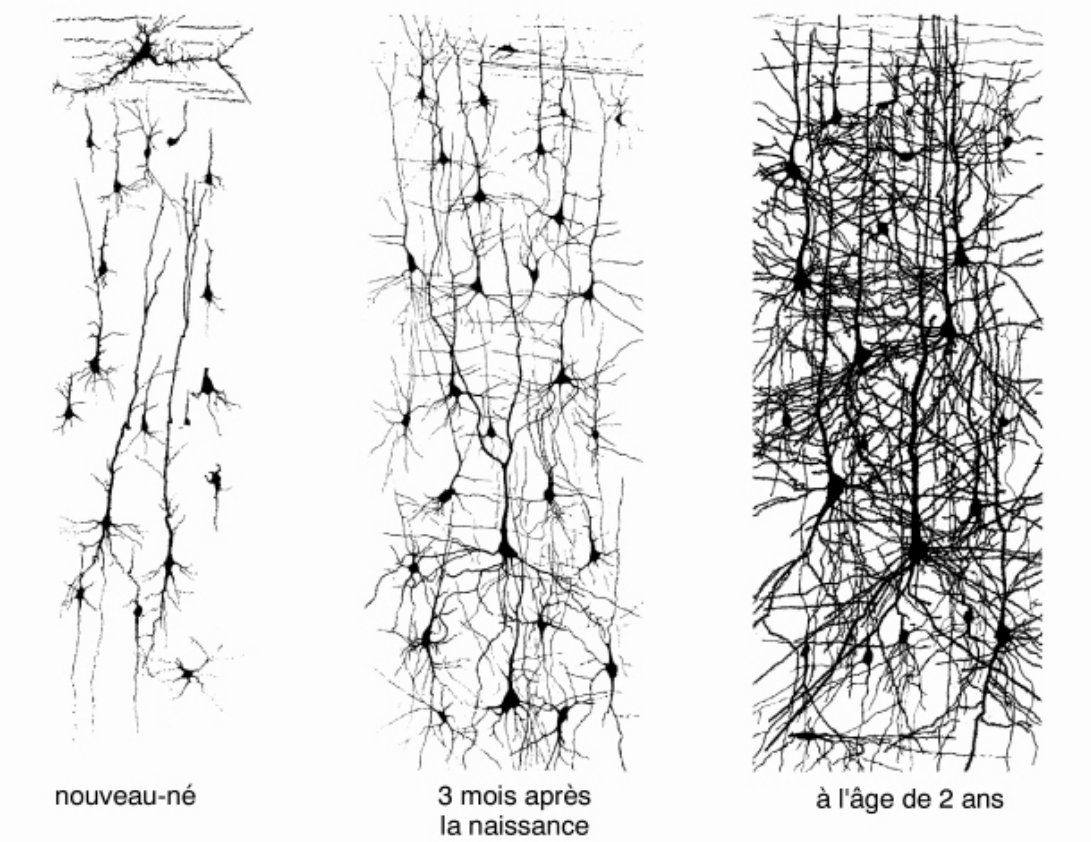
Neurone artificiel

Neuron network growth over 24 hours



In 2014, the group of Gabriel Popescu at Illinois U. visualized a growing net of stem cell neurons using spatial light interference microscopy (SLIM). Ref: `http://light.ece.illinois.edu/wp-content/uploads/2014/03/Mir_SRep_2014.pdf`
Video: `https://youtu.be/KjKsU_4s0nE`

Développement des réseaux de neurones chez l'enfant



Développement des réseaux de connections entre les neurones chez l'enfant.

Re: Museum de Toulouse `http://www.museum.toulouse.fr/-/connecte-a-vie-notre-cerveau-le-meilleur-des-reseaux-2-3-`

Formal neuron and perceptron

- $h_{perc}(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$
- $sign(a) = 1$ if $a \geq 0$ and -1 otherwise

Données d'apprentissage:

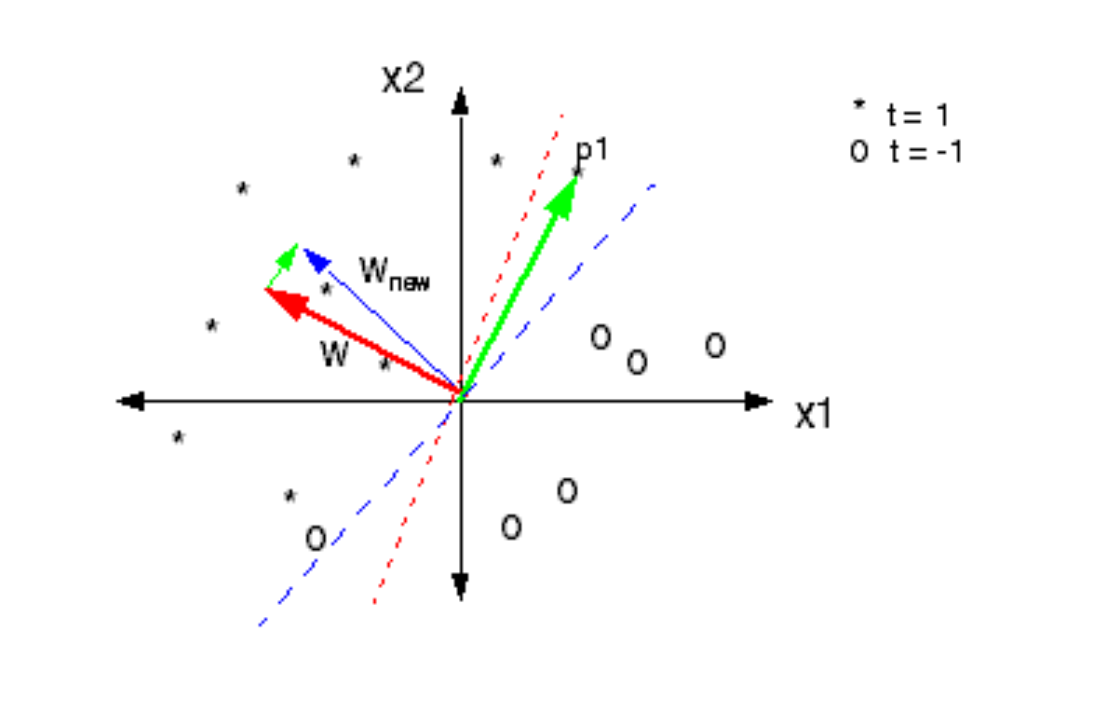
- $\mathcal{S} = \{(\mathbf{x_1}, \mathbf{y_1}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$
- $\mathbf{x}_i \in \mathbb{R}^{p+1}$: the 0^{th} component is fixed to 1.
- $y_i \in \{-1, +1\}$

Algorithme classique du perceptron

Algorithme (pseudo code)

- Continue $= 1$
- Fixer T nombre maximal d'itérations
- $\mathbf{w}_0 = 0$
- $k = 0$
- $\epsilon << 1$
- TANT QUE (continue $> \epsilon$) ou ($nk < T$) FAIRE
  - Pour i=1 à n
    - $k = k + 1$
    - Si $y_i \neq sign(\mathbf{w}^T \mathbf{x}_i)$, alors je corrige: $\mathbf{w}(k+1) = \mathbf{w}(k) + y_i.\mathbf{x}_i$
    - Sinon pas de correction
  - CONT $= \|\mathbf{w}(k+1) - \mathbf{w}(k)\|$

Correction effectuée par le perceptron



Inteprétation en terme de "loss function":

$$\ell(y, \mathbf{w}^T x) = \max(0, -y\, \mathbf{w}^T \mathbf{x}))$$

Convergence de l'algorithme du perceptron

L'algorithme converge si les données sont exactement linéairement séparables.

Théorème de convergence

Supposons qu'il existe un paramètre $\mathbf{w}^*$ tel que $\|\mathbf{w}^*\| = 1$, et $\gamma > 0$ tels que pour tout $i = 1, \ldots n$:

$$y_i(\mathbf{x}_i^T \mathbf{w}^*) \geq \gamma$$

et qu'il existe $R > 0$: $\|\mathbf{x}_i\| \leq R$,
Alors l'algorithme du perceptron converge en au plus $\frac{R^2}{\gamma^2}$ itérations.

Preuve 1/2

- $\mathbf{w}(0) = 0$
- Supposons que la k-ieme erreur est faite sur l'exemple d'indice t, nous avons:

$$\begin{aligned} \mathbf{w}(k+1)^T\mathbf{w}^* &= (\mathbf{w}(k) + y_t\mathbf{x}_t)^T\mathbf{w}^* \\ &= \mathbf{w}(k)^T\mathbf{w}^* + y_t\mathbf{x}_t^T\mathbf{w}^* \\ &\geq \mathbf{w}(k)^T\mathbf{w}^* + \gamma \end{aligned}$$

Par récurrence sur k : $\mathbf{w}(k+1)^T\mathbf{w}^* \geq k\gamma$
(Par Cauchy-Schwartz: $\|\mathbf{w}(k+1)^T\mathbf{w}^*\| \leq \|\mathbf{w}(k+1)\|\|\mathbf{w}^*\| \leq \|\mathbf{w}(k+1)\|$)
donc : $\|\mathbf{w}(k+1)\| \geq \|\mathbf{w}(k+1)^T\mathbf{w}^*\| \geq k\gamma$

Preuve 2/2

On dérive ensuite une majoration pour $\|\mathbf{w}(k+1)\|$:

$$\begin{aligned} \|\mathbf{w}(k+1)\|^2 &= \|\mathbf{w}(k) + y_t \mathbf{x}_t\|^2 \\ &= \|\mathbf{w}(k)\|^2 + y_t^2 \|\mathbf{x}_t\|^2 + 2 y_t \mathbf{x}_t^T \mathbf{w}(k) \end{aligned}$$

le terme de correction $2 y_t \mathbf{x}_t^T \mathbf{w}(k)$ est par définition négatif donc:

$$\|\mathbf{w}(k+1)\|^2 \leq \|\mathbf{w}(k)\|^2 + R^2$$

Par récurrence sur k, on a :

$$\|\mathbf{w}(k+1)\|^2 \leq kR^2$$

Au final, en prenant les deux inégalités:
on a : $k^2\gamma^2 \leq \|w(k+1)\|^2 \leq kR^2$ donc : $k \leq \frac{R^2}{\gamma^2}$
Si k est borné par $\frac{R^2}{\gamma^2}$, cela veut dire qu'en au plus $\frac{R^2}{\gamma^2}$ itérations, on n'a plus de corrections à faire.

Limitations du perceptron

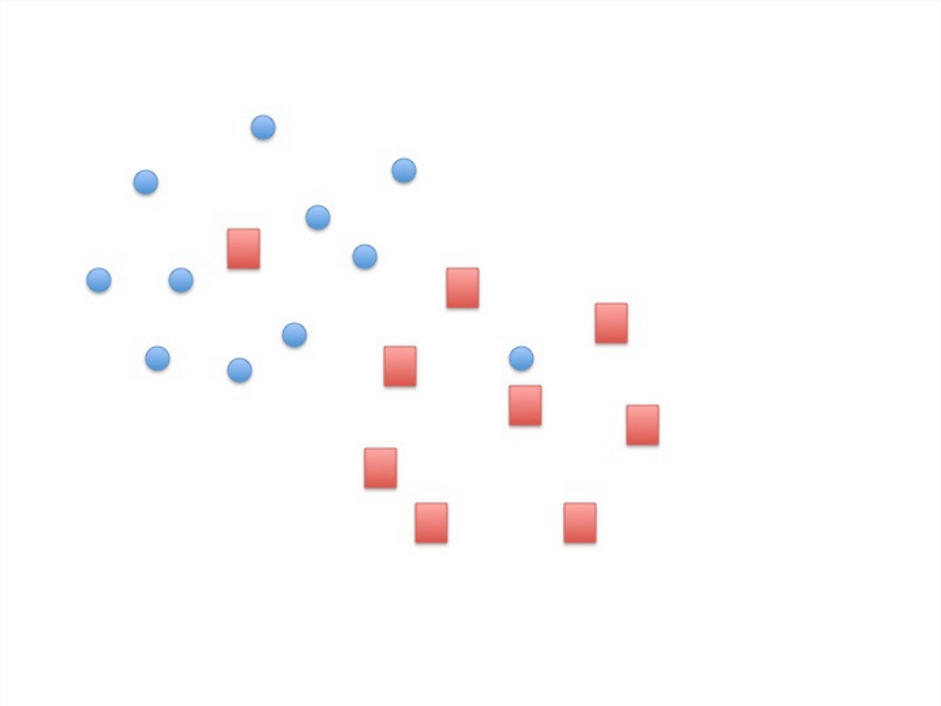
Non séparabilité linéaire:

1. Données presque " linéairement séparables": quelques "outliers" dans les données, qu'il vaut mieux éviter d'apprendre à classer : la règle du perceptron ne converge pas
2. Données séparables mais avec une frontière non linéaire : la règle du perceptron ne converge pas
3. NB : on cumule en général les deux difficultés

Limites d'un perceptron 1

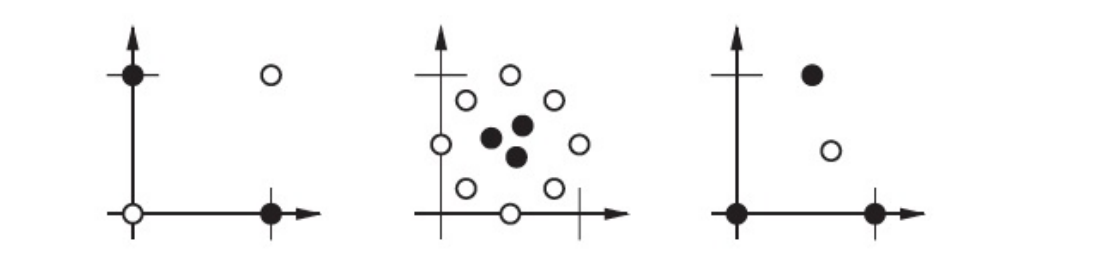
Exemple 1 de données non séparables:

Outliers dans les données : mieux vaut éviter de les



Limites d'un perceptron 2

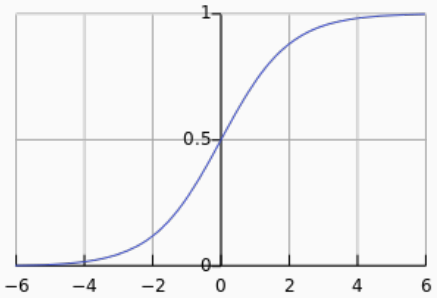
Exemple 2 : des données non séparables linéairement mais il existe une frontière non linéaire:



- **Premier problème :** XOR problem
- Un perceptron seul ne peut implémenter une fonction XOR, ni aucun des deux autres problèmes ci-dessus
- Solution :
  - Soit on rajoute une couche de neurones avant le neurone de sortie perceptron multi-couches (algorithme d'apprentissage par rétro-propagation du gradient), Werbos 1974, Le Cun 1985, Rumelhart et al. 1986.
  - Soit on transforme les données en les plongeant dans un espace où elles sont linéairement séparables (see Practical session)

Apprendre un perceptron(vue plus générale)

- Remplacer la fonction signe par une sigmoide différentiable
  - $sigm(x) = \frac{1}{1+\exp(-\frac{1}{2}x)}$



- Définir une fonction de perte différentiable
  - $\ell_i(\mathbf{w}) = (y_i - \ sigm(\mathbf{w}^T x_i))^2$
  - $L(\mathbf{w}) = \sum_i \ell_i(\mathbf{w})$

Apprendre un perceptron (vue plus générale)

Algorithme d'apprentissage du perceptron par gradient local stochastique

STOP $=$ faux

ϵ; nbIter; $j = 0$; $t = 0$

Initialiser $\mathbf{w}_0$

Jusqu'à ce que STOP soit vrai:

1. Pour j =1 jusqu'à n:
   - Tirer uniformément un indice i parmi $\{1, \ldots, n\}$
   - $\mathbf{w}^{\mathbf{t+1}} = \mathbf{w}^{\mathbf{t}} - \eta \nabla_{\mathbf{w}} \ell_{\mathbf{i}}(\mathbf{w})$
   - $t \rightarrow t + 1$
2. STOP $= (L(||\mathbf{w}(t) - \mathbf{w}(t-1)|| < \epsilon)$ et $(nbIter \leq nbMax)$

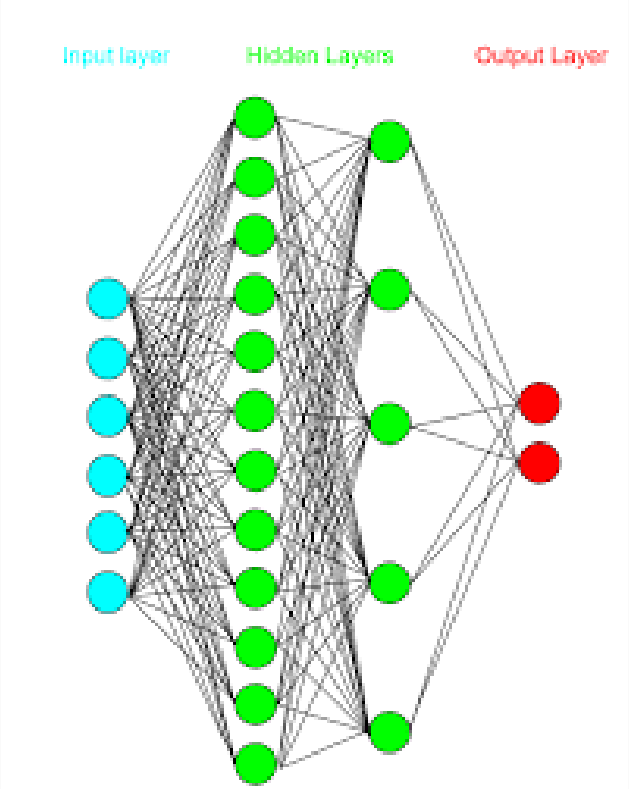
Perceptron

- Early stopping: arrêter avant de sur-apprendre (nblter petit)
- Eviter le sur-apprentissage : contrôler la norme du vecteur $\mathbf{w}$ pendant l'apprentissage
  - La fonction de perte devient : $L(\mathbf{w}) = \sum_i \ell_i(\mathbf{w}) + \lambda||\mathbf{w}||^2$
  - La mise à jour locale devient : $\mathbf{w^{t+1}} = \mathbf{w^t}(\mathbf{1} - \mathbf{2}\lambda) - \eta\nabla_{\mathbf{w}}\ell_{\mathbf{i}}(\mathbf{w})$

Passer au non linéaire

- Définir $\phi : \mathcal{X} \to \mathcal{F}$ une fonction non linéaire de re-description (en anglais *feature map*)
- Empiler les couches de neurones, chaque couche de neurone étant vue comme une redescription des entrées

Réseau de neurones formels (perceptron multi-couches)



Outline

Analyse Discriminante Linéaire (en anglais, LDA) : 2 classes

La plus simple des approches génératrices !
ICI : $\mathcal{X} = \mathbb{R}^p$

LDA

1. $p(x|Y=+1)$ and $p(x|Y=-1)$, densités supposées gaussiennes de matrice de covariance égales
2. $P(Y=+1) = 1 - P(Y=-1)$ supposés connus

$$h_{LDA}(x) = 1 \text{ if } \log\left(\frac{P(Y=+1|x)}{P(Y=-1|x)}\right) \geq 0, \text{ -1, sinon}$$

Analyse discriminante linéaire : 2 classes

Question: quelle est la forme géométrique de la frontière de décision définie par le classifieur LDA ?

Notations et définitions

- $\mu_+ \in \mathbb{R}^p$, $\mu_- \in \mathbb{R}^p$
- Σ: matrice symétrique définie positive
-

$$p(x|Y=+1) = \frac{1}{2\pi^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu_+)^T\Sigma^{-1}(x-\mu_+)) \quad (1)$$

- $P(Y=+1) = p_1$

$$p(x|Y=-1) = \frac{1}{2\pi^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu_-)^T\Sigma^{-1}(x-\mu_-)) \quad (2)$$

- $P(Y=-1) = 1-p_1$

Réponse

Formule de Bayes:

$$P(Y = i|x) = \frac{p(x|Y = i)P(Y = i)}{p(x)}$$

Puis, on cherche à définir la frontière de décision induite par le classifieur LDA:

Analyse Discriminante Linéaire

$$\log\left(\frac{P(Y=+1|x)}{P(Y=-1|x)}\right)=0$$

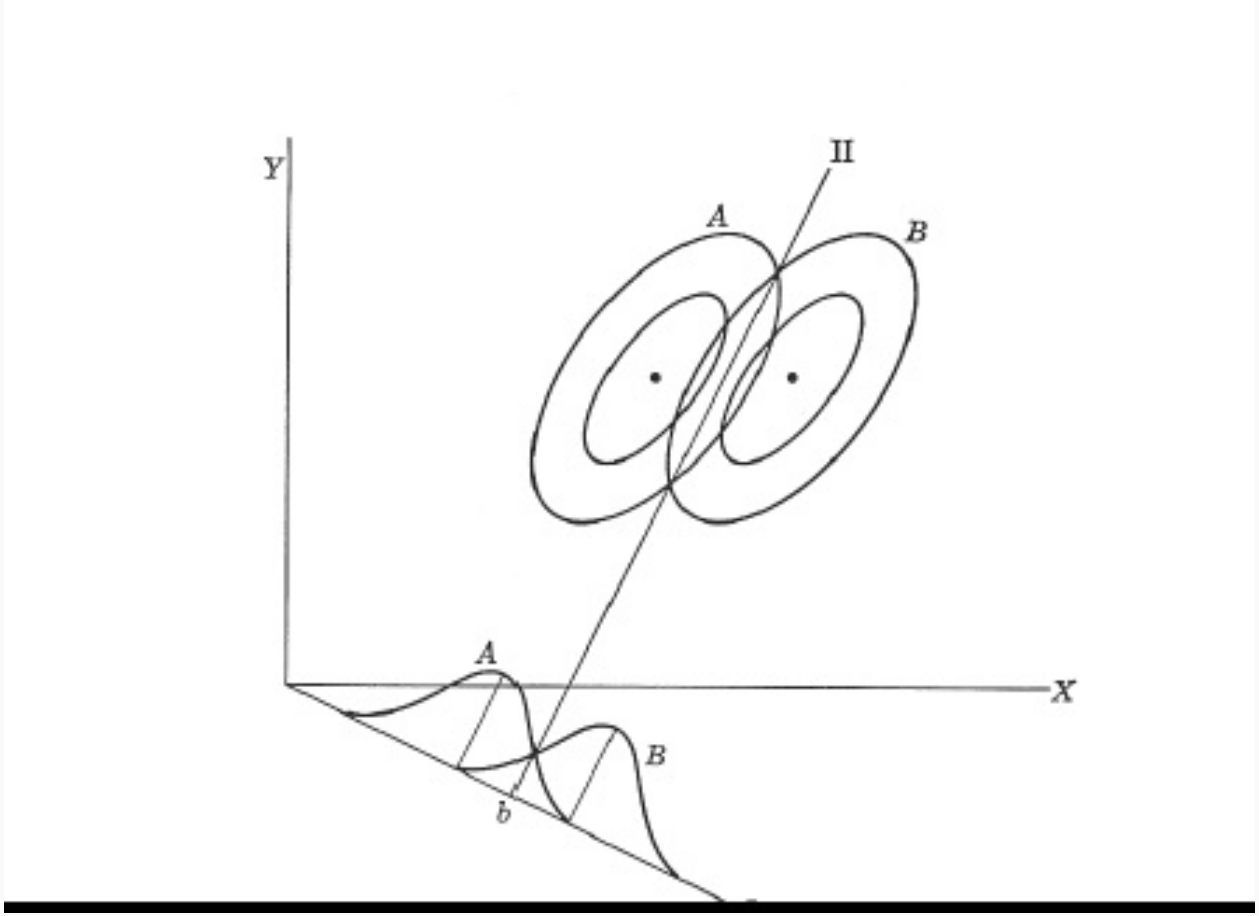
$$\text{soit}\ \ \log\left(\frac{p(x|Y=1)P(Y=1)}{p(x|Y=-1)P(Y=-1)}\right)=0$$

$$\log(\frac{p_1}{1-p_1})+\log(\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}})-\frac{1}{2}(x-\mu_+)^T\Sigma^{-1}(x-\mu_+)-\log(\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}})+\frac{1}{2}(x-\mu_-)^T\Sigma^{-1}(x-\mu_-)=0$$

$$x^T\Sigma^{-1}(\mu_+-\mu_-)+\log(\frac{p_1}{1-p_1})-\frac{1}{2}(\mu_+-\mu_-)^T\Sigma^{-1}(\mu_+-\mu_-)=0$$

Analyse Discriminante Linéaire

Le cas de deux classes aux matrices de covariances identiques



Estimation des paramètres (LDA)

Maximum de vraisemblance pour chaque sous-échantillon correspondant à une classe.
On se rappelle que la moyenne empirique (resp. la covariance empirique) est exactement l'estimateur de l'espérance par Maximum de vraisemblance.

Estimation des paramètres (LDA)

- Prendre les estimations empiriques définies à partir des données
- $S_+ = \{(x_i, y_i) \in S,\ s.t\ y_i = 1\}$
- $S_- = \{(x_i, y_i) \in S,\ s.t\ y_i = -1\}$

$$
\begin{aligned}
\hat{\mu_+} &= \frac{1}{|S_+|} \sum_{x_i \in S_+} x_i \\
\hat{\mu_-} &= \frac{1}{|S_-|} \sum_{x_i \in S_-} x_i \\
\hat{\Sigma} &= \frac{1}{2} \left(\frac{1}{|S_+|} \sum_{x_i \in S_+} (x_i - \hat{\mu}_+)(x_i - \hat{\mu}_+)^T + \frac{1}{|S_-|} \sum_{x_i \in S_-} (x_i - \hat{\mu}_-)(x_i - \hat{\mu}_-)^T \right)
\end{aligned}
$$

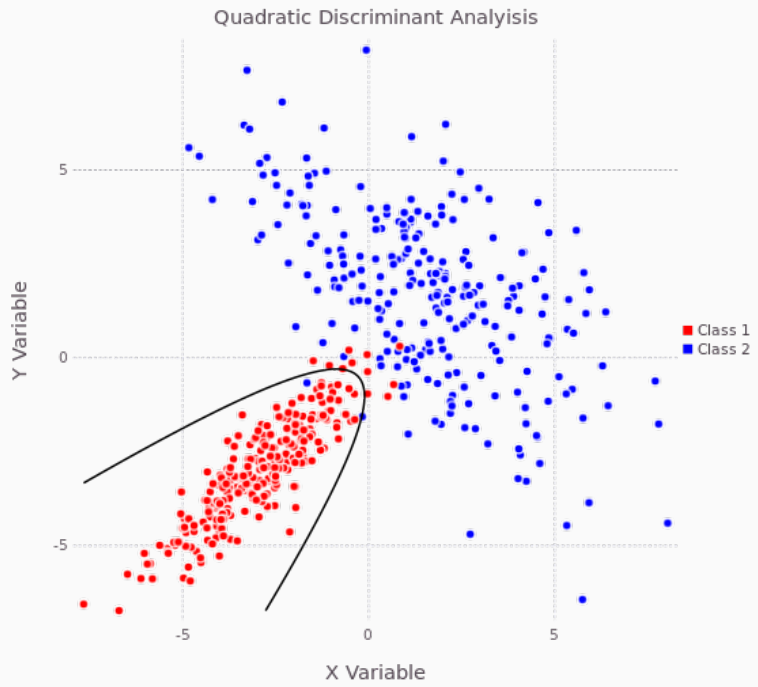
Calculs pour LDA

1. Transformer les données (sphere the data) selon la covariance commune diagonalisée:
   - $\widehat{\Sigma} = UDU^T$
   - $X^* = D^{-1/2}U^T X$
2. Les données de chaque classe ont pour covariance l'identité: simplification de la fonction f_{LDA} avec covariance = identité
3. Alors, pour prédire la classe d'un nouveau point, l'associer au "centre" de la classe la plus proche modulo l'effet des *a priori*.

Analyse Discriminante Quadratique

Cas de matrices de covariance différentes.

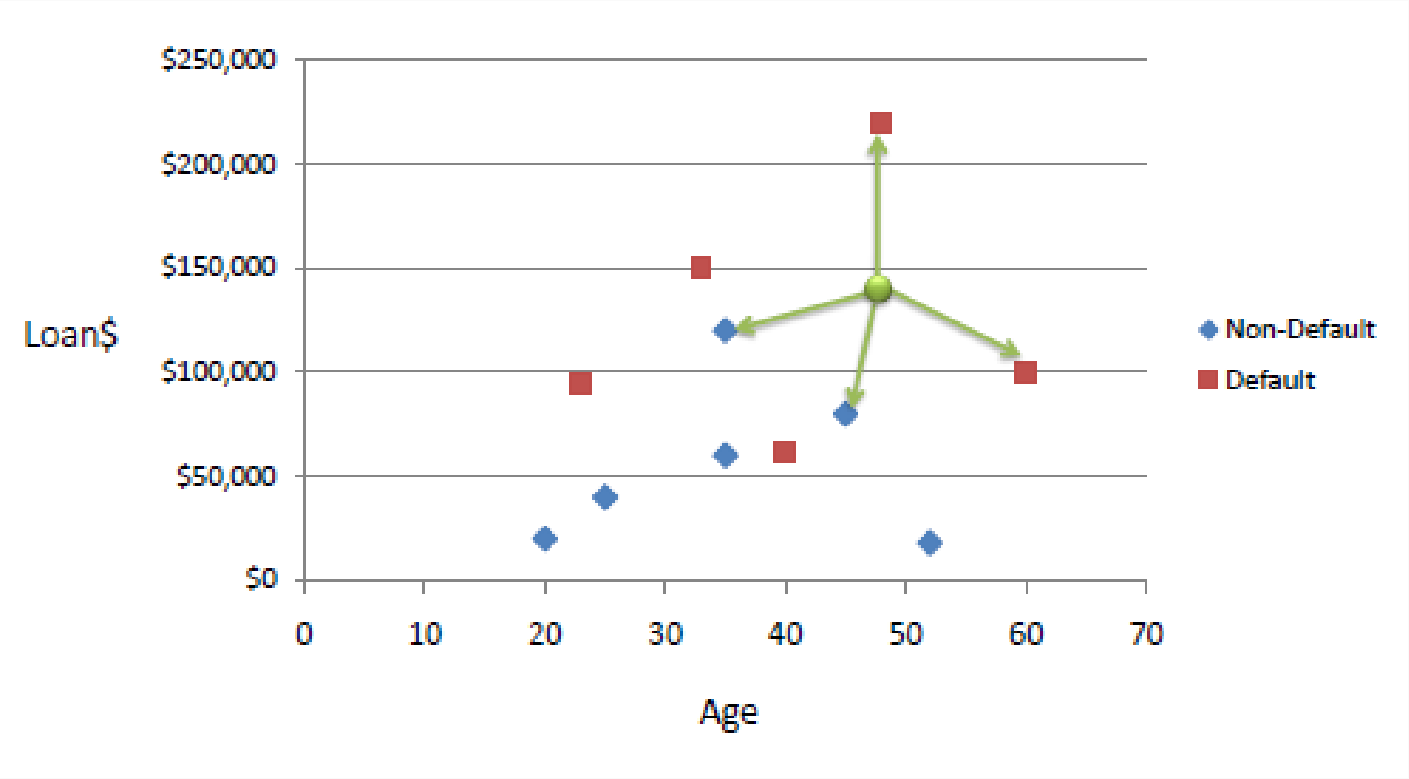
Le terme quadratique en x reste dans l'équation.



Outline

Deuxième exemple

Le classifieur des k-plus-proches voisins:



Régression par K-plus-proches voisins

K-PPV (en anglais K-Nearest neighbors: K-NN)

$$\hat{f}_{KNN}(x) = \frac{1}{L} \sum_{\ell=1}^{K} y_{(\ell)}$$

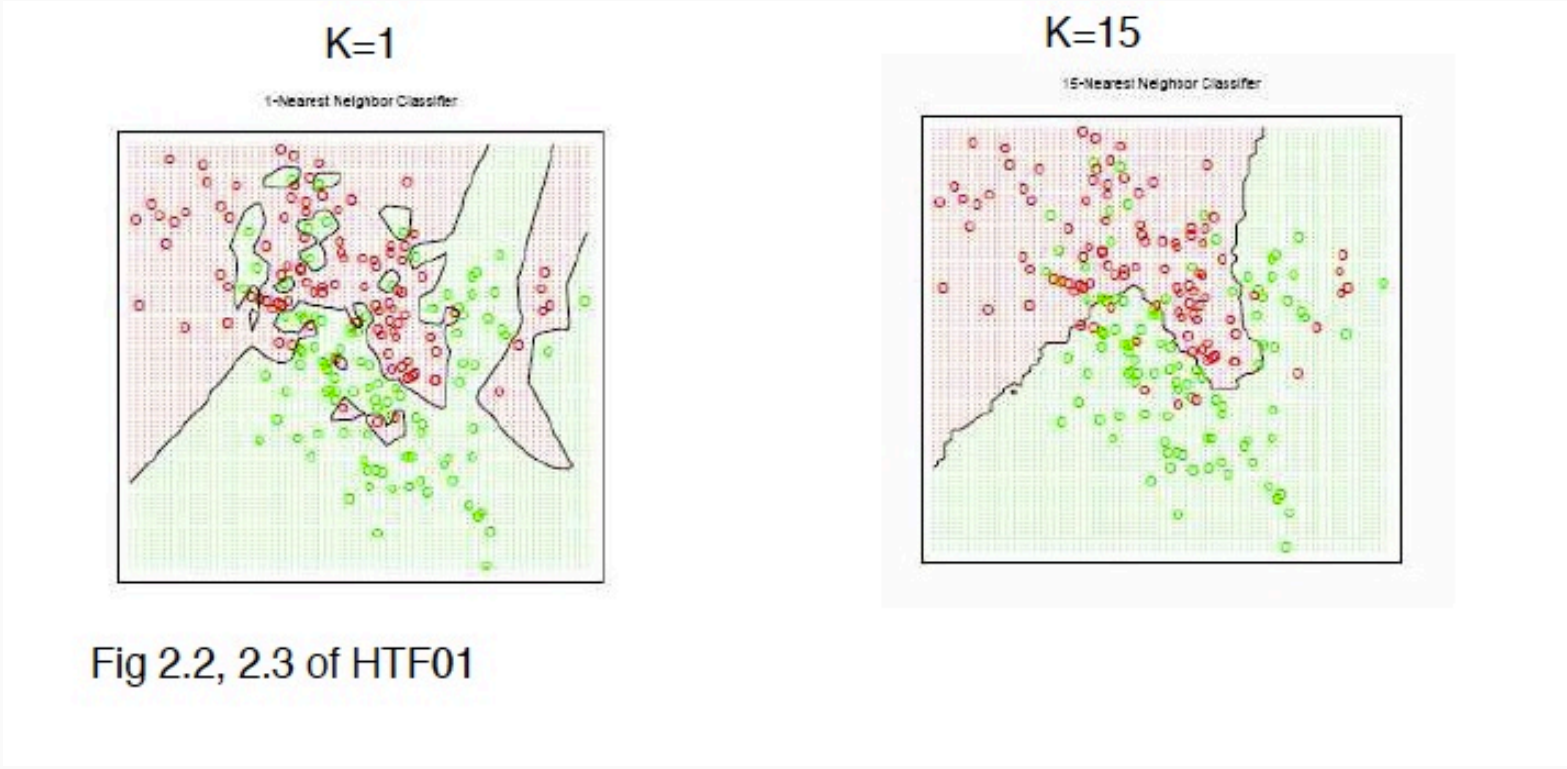
avec :

- Soit K un entier strictement positif.
- Soit d une métrique définie sur $\times$
- $S = \{(x_i, y_i), i = 1, \ldots, n\}$
- Pour une donnée x, on définit la permutation d'indices $(\cdot)$ dans $\{1, \ldots, n\}$ telle que:
  - $d(x, x_{(1)}) \leq d(x, x_{(1)}) \ldots \leq d(x, x_{(n)})$
- $S_x^K = \{x_{(1)}, \ldots, x_{(K)}\}$: K premiers voisins de x

Le paramètre de lissage K

Comment choisir K ? K: trop petit : la fonction f est trop sensible aux données : large variance

K : trop large : la fonction f devient trop peu sensible aux données : biais important



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

Question: Tracer la frontière de décision lorsque K = 50

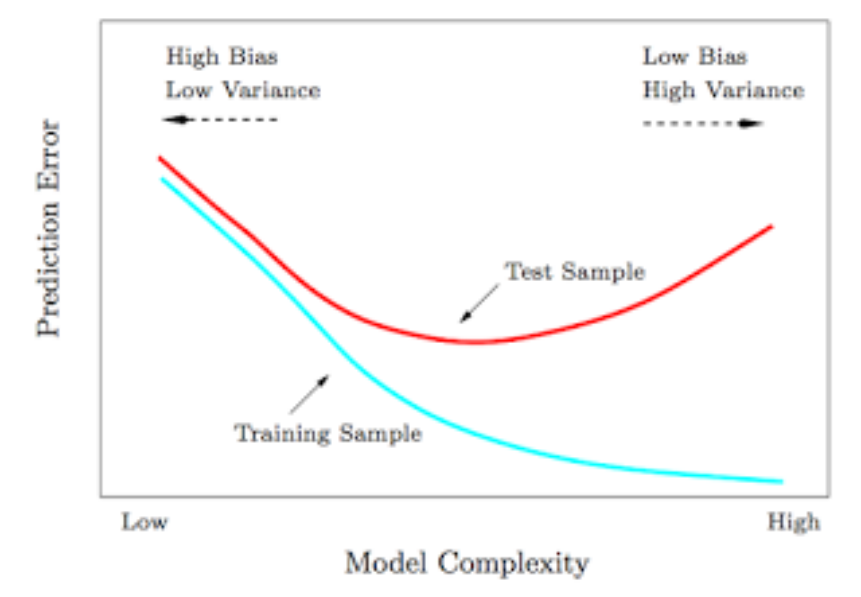
Décomposition biais-variance

On suppose: $Y = f(X) + \epsilon$ avec ϵ centré et de variance σ_ϵ^2. x est fixé.

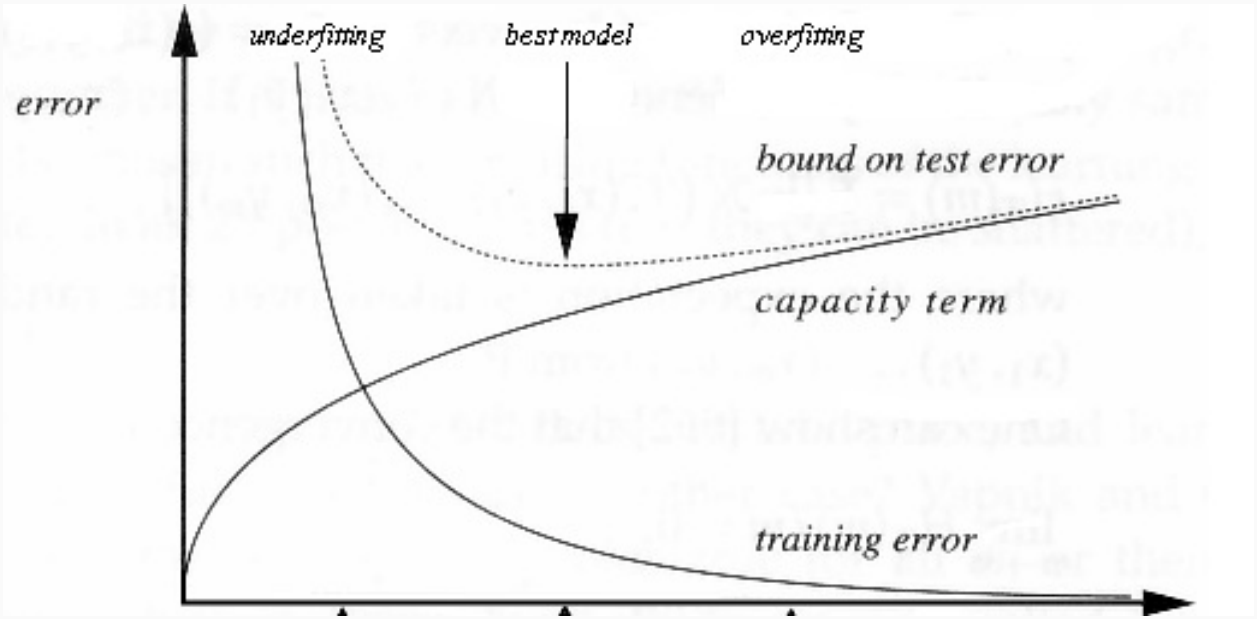
$$
\begin{aligned}
\mathbb{E}_{S,Y}[(Y - \hat{f}(x))^2] &= \mathbb{E}_{S,Y}[Y^2 + \hat{f}(x) - 2Y\hat{f}(x)] \\
&= \mathbb{E}[Y^2] + \mathbb{E}_S[\hat{f}(x)^2] - 2\mathbb{E}_S[Y\hat{f}] \\
&= Var Y + \mathbb{E}[Y]^2 + Var\hat{f}(x) + \mathbb{E}_S[\hat{f}]^2 - 2\mathbb{E}[f(x) + \epsilon]\mathbb{E}_S[\hat{f}(x)] \\
&= \sigma_\epsilon^2 + \mathbb{E}[f(x) + \epsilon]^2 + \mathbb{E}_S[\hat{f}]^2 - 2\mathbb{E}_S[f(x)]\mathbb{E}_S[\hat{f}(x)] + Var\hat{f}(x) \\
&= \sigma_\epsilon^2 + \mathbb{E}_S[\hat{f}(x) - f(x)]^2 + Var\hat{f}(x) \\
&= \sigma_\epsilon^2 + \text{Biais}^2 + \text{variance}
\end{aligned}
$$

Terme incompressible : bruit des données
Biais au carré: mesure à quel point $\hat{f}$ est loin de la cible
Variance de $\hat{f}(x)$: mesure à quel point $\hat{f}(x)$ est sensible aux données

Dilemne Biais variance



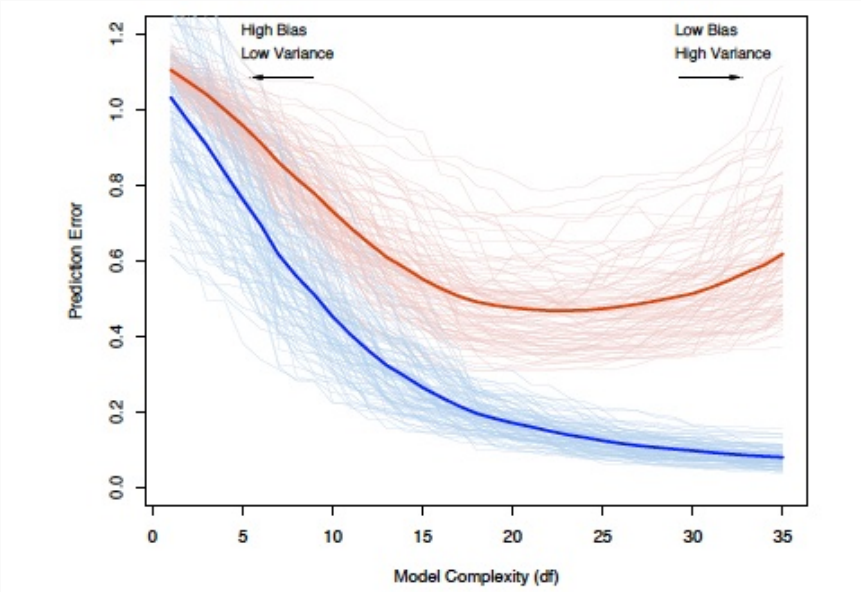
Dilemne Biais variance et risque structurel



complexité de la famille de fonctions

Biais variance - Vision empirique

Soit M datasets $S_1, \ldots, S_M$ de même taille n. Apprenons pour chacun d'entre eux, une fonction $\hat{f}$ sous différentes contrainte de complexité (ici nombre degré de libertés n/K).On a tracé sur cette figure la courbe des erreurs en apprentissage (bleue) et des erreurs en test (rouge) pour chacune des fonctions construites:



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

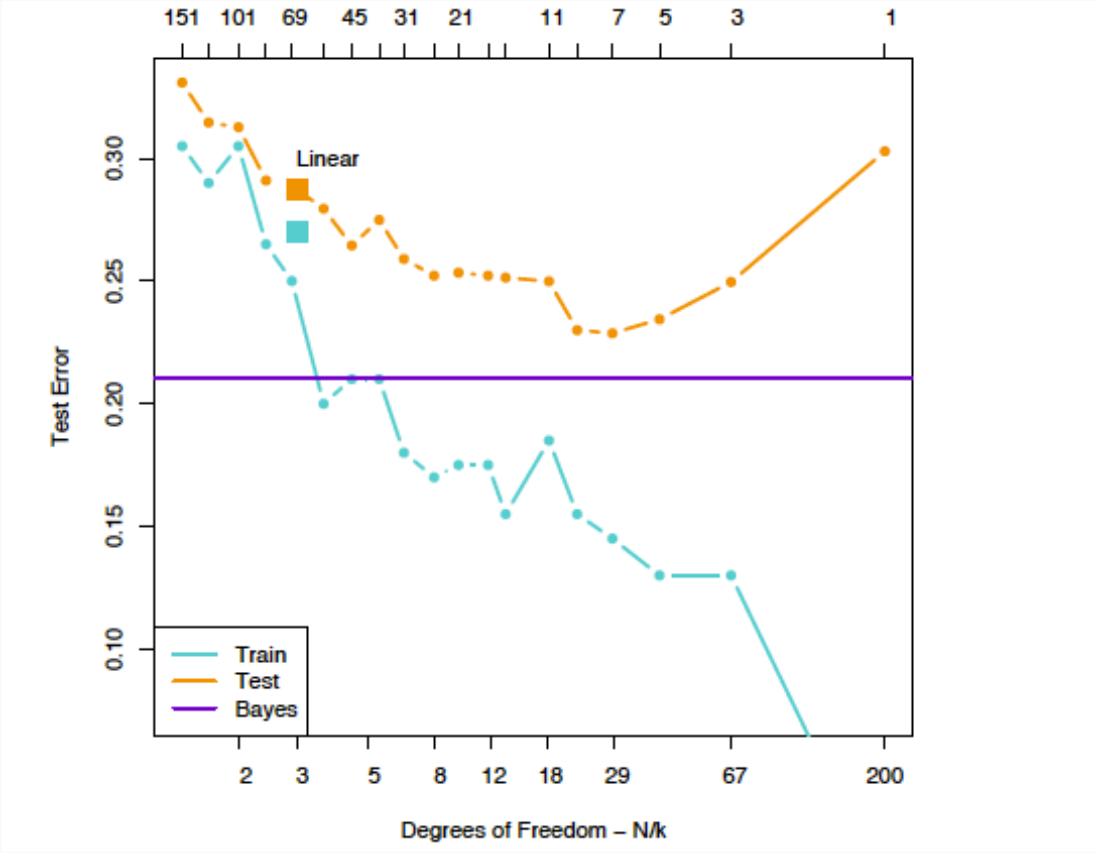
Décomposition biais-variance des k-plus-proches voisins

Posons x_0. L'aléa vient de l'échantillon utilisé pour apprendre $\hat{f}$ et de Y. On peut montrer que:

$$E_{S,Y}[(Y - \hat{f}(x_0))^2] = \sigma_\epsilon^2 + (f(x_0) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})^2 + \frac{\sigma_\epsilon}{K}$$

K contrôle le terme de variance : plus grande est la valeur de K, plus la variance décroit; mais K contrôle aussi le biais, plus petite est la valeur de K, plus petit est le biais : nous sommes en plein *dilemne biais-variance*.

Erreur de test en fonction de $\frac{n}{K}$



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

Questions

- Quelle est la force de cette méthode ?
- Quelle est sa faiblesse ?

Outline

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres
  - une méthode d'évaluation des performances

Sélection ou évaluation de modèle ?

- Estimer les performances de différents modèles afin de choisir le meilleur : sélection de modèle
- Ayant choisi un modèle, estimer son erreur en généralisation (le vrai risque) : évaluation de modèle

Aujourd'hui, nous nous concentrons sur ces deux questions.

Premier exemple

Un classifier linéaire dans le plan:

$$h(x) = \text{signe}(\beta_1 x_1 + \beta_2 x_2 + \beta_0) \tag{3}$$

Apprendre h_β en minimisant: $\sum_{i=1}^{n} L(y_i, h(x_i)) + \lambda \|\beta\|_2^2$
OU $\sum_{i=1}^{n} L(y_i, h(x_i)) + \lambda \|\beta\|_1$ Quelle valeur de λ choisir ?

Premier exemple

1. Quelle valeur de λ choisir ? Algorithme de sélection $\tilde{\lambda}$
2. Une fois que $\tilde{\lambda}$ est choisi, j'applique mon algorithme d'apprentissage et j'obtiens $\hat{h}_{\tilde{\lambda}}$: comment évaluer ses performances ?

Deuxième exemple

Le classifieur des k-plus-proches voisins: classifier paresseux : pas besoin d'algorithme d'apprentissage ! J'ai besoin des données dites d'apprentissage, d'une métrique et de la valeur de k.

- Comment choisir la valeur de K ?
- Ayant choisi $\tilde{K}$, comment estimer l'erreur en généralisation de ce K-NN ?

Validation croisée 1

Nota Bene : $\mathcal{S}$ ensemble de données dédié à la construction du modèle
En aucun cas, l'ensemble de TEST !

Diviser les données $\mathcal{S}$, réservés à la validation, en B parties de même taille et disjointes $S_{b=1}, \ldots, D_{b=B}$ avec $|D_b| = n/B$. Les données sont réparties par tirage uniforme sans remise.

Validation croisée

1. Pour $b \in \{1, \ldots, B\}$:
   - Entrainer le modèle issu de paramètre λ sur toutes les données **sauf** D_b pour obtenir un estimateur $\hat{h}^b_{\lambda,n}$
   - Calculer **sur les données restantes** D_b (test) le risque empirique

$$R_{n,b}(\lambda) = \frac{1}{n/B} \sum_{j \in D_b} L(x_j, y_j, \hat{h}^b_{\lambda,n})$$

2. Calculer le risque empirique moyen de λ (dit 'de cross-validation')

$$R^B_{n,CV}(\lambda) = \frac{1}{B} \sum_{b=1}^{k} R_{n,b}(\lambda)$$

Trouver λ

Répéter cette procédure sur tous les $\lambda \in \Lambda$ considérés (ou sur une grille sur λ est un paramètre continu) et choisir

$$\hat{\lambda}_{n,B} = \arg\min_{\lambda \in \Lambda} R_{n,CV}(\lambda). \tag{4}$$

Erreur d'apprentissage / Erreur de test

On sélectionne sur $\mathcal{S}_{val}$ On apprend sur $\mathcal{S}_{app}$ en utilisant $\tilde{\lambda}$
On teste sur $\mathcal{S}_{test}$
$\text{Err}_{CV,val}$ nous dit à quel type d'erreur en généralisation nous attendre en apprenant sur un ensemble de taille $n_{val} - n_{val}/B$. $\text{Err}_{\mathcal{S}_{app}}$ nous dit à quel point le classifieur a bien réussi à approcher les données d'apprentissage
$\text{Err}_{\mathcal{S}_{test}}$ nous dit à quel point le classifieur a bien réussi à approcher les données (nouvelles) de test

Erreur d'apprentissage / Erreur de test

Dans de nombreux articles : $\mathcal{S}_{val} = \mathcal{S}_{app}$
On sélectionne le modèle (la valeur de λ) sur $\mathcal{S}_{app}$ par validation croisée
On apprend sur $\mathcal{S}_{app}$ en utilisant $\tilde{\lambda}$ obtenu de la validation croisée
On teste sur $\mathcal{S}_{test}$
$\text{Err}_{CV,val}$ nous dit à quel type d'erreur en généralisation nous attendre en apprenant sur un ensemble de taille $n_{val} - n_{val}/B$. $\text{Err}_{\mathcal{S}_{app}}$ nous dit à quel point le classifieur a bien réussi à approcher les données d'apprentissage
$\text{Err}_{\mathcal{S}_{test}}$ nous dit à quel point le classifieur a bien réussi à approcher les données (nouvelles) de test

Outline

References

- Formal neuron / Perceptron
  - Léon Bottou: `http://cilvr.cs.nyu.edu/diglib/lsml/bottou-sgd-tricks-2012.pdf`
- Performance evaluation / model selection
  - Chapter 4 and 7 of Elements of statistical learning, Hastie, TIbshirani and Friedman.
  - A.-L. Boulesteix, Ten rules for reducing overoptimistic reporting in methodological computational research, `http://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1004191&type=printable`