Analyse, Classification, Indexation des Données

ACID

Master 1 ACID

Année 2016-2017

Présentation

- Objectif : Une initiation au « Machine learning ».
  - Comprendre et assimiler les différentes techniques permettant d'indexer ou de classifier les données (Bayles, Perceptron, Clustering, SVM, etc).

Extraire d'un ensemble de données bas niveaux une information permettant de prendre une décision ou.

Principale source bibliographique :

Pattern Classification (2nd ed)

R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons

Introduction à la classification de motifs

- La perception pour un système numérique
- Un exemple
- Système de reconnaissance de motifs (forme, objets)
- Un schéma de conception
- Apprentissage et adaptabilité
- Conclusion

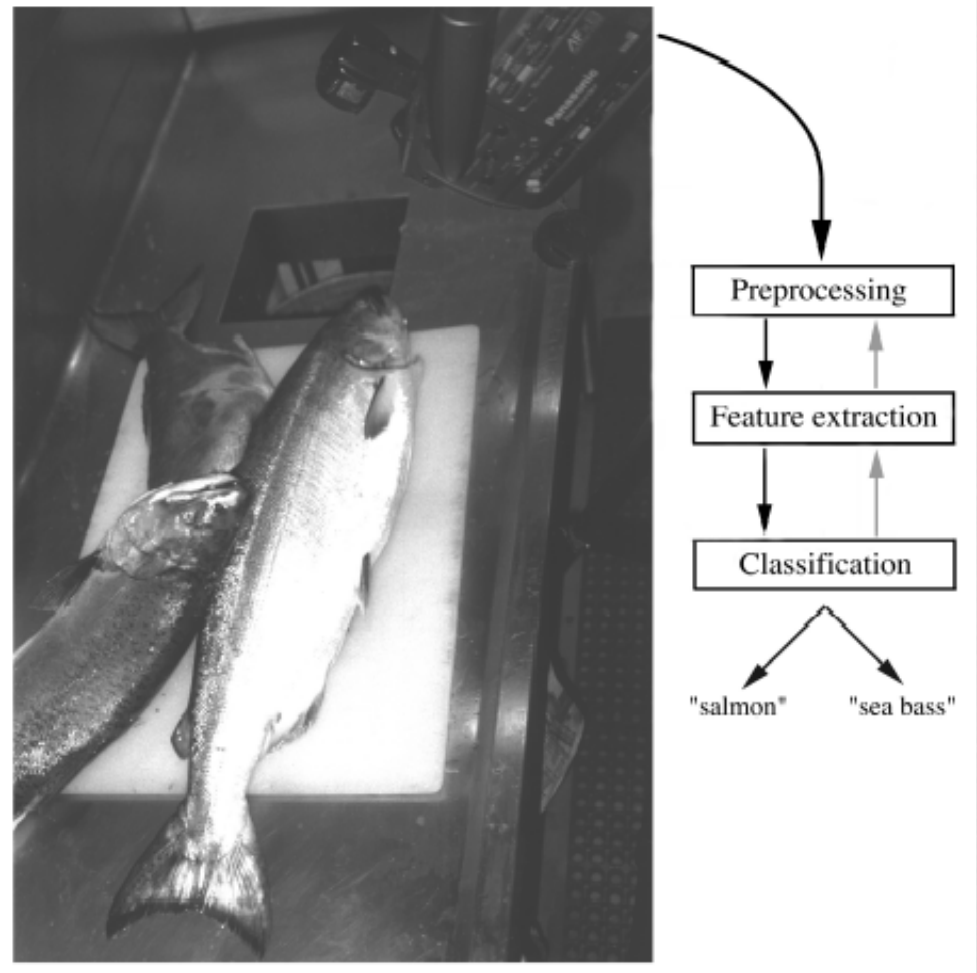
La perception pour un système numérique

- Concevoir un système qui peut :
  - *Reconnaissance de la parole, des empreintes digitales, OCR, etc.*
- En utilisant un ou des capteurs numériques:
  - *Caméra, micro, scanner, laser, etc.*
- Comment traduire le plus fidèlement possibles des connaissances haut niveau en utilisant des informations bas niveaux.
  - *« Semantic gap ».*
  - *Fusion de données hétérogènes précoce ou tardive*

Un exemple

- Concevoir un système capable de trier deux espèces de poissons (bar et saumon).
  - On dispose d'un tapis roulant, d'une caméra, et d'une vanne d'aiguillage. Les saumons vont à droite et les bars à gauche.
  - Il nous faut trouver un moyen de distinguer les bars d'un saumon en fonction de l'acquisition et ensuite piloter la vanne en fonction de la reconnaissance.
  - Et si les clients nous pénalisaient en cas d'erreurs ?

Une ébauche de solution



Traitement de la séquence vidéo

- Positionner et calibrer la caméra pour extraire des données sur des séquences d'images.
  - Prétraitement pour améliorer la qualité des images (équilibrage contraste et illumination, suppression du bruit, etc)
  - Segmentation des zones correspondant à un poissons
  - Extraction de descripteurs
    - géométriques (longueur , largeur, etc)
    - colorimétriques (brillance, couleur, etc)
    - plus haut-niveaux(nombre et forme des nageoires, position de la bouche, etc). Classification basée sur les descripteurs précédents.

Chaine de traitement

Acquisition(s)	Sources: Hétérogénes, multimodale, etc..
Segmentation	Extraction des éléments déterminants
Calcul de descripteurs	Extraction des descripteurs les plus pertinents
Classification	Objet de ce cours
Post-traitements	Exploitation du contexte, (connaissances à postériori)

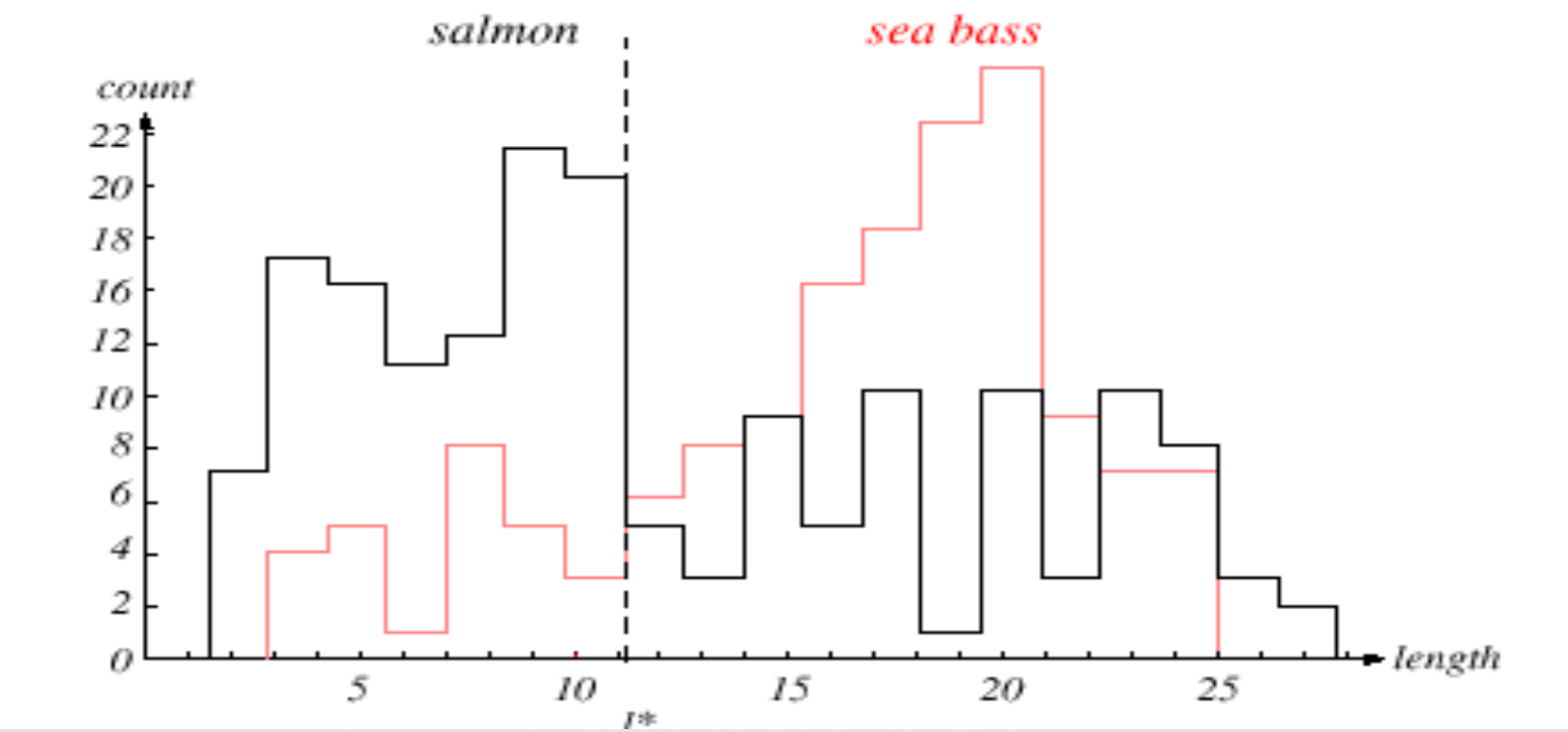
Descripteur

Taille du poisson

	2	4	8	10	12	14
bass	0	1	3	8	10	5
salmon	2	5	10	5	1	0

Est ce que la longueur est discriminante ?

En moyenne les saumons sont plus petits que les bars….
Mais est-ce que cela peut nous aider à prendre une décision ???



Un premier classifieur

Choisir un seuil L en fonction d'un critère

Si longueur mesurée < L c'est un saumon

Sinon c'est un bar

Comment choisir L ??

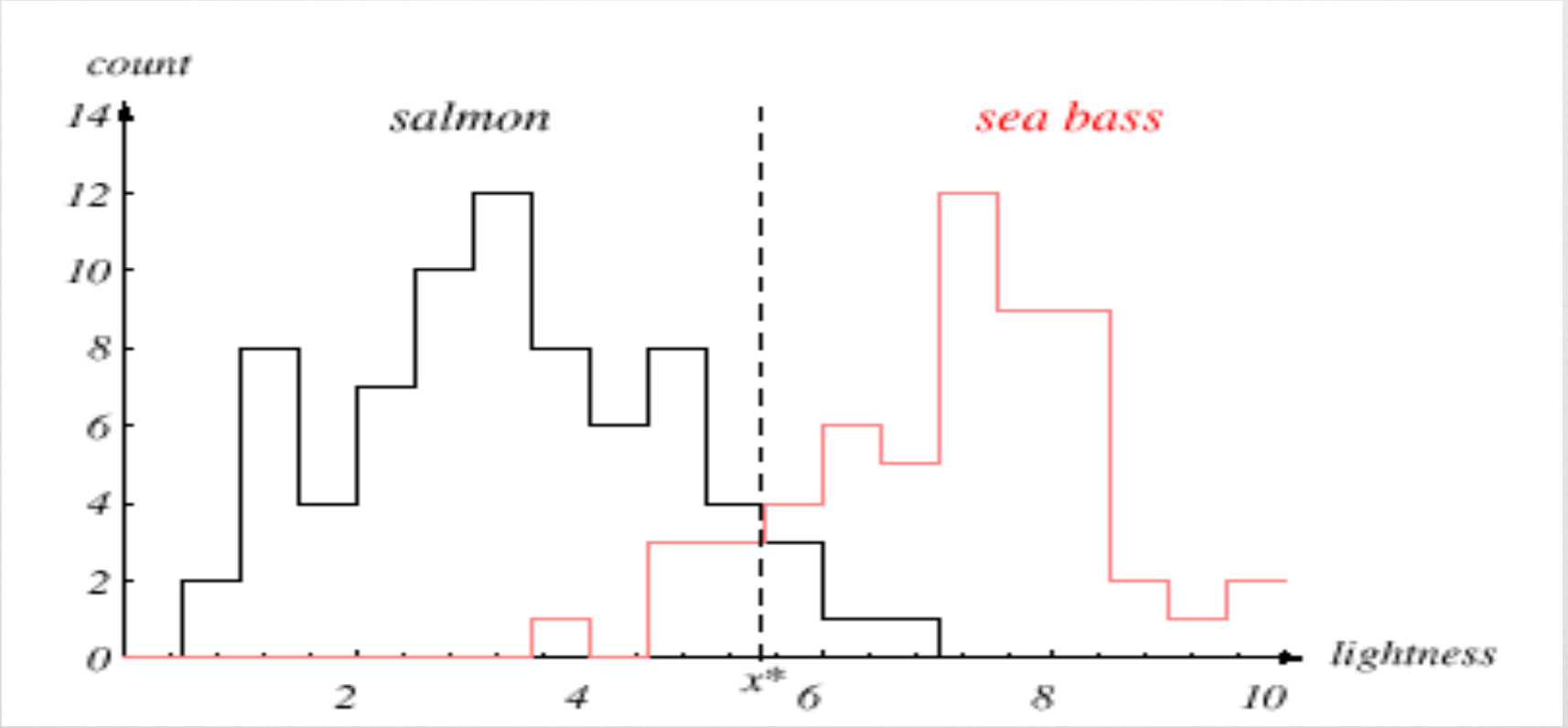
	2	4	8	10	12	14
bass	0	1	3	8	10	5
salmon	2	5	10	5	1	0

Comment mesurer la qualité de la classification ??

- Avec 2 classes
- Avec n classes

Est ce que la brillance est plus discriminante ?

Les saumons sont plus sombre que les bars, mais il existe tout de même une zone d'incertitude.



Un deuxième classifieur

On a maintenant un deuxième classifieur

	1	2	3	4	5
bass	0	1	2	10	12
salmon	6	10	6	1	0

Est-il meilleur que le précédent?.

Point d'avancement.

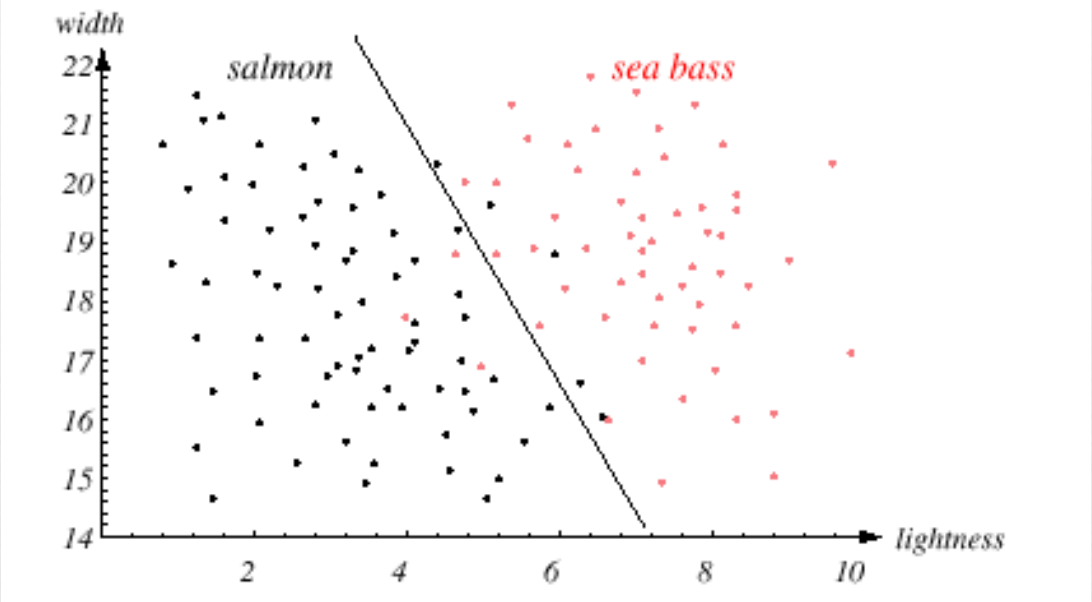
- Est-ce que l'échantillon étudié est représentatif de la réalité.(Est-ce que 200 poissons suffisent).
- Peut-on obtenir une loi de probabilité à partir de ces distribution.
- La brillance semble plus discrimante que la longueur. Est-ce vrai.
- Qu'elle stratégie adopter si il est préférable de mal classé un saumon qu'un bar (coût de pénalité). Dans ce cas, notre décision doit minimiser une fonction de cout.

Deux classifieurs et deux descripteurs

- Comment les combiner ?
  1. Chacun vote mais que faire si ils ne sont pas d'accord?
  2. On fusionne les deux descripteurs pour n'en faire qu'un.

En combinant les deux descripteurs

$X = (x_1, x_2)$ avec x_1 la brillance et la $x_2$longueur



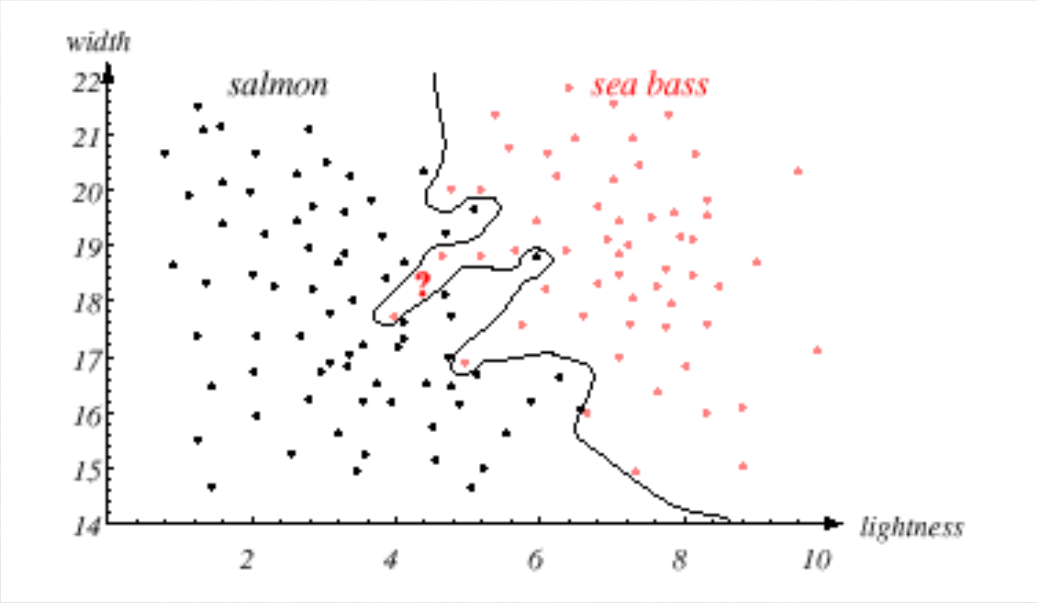
On peut chercher la droite qui sépare au mieux les saumons des bars en laissant le moins de poisson mal classés.

Meilleure solution seulement 4% de mal classé.

Comment améliorer le classifieur

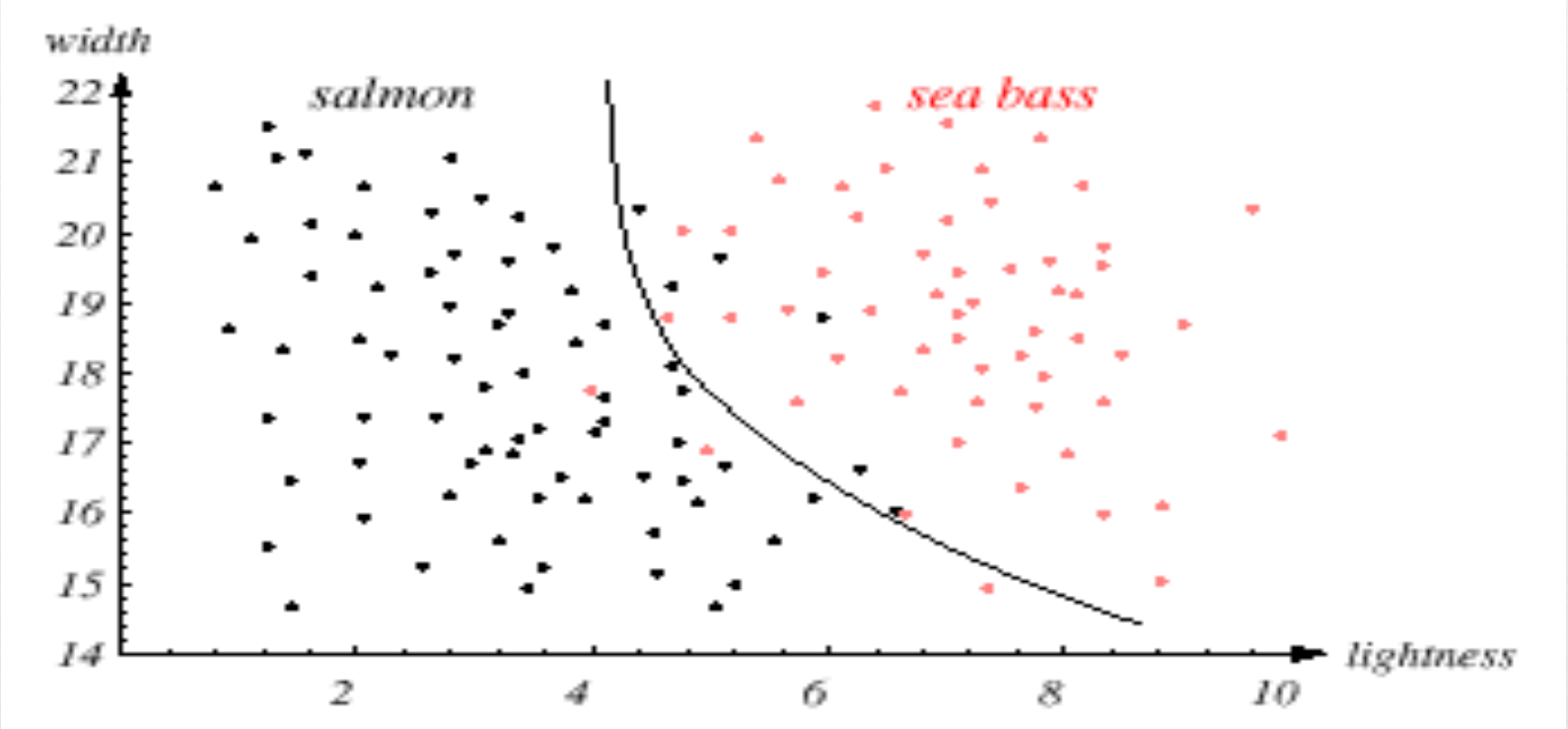
- Augmenter l'ensemble d'apprentissage afin d'avoir une meilleure distribution et donc un classifieur plus robuste (mais ce n'est pas toujours possible de disposer d'un ensemble annoté de grande taille).
  - ⇒ Problème d'une vérité terrain représentative de la distribution, annotée et conséquente.
- Augmenter le nombre de descripteurs indépendants, mais il y a un risque que la frontière deviennent plus imprécises.
  - ⇒ Problème du choix des descripteurs, de la réduction en dimension et de l'espace de représentation
- Changer la forme de la courbe (ellipse, polynôme, etc)
  - ⇒ Problème du choix du noyau.

Est-ce la bonne solution??



Fortement contrainte par la vérité terrain,
Difficile de considérer cette séparation avec un nouveau descripteur
Difficile à définir parce que très complexe
Sera t'elle encore pertinente avec un nouvel élément ??
(Le point d'interrogation peut devenir un saumon)
overfitting
Quel est le taux d'erreur ???

Une meilleure solution ??



On a une meilleure séparation qu'avec une simple droite.
La courbe est plus simple d'expression
Le meilleur compromis entre efficacité et pertinence.

Schéma général d'un système d'apprentissage

Collecte des données

Sélection des descripteurs

Sélection du modèle de classification

Entrainement

Evaluation

Schéma général d'un système d'apprentissage

Comment être sûr que les données collectées sont représentatives des éléments qui seront traités par la suite.

Schéma général d'un système d'apprentissage

Collecte des données

Etude des descripteurs , de leur distribution, afin de faire émerger les plus discriminants

Indépendant de ?????? (du bruit, de transformation, etc)

Sélection des descripteurs

Sélection du modèle de classification

Entrainement

Evaluation

Schéma général d'un système d'apprentissage

Collecte des données

Sélection des descripteurs

Comment sont réparties les descripteurs ?
On connait leur distribution ?
Est-ce que la séparation est linéaire ?

Sélection du modèle de classification

Entrainement

Evaluation

Schéma général d'un système d'apprentissage

Collecte des données

Sélection des descripteurs

Sélection du modèle de classification

Entrainement

Evaluation

Quels ensemble choisir ?
Comment ?
Une seule fois ? Plusieurs fois ?

Schéma général d'un système d'apprentissage

Collecte des données

Sélection des descripteurs

Sélection du modèle de classification

Entrainement

Evaluation

Comment évaluer ?

Prise en compte de connaissance a priori

- Décision Bayesienne
  - Connaissance de la distribution des descripteurs
  - Pas besoin de vérité terrain annotée
  - On peut créer un classifieur optimal
- Maximum de vraisemblance et estimation de paramètres bayesien
  - On connait la forme de la distribution mais pas les paramètres de la distribution.
  - On a besoin d'une vérité terrain annotée.

Prise en compte de connaissances à priori

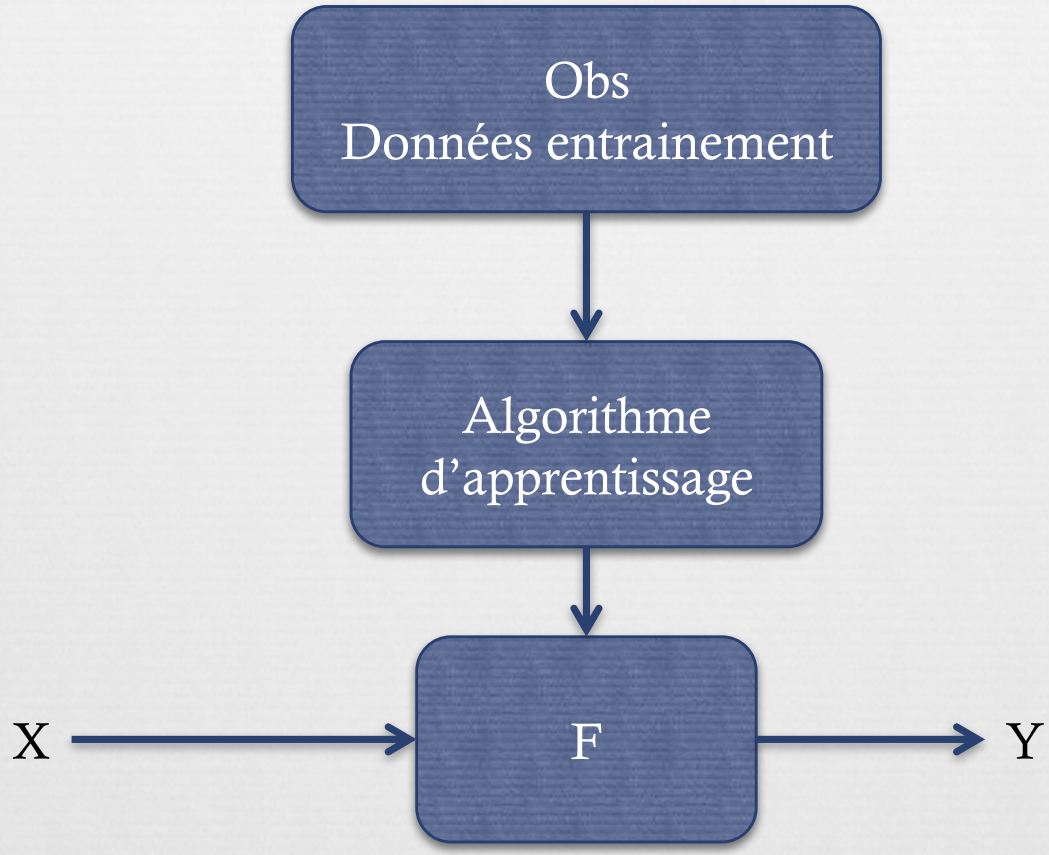
- Fonction de Séparation connue
  - Estimer les paramètres des courbes de séparation (droite, polynome, etc)
  - Besoin d'une vérité terrain annotée
- Méthodes non paramétriques
  - Pas d'apriori sur le modèle de la distribution, on a besoin d'une vérité terrain annotée
- Apprentissage non supervisée.

Analyse, Classification,Indexation des Données
ACID

Master I – UE ACID

Année 2016-2107

Régression Linéaire

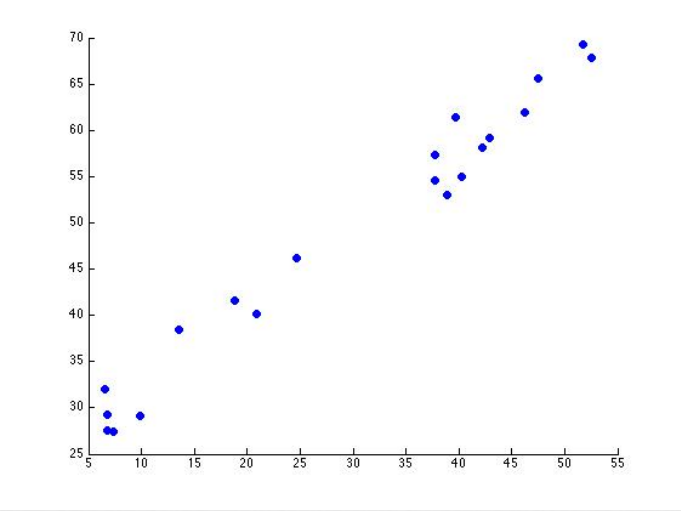


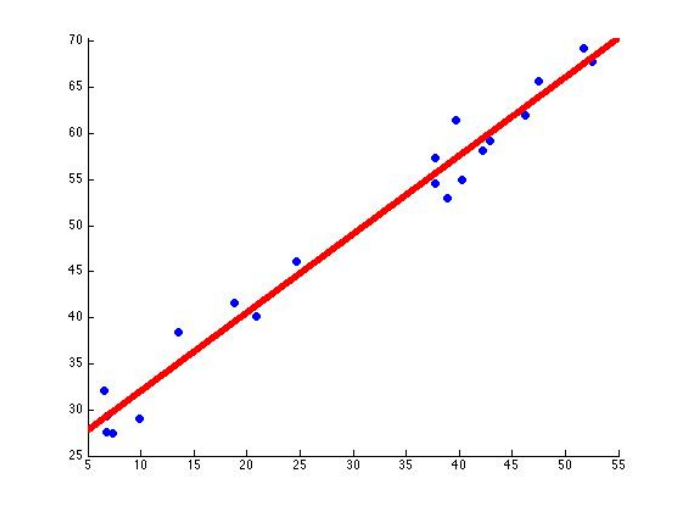
Régression Linéaire

X	Y
37,79	54,62
6,79	29,24
47,46	65,62
51,70	69,31
38,94	53,02
42,89	59,21
42,16	58,18
24,61	46,15
37,77	57,31
13,56	38,39
40,30	55,00
6,59	32,07
18,85	41,63
7,31	27,47
9,86	29,08
46,17	61,92
39,74	61,39
20,85	40,09
52,51	67,86
6,72	27,62

Etape 1 : Nous avons une observation composée d'un ensemble de m couples de mesures (x^i,y^i). $Obs = \{(x^1,y^1).....(x^m,y^m)\}$

Etape 2 : On cherche à comprendre la relation qui existe entre un x^i et un y^i.





Etape 3 : On cherche à définir la fonction h telle que : h(x) = y;

Etape 4 : On suppose que la relation entre les variables (x^i,y^i) est linéaire.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Régression linéaire

$h_\theta(x) = \theta_0 + \theta_1 x$ on pose $x_0 = 1$ et $x_1 = x$

on peut réécrire l'équation sous la forme

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Cette écriture permet de généraliser notre problème avec un nombre quelconque de variables x^1……..x^n, …….

Sachant que h est une fonction linéaire, on doit chercher le paramètre Θqui convient le mieux à notre problème. Par exemple, la somme des distances des points de l'échantillon à la droite sont le plus petit possible. On doit maintenant chercher la valeur de Θ qui minimise J.

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

Rappel

Soit une fonction J (*Θ)*, les valeurs qui annulent les dérivées partielles de la fonction J correspondent à des extréma (maxima ou minima) locaux.

Les solutions de l'équation $\frac{\partial J(\theta)}{\partial \theta} = 0$ sont les extrema.

$$\frac{\partial J(\theta)}{\partial \theta} = \vec{\nabla}_\theta(J) = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_i} \\ \\ \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{pmatrix}$$

Dans notre cas, $\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$, donc $\frac{\partial J(\theta)}{\partial \theta} = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_0} = 0 \\ \frac{\partial J(\theta)}{\partial \theta_1} = 0 \end{pmatrix}$

.

Trouvez le minimum

Une fois le gradient calculé, on a deux façons de procéder :

1. Descente de Gradient: On se déplace dans le ***"sens de la descente de la pente"***, jusqu'à ce qu'il n'y plus de pente ou que l'on ne puisse plus avancer.

2. Résoudre l'équation :

$$\frac{\partial J(\theta)}{\partial \theta} = 0$$

Descente de Gradient

Descente de Gradient pour une fonction $J(\theta)$.

On part d'un θ_0 choisit (aléatoirement ou pas)

$l = 1;$

Répéter

$$\theta_{l+1} = \theta_l - \frac{\partial J(\theta)}{\partial \theta}(\theta_l)$$

$$l = l + 1$$

jusqu'à $\theta_{l+1} - \theta_{l+1} < \varepsilon$

Descente de Gradient pour une fonction $J(\theta)$.

On part d'un θ_0 choisit (aléatoirement ou pas)

$l = 1;$

Répéter

Pour i = 1 to n

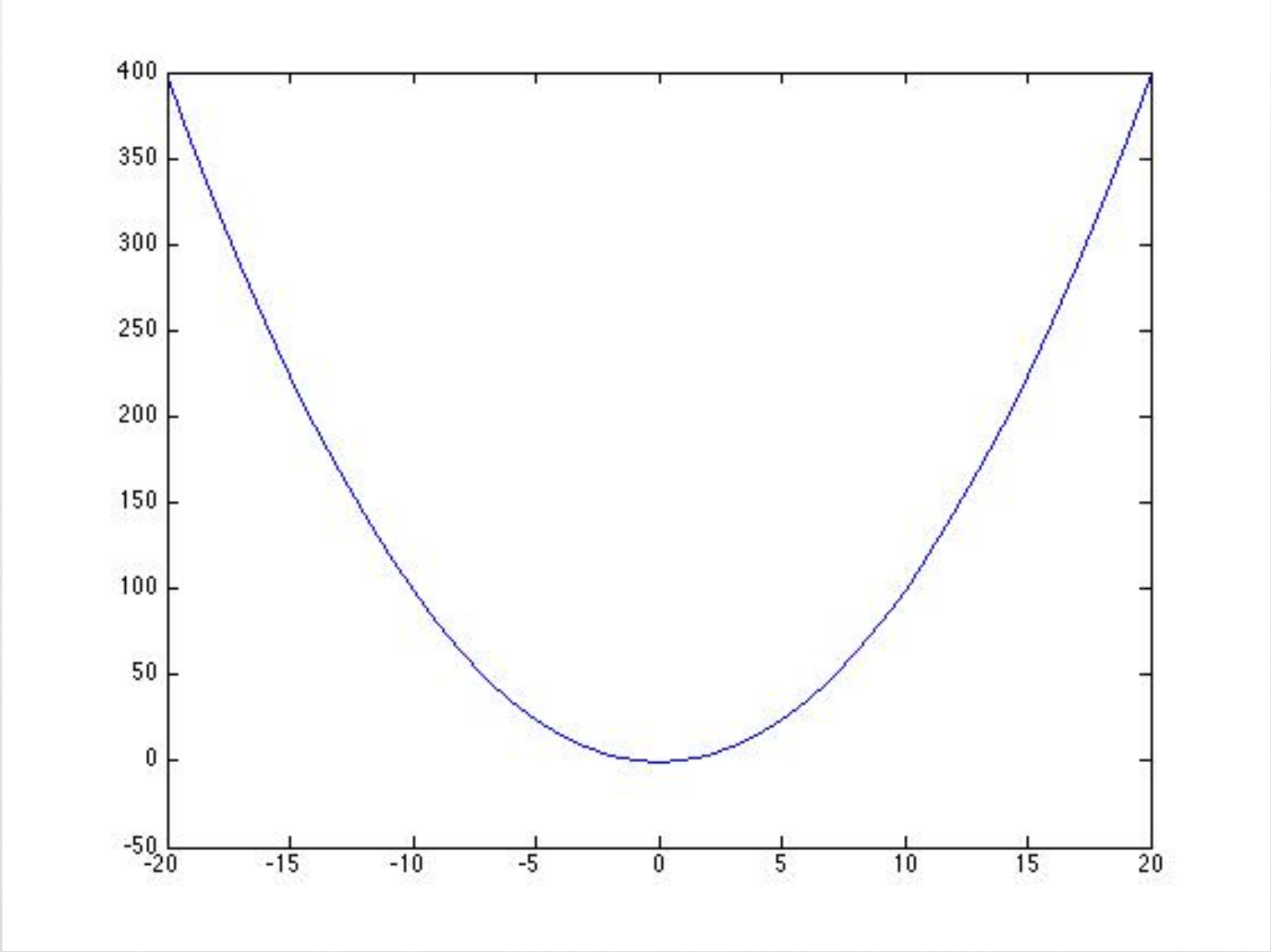
$$\theta^i{}_{l+1} = \theta^i{}_l - \frac{\partial J(\theta)}{\partial \theta_i}(\theta_l)$$

fin pour

$$l = l + 1$$

jusqu'à $[\![\theta_{l+1} - \theta_{l+1}]\!] < \varepsilon$

Un exemple de descente



Descente de Gradient

Descente de Gradient pour une fonction $J(\theta)$ avec amortissement

On part d'un θ_0 choisit (aléatoirement ou pas)

$l = 1;$

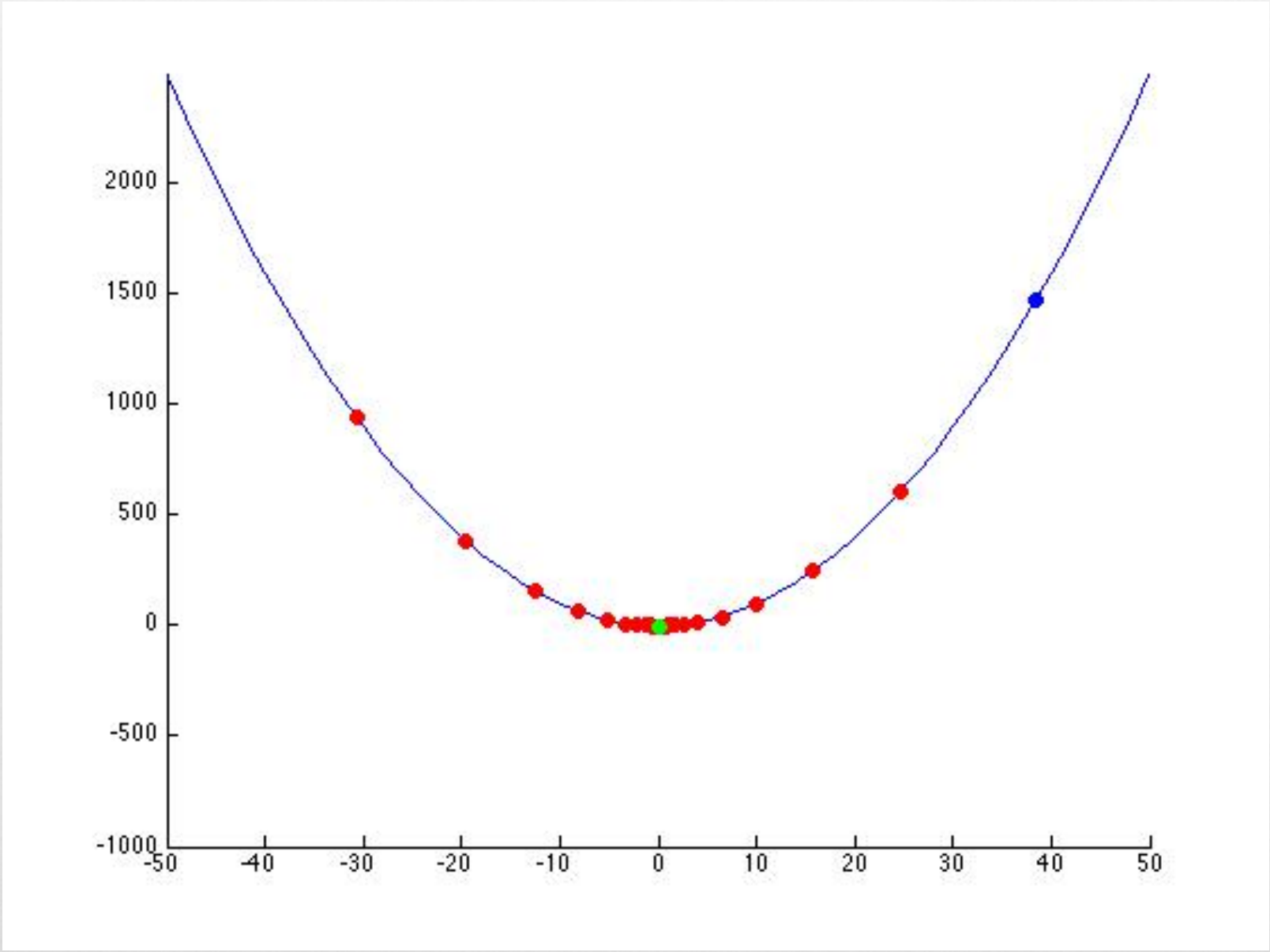
Répéter

$$\theta_{l+1} = \theta_l - \eta(l)\frac{\partial J(\theta)}{\partial \theta}(\theta_l)$$

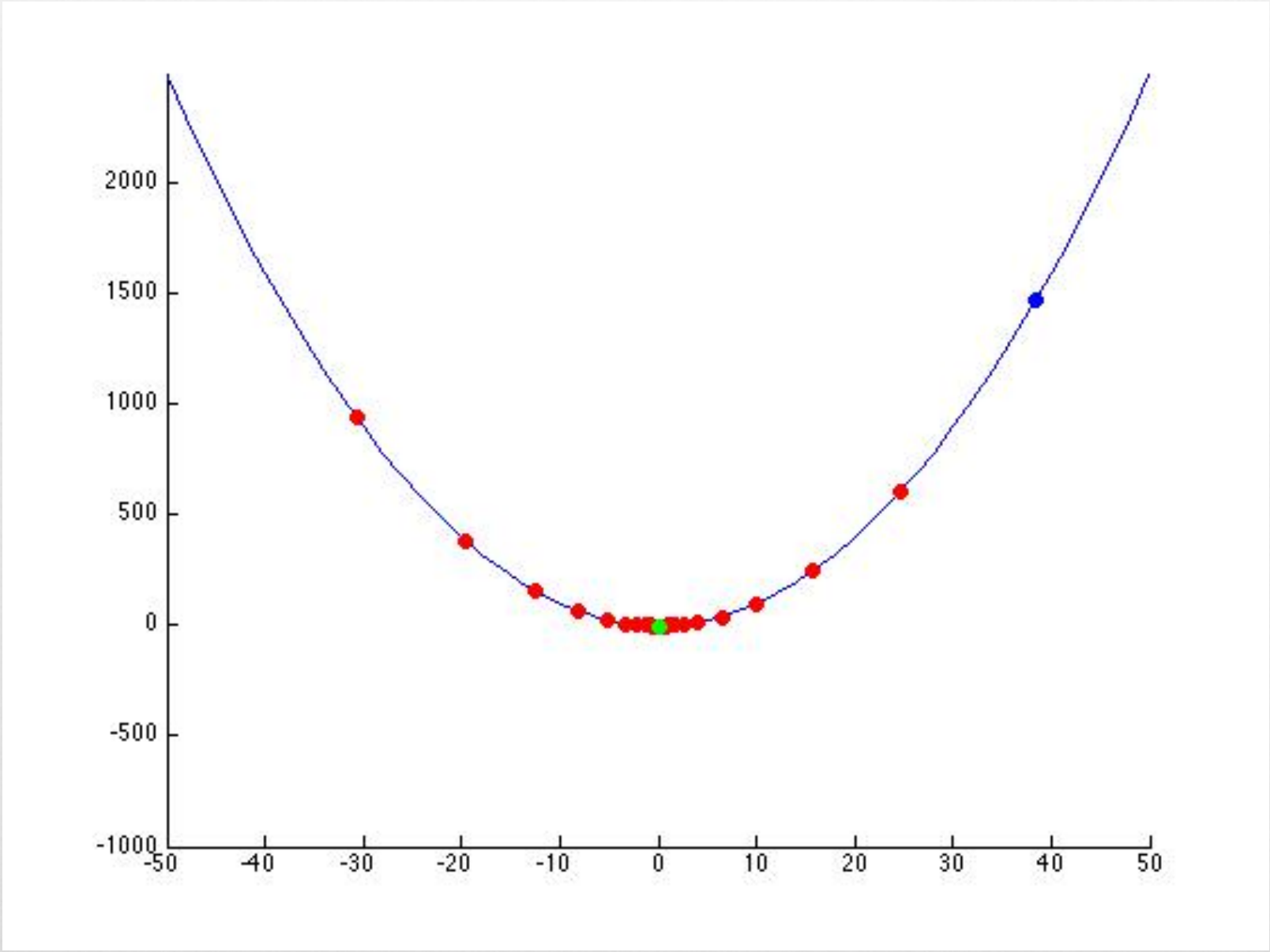
$$l = l + 1$$

jusqu'à $\theta_{l+1} - \theta_{l+1} < \varepsilon$

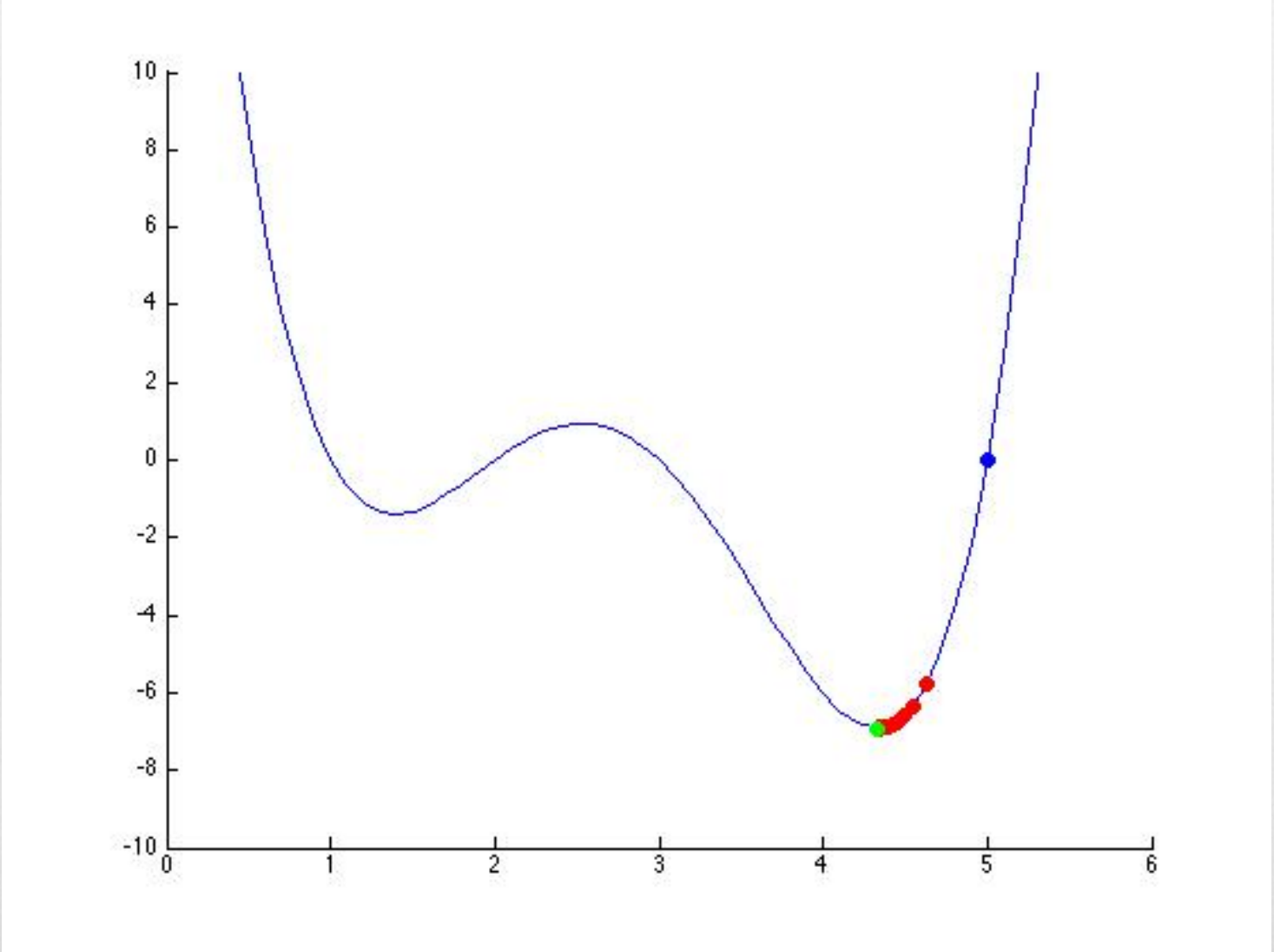
Un exemple de descente



Un exemple de descente



D’autres exemples



Calcul du Gradient

$$J(\theta) = \frac{1}{2}\sum_{k=1}^{m}(h_\theta(x^{(k)}) - y^{(k)})^2$$

$$\frac{\partial J(\theta)}{\partial \theta_i} = \frac{1}{2}\sum_{k=1}^{m}(\sum_{j=0}^{n}\theta_j x_j^k - y^k)^2$$

On calcule

$$\frac{\partial(\sum_{j=0}^{n}\theta_j x_j^k - y^k)^2}{\partial \theta_i} = 2x_i^k(\sum_{j=0}^{n}\theta_j x_j^k - y^k)$$

$$\frac{\partial(\sum_{j=0}^{n}\theta_j x_j^k - y^k)^2}{\partial \theta_i} = 2x_i^k(h_\theta(x^{(k)}) - y^{(k)})$$

donc

$$\frac{\partial J(\theta)}{\partial \theta_i} = \sum_{k=1}^{m} x_i^k(h_\theta(x^{(k)}) - y^{(k)})$$

Calcul du Gradient

$$\frac{\partial J(\theta)}{\partial \theta_i} = \sum_{k=1}^{m} x_i^k (h_\theta(x^{(k)}) - y^{(k)})$$

pour i = 0 on a $x_0^{(k)} = 1$

$$\frac{\partial J(\theta)}{\partial \theta_0} = \sum_{k=1}^{m} h_\theta(x^{(k)}) - y^{(k)}$$

Pour i = 1 on a $x_1^{(k)} = x^{(k)}$

$$\frac{\partial J(\theta)}{\partial \theta_1} = \sum_{k=1}^{m} x_1^k (h_\theta(x^{(k)}) - y^{(k)})$$

Descente de Gradient

Si on revient à notre problème

On part d'un θ_0 choisit (aléatoirement ou pas)

$l = 1$;

Répéter

Pour i = 0 to 2

$$\theta^i{}_{l+1} = \theta^i{}_l - \eta(l)\sum_{k=1}^{m} x_i^k (h_\theta(x^{(k)}) - y^{(k)})$$

fin pour

$l = l + 1$

jusqu'à $[\![\theta_{l+1} - \theta_{l+1}]\!] < \varepsilon$

Si on revient à notre problème

On part d'un θ_0 choisit (aléatoirement ou pas)

$l = 1$;

Répéter

Pour i = 1 to 2

Pour tout les couples $(x^{(k)}, y^{(k)}) \in Obs$

gradi += $(x_i^k \theta_n^T x^{(k)} - y^{(k)})$

fin pour

$\theta^i{}_{l+1} = \theta_l{}^i - \eta(l)$gradi

fin pour

$l = l + 1$

jusqu'à $[\![\theta_{l+1} - \theta_{l+1}]\!] < \varepsilon$

Descente de Gradient

On part d'un $\theta_0 = \begin{pmatrix} \theta_0^0 \\ \theta_0^1 \end{pmatrix}$ choisit (aléatoirement ou pas)

$l = 1$;

Répéter

$$\nabla_{\theta_l} = \begin{pmatrix} \nabla_{\theta_0} = 0 \\ \nabla_{\theta_1} = 0 \end{pmatrix}$$

Pour tout les couples $(x^{(k)}, y^{(k)}) \in Obs$

$$\nabla_{\theta_l} = \begin{pmatrix} \nabla_{\theta_0} += \theta_n^T x^{(k)} - y^{(k)} \\ \nabla_{\theta_1} += x_1^k \theta_n^T x^{(k)} - y^{(k)} \end{pmatrix}$$

fin pour

$$\theta_{l+1} = \theta_l - \eta(l) \nabla_{\theta_l}$$

fin pour

$l = l + 1$

jusqu'à $[\![\theta_{l+1} - \theta_{l+1}]\!] < \varepsilon$

Formalisation différente

On crée une matrice X dont les lignes sont les échantillons $x^{(i)}$

$$X = \begin{pmatrix} x_1^{(1)} & & x_n^{(1)} \\ & x_2^{(2)} & \\ & & \\ x_1^{(m)} & & x_n^{(m)} \end{pmatrix}$$

On note $\vec{y}$ le vecteur obtenu en concaténant les $y^{(i)}$

$$\vec{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \\ \\ y^{(m)} \end{pmatrix}$$

on a

$$X\theta - \vec{y} = \begin{pmatrix} x_1^{(1)} & & x_n^{(1)} \\ & x_2^{(2)} & \\ & & \\ x_1^{(m)} & & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta_1 \\ : \\ : \\ : \\ \theta_n \end{pmatrix} - \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \\ \\ y^{(m)} \end{pmatrix}$$

Formalisation différente

$$X\theta - \vec{y} = \begin{pmatrix} \mathrm{x}_1^{(1)}\theta_1 & & \mathrm{x}_n^{(1)}\theta_n \\ & \mathrm{x}_2^{(2)}\theta_2 & \\ \mathrm{x}_1^{(m)}\theta_1 & & \mathrm{x}_n^{(m)}\theta_n \end{pmatrix} - \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \\ \\ y^{(m)} \end{pmatrix}$$

$$X\theta - \vec{y} = \begin{pmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \\ \\ h_\theta(x^{(n)}) - y^{(n)} \end{pmatrix}$$

Regression

$$\begin{aligned}\frac{1}{2}(X\theta-\vec{y})^T(X\theta-\vec{y}) &= \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2 \\ &= J(\theta)\end{aligned}$$

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta-\vec{y})^T(X\theta-\vec{y}) \\ &= \frac{1}{2}\nabla_\theta\left(\theta^T X^T X\theta-\theta^T X^T\vec{y}-\vec{y}^T X\theta+\vec{y}^T\vec{y}\right) \\ &= \frac{1}{2}\nabla_\theta \operatorname{tr}\left(\theta^T X^T X\theta-\theta^T X^T\vec{y}-\vec{y}^T X\theta+\vec{y}^T\vec{y}\right) \\ &= \frac{1}{2}\nabla_\theta\left(\operatorname{tr}\theta^T X^T X\theta-2\operatorname{tr}\vec{y}^T X\theta\right) \\ &= \frac{1}{2}\left(X^T X\theta+X^T X\theta-2X^T\vec{y}\right) \\ &= X^T X\theta-X^T\vec{y}\end{aligned}$$

Régression

$$(X\theta - \vec{y})^T (X\theta - \vec{y}) = \begin{pmatrix} h_\theta(x^{(1)}) - y^{(1)} & \cdots\cdots & h_\theta(x^{(n)}) - y^{(n)} \end{pmatrix} \begin{pmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ \vdots \\ h_\theta(x^{(n)}) - y^{(n)} \end{pmatrix}$$

$$\begin{aligned} \frac{1}{2}(X\theta - \vec{y})^T (X\theta - \vec{y}) &= \frac{1}{2}\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \\ &= J(\theta) \end{aligned}$$

Régression

$$X^T X\theta = X^T \vec{y}$$

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

Décision Bayesienne

Introduction à la théorie de la décision Baysienne

- La théorie de la décision Baysienne est basée sur un compromis entre une approche probabiliste de la décision et le coût qui accompagne cette décision.
  - Elle s'appuie sur une connaissance sur la connaissance des probabilités qui influent sur la décision.

Introduction à la théorie de la décision Baysienne

- Un ensemble d'états ou de classes qui sont considérées comme des variables aléatoires.

$$W = \{\omega_i \mid i = 1..n\}$$

  - Sur notre exemple:

ω_1 représente la classe des bars

ω_2 représente la classe des saumons

Introduction à la théorie de la décision Baysienne

- Il nous faut un “a priori”sur la probalité des états.

$P(\omega_i)$ représente la probabilité à priori de la classe ω_i

$$\sum_{i=1}^{i=n} P(\omega_i) = 1$$

Sur notre exemple:

$P(\omega_1)$ représente la probabilité à priori d'avoir un bar

$P(\omega_2)$ représente la probabilité à priori d'avoir un saumon

$$P(\omega_1) + P(\omega_2) = 1$$

Comment déterminer ces deux probabilités ?

Introduction à la théorie de la décision Baysienne

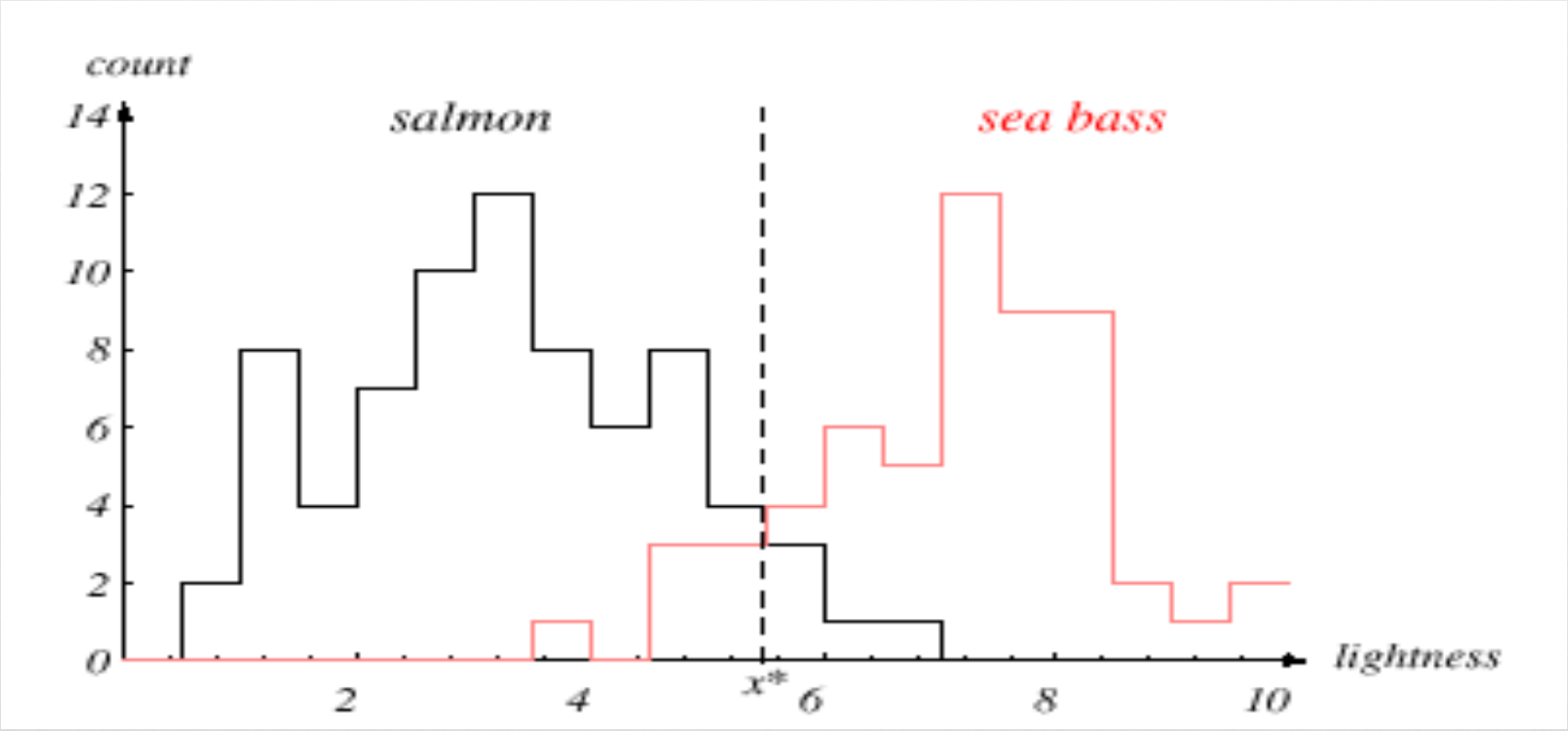
Sans information suplémentaire le comportement du classifieur serait:

Si $P(\omega_1) > P(\omega_2)$ alors $\omega = \omega_1$(bar) sinon $\omega = \omega_2$(saumon)

Est-ce la bonne décision ???
Que se passe t'il si les deux probabilités sont égales??

Introduction à la théorie de la décision Baysienne

Et si on utilisait plus d'information comme la brillance



Introduction à la théorie de la décision Baysienne

- On utilise le descripteur x

$P(x \mid \omega_i)$ la densité de probabilité de ω_i au point x

Autrement dit qu'elle est la probabilité d'avoir x sachant que j'ai ω_i

Sur notre exemple

$P(x \mid \omega_1)$ la probabilité d'avoir x si l'objet est un saumon

$P(x \mid \omega_2)$ la probabilité d'avoir x si l'objet est un bar

Introduction à la théorie de la décision Baysienne

Comment utiliser l'information de luminance pour obtenir une probabilité conditionnelle

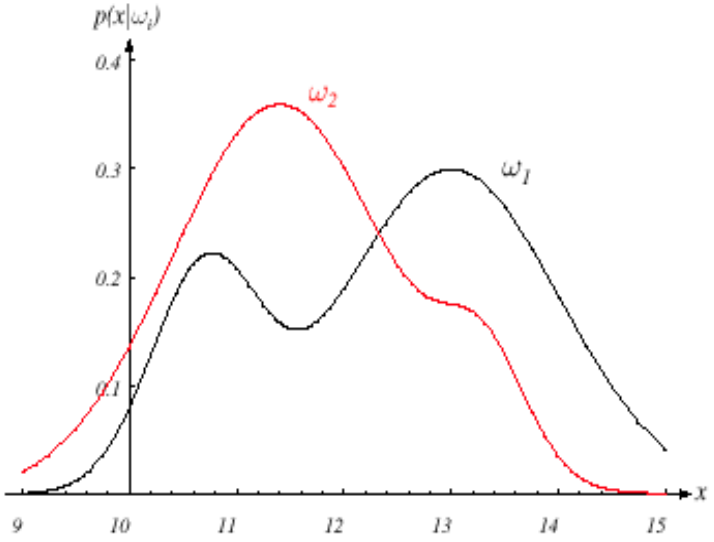


FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i. If x represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Introduction à la théorie de la décision Baysienne

Le théorème de Bayles ou comment renverser les probabilités quand cela nous arrange.

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

$$P\left(\omega_i \mid x\right) = \frac{P\left(x \mid \omega_i\right)P\left(\omega_i\right)}{P\left(x\right)}$$

avec $P\left(x\right) = \sum_{i=1}^{i=C} P\left(x \mid \omega_i\right)P\left(\omega_i\right)$ cette valeur est nécessaire pour avoir $\sum_{i=1}^{i=C} P\left(\omega_i \mid x\right) = 1$

$P\left(\omega_i \mid x\right)$ est la probabilité a posteriori que x détermine ω_i

Introduction à la théorie de la décision Baysienne

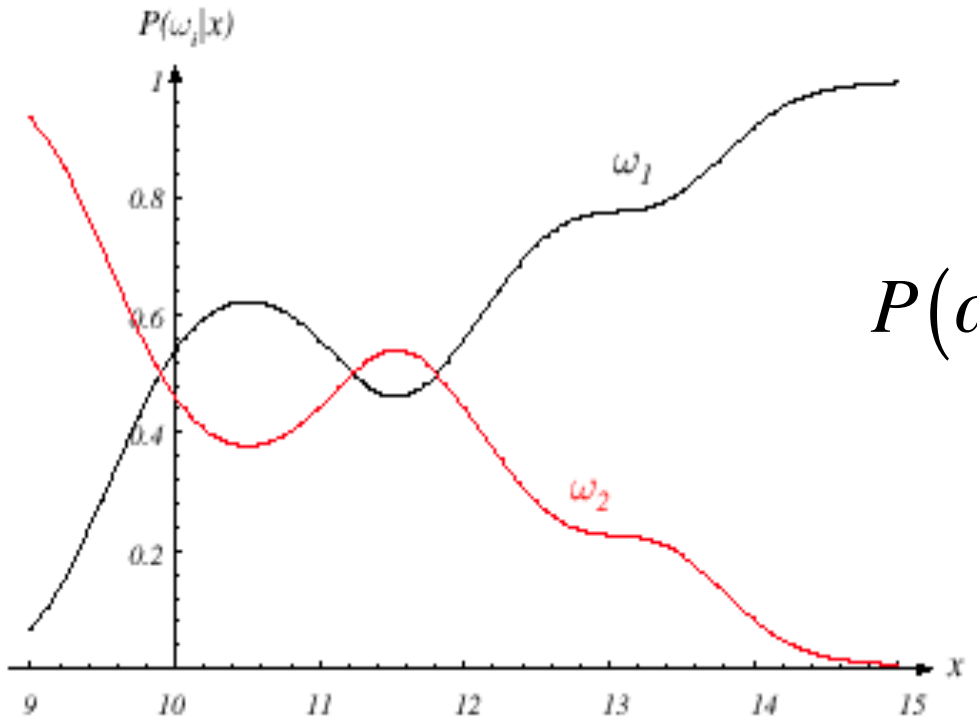


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

$$P(\omega_i \mid x) = \frac{P(x \mid \omega_i) P(\omega_i)}{P(x)}$$

Un nouveau classifieur

- On peut maintenant tenir compte des probabilités conditionnelles

Pour un x observé

Si $P(\omega_1 \mid x) \geq P(\omega_2 \mid x)$ alors $\omega = \omega_1$(bar) sinon $\omega = \omega_2$(saumon)

Ou encore si $P(x \mid \omega_1)P(\omega_1) \geq P(x \mid \omega_2)P(\omega_2)$ alors $\omega = \omega_1$(bar) sinon $\omega = \omega_2$(saumon)

- Quelle est l'erreur commise.

Quelle est l'erreur commise

- La probabilité de l'erreur est alors :

$$P(error \mid x) = P(\omega_1 \mid x) \text{si on a décidé } \omega_2$$
$$P(error \mid x) = P(\omega_2 \mid x) \text{si on a décidé } \omega_1$$

- Avec le classifieur précédent:

$$P(error \mid x) = \min\left[P(\omega_1 \mid x), P(\omega_2 \mid x)\right]$$

Généralisation de l'approche bayesienne.

- Utilisation d'un ensemble de descripteurs
- Utilisation de plus de deux classes
- Permettre des actions et non pas simplement donner un état de la nature.
- Introduire une fonction de cout qui est plus générale que la probabilité de l'erreur

Généralisation de l'approche bayesienne.

$W = \{w_i \mid i = 1..c\}$ l'ensemble des états (classes)

$A = \{\alpha_i \mid i = 1..c\}$ l'ensemble des actions associés aux états (classes)

Soit $\lambda(\alpha_i \mid \omega_j)$ la perte (pénalité) de faire l'action α_i sachant que l'on a vraiment un ω_j

$R(\alpha_i \mid x) = \sum_{j=1}^{j=c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid x)$ la perte de faire l'action α_i sachant que x est observé

$R(\alpha_i \mid x)$ est le risque conditionnel.

Le risque total est R = $\sum_{i=1}^{i=c} R(\alpha_i \mid x)$

Pour minimiser le risque total associé à un x observé

il faut choisir l'action associée $\alpha_i = \min_{i \in 1..c} R(\alpha_i \mid x)$

Généralisation de l'approche bayesienne.

Illustration sur un exemple à 2 classes c =2

On note $\lambda_{ij} = \lambda(\alpha_i \mid \omega_j)$

$$R(\alpha_1 \mid x) = \lambda_{11} P(\omega_1 \mid x) + \lambda_{12} P(\omega_2 \mid x)$$
$$R(\alpha_2 \mid x) = \lambda_{21} P(\omega_1 \mid x) + \lambda_{22} P(\omega_2 \mid x)$$

Le classifieur est alors le suivant :

$$\text{Si } R(\alpha_1 \mid x) < R(\alpha_2 \mid x) \text{ alors } \alpha_1 \text{ sinon } \alpha_2$$

Classifieur

$$R(\alpha_1 \mid x) < R(\alpha_2 \mid x) \Leftrightarrow (\lambda_{21} - \lambda_{11}) P(x \mid \omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) P(x \mid \omega_2) P(\omega_2)$$

Donc si $(\lambda_{21} - \lambda_{11}) P(x \mid \omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) P(x \mid \omega_2) P(\omega_2)$ alors α_1 sinon α_2

$$\frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \times \frac{P(\omega_2)}{P(\omega_1)} \text{ alors } \alpha_1 \text{ sinon } \alpha_2$$

Exercise

Application directe avec deux états

$$P(x \mid \omega_1) = N(2,\ 0.5)$$

$$P(x \mid \omega_2) = N(1.5,\ 0.2)$$

$$P(\omega_1) = 2/3$$

$$P(\omega_2) = 1/3$$

$$\lambda = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Minimisation de l'erreur de classement

- Une erreur de classement est faite lorsque, on choisit wi au lieu de wj.

$$\lambda(\alpha_i,\omega_j)=\begin{cases}0 & i=j\\ 1 & i\neq j\end{cases} \qquad i,j=1,\ldots,c$$

$$R(\alpha_i \mid x)=\sum_{j=1}^{j=c}\lambda(\alpha_i \mid \omega_j)P(\omega_j \mid x)$$

$$=\sum_{j\neq 1}P(\omega_j \mid x)=1-P(\omega_i \mid x)$$

Minimisation de l'erreur de classement

- Minimiser le risque revient à choisir le

$$P(\omega_j \mid x) \text{ maximum.}$$

- Le classifieur est alors

$$\omega_i \text{ tel que } i = \arg\max_{j \in 1..c} P(\omega_j \mid x)$$

Minimisation de l'erreur de classement

- Revenons au cas à deux classes

$$\frac{P(x\mid\omega_1)}{P(x\mid\omega_2)} > \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \times \frac{P(\omega_2)}{P(\omega_1)} \text{ alors } \alpha_1 \text{ sinon } \alpha_2$$

$$\textit{Soit} \quad \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} . \frac{P(\omega_2)}{P(\omega_1)} = \theta_\lambda \text{ alors on choisit } \omega_1 \text{ si } \frac{P(x\mid\omega_1)}{P(x\mid\omega_2)} > \theta_\lambda$$

$$\text{Avec } \lambda = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \theta_\lambda = \frac{P(\omega_2)}{P(\omega_1)} = \theta_a$$

$$\text{Avec } \lambda = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \quad \theta_\lambda = \frac{2P(\omega_2)}{P(\omega_1)} = \theta_b$$

Minimisation de l'erreur de classement

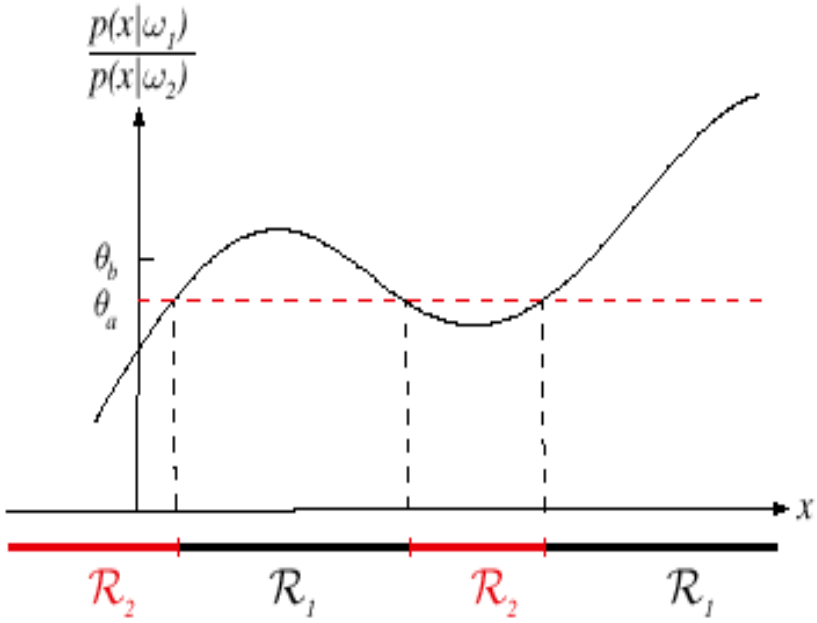


FIGURE 2.3. The likelihood ratio $p(x|\omega_1)/p(x|\omega_2)$ for the distributions shown in Fig. 2.1. If we employ a zero-one or classification loss, our decision boundaries are determined by the threshold θ_a. If our loss function penalizes miscategorizing ω_2 as ω_1 patterns more than the converse, we get the larger threshold θ_b, and hence $\mathcal{R}_1$ becomes smaller. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Classifieurs, fonctions discriminantes et surface de séparation

- Le cas de classification multi-classes.
  - Ensemble de fonctions discriminantes: $g_i(x), i = 1 \; .. \; c$
  - Le classifieur assigne un vecteur x à une classe w_i

$$\text{Si } \forall j \neq i \;\; g_i(x) > g_j(x) \text{ alors}$$

Schéma d'un classifieur multiclasse

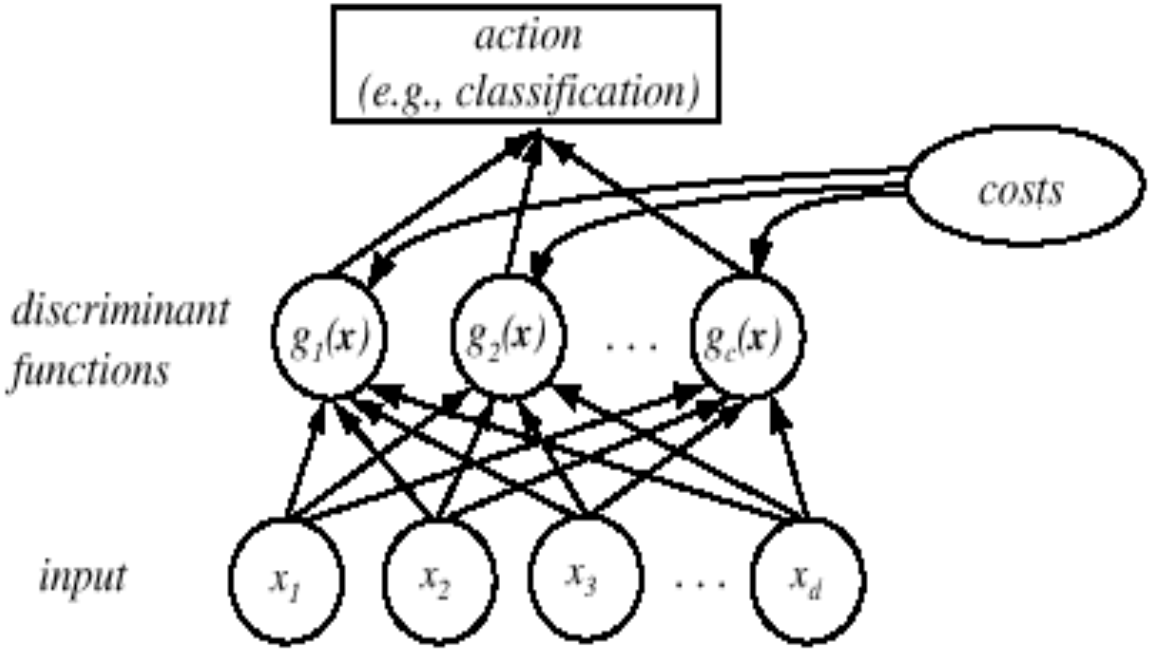


FIGURE 2.5. The functional structure of a general statistical pattern classifier which includes d inputs and c discriminant functions $g_i(\mathbf{x})$. A subsequent step determines which of the discriminant values is the maximum, and categorizes the input pattern accordingly. The arrows show the direction of the flow of information, though frequently the arrows are omitted when the direction of flow is self-evident. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Différentes illustrations

- Cas général avec risques:

$$g_i(x) = -R(\alpha_i \mid x)$$

- On peut remplacer $g_i(x)$ par
  $f(g_i(x))$ si f est une fonction monotone croissante.

$$g_i(x) = P(\omega_i \mid x) = \frac{P(x \mid \omega_i)P(\omega_i)}{\sum_{i=1}^{i=c} P(x \mid \omega_i)P(\omega_i)}$$

$$g_i(x) = P(x \mid \omega_i)P(\omega_i)$$

$$g_i(x) = \log(P(x \mid \omega_i)) + \log(P(\omega_i))$$

Classifier à deux classes

$$g(x) = g_1(x) - g_2(x)$$

Si $g(x) > 0$ alors ω_1 sinon ω_2

Minimisation de l'erreur de classification

$$g(x) = P(\omega_1 \mid x) - P(\omega_2 \mid x)$$

$$= \ln \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

Moyenne et Espérance

La moyenne $\mu = E[x] = \begin{pmatrix} \mu_i \\ \\ \\ \mu_d \end{pmatrix}$

dans le cas continue $E[x] = \int xp(x)dx$

dans le cas discret $E[x] = \frac{1}{n}\sum_{i=1}^{i=n} x_i p(x_i)$

Matrice de Covariance

La matrice de covariance $\Sigma = E[(x-\mu)(x-\mu)^t]$

dans le cas continue $E[(x-\mu)(x-\mu)^t] = \int (x-\mu)(x-\mu)^t \, p(x)dx$

dans le cas discret

$$E[(x-\mu)(x-\mu)^t] = \begin{bmatrix} Var(x_1) & Cov(x_1,x_2) & \dots & Cov(x_1,x_n) \\ Cov(x_2,x_1) & Var(x_2) & \dots & Cov(x_2,x_n) \\ \dots & & & \\ Cov(x_n,x_1) & Cov(x_n,x_2) & \dots & Var(x_n) \end{bmatrix}$$

$$\mathrm{Var}(x_i) = \frac{1}{n}\sum_{k=1}^{k=n}(x_i^k - \mu_i)^2 \text{ et } Cov(x_i,x_j) = Cov(x_j,x_i) = \frac{1}{n}\sum_{k=1}^{k=n}(x_i^k - \mu_i)(x_j^k - \mu_j)$$

Loi normale en dimension d

$$P(x) = \frac{1}{(2\pi)^{d/2} \left|\Sigma\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]$$

avec x et μ des vecteurs de dimension d

Σ la matrice de covariance

$\left|\Sigma\right|$ le déterminant de la matrice de covariance

Σ^{-1} l'inverse de la matrice de covariance

Loi Normale Multivarié

Pour une loi multivariée la probabilité P(x) est inversement proportionnelle à

$$(x-\mu)^T \Sigma^{-1} (x-\mu)$$

La distance de Malahanobis

$\text{dist}(x,y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

mesure la distance entre (x,y) en fonction de la distribution définie par Σ.

Si $\Sigma = \sigma^2 I$ on retrouve la distance euclidienne.

Si Σ est diagonale on retrouve la distance euclienne normalisée.

$$\text{dist}(x,y) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

Loi Normale Multivarié

$$(x-\mu)^T \Sigma^{-1}(x-\mu)$$

La matrice Σ est:

$1-$ définie positive donc Σ^{-1} existe

$2-$ Ses valeurs propres sont réelles.

Λ la matrice diagonales des valeurs propres

$3-$ Ses vecteurs propres sont orthogonaux

Φ la matrice dont les colonnes sont les

vecteurs propres normalisés on a $\Phi^T = \Phi^{-1}$

$4-\ \Sigma\Phi = \Phi\Lambda$ donc $\Sigma = \Phi\Lambda\Phi^{-1}$et $\Sigma^{-1} = \Phi\Lambda^{-1}\Phi^{-1}$

$5-$ on pose $\Lambda^{-1} = \Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}$

on a $\Sigma^{-1} = \Phi\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\Phi = (\Phi\Lambda^{-\frac{1}{2}})(\Phi\Lambda^{-\frac{1}{2}})^T$

$= MM^T$

Loi Normale Multivarié

$$(x-\mu)^T \Sigma^{-1}(x-\mu)$$

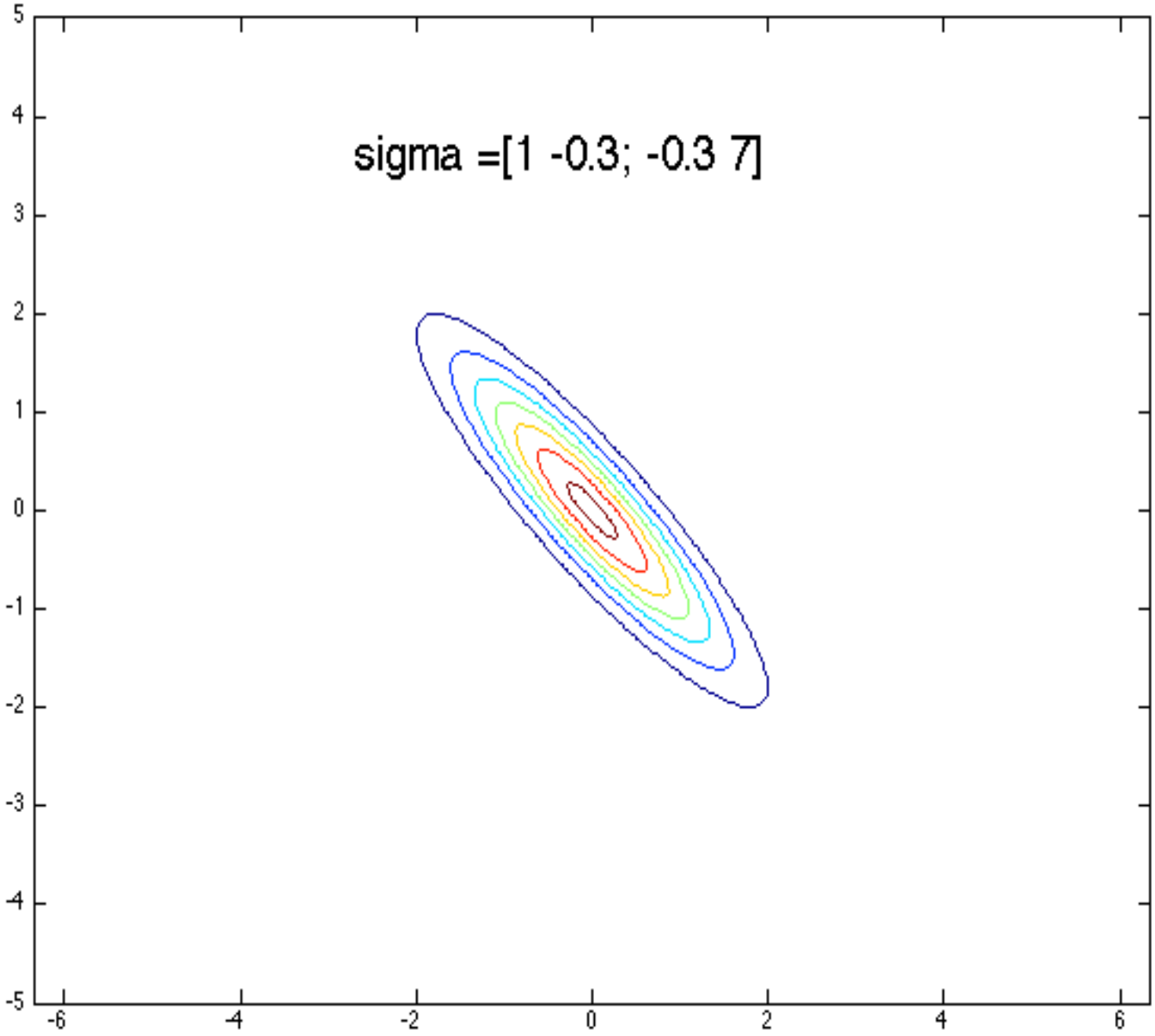
On a $(x-\mu)^T \Sigma^{-1}(x-\mu) = (x-\mu)^T (\mathrm{M}\mathrm{M}^T)(x-\mu)$

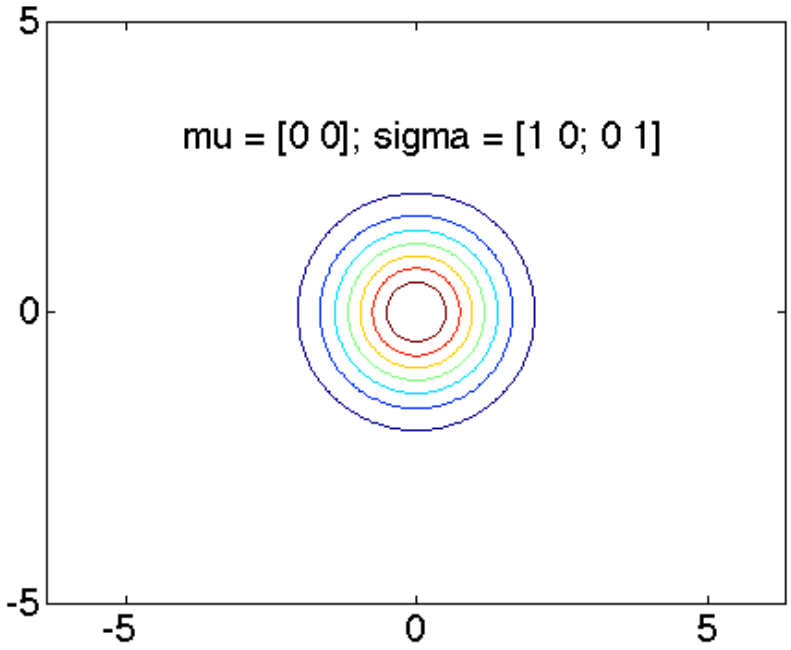
$$= (\mathrm{M}^T (x-\mu)^T)^T (\mathrm{M}^T (x-\mu)^T)$$

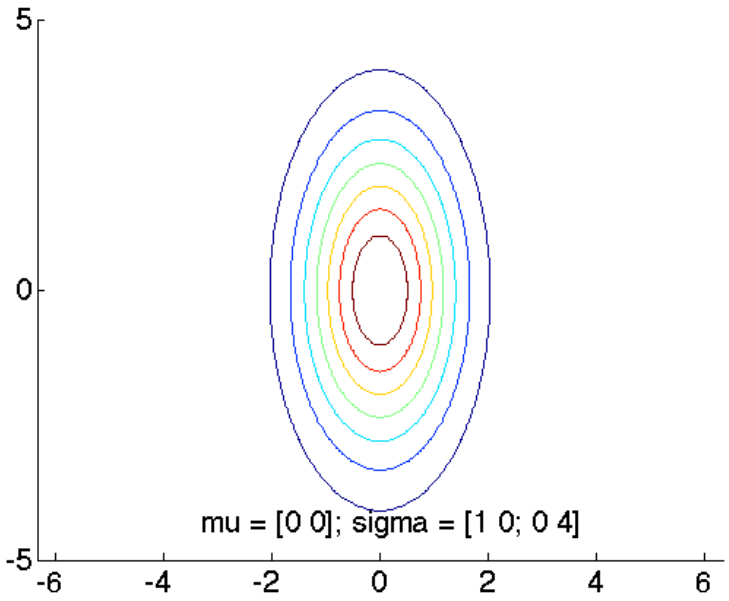
$$= \left\| \mathrm{M}^T (x-\mu) \right\| \left\| \mathrm{M}^T (x-\mu) \right\|$$

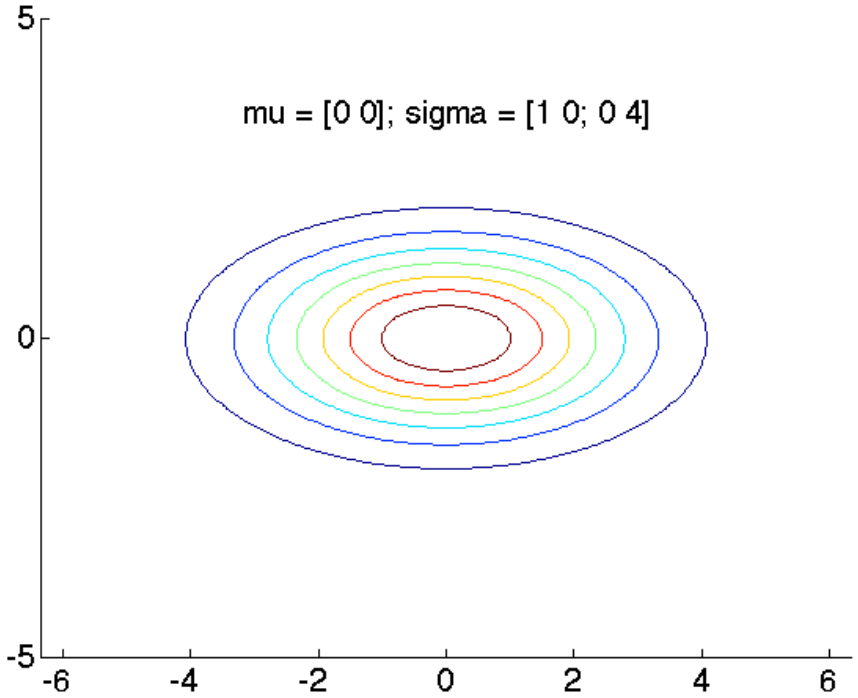
$$= \left\| \mathrm{M}^T (x-\mu) \right\|^2$$

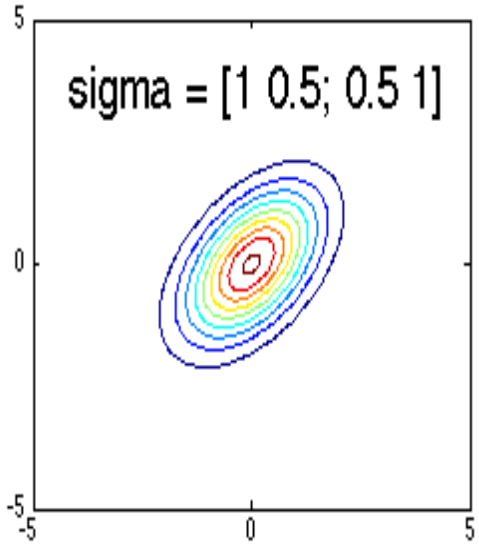
Les points de même probabilité sont situés sur une ellipse. La distance de Malahanobis permet de subsituer à la distance Euclidienne.

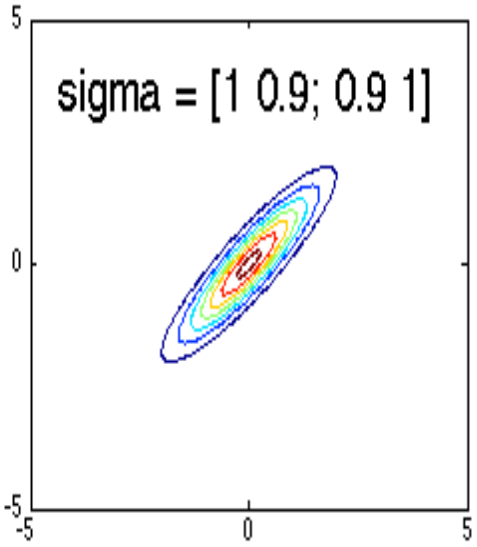

sigma =[1 -0.3; -0.3 7]
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6

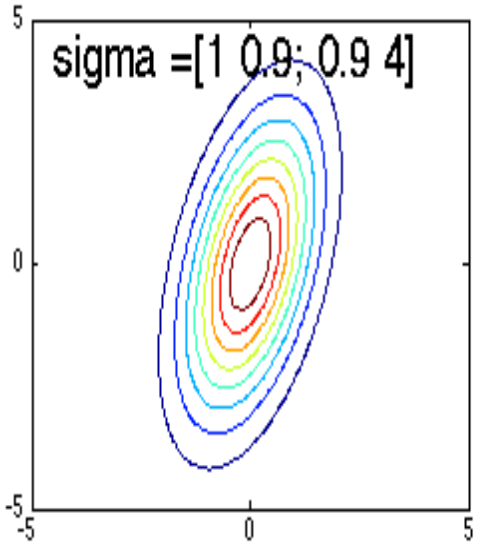

mu = [0 0]; sigma = [1 0; 0 1]
5
0
-5
-5
0
5

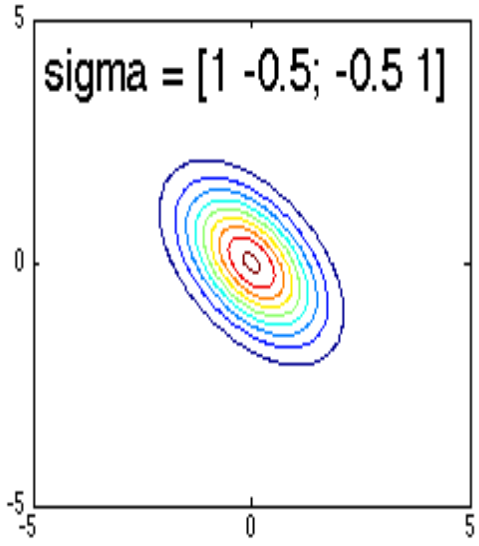

mu = [0 0]; sigma = [1 0; 0 4]
5
0
-5
-6
-4
-2
0
2
4
6

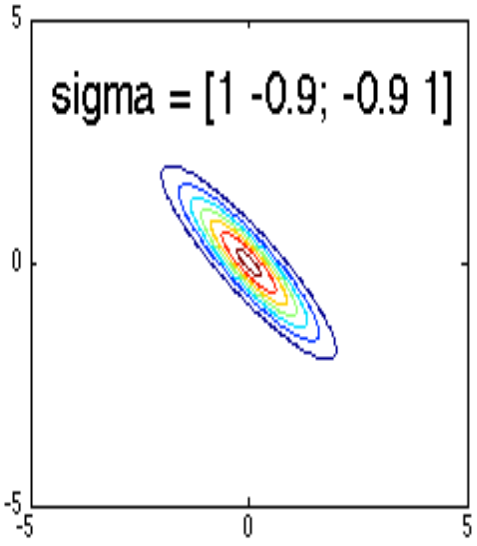

mu = [0 0]; sigma = [1 0; 0 4]
5
0
-5
-6
-4
-2
0
2
4
6

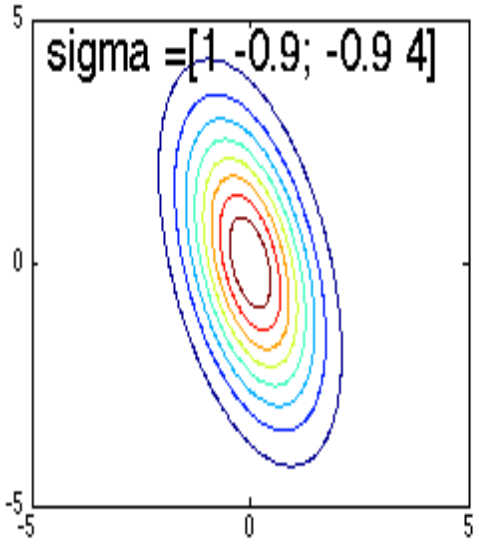

sigma = [1 0.5; 0.5 1]
5
0
-5
-5
0
5


sigma = [1 0.9; 0.9 1]
5
0
-5
-5
0
5


sigma =[1 0.9; 0.9 4]
5
0
-5
-5
0
5


sigma = [1 -0.5; -0.5 1]
5
0
-5
-5
0
5


sigma = [1 -0.9; -0.9 1]
5
0
-5
-5
0
5


sigma =[1 -0.9; -0.9 4]
5
0
-5
-5
0
5

Recentrage

Si l'ensemble X suit une loi N(μ,Σ)

alors l'ensemble AX suit une loi N($A^T\mu$, $A^T\Sigma A$)

On donc trouver A tel que $A^T\Sigma A = Id$;

On a vu que $\Sigma = \Phi\Lambda\Phi^{-1} = \Phi\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\Phi^{-1}$

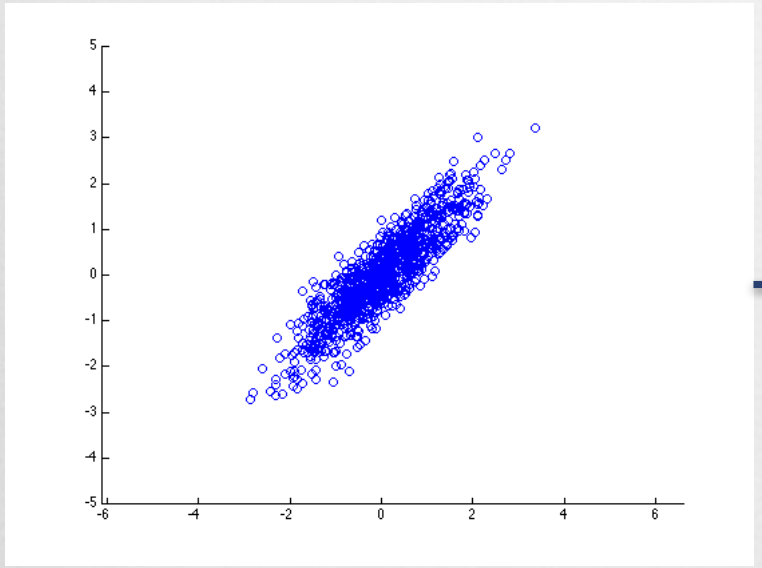
On cherche A tel que $A^T\Phi\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\Phi^{-1}A = Id$;

Si on prend $A = \Phi^{-1}\Lambda^{-\frac{1}{2}} = \Phi\Lambda^{-\frac{1}{2}}$

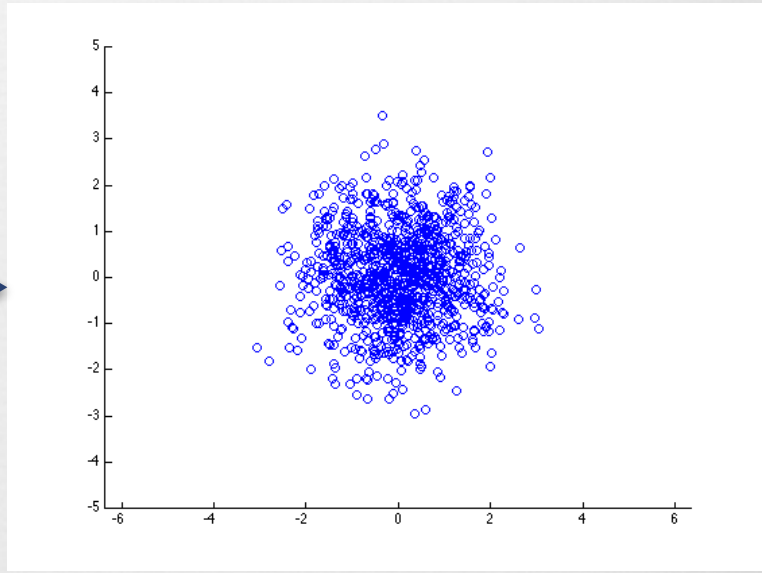
alors $A^T = \Lambda^{-\frac{1}{2}}\Phi^{-1} = \Lambda^{-\frac{1}{2}}\Phi$

Donc AX suit une loi N(μ,Id)

Recentrage



$$\Phi\Lambda^{-\frac{1}{2}}$$



mu = [0 0];
sigma = [1 0.9; 0.9 1];

mu = [0 0];
sigma = [1 0; 0 1];

```
mu    = [0 0];
sigma = [1 0.9; 0.9 1];

V1 = mvnrnd(mu,sigma, 1000);

sigmaCalc = cov(V1);
[vecp,valp] = eig(sigmaCalc);

Alph = sqrt(inv(valp));
W = vecp*Alph;
VCentre = W'*V1';

cov(Vcentre);
```

Fonctions discriminantes pour une loi normale multi-variée

Rappel : Pour un classifieur qui minimise l'erreur de classification, la fonction discriminante associée à chaque classe ω_i s'exprime comme :

$$g_i(x) = \ln P\left(\omega_i \mid x\right) + \ln P(\omega_i)$$

Si chaque probabilité $P\left(\omega_i \mid x\right)$ suit une distribution normale $N(\mu_i, \Sigma_i)$ alors

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

Fonctions discriminantes pour une loi normale multi-variée

Rappel : Pour un classifieur qui minimise l'erreur de classification, la fonction discriminante associée à chaque classe ω_i s'exprime comme :

$$g_i(x) = \ln P\left(x \mid \omega_i\right) + \ln P(\omega_i)$$

Si chaque probabilité $P\left(x \mid \omega_i\right)$ suit une distribution normale $N(\mu_i, \Sigma_i)$ alors

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

$Cas : \Sigma_i = \sigma^2 I$

Dans le cas où, tous les w_i sont indépendants et ont la même variance,

la matrice de convariance est de la forme $\Sigma_i = \sigma^2 I$

On a $\Sigma_i^{-1} = \frac{1}{\sigma^2} I$ et $\left|\Sigma_i\right| = \sigma^{2d}$

On a alors

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \frac{1}{\sigma^2} I(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\sigma^{2d} + \ln P(\omega_i)$$

On peut enlever les constantes additives, puisque toutes les classes ont les mêmes

$$g_i(x) = -\frac{\left\|x-\mu_i\right\|^2}{2\sigma^2} + \ln P(\omega_i)$$

$Cas : \Sigma_i = \sigma^2 I$

$$g_i(x) = -\frac{\|x-\mu\|^2}{2\sigma^2} + \ln P(\omega_i)$$

$$\|x-\mu\|^2 = (x-\mu_i)^t(x-\mu_i) = x^t x - 2\mu_i^t x + \mu_i^t \mu_i$$

On peut supprimer $x^t x$ puisque c'est la même valeur pour toutes les classes.

On obtient alors:

$$g_i(x) = -\frac{-2\mu_i^t x + \mu_i^t \mu_i}{2\sigma^2} + \ln P(\omega_i)$$

ou alors

$g_i(x) = w_i^t x + w_{i0}$ (fonction discriminante linéaire)

avec:

$$w_i = \frac{\mu_i}{\sigma^2}; \quad w_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + \ln P(\omega_i)$$

(ω_{i0} est appelé le seuil de la ième classe),

on obtient ainsi l'équation d'une hyper-droite.

$$Cas : \Sigma_i = \sigma^2 I$$

Equation de l'hyperplan séparant Ri and Rj

On pose $g_i(x) = g_j(x)$

$$w_i^t x + w_{i0} = w_j^t x + w_{j0} \Leftrightarrow (w_i^t - w_j^t)x + (w_{i0} - w_{j0}) = 0$$

$$w^t (x - x_0) = 0 \quad \text{avec} \quad w = \mu_i - \mu_j$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$

Toujours orthogonal à la droite joignant les moyennes.

$$Si \quad P(\omega_i) = P(\omega_j) \quad \text{alors} \quad x_0 = \frac{1}{2}(\mu_i + \mu_j)$$

$$Cas : \Sigma_i = \sigma^2 I$$

- Un classifieur qui utilise une fonction discrimante linéaire est appelé un “classifieur linéaire”
- Les surface de décision pour un classifieur linéaires sont des portions d’hyperplan définies par:

$$g_i(x) = g_j(x)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\left\|\mu_i - \mu_j\right\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$

$Cas : \Sigma_i = \Sigma$

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma| + \ln P(\omega_i)$$

On peut enlever la constante additive $-\frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma|$

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma^{-1}(x-\mu_i) + \ln P(\omega_i)$$

Si on considère

$$(x-\mu_i)^t \Sigma^{-1}(x-\mu_i) = x^t\Sigma^{-1}(x-\mu_i) - \mu_i^t\Sigma^{-1}(x-\mu_i)$$

$$= x^t\Sigma^{-1}x + x^t\Sigma^{-1}\mu_i - \mu_i^t\Sigma^{-1}x + \mu_i^t\Sigma^{-1}\mu_i$$

$$g_i(x) = -\frac{x^t\Sigma^{-1}\mu_i - \mu_i^t\Sigma^{-1}x + \mu_i^t\Sigma^{-1}\mu_i}{2} + \ln P(\omega_i)$$

ou alors

$$g_i(x) = w_i^t x + w_{i0}$$

$$w_i = \Sigma^{-1}\mu_i; \quad w_{i0} = -\frac{1}{2}\mu_i^t\Sigma^{-1}\mu_i + \ln P(\omega_i)$$

$$Cas : \Sigma_i = \Sigma$$

Equation de l'hyperplan séparant Ri and Rj

On pose $g_i(x) = g_j(x)$

$$w_i^t x + w_{i0} = w_j^t x + w_{j0} \Leftrightarrow (w_i^t - w_j^t)x + (w_{i0} - w_{j0}) = 0$$

$$w^t (x - x_0) = 0 \quad \text{avec}$$

$$w=\Sigma^{-1}(\mu_i - \mu_j)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln\left[P(\omega_i)/P(\omega_j)\right]}{(\mu_i - \mu_j)^t \Sigma^{-1}(\mu_i - \mu_j)}.(\mu_i - \mu_j)$$

L'hyperplan séparant Ri and Rj n'est pas en général orthogonal à la droite passant par les moyennes.

$Cas : \Sigma_i = quelconque$

Les matrices de covariances sont différentes pour chaque catégorie.

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

On peut enlever la constante additive $-\frac{d}{2}\ln 2\pi$

$$g_i(x) = x^t W_i x + w_i^t x + w_{i0}$$

avec :

$$W_i = -\frac{1}{2}\Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1}\mu_i$$

$$w_{i0} = -\frac{1}{2}\mu_i^t \Sigma_i^{-1}\mu_i - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

Exemple

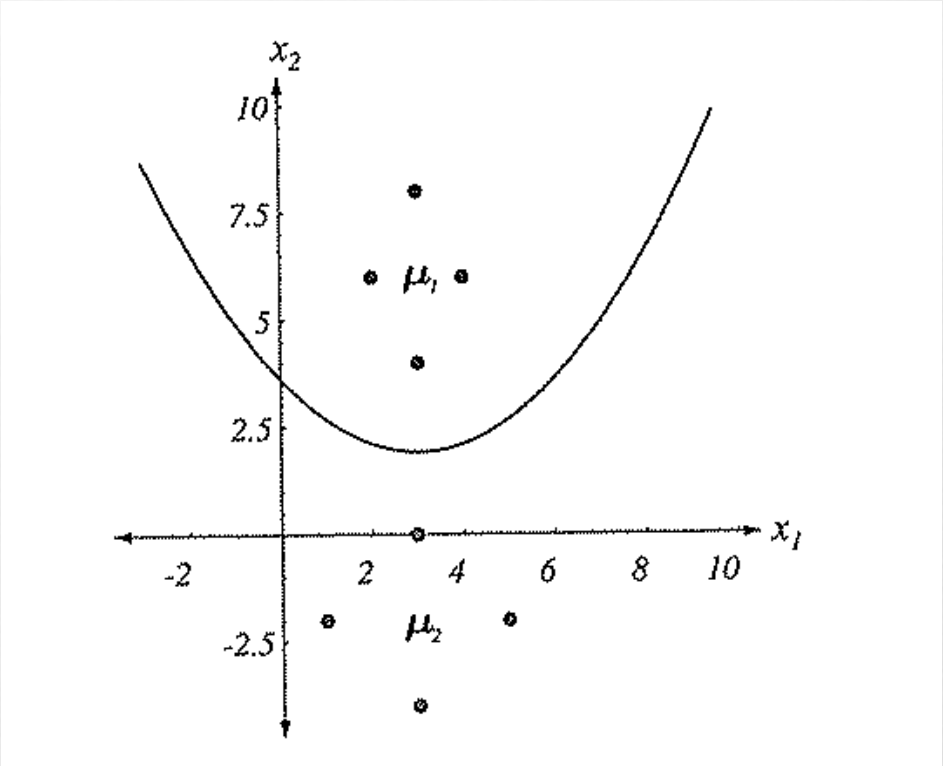
$S_1 = (3,3),(3,8),(2,6),(4,6)$

$S_1 = (1,-2),(3,0),(3,-4),(5,-2)$

$$\mu_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\Sigma_1^{-1} = \begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix} \quad \Sigma_2^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$$



Si $P(\omega_1) = P(\omega_2) = 0,5$

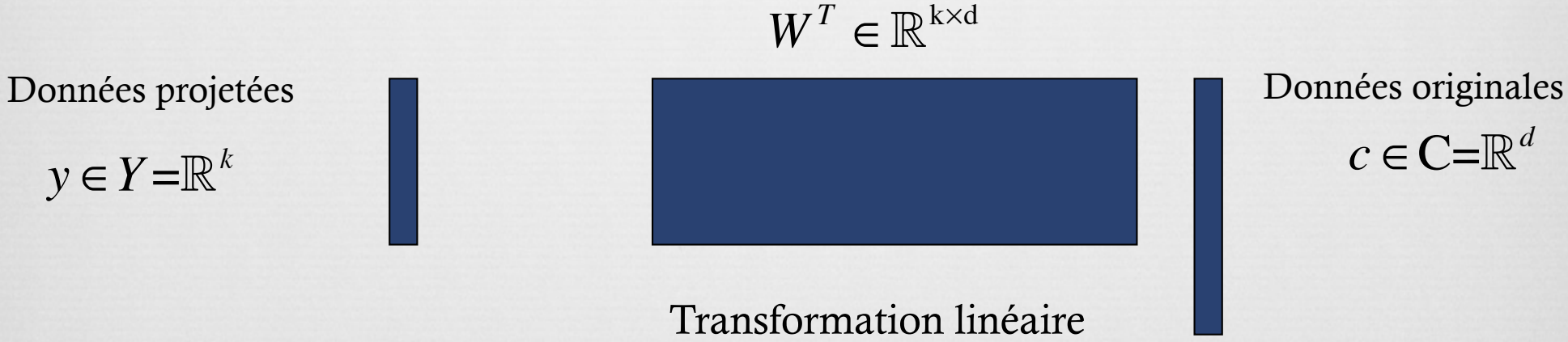
La frontière est représentée par l'équation

$x_2 = 3,514 - 1,125x_1 + 0,1875x_1^2$

ACP et ACI

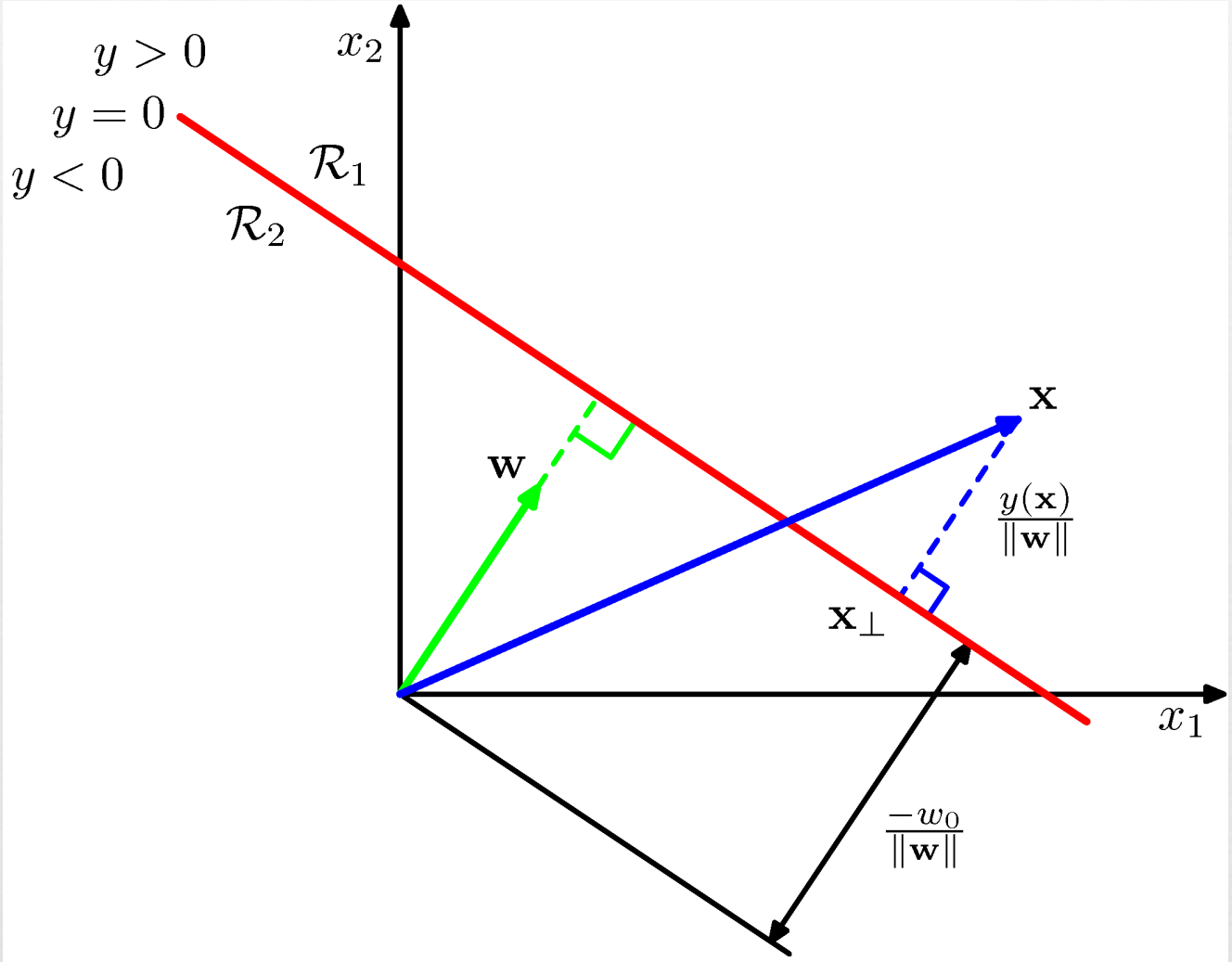
Réduction en dimension

- Les points de l'échantillon sont des éléments d'un espace affine de dimension **d**, on cherche un sous-espace vectoriel de dimension **k** qui satisfait un certain critère. On projette alors les éléments de l'échantillon dans ce nouvel espace. C'est dans ce nouvel espace que se fera la classification.

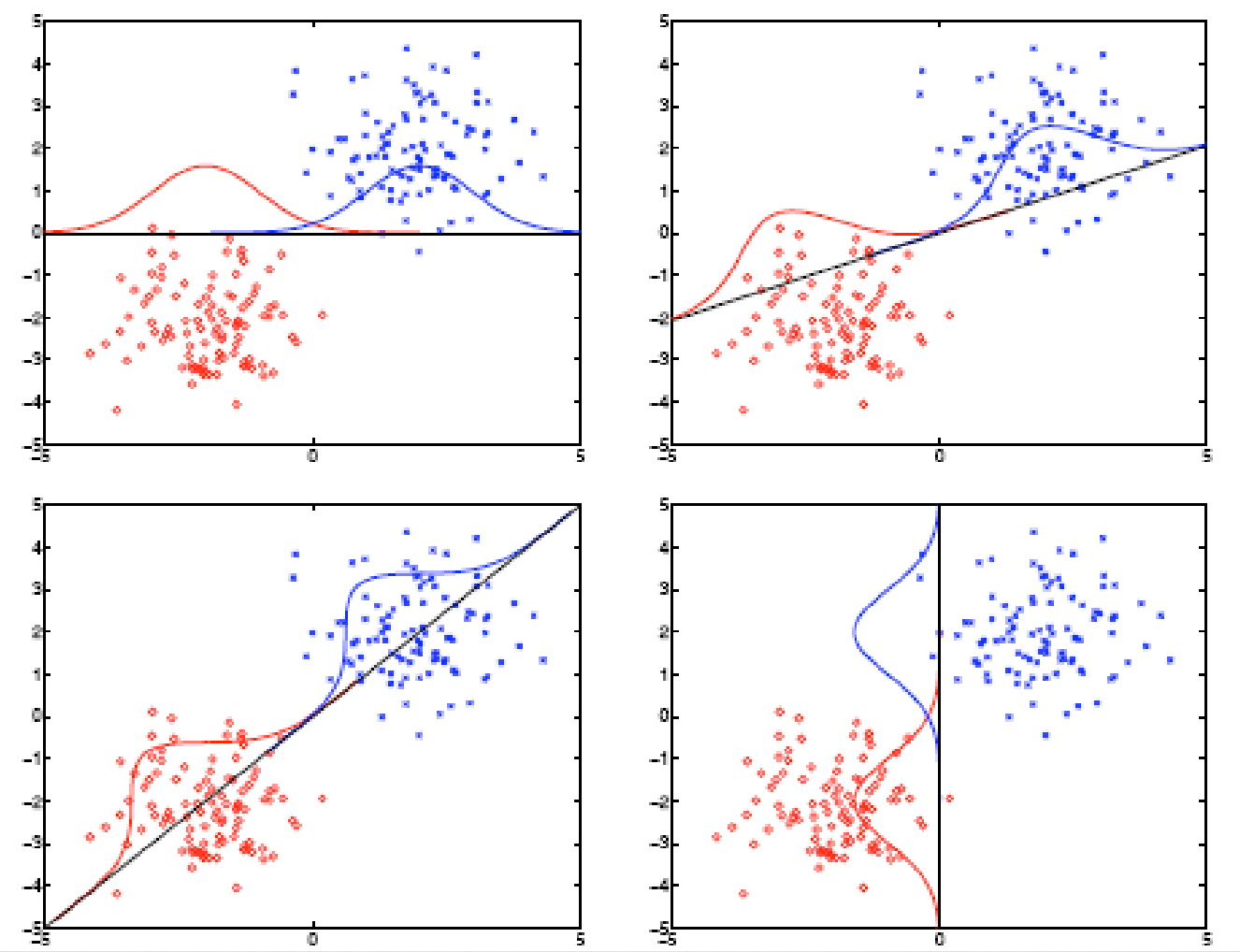


$$W \in \mathbb{R}^{d \times k} : C \to Y = W^T X \in \mathbb{R}^k$$

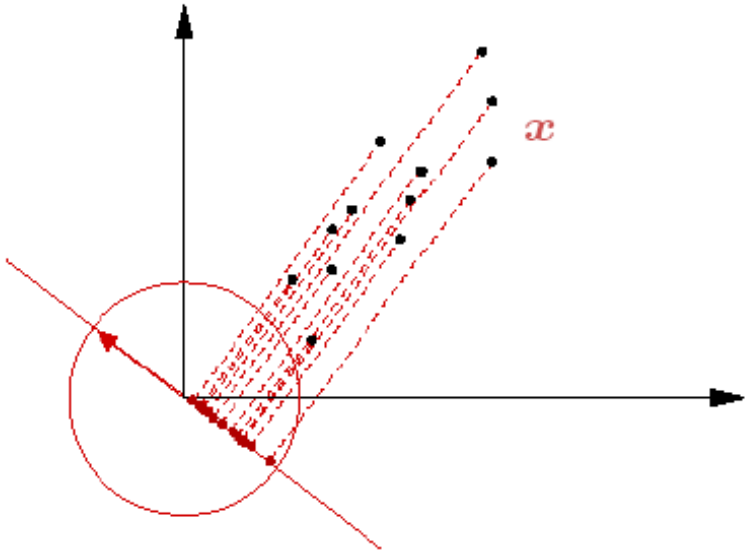
Résumé sur la géométrie



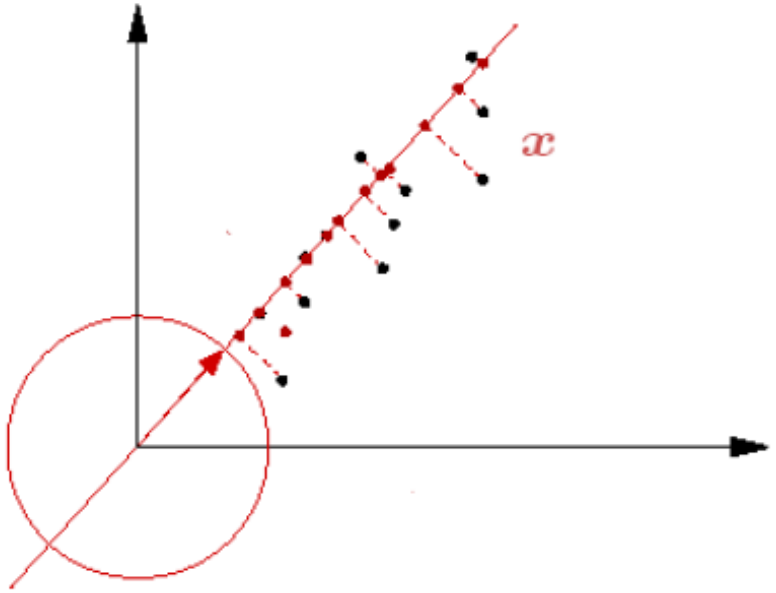
Réduction en dimension



Deux exemples de critères



Projection basée sur le critère correspondant à la meilleure discrimination.

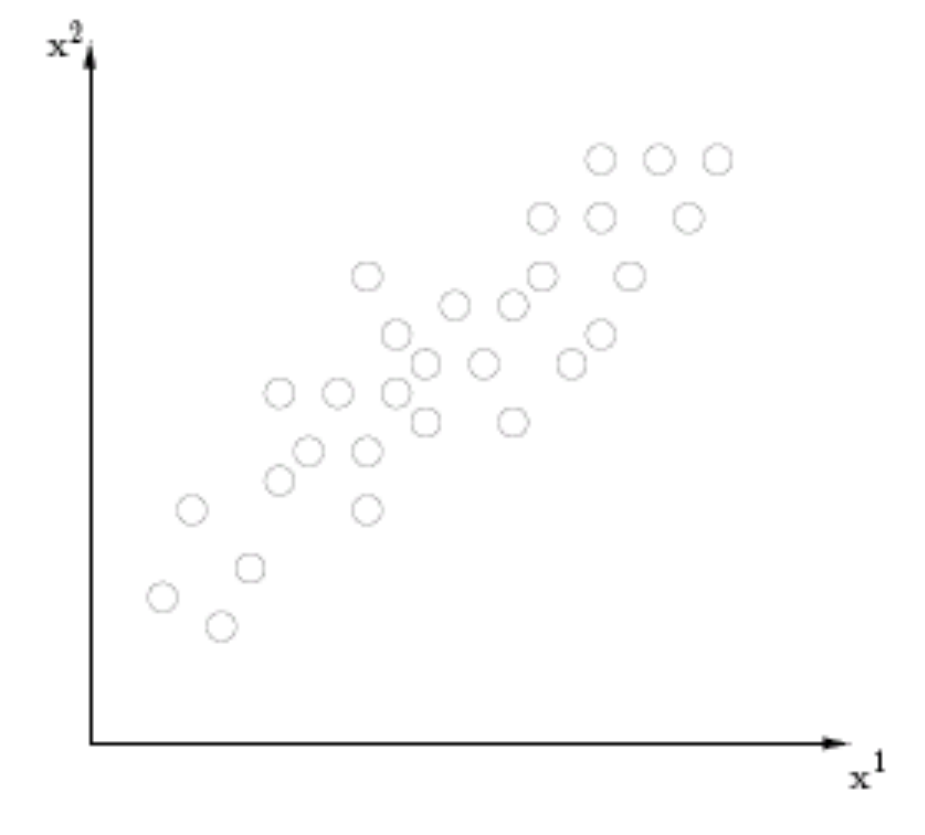


Projection basée sur le critère Permettant de conserver le plus d'information possible.

ACP

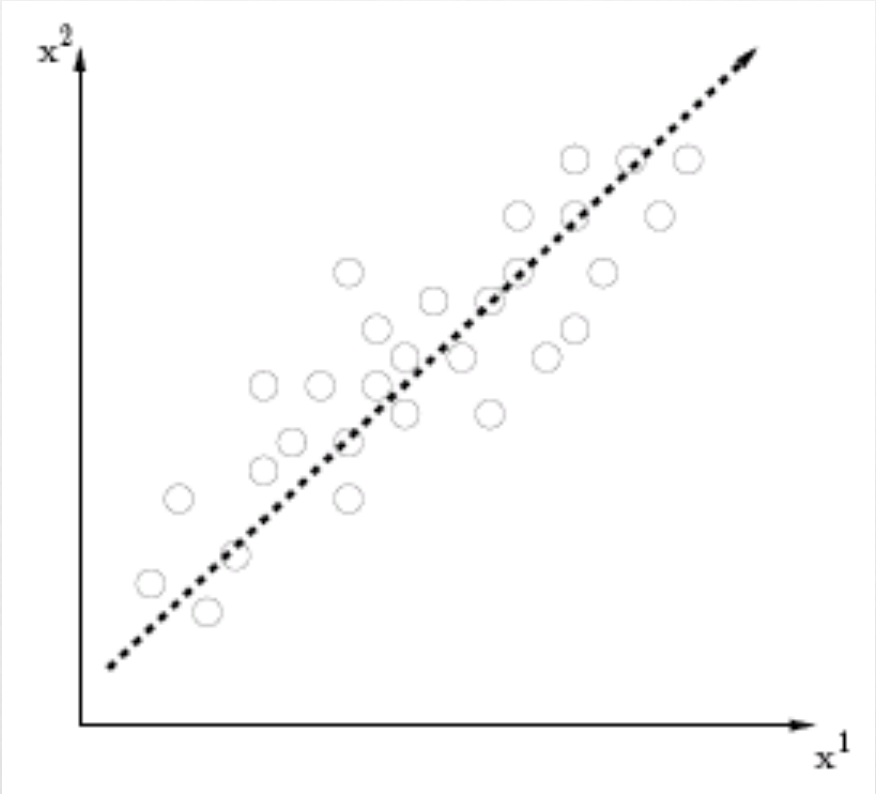
PCA :Exemple en dimension 2

- On veut passer de 2 variables à 1 seule.



PCA : Exemple en dimension 2

- On cherche la direction qui différencie le plus les points entre eux.



L'analyse en composantes principales

- L’objectif principal de l’analyse en composantes principales est de conserver le plus d’information possible dans la projection, ce qui revient à essayer d’avoir la plus grande variance sur l’ensemble des points projetés. On cherche l’axe de plus forte variance
- Pour cela, il faut que la somme des erreurs de projections soit minimale.

Formalisation

Pour pouvoir travailler dans un espace vectoriel, il faut recentrer
le nuage de point à l'origine, on calcule la moyenne des points de l'échatillon
on note μ cette moyenne est on transforme tous les points p_i
de l'échantillon en c_i tel que $c_i = p_i - \mu$

$$\begin{pmatrix} y_1 \\ : \\ y_k \end{pmatrix} = \begin{pmatrix} e_{11} & \cdots & e_{1d} \\ \vdots & \ddots & \vdots \\ e_{k1} & \cdots & e_{kd} \end{pmatrix} \begin{pmatrix} c_1 \\ : \\ : \\ c_d \end{pmatrix}$$

avec $\begin{pmatrix} c_1 \\ : \\ : \\ c_d \end{pmatrix}$ un élement de l'échantillon dans l'espace de dimension d;

$W^T = \begin{pmatrix} e_{11} & \cdots & e_{1d} \\ \vdots & \ddots & \vdots \\ e_{k1} & \cdots & e_{kd} \end{pmatrix}$ la matrice de projection de $\mathbb{R}^d \to \mathbb{R}^k$;

$\begin{pmatrix} y_1 \\ : \\ y_k \end{pmatrix}$ le vecteur résultant de la projection qui sera utilisé pour la classification, il est de dimension k.

$$\begin{pmatrix} y_1 \\ : \\ y_k \end{pmatrix} = W^T \begin{pmatrix} c_1 \\ : \\ : \\ c_d \end{pmatrix}$$

Formalisation

- Les colonnes de la matrice W forment une base du sous-espace vectoriel de dimension k. On notera :

$$\vec{e}_i = \begin{pmatrix} e_{1,i} \\ : \\ e_{2,i} \\ : \\ e_{d,i} \end{pmatrix}$$ le ième vecteur de la base

On considérera que nous avons une base orthonormée c'est à dire

$\forall$ i $\neq$ j, $\vec{e}_i . \vec{e}_j = 0$

$\forall$ i, $\vec{e}_i . \vec{e}_i = 1$

L'analyse en composantes principales

Soit $B = \{\vec{e}_1, \ldots \vec{e}_k\}$ la base orthormée représentant la projection.

Soit $Ec = \{c_1, \ldots, c_N\}$ l'ensemble des échantillons

Soit c_i un élément de l'échantillon,

Soit $(\alpha_{i,1}, \ldots \alpha_{i,k})$ les coordonnées de c_i dans la base B

Soit $\tilde{c}_i = \sum_{j=1}^{j=k} \alpha_{i,j} \vec{e}_j$ le projeté de c_i dans le sous espace défini par B.

L'erreur commise pour un échantillon s'exprime alors par $[\![c_i - \tilde{c}_i]\!]_2 = [\![c_i - \sum_{j=1}^{j=k} \alpha_{i,j} \vec{e}_j]\!]_2$

on doit minimiser $\sum_{i=1}^{i=N} [\![c_i - \sum_{j=1}^{j=k} \alpha_{i,j} \vec{e}_j]\!]_2$

Soit $J(\vec{e}_1, \ldots \vec{e}_k, \alpha_{1,1 \ldots\ldots}, \alpha_{n,k}) = \sum_{i=1}^{i=N} [\![c_i - \sum_{j=1}^{j=k} \alpha_{i,j} \vec{e}_j]\!]_2$ la fonction à minimiser.

L'analyse en composantes principales

$$J(\vec{e}_1,\dots \vec{e}_k,\alpha_{1,1\dots\dots},\alpha_{n,k}) = \sum_{i=1}^{i=N} [\![c_i - \sum_{j=1}^{j=k} \alpha_{i,j}\vec{e}_j]\!]_2$$

$$= \sum_{i=1}^{i=N} [\![c_i]\!]^2 - 2\sum_{i=1}^{i=N} c_i^t (\sum_{j=1}^{j=k} \alpha_{i,j}\vec{e}_j) + \sum_{i=1}^{i=N}\sum_{j=1}^{j=k} \alpha^2{}_{i,j}$$

$$= \sum_{i=1}^{i=N} [\![c_i]\!]^2 - 2\sum_{i=1}^{i=N}\sum_{j=1}^{j=k} \alpha_{i,j} c_i^t \vec{e}_j + \sum_{i=1}^{i=N}\sum_{j=1}^{j=k} \alpha^2{}_{i,j}$$

On calcule maintenant les dérivées partielles avec en $\alpha_{i,j}$:

$$\frac{\partial(\sum_{i=1}^{i=n} [\![c_i]\!]^2 - 2\sum_{i=1}^{i=n}\sum_{j=1}^{j=k} \alpha_{i,j} c_i^t \vec{e}_j + \sum_{i=1}^{i=n}\sum_{j=1}^{j=k} \alpha^2{}_{i,j})}{\partial \alpha_{i,j}} = -2c_i^t \vec{e}_j + 2\alpha_{i,j}$$

Le minimum annule les dérivées partielles donc :

$$\frac{\partial(\sum_{i=1}^{i=n} [\![c_i]\!]^2 - 2\sum_{i=1}^{i=n}\sum_{j=1}^{j=k} \alpha_{i,j} c_i^t \vec{e}_j + \sum_{i=1}^{i=n}\sum_{j=1}^{j=k} \alpha^2{}_{i,j})}{\partial \alpha_{i,j}} = 0 \Rightarrow$$

$$-2c_i^t \vec{e}_j + 2\alpha_{i,j} = 0 \Rightarrow$$

$$\alpha_{i,j} = c_i^t \vec{e}_j$$

L'analyse en composantes principales

$$J(\vec{e}_1,\dots \vec{e}_k,\alpha_{1,1\dots\dots},\alpha_{N,k}) = \sum_{i=1}^{i=N}[\![c_i]\!]^2 - 2\sum_{i=1}^{i=N}\sum_{j=1}^{j=k}\alpha_{i,j}c_i^t\vec{e}_j + \sum_{i=1}^{i=N}\sum_{j=1}^{j=k}\alpha^2{}_{i,j})$$

On injecte maintenant la condition $\alpha_{i,j} = c_i^t\vec{e}_j$ dans l'équation précédente:

$$J(\vec{e}_1,\dots \vec{e}_k) = \sum_{i=1}^{i=N}[\![c_i]\!]^2 - 2\sum_{i=1}^{i=N}\sum_{j=1}^{j=k}c_i^t\vec{e}_j c_i^t\vec{e}_j + \sum_{i=1}^{i=N}\sum_{j=1}^{j=k}(c_i^t\vec{e}_j)^2$$

$$J(\vec{e}_1,\dots \vec{e}_k) = \sum_{i=1}^{i=N}[\![c_i]\!]^2 - \sum_{i=1}^{i=N}\sum_{j=1}^{j=k}(c_i^t\vec{e}_j)^2$$

Comme $(c_i^t\vec{e}_j)^2 = (c_i^t\vec{e}_j)(c_i^t\vec{e}_j) = (\vec{e}_j{}^t c_i)(c_i^t\vec{e}_j) = \vec{e}_j{}^t(c_i)(c_i^t)\vec{e}_j$

$$J(\vec{e}_1,\dots \vec{e}_k) = \sum_{i=1}^{i=N}[\![c_i]\!]^2 - \sum_{j=1}^{j=k}\vec{e}_j{}^t(\sum_{i=1}^{i=N}c_i c_i^t)\vec{e}_j$$

Comme $c_i = p_i - \mu$ on a $S = \sum_{i=1}^{i=N}c_i c_i^t = \sum_{i=1}^{i=N}(p_i-\mu)(p_i-\mu)^t$

S est appelée la "scatter matrice" elle est égale N-1 fois la matrice de covariance.

On a $J(\vec{e}_1,\dots \vec{e}_k) = \sum_{i=1}^{i=N}[\![c_i]\!]^2 - \sum_{j=1}^{j=k}\vec{e}_j{}^t\ S\ \vec{e}_j$

L'analyse en composantes principales

$$J(\vec{e}_1,\dots \vec{e}_k) = \sum_{i=1}^{i=N} [\![c_i]\!]^2 - \sum_{j=1}^{j=k} \vec{e}_j{}^t \; S \, \vec{e}_j$$

On sait que $\sum_{i=1}^{i=N} [\![c_i]\!]^2$ est constant et ne dépend donc pas des $\vec{e}_i$

Donc minimiser $J(\vec{e}_1,\dots \vec{e}_k)$ est équivalent à maximiser $\tilde{J}(e_1,\dots e_k) = \sum_{j=1}^{j=k} e_j{}^t \; S \, e_j$

Sachant que $B = \{ \vec{e}_1,\dots \vec{e}_k \}$ est une base orthormée on peut injecter un nouveau terme basé sur le principe du Lagrangien $\sum_{j=1}^{j=k} \lambda_j (e_j^t e_j - 1) = 0, \forall \lambda_j$

On doit donc maximiser :

$$\tilde{J}(e_1,\dots e_k) = \sum_{j=1}^{j=k} e_j{}^t \; S \, e_j - \sum_{j=1}^{j=k} \lambda_j (e_j^t e_j - 1)$$

S est une matrice symétrique définie et semi-positive donc $\frac{\partial (e_j{}^t S e_j)}{\partial e_j} = 2 S e_j$

$$\frac{\partial \tilde{J}(e_1,\dots e_k)}{\partial e_j} = \frac{\partial (\sum_{j=1}^{j=k} e_j{}^t \; S \, e_j - \sum_{j=1}^{j=k} \lambda_j (e_j^t e_j - 1))}{\partial e_j} = 2 S e_j - 2 \lambda_j e_j = 2 (S e_j - 2 \lambda_j e_j)$$

L'analyse en composantes principales

On a $\frac{\partial \tilde{J}(e_1,\dots e_k)}{\partial e_j} = 2(Se_j - \lambda_j e_j)$;

Il faut résoudre le système suivant :

$$\frac{\partial \tilde{J}(e_1,\dots e_k)}{\partial e_j} = 0 \Leftrightarrow 2(Se_j - \lambda_j e_j) = 0 \Leftrightarrow Se_j = \lambda_j e_j$$

Les solutions du systèmes sont les vecteurs propres de la matrice S.

La base B est donc constituée des vecteurs propres de la matrice S.

Si on revient à la fonction $J(e_1,\dots e_k) = \sum_{i=1}^{i=n} [\![c_i]\!]^2 - \sum_{j=1}^{j=k} e_j^{\,t}\ S\, e_j$

$$J(e_1,\dots e_k) = \sum_{i=1}^{i=n} [\![c_i]\!]^2 - \sum_{j=1}^{j=k} e_j^{\,t} \lambda_j e_j^{\,t} = \sum_{i=1}^{i=n} [\![c_i]\!]^2 - \sum_{j=1}^{j=k} \lambda_j \quad (\text{car } e_j^{\,t} e_j = 1)$$

Pour minimiser J il faut prendre les valeurs propres de plus grande valeur.

L'espace vectoriel de dimension k qui satisfait le mieux au critère est celui constitué des k vecteurs propres des plus grandes valeurs propres.

L'analyse en composantes principales

Soit $P = \{p_1, \ldots, p_N\}$ l'ensemble à traiter

1: Calculer la moyenne $\mu = \dfrac{\sum_{j=1}^{j=N} p_j}{N}$

2 : Recentrer P en le transformant en $C = \{c_1 = (p_1 - \mu), \ldots c_N = (p_N - \mu)\}$

3 : Calculer S avec $S = \sum_{j=1}^{j=N} c_j c_j^t$

4 : Calculer les couples (valeurs propres, vecteurs propres) de S, les (λ_i, e_i);

5 : Retenir les k vecteurs propres $(e_i, \ldots e_k)$ de plus grandes valeurs propres avec $\lambda_1 \leq \ldots \leq \lambda_k$;

6: Former $W = \begin{pmatrix} e_{11} & \cdots & e_{1d} \\ \vdots & \ddots & \vdots \\ e_{k1} & \cdots & e_{kd} \end{pmatrix}$ la matrice de projection à partir des $(e_i, \ldots e_k)$

7 : Effectuer la projection des c_i

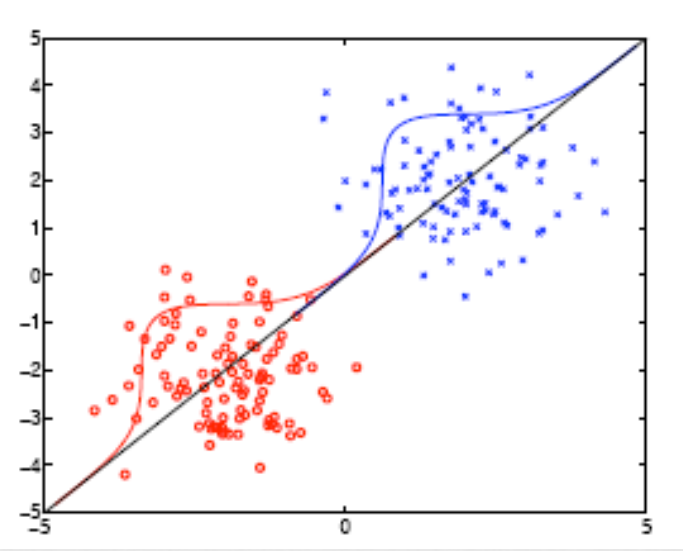
L'analyse en composantes principales
Code Matlab

```
load fisheriris.mat;
P = [meas(:,1),meas(:,2),meas(:,3)];

mu = mean(P);                          % Etape 1
C    = P - repmat(mu, size(P,1),1);    % Etape 2

S = (size(C,1) - 1) * cov(C);          %Etape 3

[e,lambda]=eigs(S);                    %Etape 4

W2 = transpose([e(:,1),e(:,2)])        % Etape 5 et Etape 6
Cproj2 = transpose(W2*C');             % Etape 7

W1 = transpose([e(:,1)])               % Etape 5 et Etape 6
Cproj1 = transpose(W1*C');             % Etape 7


afficherPlan(C,W2,mu);
afficherDroite(C,W1,mu);
```

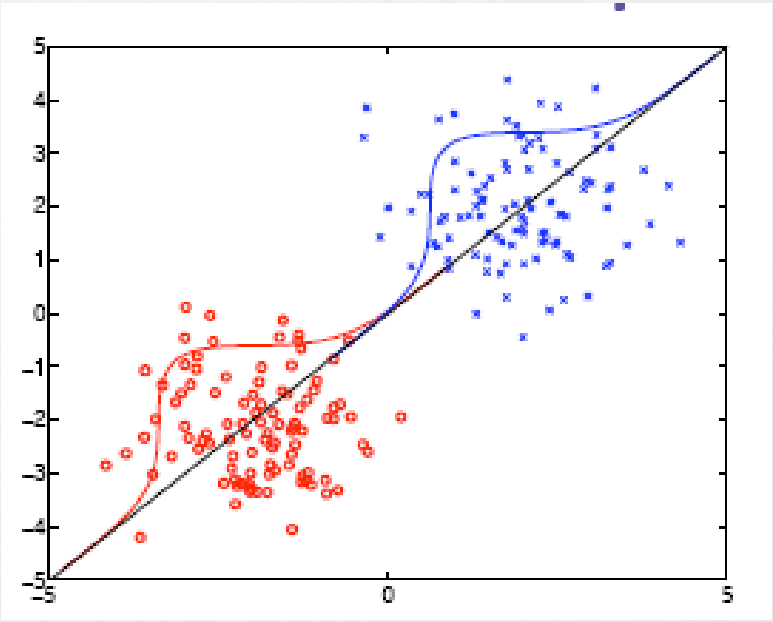
ACI

Choisir la meilleure Projection

- On veut trouver la droite qui sépare le mieux les projections des deux classes



Fisher Linear Discriminant: Preliminaries



On a deux classes C_1 et C_2

On note dans $\mathbb{R}^d$:

(μ_1, s_1^2) le couple associé à C_1

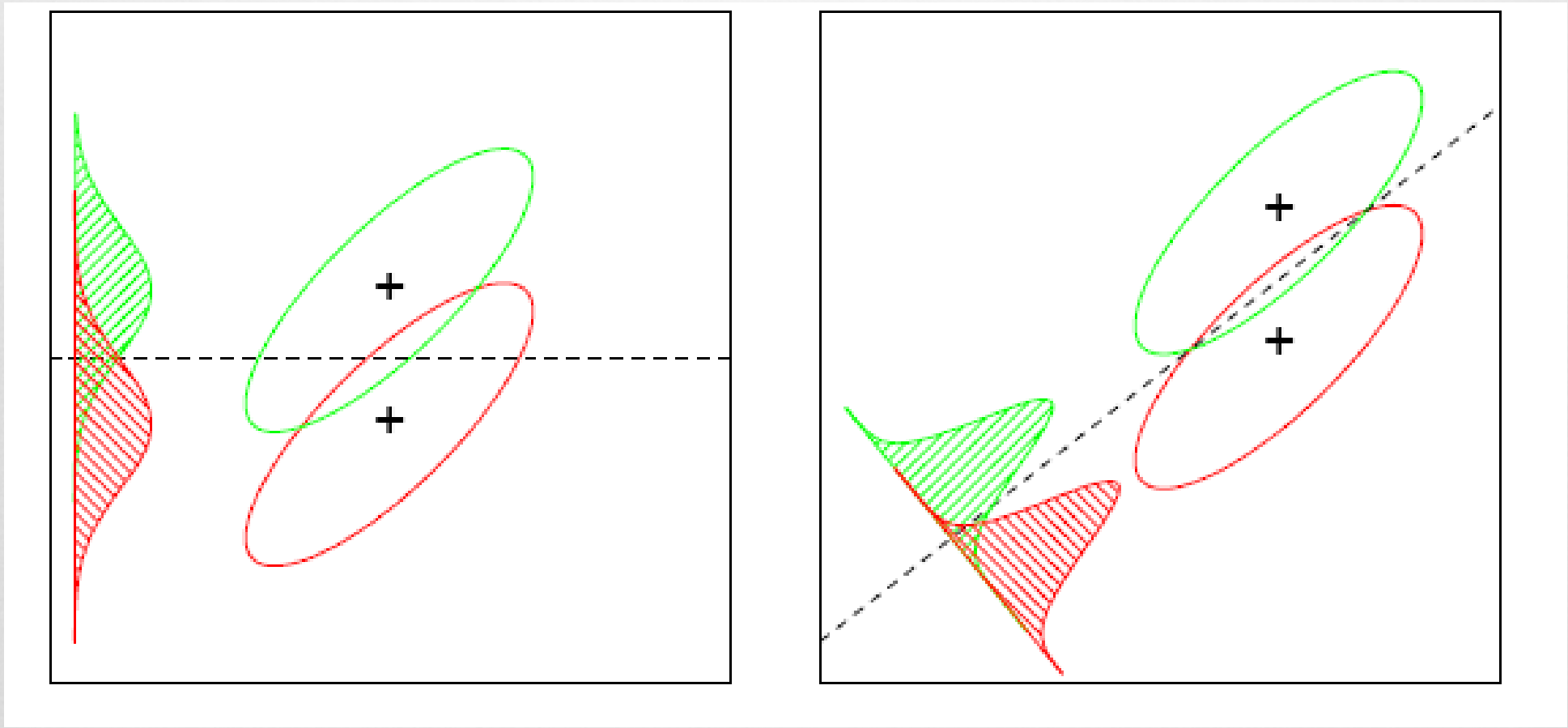
(μ_2, s_2^2) le couple associé à C_2

On note dans $\mathbb{R}$:

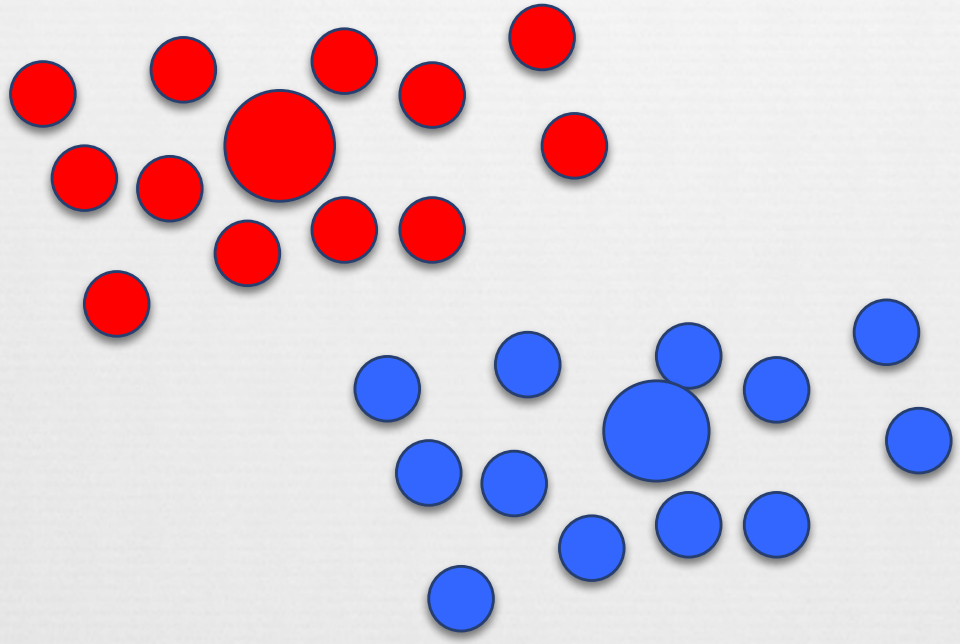
$(\hat{\mu}_1, \hat{s}_1^2)$ le couple associé à la projection de C_1

$(\hat{\mu}_2, \hat{s}_2^2)$ le couple associé à la projection de C_2

Quelle est la meilleure projection?

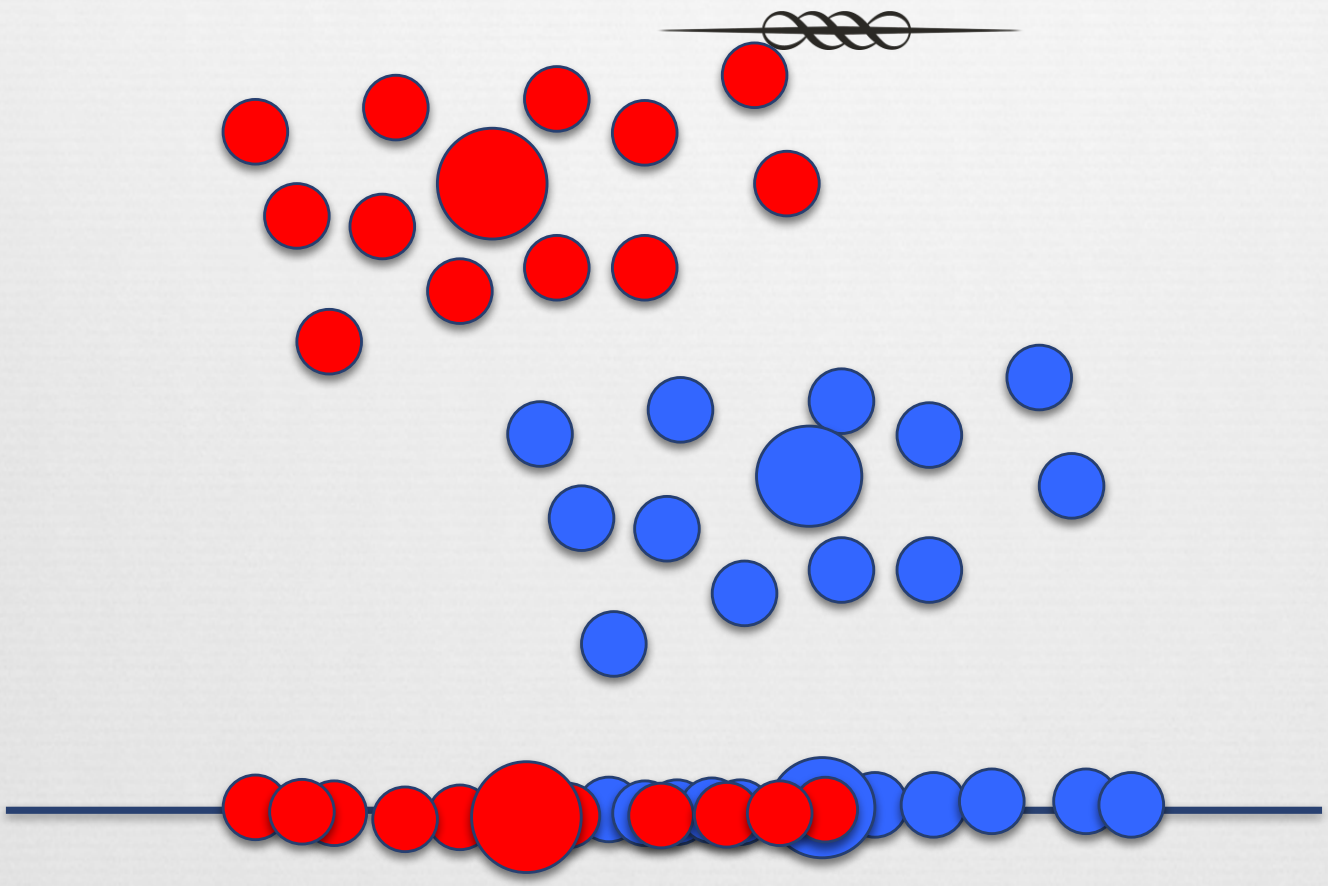


Comment choisir



La distance entre la projection des moyennes $\left|\hat{\mu}_1 - \hat{\mu}_2\right|$ semble un bon critère, plus la distance est grande plus les classes seront éloignées.

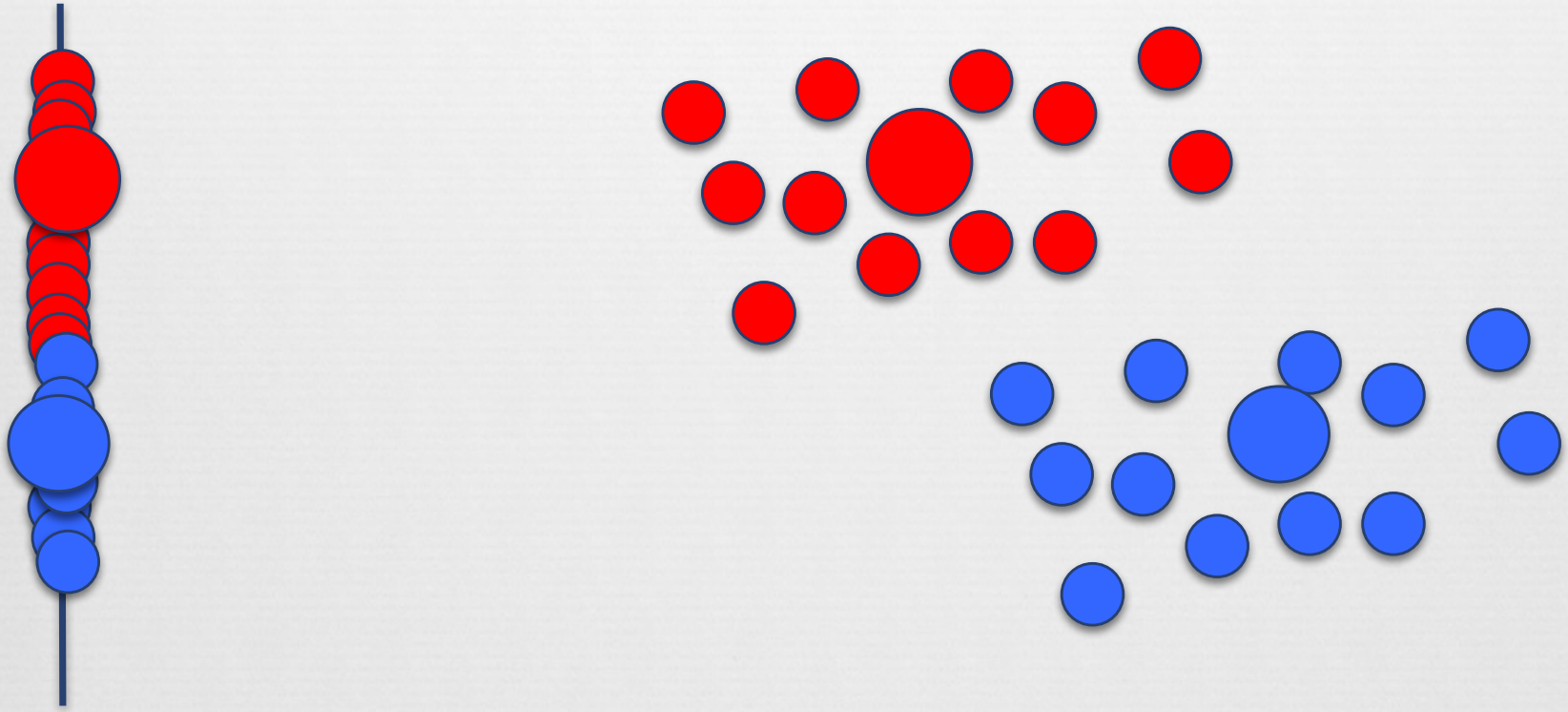
Comment choisir



Il faut aussi tenir compte de la variance des classes projetés.

Plus $\hat{s}_1^2 + \hat{s}_2^2$ est petit, plus la distance entre les moyennes est pertinente.

Comment choisir.



Le critère que nous allons retenir est $J(W) = \frac{(\hat{\mu}_1 - \hat{\mu}_2)^2}{\hat{s}_1^2 + \hat{s}_2^2}$

et nous devons le maximiser J(W)

Fisher: analyse discriminante

$$\mu_k = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|} c_i \text{ et } \hat{\mu}_k = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|} y_i = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|} W^T c_i$$

$$\hat{\mu}_k = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|} W^T c_i = W^T\left(\frac{1}{|C_k|}\sum_{i=1}^{|C_k|} c_i\right) = W^T \mu_k$$

Donc

$$(\hat{\mu}_1 - \hat{\mu}_2)^2 = (W^T\mu_1 - W^T\mu_2))^2 = W^T(\mu_1-\mu_2)(\mu_1-\mu_2)^T W$$

$$= W^T S_B W \quad \text{avec } S_B = (\mu_1-\mu_2)(\mu_1-\mu_2)^T$$

Fisher: analyse discriminante

$$S_k = \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} (c_i - \mu_k)(c_i - \mu_k)^T$$

On note

$S_W = S_1 + S_2$ est appelé la scatter matrice interclasse.

On calcule $\hat{s}_k^2 = \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} (y_i - \hat{\mu}_k)^2$

$$= \frac{1}{|C_k|} \sum_{i=1}^{|C_i|} (W^T c_i - W^T \mu_k)^2$$

$$= \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} (W^T (c_i - \mu_k))^2 = \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} (W^T (c_i - \mu_k))^T (W^T (c_i - \mu_k))$$

$$= \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} W^T (c_i - \mu_k)^T (c_i - \mu_k) W = W^T \frac{1}{|C_k|} \sum_{i=1}^{|C_k|} (c_i - \mu_k)^T (c_i - \mu_k) W$$

$$= W^T S_k W$$

Fisher: analyse discriminante

$$\hat{s}_1^2 + \hat{s}_2^2 = W^T \mathrm{S}_1 W + W^T \mathrm{S}_2 W = W^T \mathrm{S}_W W$$

$$(\hat{\mu}_1 - \hat{\mu}_2)^2 = W^T S_B W \quad \text{avec } S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

On doit maximiser J(W) = $\dfrac{(\hat{\mu}_1 - \hat{\mu}_2)^2}{\hat{s}_1^2 + \hat{s}_2^2} = \dfrac{W^T S_B W}{W^T \mathrm{S}_W W}$

On calcule la dérivée: $\dfrac{\partial(\mathrm{J(W)})}{\partial(W)} = \dfrac{\partial(\dfrac{W^T S_B W}{W^T \mathrm{S}_W W})}{\partial(W)}$

$$= W^T \mathrm{S}_W W \ \frac{\partial(W^T S_B W)}{\partial(W)} - W^T S_B W \frac{\partial(W^T S_W W)}{\partial(W)}$$

$$= W^T \mathrm{S}_W W (2 S_B W) - W^T S_B W (2 S_W W)$$

On résout $\dfrac{\partial(\mathrm{J(W)})}{\partial(W)} = W^T \mathrm{S}_W W (2 S_B W) - W^T S_B W (2 S_W W) = 0$

On divise par $2(W^T \mathrm{S}_W W)$

$$\frac{\partial(\mathrm{J(W)})}{\partial(W)} = \frac{W^T \mathrm{S}_W W (2 S_B W)}{2(W^T \mathrm{S}_W W)} - \frac{W^T S_B W (2 S_W W)}{2(W^T \mathrm{S}_W W)} = S_B W - \mathrm{J(W)} S_W W = 0$$

$$\frac{\partial(\mathrm{J(W)})}{\partial(W)} = S_B W - \mathrm{J(W)} S_W W = 0 \Rightarrow S_B W = \mathrm{J(W)} S_W W$$

Fisher: analyse discriminante

$$\frac{\partial(\mathrm{J(W)})}{\partial(W)} = S_B W - \mathrm{J(W)} S_W W = 0 \Rightarrow S_B W = \mathrm{J(W)} S_W W$$

$S_B W = \mathrm{J(W)} S_W W$ J(W) est un scalaire, on pose J(W) =λ

$$S_B W = S_W \lambda W$$

$$S_W^{-1} S_B W = \lambda W$$

W est un vecteur propre de la matrice $S_W^{-1} S_B$

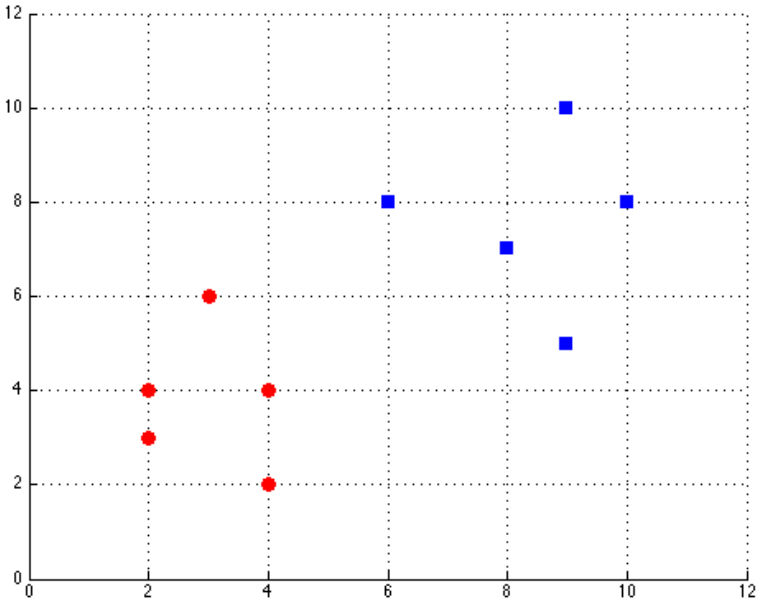
On sait que $S_B c = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T x = \alpha(\mu_1 - \mu_2)$

$$S_W^{-1} S_B W = S_W^{-1}(\alpha(\mu_1 - \mu_2))$$

donc W=$S_W^{-1}(\mu_1 - \mu_2)$ est solution.

Exemple en Matlab

```
c1 = [4,2;2,4;2,3;3,6;4,4]
c2 = [9,10; 6,8; 9,5; 8,7;10,8]

hold on;
grid on;
scatter(c1(:,1),c1(:,2),80,'filled','or');
scatter(c2(:,1),c2(:,2),80,'filled','sb');
axis([0 12 0 12])
```

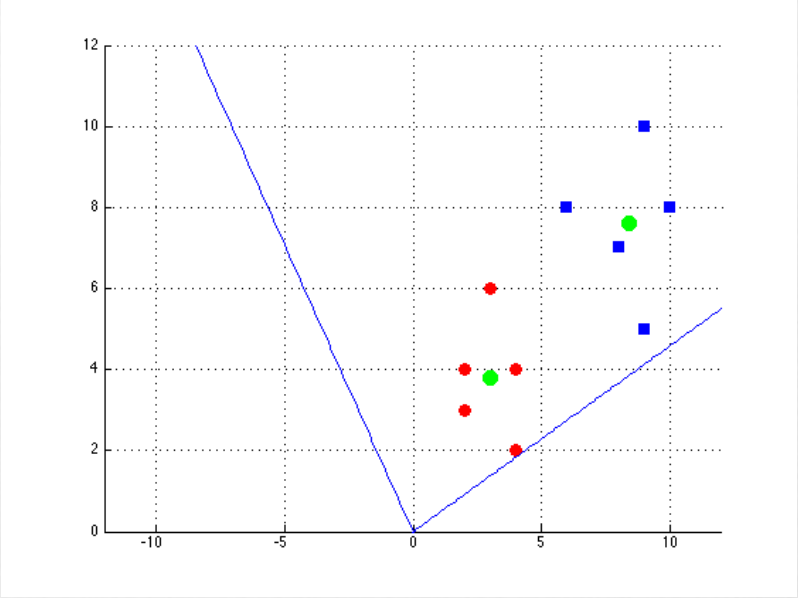



```
mu1 = mean(c1)' %% 3.0000    3.8000
mu2 = mean(c2)' %% 8.4000    7.6000
S1 = cov(c1)
%% 1.0000   -0.2500
%%-0.2500    2.2000
S2 = cov (c2)
%%  2.3000   -0.0500
%% -0.0500    3.3000
Sw = S1 + S2
%% 3.3000   -0.3000
%% -0.3000    5.5000
Sb = (mu1-mu2)*(mu1-mu2)'
%% 29.1600   20.5200
%% 20.5200   14.4400
```

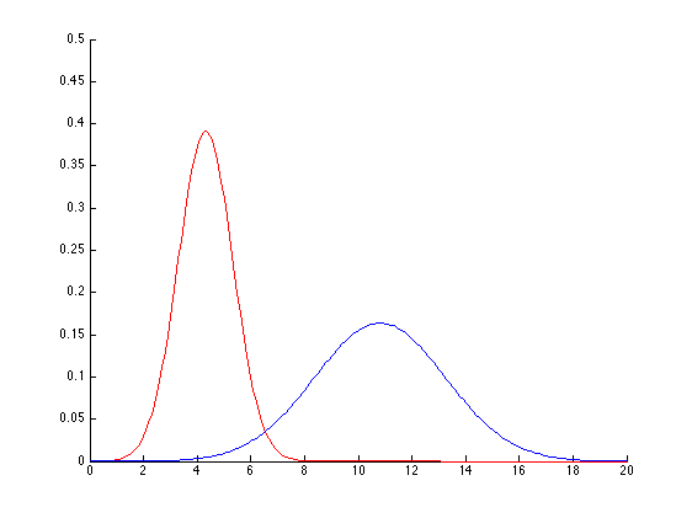
Exemple en Matlab

```
invSw = inv(Sw)
%%0.3045    0.0166
%%0.0166    0.1827
invSwSb = invSw*Sb
%%9.2213    6.4890
%%4.2339    2.9794
```

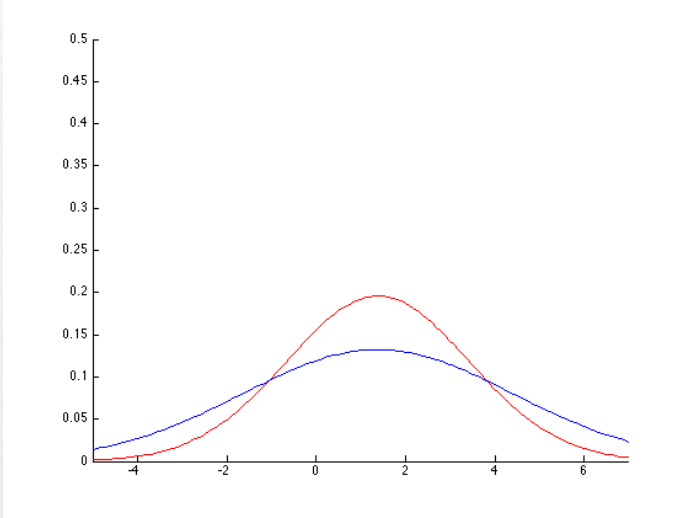
```
[V,d] = eig(invSwSb)
%vecteurs propres
%%0.9088   -0.5755
%%0.4173    0.8178
% valeurs propres
%%12.2007    0
%%0          0.0000
```



Exemple en Matlab




```
figure
hold on;
cproj1 = V(:,1)'*c1';
cproj2 = V(:,1)'*c2';

muproj1 = mean(cproj1)
muproj2 = mean(cproj2)
Sproj1 =cov(cproj1)
Sproj2 = cov(cproj2)
x=0:0.1:20;
plot(x,normpdf(x,muproj1,Sproj1),'r');
plot(x,normpdf(x,muproj2,Sproj2),'b');
axis([0 20 0 0.5])
```




```
figure
hold on;
cproj1 = V(:,2)'*c1';
cproj2 = V(:,2)'*c2';

muproj1 = mean(cproj1)
muproj2 = mean(cproj2)
Sproj1 =cov(cproj1)
Sproj2 = cov(cproj2)
x=-5:0.1:7;
plot(x,normpdf(x,muproj1,Sproj1),'r');
plot(x,normpdf(x,muproj2,Sproj2),'b');
axis([-5 7 0 0.5])
```

Fisher à P classes.

On passe maintenant à P classes $C=\{C_1,.....C_p\}$

Dans le cas de P classes on peut réduire de 1 à P-1 dimensions

On a maintenant P-1 projections $[y_1,....y_{P-1}]$

On pose $W = [w_1 \,||\, w_2 .....|| \, w_{P-1}]$

$y_i = w_i^T c \Rightarrow y = W^T$ avec

on a $c=\begin{bmatrix} c_1 \\ : \\ : \\ c_d \end{bmatrix}$ et $y=\begin{bmatrix} c_1 \\ : \\ : \\ c_{P-1} \end{bmatrix}$

Si on n échantillons de dimension d on peut les mettre dans une matrice

$E_{d\times n}=\begin{pmatrix} c_1^1 & .... & .... & c_1^n \\ .. & .... & .... & .. \\ .. & .... & .... & .. \\ c_d^1 & .... & .... & c_d^n \end{pmatrix}$ et les projetés dans $Y_{P-1\times n} = \begin{pmatrix} y_1^1 & .... & .... & y_1^n \\ .. & .... & .... & .. \\ .. & .... & .... & .. \\ y_{P-1}^1 & .... & .... & y_{P-1}^n \end{pmatrix}$

On a donc

$$Y = W^T E$$

Fisher à P classes.

A deux classes : $S_W = S_1 + S_2$

A P classes $Sw = \sum_{k=1}^{P} S_k$

avec $S_k = \frac{1}{|C_k|} \sum_{c \in C_k} (c - \mu_k)(c - \mu_k)^T$

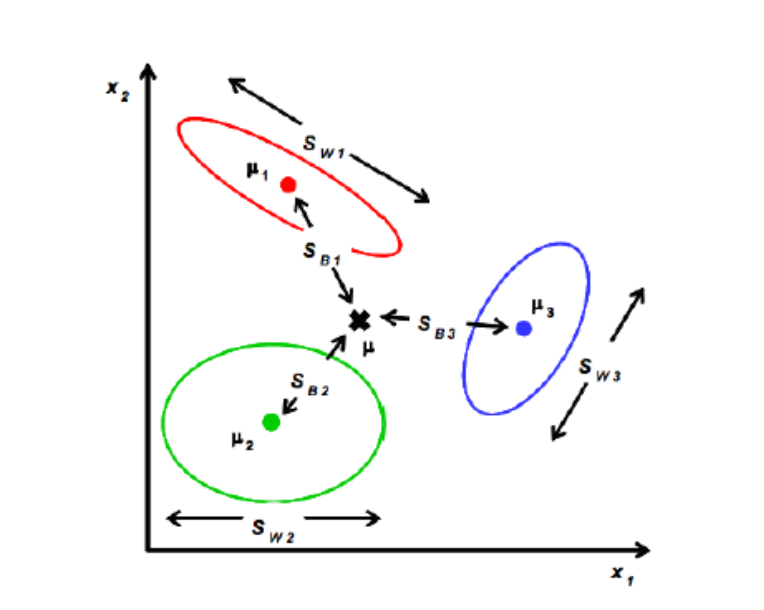
et $\mu_k = \frac{1}{|C_k|} \sum_{c \in C_k} c$

A deux classes : $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

A P classes $S_B = \sum_{k=1}^{P} |C_k|(\mu_k - \mu)(\mu_k - \mu)^T$

avec $\mu = \frac{1}{N} \sum_{c \in C} c = \frac{1}{N} \sum_{k=1}^{P} |C_k| \mu_k$

et $\mu_k = \frac{1}{|C_k|} \sum_{c \in C_k} c$



$$\hat{\mu}_k = \frac{1}{|C_k|} \sum_{c \in C_k} W^T c = W^T \left(\frac{1}{|C_k|} \sum_{c \in C_k} c \right) = W^T \mu_k$$

et $\hat{\mu} = \frac{1}{N} \sum_{c \in C} W^T c = \frac{1}{N} \sum_{k=1}^{P} |C_k| \hat{\mu}_k$

Et donc

$$\hat{S}_W = \sum_{i=1}^{P} \hat{S}_i = \sum_{i=1}^{P} \sum_{y} (y - \hat{\mu}_k)(y - \hat{\mu}_k)^T = \sum_{i=1}^{P} \sum_{c \in C_k} (W^T c - \hat{\mu}_k)(W^T c - \hat{\mu}_k)^T$$

$$\hat{S}_B = \sum_{i=1}^{P} (\hat{\mu}_i - \hat{\mu})(\hat{\mu}_i - \hat{\mu})^T$$

Fisher à P classes.

On a $\hat{S}_W = W^T S_W W$

et $\hat{S}_B = W^T S_B W$

$J(W) = \dfrac{\det(W^T S_B W)}{\det(W^T S_W W)}$ à maximiser.

On retrouve le même problème de valeur propre.

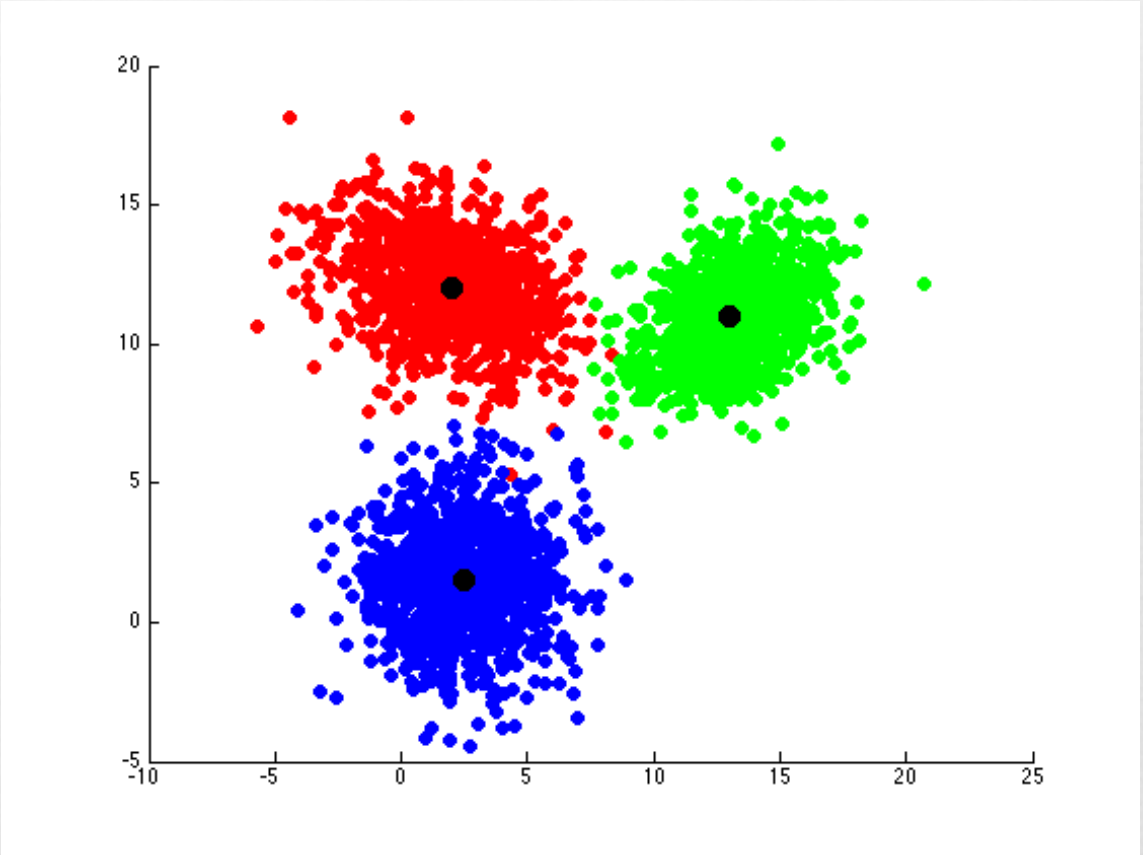
$S_B W = \lambda S_w W$

Comme $S_B = \sum_{k=1}^{P} |C_k| (\mu_k - \mu)(\mu_k - \mu)^T$ est de rang p-1

on a que p-1 valeurs propres disctinces.

Fisher à P classes.

```
Mu = [5;5]
Mu1 = [Mu(1)-3; Mu(2)+7]
covM1 = [5 -1; -1 3]
x1 = mvnrnd(Mu1,covM1,1000);
%%%%%%%%%%%
Mu = [5;5]
Mu2 = [Mu(1)-2.5; Mu(2)-3.5]
covM2 = [4 0; 0 4]
x2 = mvnrnd(Mu2,covM2,1000);
%%%%%%%%%%%
Mu = [5;5]
Mu3 = [Mu(1)+8; Mu(2)+6]
covM3 = [3.5 1; 1 2.5]
x3 = mvnrnd(Mu3,covM3,1000);
%%%%%%%%%%%
figure;
hold on;

scatter(x1(:,1),x1(:,2),'filled','r')
scatter(x2(:,1),x2(:,2),'filled','b')
scatter(x3(:,1),x3(:,2),'filled','g')
scatter(Mu1(1), Mu1(2),100,'filled','k');
scatter(Mu2(1), Mu2(2),100,'filled','k');
scatter(Mu3(1), Mu3(2),100,'filled','k');
```



Fisher à P classes.

```
Mu1 = mean(x1);
Mu2 = mean(x2);
Mu3 = mean(x3);
Mu = (Mu1 + Mu2 + Mu3) ./3;

S1 = cov(x1);
S2 = cov(x2);
S3 = cov(x3);
Sw = S1+S2+3;

N1 = size(x1,1);
N2 = size(x2,1);
N3 = size(x3,1);
SB1 = N1.* (Mu1-Mu)'*(Mu1-Mu);
SB2 = N2.* (Mu2-Mu)'*(Mu2-Mu);
SB3 = N3.* (Mu3-Mu)'*(Mu3-Mu);

SB = SB1 + SB2 + SB3;
invSw = inv(Sw);
invSwSB = invSw*SB;

[V,D] = eig(invSwSB)
```

V =

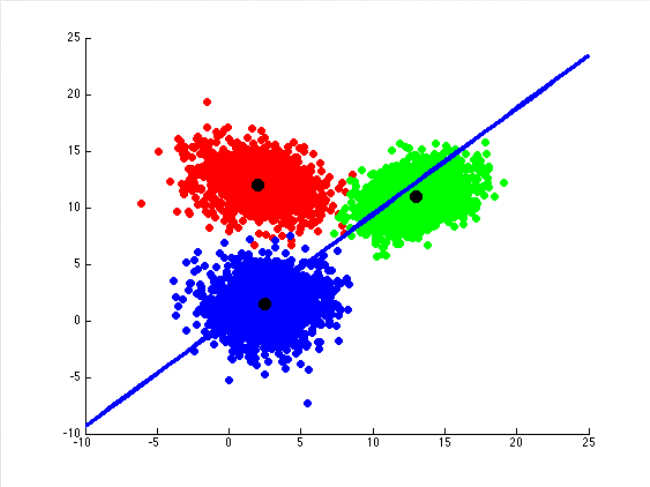
0.6970	-0.6435
0.7171	0.7654

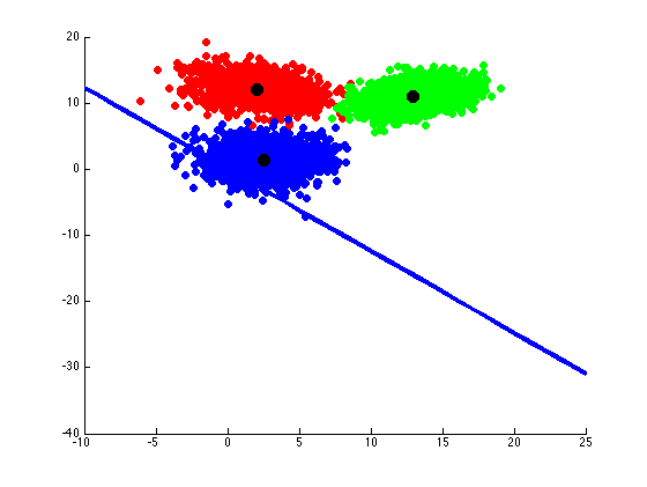
D =

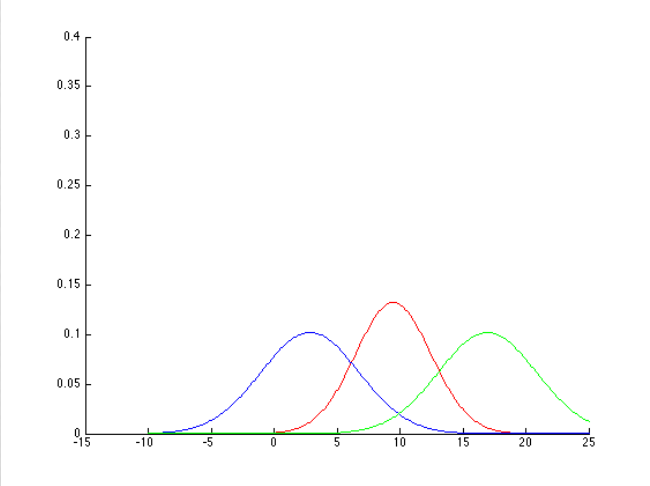
1.0e+04 *

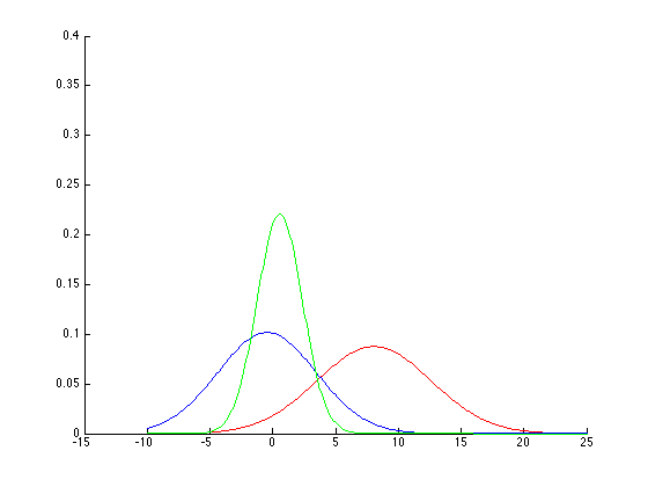
1.3375	0
0	0.6347

Fisher à P classes.









Les classifieurs linéaires

Classifieurs linéaires

On dispose d'une vérité terrain composée de N classes

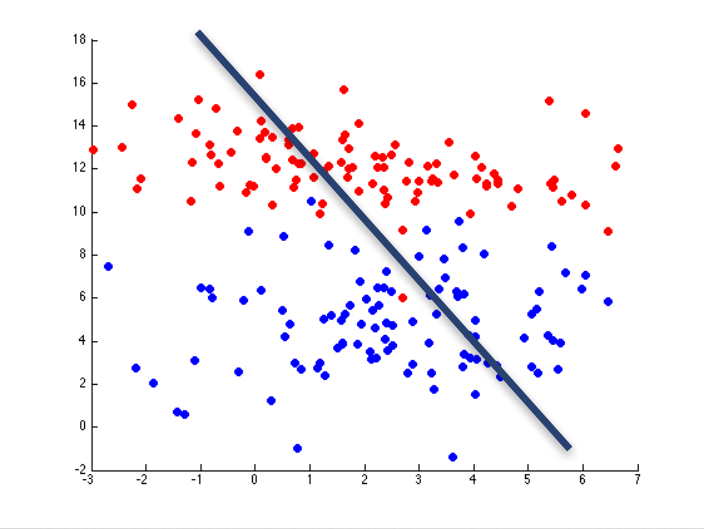
$C = \{C_1 .....C_n\}$ où chaque

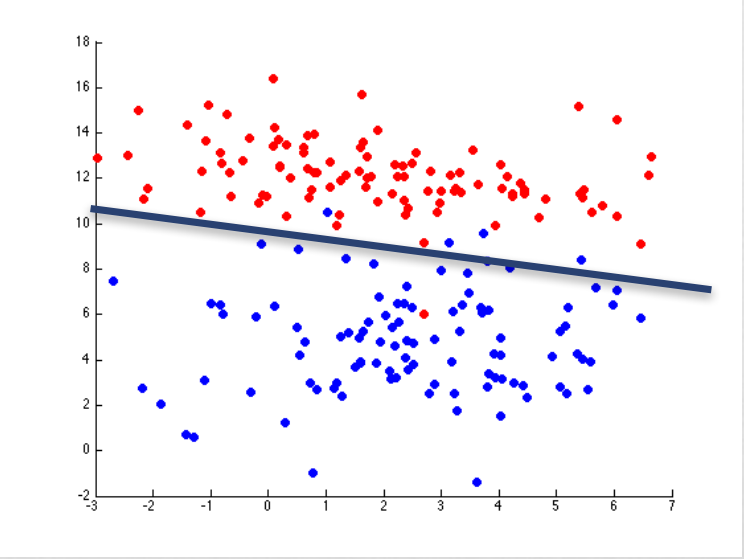
$C_i = \{x^{(1,i)}, ...x^{(k_i,i)}\}$ est un ensemble d'échantillons

On suppose que les différentes classes peuvent être séparées par un hyper-plan P(W).

(si d est la dimension de l'espace, la dimension de l'hyper plan est d-1).

Il faut définir une fonction J(W) qui permet de caractériser la meilleure valeur de W.

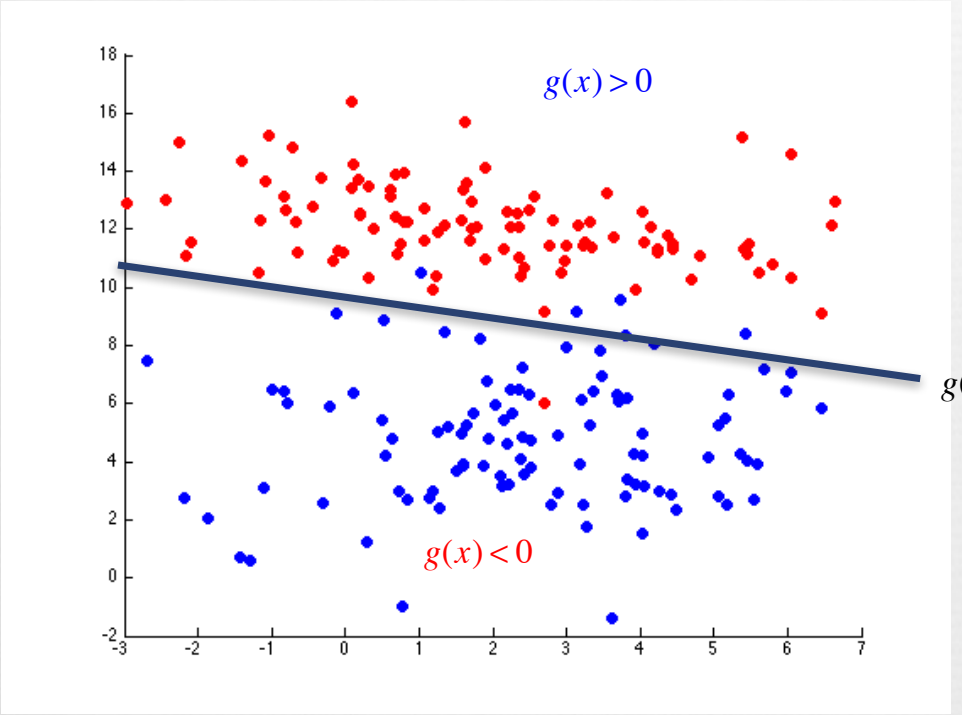




Classifieurs linéaires

- Les fonctions discriminantes peuvent être autres que linéaires (même philosophie)
- Pourquoi des fonctions discriminantes linéaires ?
  - Modèle simple et facilement solvable analytiquement.
  - Optimale pour les distributions gaussiennes avec une même matrice de covariance (cours précédent)
  - Pas optimale pour d'autres distribution mais relativement efficace.
- Ne demande pas d'autres types d'information (modèle de probabilités…)

Classifieur Linéaire à deux classes.



$g(x) = W^T x + W_0$ (l'équation d'une droite).

Prendre une décision en fonction de la valeur de g(x)

$g(x) > 0 \Rightarrow x \in C_1$

$g(x) < 0 \Rightarrow x \in C_2$

$g(x) = 0 \Rightarrow x \in C_1$ ou $x \in C_2$

Classifieur linéaire

Dans $\mathbb{R}^n$, l'hyperplan défini par $\begin{bmatrix} w \\ w_0 \end{bmatrix}$ avec $w = \begin{bmatrix} w_1 \\ : \\ w_n \end{bmatrix} \in \mathbb{R}^n$ est l'ensemble de $x = \begin{bmatrix} x_1 \\ : \\ x_n \end{bmatrix} \in \mathbb{R}^n$,

tel que $g(x) = w^T x + w_0 = w_1 x_1 + w_2 x_2 + ... + w_l x_l + w_0 = 0$

Soit x_1 et x_2 deux points de l'hyper plan de séparation:

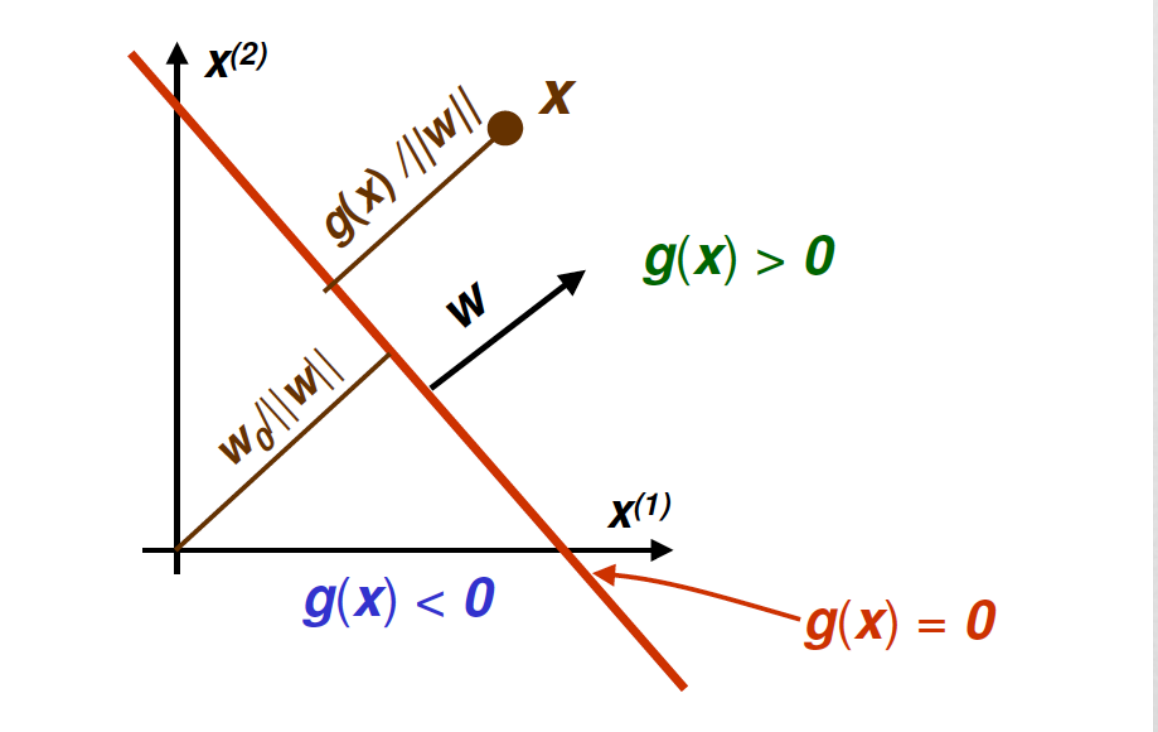
$w^T x_1 + w_0 = w^T x_2 + w_0 = 0 \Rightarrow \quad \forall x_1, x_2 \; w^T (x_1 - x_2) = 0$

Classifieur linéaire

$g(x) = W^T x + W_0$ (l'équation d'une droite).

W définit la direction orthogonale à l'hyper plan (espace vectorie).

W_0 positionne l'hyper plan dans l'espace affine.



Classifieurs linéaires

Supposons que nous avons M classes. Nous avons donc M fonctions discriminantes.

$g_i(x) = W_i^T x + W_0$

Pour un x donné, nous devons prendre une décision en fonction de la valeur des $g_i(x)$.

On choisit le classe C_j telle que

$\forall\ j \neq i\ g_j(x) \geq g_i(x)$

Ce type de classifieur est appelé "Machine linéaire", il divise l'espace en différentes régions associées aux classes.

Classifieurs linéaires

Deux régions contiguës R_i et R_j sont séparées, par une portion d'hyperplan défini par : $g_j(x) = g_i(x)$

$$\Rightarrow W_i^T x + W_0 = W_j^T x + W_0$$

$$\Rightarrow W_i^T x + W_{i,0} = W_j^T x + W_{j,0}$$

$$\Rightarrow (W_i^T - W_j^T)x + (W_{i,0} - W_{j,0}) = 0$$

$(W_i^T - W_j^T)$ est le vecteur normal à l'hyperplan

La distance d'un point x est défini par $\dfrac{\left| g_j(x) - g_i(x) \right|}{\left\| (W_i^T - W_j^T) \right\|}$

Classifieurs linéaires (normalisation)

Dans $\mathbb{R}^n$, l'hyperplan défini par $\begin{bmatrix} w \\ w_0 \end{bmatrix}$ avec $w = \begin{bmatrix} w_1 \\ : \\ w_n \end{bmatrix} \in \mathbb{R}^n$ est l'ensemble de $x = \begin{bmatrix} x_1 \\ : \\ x_n \end{bmatrix} \in \mathbb{R}^n$,

On modifie l'équation $g(x) = w^T x + w_0$ en

$g(x) = [w_0 \; w^T] \begin{bmatrix} 1 \\ x \end{bmatrix}$

et on pose

$y = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ : \\ x_n \\ \\ \end{bmatrix}$ et $a = \begin{bmatrix} w \\ w_0 \end{bmatrix}$ ce qui ramène notre problème à $g(y) = a^T y$

Classifieurs linéaires (normalisation)

On a transformé notre ensemble d'échantillon en $y = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ : \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$,

On considère que nous avons deux classes et $g(y) > 0 \Rightarrow y \in C_1$ et $g(y) < 0 \Rightarrow y \in C_2$

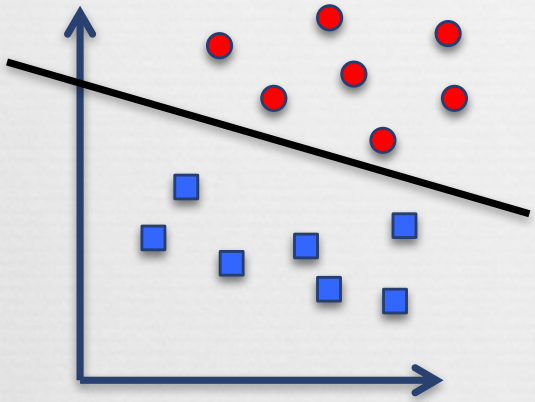
Soit $C_1 = \{y_1^{(1)}, ..., y_l^{(1)}\}$ et $C_2 = \{y_1^{(2)}, ..., y_k^{(2)}\}$, notre classifieur commet aucune erreur si

$\forall y_i^{(1)} \in C_1, g(y_i^{(1)}) > 0 \Leftrightarrow a^T y_i^{(1)} > 0$ et $\forall y_i^{(2)} \in C_2, g(y_i^{(2)}) > 0 \Leftrightarrow a^T y_i^{(2)} < 0$

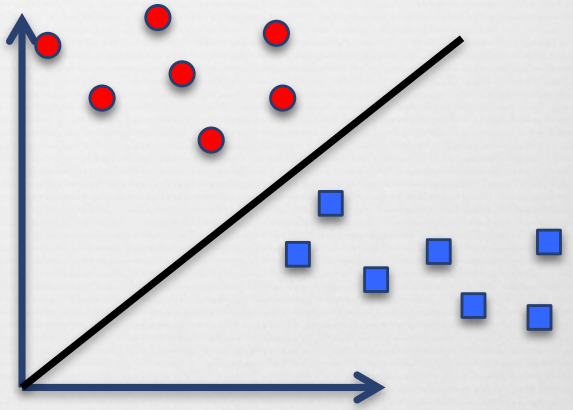
Si on transforme les $y^{(2)}$ de la classe C2 en $-y^{(2)}$ appartenant à la classe $\tilde{C}_2$, on obtient alors

$$\forall y_i \in C_1 \cup \tilde{C}_2, a^T y_i > 0$$

Classifieurs linéaires (normalisation)



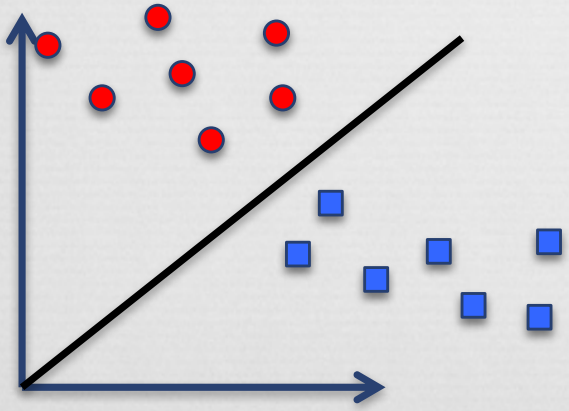
$$y = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ : \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1},$$

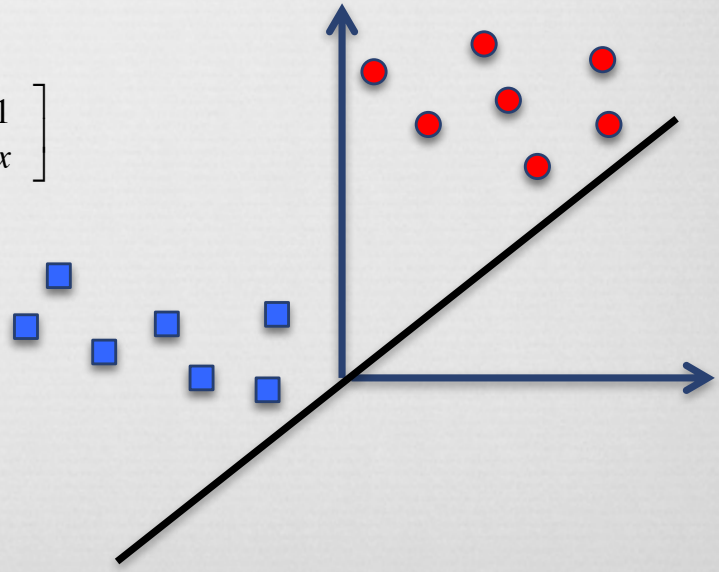


Si $y = \begin{bmatrix} 1 \\ x \end{bmatrix} \in C_2$,

on le transforme en $\tilde{y} = \begin{bmatrix} -1 \\ -x \end{bmatrix}$

les $y \in C_1$ ne changent pas





Définir un critére J(a)

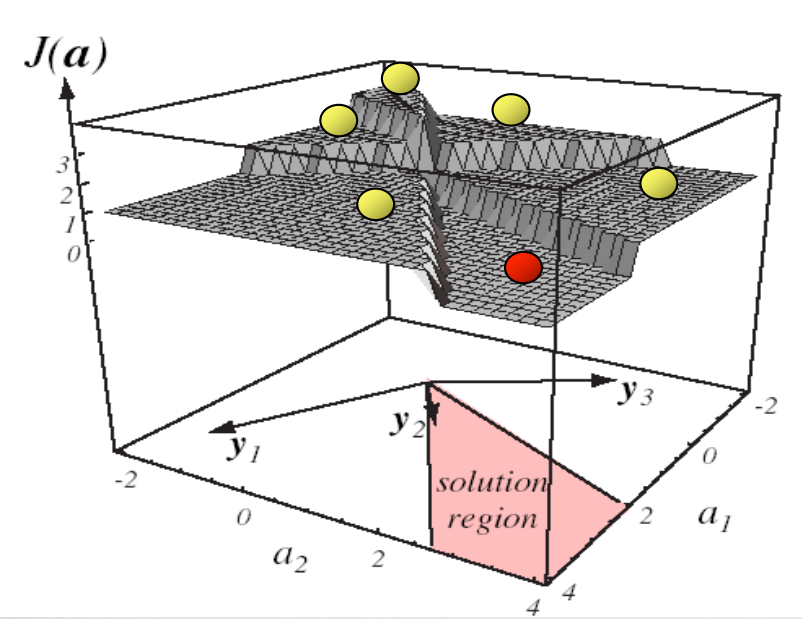
Si le problème est liéairement séparable, il existe au moins un vecteur a

tel que $\forall y \in C_1 \cup \tilde{C}_2 , a^T y > 0$ avec y = $\begin{bmatrix} y_1=1 \\ .. \\ ... \\ y_{n+1}=x_n \end{bmatrix} \in \mathbb{R}^{n+1}$

$a^T y > 0 = \sum_{i=1}^{n+1} a_i y_i$

On notera $Y_M(a) = \{y \in C_1 \cup \tilde{C}_2$ tel que $a^T y < 0\}$, l'ensemble des échantillons mal classés par a.

Définir un critére J(a)



On peut prendre comme critère

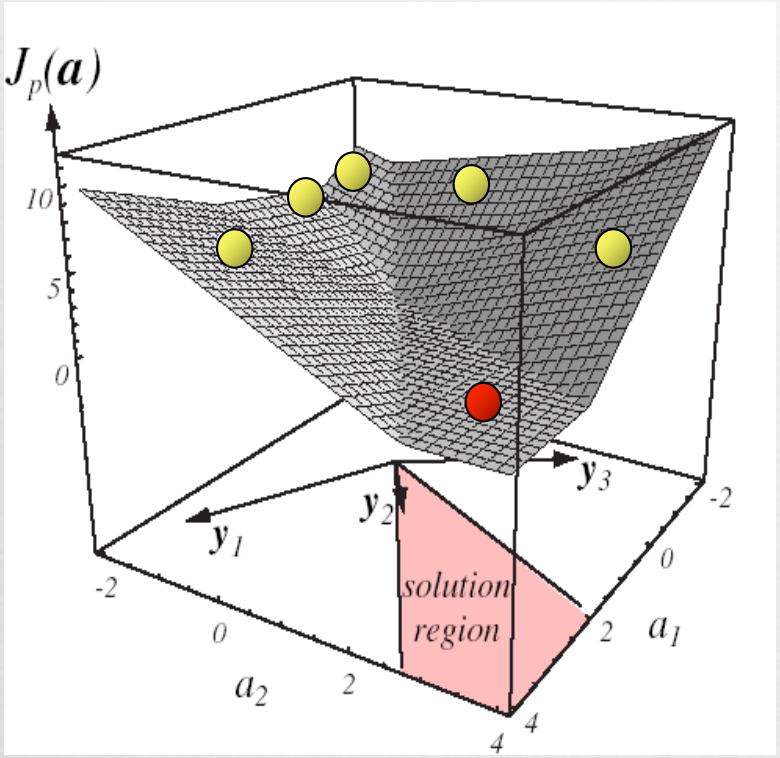
$$J(a) = \left|Y_M(a)\right|$$

c'est à dire le nombre de mal classé

Si J(a) = 0 alors a est un vecteur solution.

J(a) est une fonction constante par morceaux et on ne peut facilement la minimiser.

Définir un critére J(a)



On peut prendre comme critère celui du ***Perceptron***

$$J(a) = \sum_{y \in Y_M} \left(\frac{-a^T y}{||a||}\right)$$

Si y est mal classe $a^T y < 0$ est donc

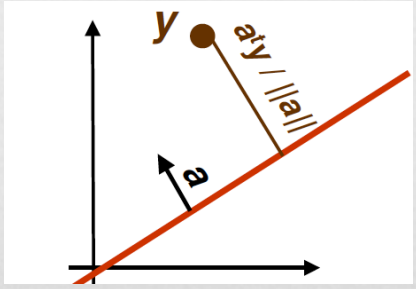
$\frac{-a^T y}{||a||}$ represente la distance de y à l'hyperplan normal à a.

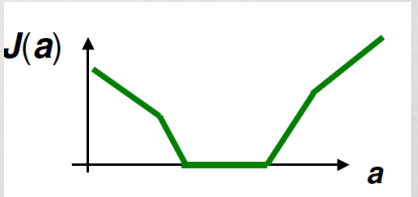
On peut simplifier par $||a||$ et on garde le critère suivant :

$$J(a) = \sum_{y \in Y_M} (-a^T y)$$

J(a) est une fonction linéaire par morceaux et

on peut facilement la minimiser.





Perceptron

Perceptron : $J(a) = \sum_{y \in Y_M} (-a^T y)$

On calcule le gradient $\nabla J(a) = \dfrac{\partial \left(\sum_{y \in Y_M} (-a^T y) \right)}{\partial a}$

$\nabla J(a) = \sum_{y \in Y_M} (-y)$

On ne peut pas résoudre analytiquement $\nabla J(a) = \sum_{y \in Y_M} (-y) = 0$

car on ne peut exprimer analytiquement Y_M.

On chercher a par descente de gradient :

$a^{k+1} = a^k - \eta^k \nabla J(a^k) = a^k + \eta^k \sum_{y \in Y_M} (y)$

Perceptron

Perceptron : $\mathrm{J(a)} = \sum_{y \in Y_M} (-a^T y)$

On chercher a par descente de gradient :

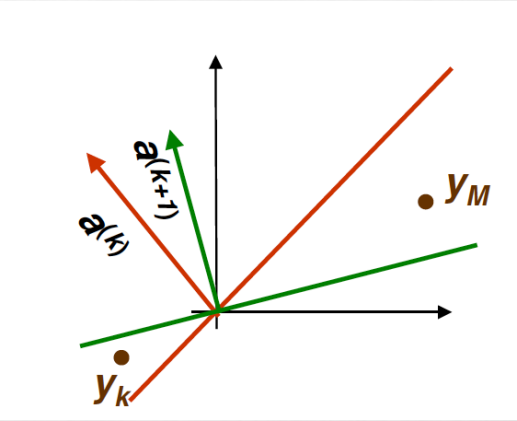
$$a^{k+1} = a^k - \eta^k \nabla \mathrm{J}(a^k) = a^k + \eta^k \sum_{y \in Y_M} (y)$$

On peut faire la descente élément par élément.

Pour un $\mathrm{y}_i \in Y_M(a^k)$

$$a^{k+1} = a^k + \mathrm{y}_i$$

recalculer $Y_M(a^{k+1})$



Il faut régler correctement la valeur de η^k