

# Systemes Intelligents : Raisonnement et Reconnaissance

James L. Crowley

Deuxième Année ENSIMAG

Deuxième Semestre 2007/2008

Seance 10

14 mai 2008

## Discrimination Linéaire

Notations .....	2
Fonctions de Discrimination (rappel).....	3
La Classification Linéaire .....	5
Cas simple : si le bruit d'observation domine.....	5
Si la variation interclasse domine.....	5
Détection par classification linéaire.....	6
Vecteur entre les Centres de Gravité.....	9
La discriminante linéaire de Fisher.....	10
Estimation par Moindres de Carrées.....	14
A Committee of Boosted Classifiers.....	15
Résumé de l'algorithme de Boosting.....	15
Détection par une Comité de Classifiers :.....	16
Perceptrons .....	17
Méthodes à Noyaux (Kernel Methods).....	19

Sources Bibliographique :

N. Cristianini, J. Shawe-Taylor, "Support Vector Machine and other Kernel based learning methods", Cambridge University Press, 2000.

**Notations**

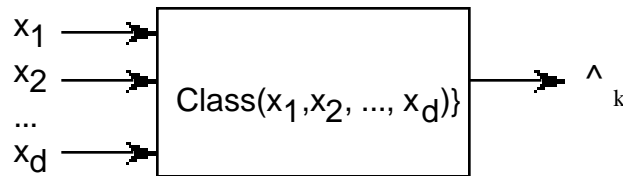
$x$	Une variable
$X$	Une valeur aléatoire (non-prévisible).
$N$	Le nombre de valeurs possibles pour $x$ ou $X$
$x$	Un vecteur de $D$ variables
$X$	Un vecteur aléatoire (non-prévisible).
$D$	Nombre de dimensions de $x$ ou $X$
$T_k$	La classe $k$
$k$	Indice d'une classe
$K$	Nombre de classes
$M$	Nombre totale d'exemples de toutes les classes
$E_m$	L'événement $m$ .
$X_m$	Le vecteur de caractéristiques pour $E_m$
$Y_m$	Une variable d'Indication pour $E_m$

$E_m : (X_m, Y_m)$  L'événement  $E_m$  est décrit par son vecteur de caractéristiques et son classes  $Y_m$

## Fonctions de Discrimination (rappel)

Soit un événement (ou observation)  $E$  décrit par un vecteur de caractéristiques  $X$ .  
 Soit  $K$  classes d'événements  $\{k\} = \{1, 2, \dots, K\}$  avec une classe  $k \in \{k\}$

La classification est un processus d'estimation de l'appartenance d'un événement à une des classes  $k$  fondées sur les caractéristiques de l'événement,  $X$ .



$\hat{k}$  est la proposition que  $(E = k)$ :  $\hat{k} = d\{g(X)\}$ .

La fonction de classification est composée de deux parties  $d()$  et  $g_k()$ :

- $g(X)$  : Une fonction de discrimination :  $\mathbb{R}^D \rightarrow \mathbb{R}^K$
- $d()$  : Une fonction de décision :  $\mathbb{R}^K \rightarrow \{K\}$

Dans les séances précédentes, nous avons développé

La règle de décision :  $\hat{k} = \arg\text{-max}_k \{ p(k | X) \}$

en utilisant la règle de Bayes :  $p(k | X) = \frac{p(X | k)p(k)}{p(X)}$

La règle de décision est :  $\hat{k} = \arg\text{-max}_k \{ p(X | k) p(k) \} = \arg\text{-max}_k \{ g_k(X) \}$

en utilisant la loi Normale :

$$p(X | k) = \mathcal{N}(X; \mu_k, C_k) = \frac{1}{(2\pi)^{\frac{D}{2}} \det(C_k)^{\frac{1}{2}}} e^{-\frac{1}{2}(X - \mu_k)^T C_k^{-1} (X - \mu_k)}$$

on obtient :

$$g_k(x) = -\frac{1}{2} \text{Log}\{\det(C_k)\} - \frac{1}{2}(X - \mu_k)^T C_k^{-1} (X - \mu_k) + \text{Log}\{p(\mu_k)\}$$

Forme canonique de la fonction de discrimination :

$$g_k(X) = X^T D_k X + W_k^T X + B_k.$$

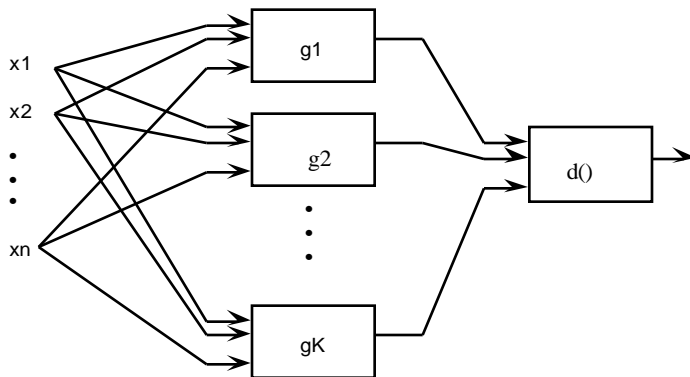
avec

$$D_k = \frac{1}{2} C_k^{-1}$$

$$W_k = C_k^{-1} \mu_k$$

$$B_k = -\frac{1}{2}(\mu_k^T C_k^{-1} \mu_k) - \frac{1}{2} \text{Log}\{\det(C_k)\} + \text{Log}\{p(\mu_k)\}$$

Dans cette forme, le classificateur est une machine qui calcule K fonctions  $g_k(x)$  suivie d'une sélection de la décision.



## La Classification Linéaire

Nous allons regarder des méthodes linéaires pour concevoir les  $g_k(X)$ .

Rappeler que chaque observation est corrompue par deux sources aléatoires de variation :

- 1) Le bruit d'observation :  $B_o$
- 2) Les variations des individus à l'intérieur de la classe :  $B_i$

$$X = x + B_i + B_o$$

Le bruit d'observation :  $B_o$ , et typiquement Normale.

### Cas simple : si le bruit d'observation domine.

En Général le Bruit d'observation est Normale.

Si  $B_o \gg B_i$  alors pour toutes les classes :  $j, i \in C_i \cap C_j$  et  $D_i \cap D_j$

Dans ce cas, Les fonction de discrimination devient linéaire

$$g_k(X) = W_k^T X + B_k.$$

Ce cas est fréquent dans la communication, et en analyse de signale.

### Si la variation interclasse domine

Dans beaucoup de cas réel, la variation  $B_i$  domine  $B_o$  et  $B_i$  n'est pas normal.

$$B_i \gg B_o. \text{ et } B_i \sim \mathcal{N}(X; \mu, C)$$

Que faire ?

Nous allons regarder plusieurs techniques pour accommoder les variations  $B_i$  en utilisant les fonctions de discrimination linéaire.

- 1) Boosting : Classification par comité de classifieurs linéaires

- 2) Une Cascade de comités (AdaBoost)
- 3) Les méthodes à noyau.

**Détection par classification linéaire**

La Détection : Discrimination Binaire :  $K=2$  : (Classe 1 et les "autres")  
 exemple détection de visage dans les images.

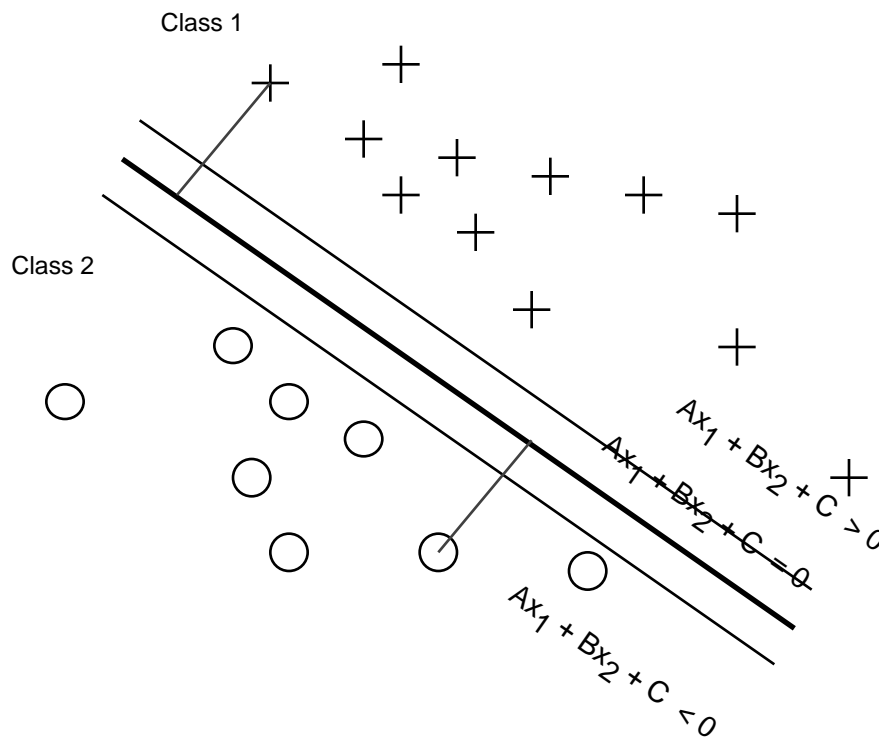
Nous allons considérer le cas d'une "détection" pour les événements d'une classe.

Il y a que deux classes d'événements :  $k=1$  (la classe à détecter) et  $k=2$  (les autres).  
 Dans nos exemples, la classe est indiquée par une variable d'indication :  $Y_m$ .

- $Y_m = +1$  pour les événements de  $T_1$ , et
- $Y_m = -1$  pour les événements de  $T_2$

Soit  $M$  exemples  $\{X_m, Y_m\}$

Notre objective est d'estimer un plan (ou hyperplan si  $D > 2$ ) qui sépare les deux classes.  
 dans ce cas , la fonction de décision devient  $d() = \text{sgn}()$ .



Un (hyper)plan est un ensemble de point tel quel

$$w_1x_1 + w_2x_2 + \dots + w_Dx_D + B = 0$$

En forme de vecteur :  $W \cdot X + B = 0$

ou  $W = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_D \end{pmatrix}$  est la norme du plan.

Si  $\|W\| = 1$ , alors pour tous les points hors du plan,

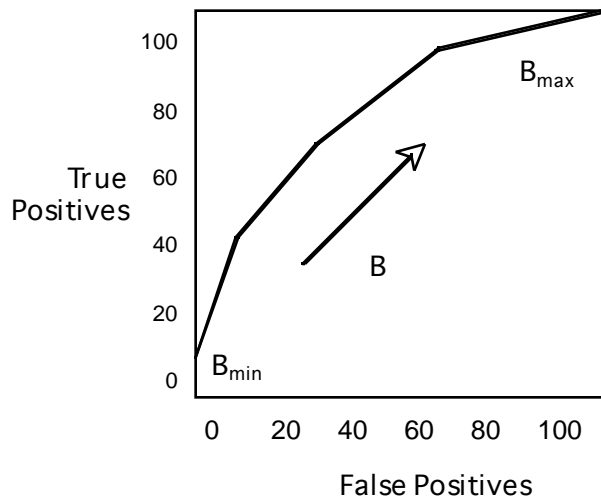
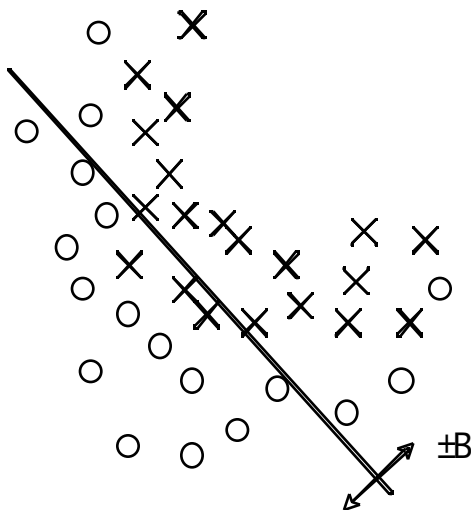
$B = -W \cdot X$  est la distance perpendiculaire à l'origine.

si  $\|W\| \neq 1$  alors utilise  $W' = \frac{W}{\|W\|}$ , et  $B' = \frac{B}{\|W\|}$

Dans ce cas, on peut dire que  $Y = W \cdot X + B$

projete les caractéristiques d'un événement sur une norme,  $W$ .

$B$  est un "biais". La valeur  $B$  (le biais) est une variable libre qui permet de contrôler le rapport entre les détections manquées et les fausses détections.



Ceci s'appelle une courbe "ROC". (Receiver Operating Characteristics)

Un détecteur est caractérisé par son ROC (et non pas par la réponse à une valeur  $B$ ).

Comment calculer le plan. ?

Plusieurs techniques :

- 1) Vecteur entre les centre de Gravités.
- 2) LDA (Méthode de Fisher).
- 3) Régression
- 4) Perceptrons



**Vecteur entre les Centres de Gravité.**

Soit  $g_1(X) = W_1^T X + B_1$ , et  $g_2(X) = W_2^T X + B_2$ .

ou :

$$W_k = C_k^{-1} \mu_k$$

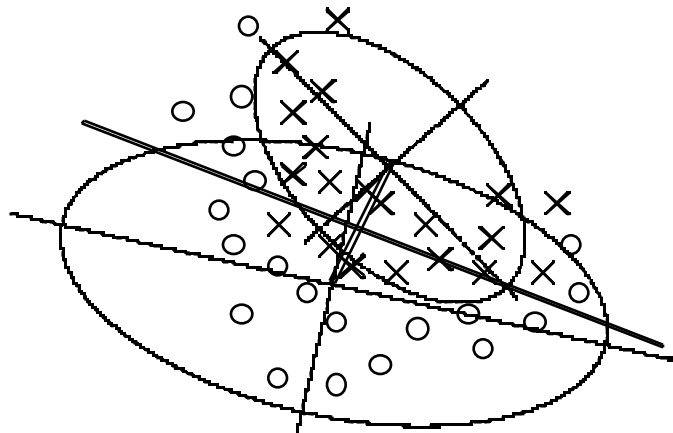
$$B_k = -\frac{1}{2}(\mu_k^T C_k^{-1} \mu_k) - \frac{1}{2} \text{Log}\{\det(C_k)\} + \text{Log}\{p(\mu_k)\}$$

La frontière entre les deux est :

$$g_1(X) - g_2(X) = 0$$

$$(W_1^T - W_2^T)X + B_1 - B_2 = 0$$

$$(C_1^{-1} \mu_1 - C_2^{-1} \mu_2) + B_1 - B_2 = 0$$



La direction est déterminée par le vecteur entre centre de gravité, pondéré par les  $C^{-1}$ . Ceci n'est pas optimum.

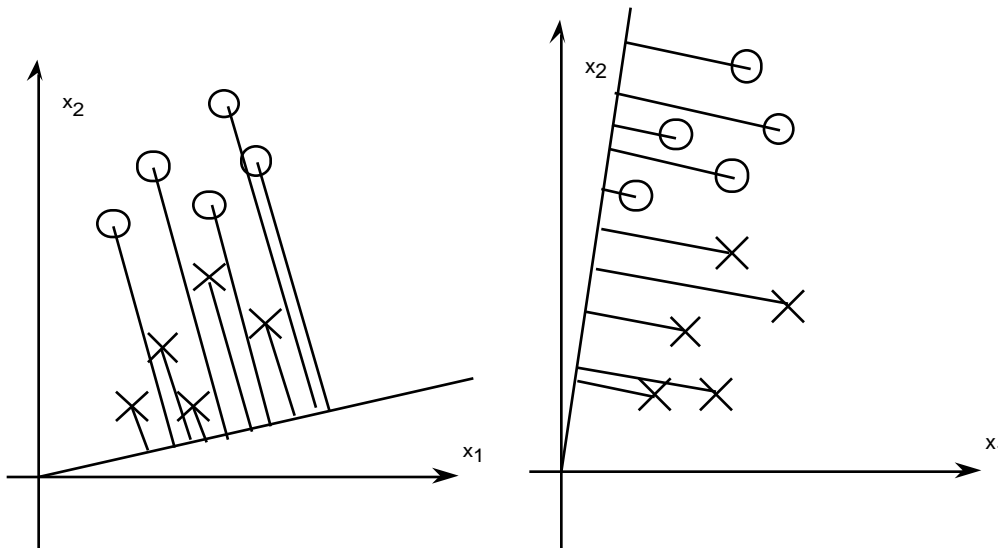
## La discriminante linéaire de Fisher

Le principe de Fisher est de projeter le vecteur de caractéristique,  $X$  de  $D_x$  dimensions vers un espace  $Z$  de  $D_z$  par une transformation linéaire  $F$  choisit tel quel

- 1)  $D_z \ll D_x$  et
- 2) Les exemples des classes  $A_k$  sont séparés.

$$z = F^T x$$

En général, s'il y a  $K$  classes, nous allons chercher  $D_z = K-1$



La discriminabilité des classes dépend de la direction de  $F$

Pour déterminer la meilleure projection, on appuie sur une mesure de la séparation entre classes.

Soit  $K=2$  classes  $T_1$  et  $T_2$  représenté par les exemples  $X_{1m}$  et  $X_{2m}$

Dans ce cas,  $D_z = 2-1 = 1$ .  $Z$  est un scalaire

Pour chaque exemple :

$$Z_{km} = F^T X_{km}$$

Nota que  $F$  est une projection telle que  $\|F\| = 1$

Les moyennes des exemples pour chaque classe sont

$$\mu_k = E\{X_{km}\} = \frac{1}{M_k} \sum_{m=1}^{M_k} X_{km}$$

Les moments sont les invariants affines. Donc, la moyenne (1ere moment) d'une projection est la projection de la moyenne.

$$\tilde{\mu}_k = E\{Z_{km}\} = \frac{1}{M_k} \sum_{m=1}^{M_k} Z_{km} = \frac{1}{M_k} \sum_{m=1}^{M_k} F^T X_{km} = F^T \mu_k$$

La distance entre les classes est  $d_{12} = \|\tilde{\mu}_1 - \tilde{\mu}_2\| = \|F^T(\mu_1 - \mu_2)\|$

On veut rendre la distance entre classes aussi grandes que possible, sans disperser les classes.

La dispersion ("Scatter") pour une ensemble  $\{X_{km}\}$  d'exemples et pour une classe  $k$  est une matrice

$$S_k = M_k C_k = \sum_{m=1}^{M_k} (X_{km} - \mu_k)(X_{km} - \mu_k)^T$$

La transformation  $F$  projet le vecteur  $X$  sur la scalaire  $Z$ . La dispersion ("Scatter") pour la projection des exemples de la classe  $k$  est

$$\tilde{S}_k = \sum_{m=1}^{M_k} (Z_{km} - \tilde{\mu}_k)^2$$

Le critère de Fisher est de maximiser le ratio de la séparation des deux classes par rapport à leurs dispersions.

$$J(F) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{S}_1 + \tilde{S}_2} = \frac{\|F^T(\mu_1 - \mu_2)\|^2}{\tilde{S}_1 + \tilde{S}_2}$$

Fisher cherche la transformation  $F^T$  tel quel

$$F = \arg\text{-max}_F \left\{ \frac{\|F^T(\mu_1 - \mu_2)\|^2}{\tilde{S}_1 + \tilde{S}_2} \right\}$$

Soit  $M = \sum_{k=1}^K M_k$  exemples,  $X_{km}$ .

Pour  $K=2$ ,  $M = M_1 + M_2$

La moyenne de chaque classe est

$$\mu_k = \frac{1}{M_k} \sum_{m=1}^{M_k} X_{km}$$

La moyenne de TOUS les exemples est

$$\mu = \frac{1}{M} \sum_{k=1}^K M_k \mu_k = \frac{1}{M} (M_1 \mu_1 + M_2 \mu_2)$$

La matrice de dispersion inter-classes  $S_B$  (B voudrait dire "between") est la dispersion des moyennes des classes.

$$S_B = (\mu_1 - \mu)(\mu_1 - \mu)^T + (\mu_2 - \mu)(\mu_2 - \mu)^T$$

La dispersion intra-classe  $S_w$  (En Anglais W pour "within") est la covariance moyenne.

$$S_w = \sum_{k=1}^K S_k = \sum_{k=1}^K M_k C_k = S_1 + S_2$$

La meilleure transformation  $F$  est celle que

$$F = \arg\text{max}_F \left\{ \frac{\|F^T S_B F\|}{\|F^T S_w F\|} \right\}$$

Dans notre exemple avec  $K=2$ ,  $D_z = 1$  (donc  $F^T S_B$  est une scalaire)

On définit :

$$J(F) = \frac{F^T S_B F}{F^T S_w F}$$

en physique, ceci est connu comme le quotient de Rayleigh. Il est possible de montrer que

$$S_B F = S_w F.$$

donc

$$S_w^{-1} S_B F = F.$$

Le facteur d'échelle n'est pas important est-on peut déterminer directement

$$\text{Donc } F = S_w^{-1} S_B = S_w^{-1}(\mu_1 - \mu_2)$$

Ceci est la discriminant linéaire de Fisher pour deux classes.

Il maximise la dispersion entre les classes.

On rappelle que la surface de décision linéaire entre deux classes a la forme :

$$F^T X + b_0 = 0 \quad \text{où } F = C^{-1}(\mu_1 - \mu_2)$$

et  $b_0$  est un constant

**Estimation par Moindres de Carrés**

Soit  $M$  exemples  $\{y_m, X_m\}$  tel que  $y=+1$  pour les événements de  $T_1$ ,  
et  $y = -1$  pour les événements de  $T_2$

On cherche  $y = g(X) = W X$

qui minimise la fonction de "Loss" :  $L(\hat{W}) = \sum_{m=1}^M (y_m - \hat{X}_m \hat{W})^2$

Pour l'ensemble des  $M$  exemples  $\{y_m, X_m\}$ . on compose la matrice  $X$  et un vecteur  $Y$

$$X = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_M) \text{ (taille } D \text{ lignes, et } M \text{ colonnes)}$$

$$Y = (y_1, y_2, \dots, y_M) \text{ (taille } M \text{ lignes).}$$

$$\text{on a } L(\hat{W}) = (Y - X \hat{W})^T (Y - X \hat{W})$$

Le Minimum est trouvé quand :

$$\frac{\partial L(\hat{W})}{\partial \hat{W}} = -2 X^T Y + 2 X^T X \hat{W} = 0$$

$$\text{Donc : } X^T Y = X^T X \hat{W} \text{ et donc } \hat{W} = (X^T X)^{-1} X^T Y$$

$$\text{et } E_1 \text{ si } \hat{Y} = W X > 0$$

$$\text{on peut ajouter un biais : } E_1 \text{ si } W X + B > 0$$

## A Committee of Boosted Classifiers

Une des idées les plus innovantes en apprentissage des dernières années est le "boosting". Le principe est de composer un comité de classificateurs linéaires faibles.

Leur combinaison par vote donne un classificateur fort.

Avec l'algorithme ADA Boost, on peut déterminer une ensemble de classifieurs linéaire faible pour lequel le taux d'erreur est arbitrairement limité.

L'idée est d'appliquer un poids aux exemples, et de renforcer le poids des exemples mal classés, et ré-estimer une nouvelle classifieur.

### Résumé de l'algorithme de Boosting

Soit une ensemble des exemples  $\{Y_m, X_m\}$

1) Initialiser un vecteur de M coefficients  $a_m = 1$ . Initialiser  $S = M$ .

( $a_m$  serait le poids de chaque exemple,  $\{Y_m, X_m\}$ .  $S$  est la somme des poids.

2) Pour l'ensemble des M exemples  $\{Y_m, X_m\}$ . on compose le matrice  $\mathbf{X}$  et une vecteur  $\mathbf{Y}$  pour calculer le Nième fonction de discrimination.

$$\mathbf{X} = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_M) \text{ (taille D lignes, et M colonnes)}$$

$$\mathbf{Y} = (Y_1, Y_2, \dots, Y_M) \text{ (taille M lignes).}$$

On trouve une classification, par exemple avec :

$$\hat{\mathbf{W}} = (\mathbf{X} \mathbf{X})^{-1} \mathbf{X} \mathbf{Y}$$

$$\hat{\mathbf{Y}} = \hat{\mathbf{W}}^T \mathbf{X}$$

3) Pour chaque exemple en  $\{X_m\}$ , si  $\hat{\mathbf{W}}^T X_m - Y_m > 0$  alors error :

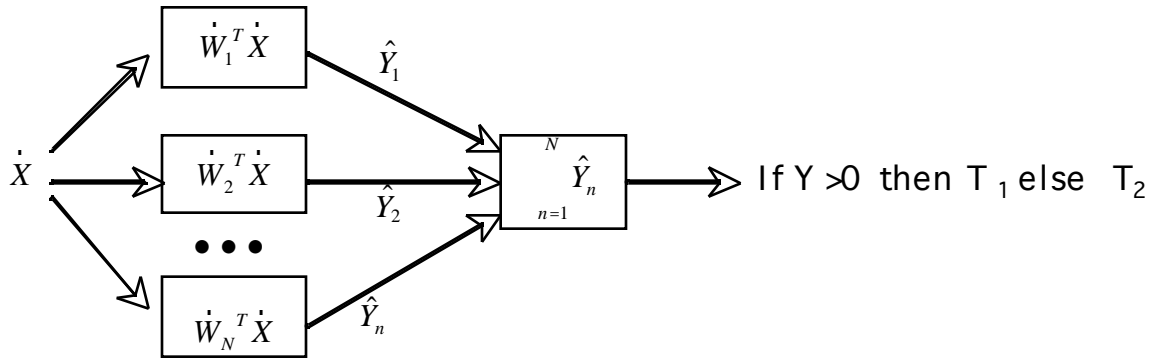
$$a_m = a_m + 1, S = S + 1$$

4) Repeter étape 2 pour classifier N+1.

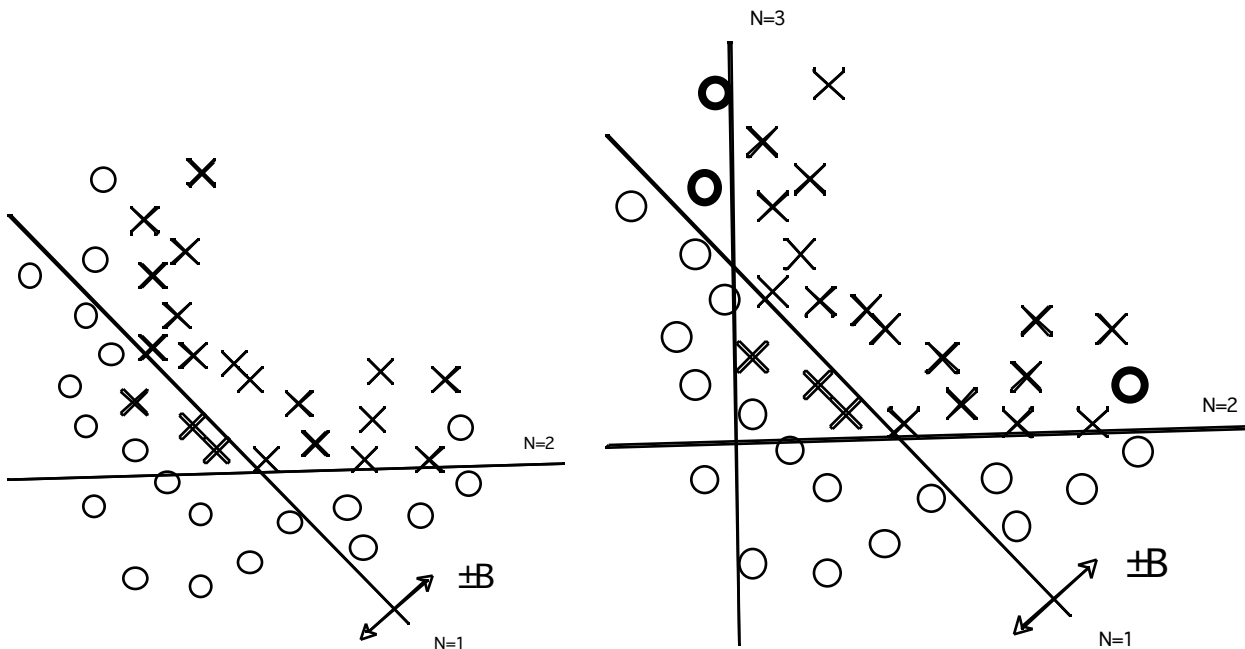
**Détection par une Comité de Classifiers :**

Avec un comité de classifieurs, la décision sont faites par vote

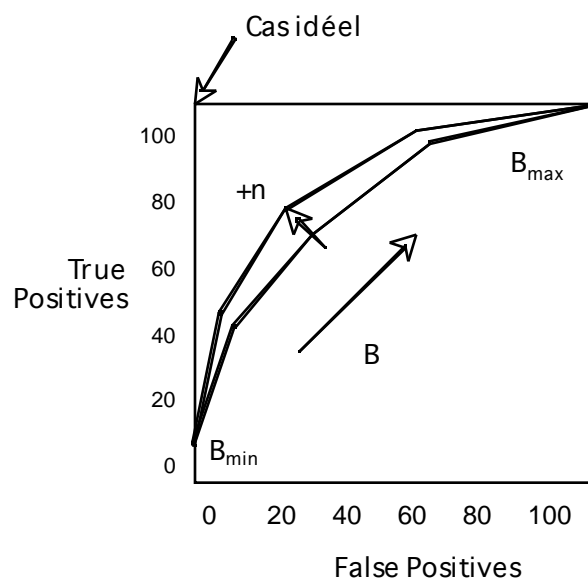
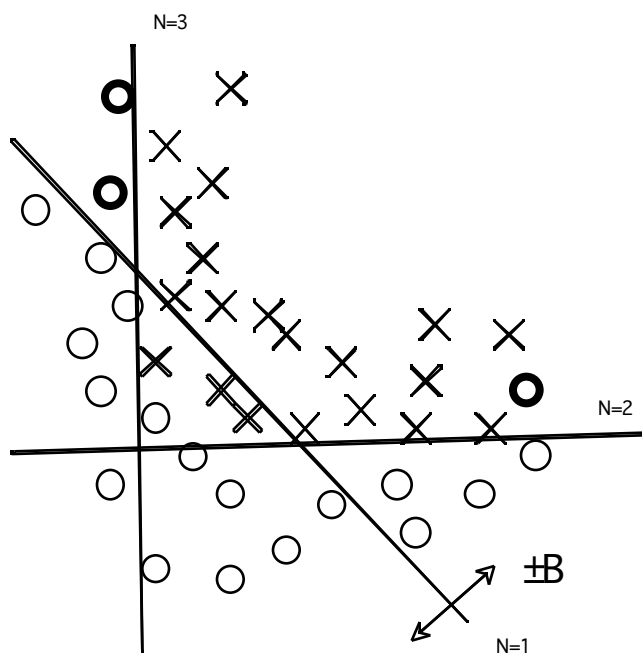
Pour N détecteurs : 
$$\hat{Y} = \sum_{n=1}^N \hat{W}_n^T X$$
 si  $Y > 0$  then  $T_1$  else  $T_2$ .



**Boosting Théorème :** l'ajout d'un détecteur avec les données pondérées par boosting améliore toujours la courbe ROC.







## Perceptrons

Un "perceptron" est une méthode incrémentale d'apprentissage inventé par Frank Rosenblatt en 1956. Il s'agit d'une méthode "en-ligne", dirigé par les erreurs. Une perception génère une ensemble d'Hyperplans pour séparer les exemples des classes. Si les exemples peuvent être parfaitement séparé, on dit que les classes sont "séparables". Sinon, ils sont "non-separable".

Le "marge". , est le plus petit séparation entre deux classes.

Si les exemples sont "séparables", l'algorithme d'apprentissage utilise les erreurs pour une mise a jour du plan jusqu'à l'il n'y a plus d'erreur. Si les exemples ne sont pas séparables, la méthode ne convergera pas, et il faut arrêter après un certain nombre de cycles.

À chaque cycle, on utilise les erreurs pour adapter le plan de séparation.

Note que pour tous les M exemples :

$$y_m(W \cdot X_m + b) = \begin{cases} 1 & \text{Si la classification est correcte} \\ -1 & \text{Si la classification est en erreur} \end{cases}$$

l'algorithme de perceptron utilise un "gain" positif pour déterminer la vitesse d'apprentissage.

Algorithme :

```

W0 = 0; b0 = 0; i = 0;
R = max { || Xm || }
REPEAT
  FOR m = 1 TO M DO
    IF ym(Wi · Xm + bi) ≤ 0 THEN
      Wi+1 = Wi + ym Xm;
      bi+1 = bi + ym R2;
      i = i + 1;
    END IF
  END FOR
UNTIL no mistakes in FOR loop.
```

La marge pour chaque exemple est :

$$m = y_m(W_i \cdot X_m + b_i)$$

Si les coefficients sont normalisés, le marge devient "géométrique" ou "Euclidienne".

$$W' = \frac{W}{\|W\|}, \quad b' = \frac{b}{\|W\|}$$

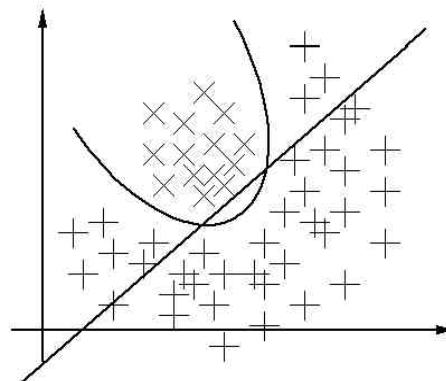
Le "qualité" d'un perceptron est donnée par la distribution des, par exemple, par un histogramme des marges géométrique.

La règle de décision est  $d(g(X)) = \text{sgn}(g(X))$

## Méthodes à Noyaux (Kernel Methods)

Parce que les fonctions de discrimination linéaire sont si simples à estimer, il est intéressant de voir si on peut les appliquer dans les cas où les données ne sont pas séparées par les plans.

Par exemple, si les covariances ne sont pas égales, une frontière quadratique donne une meilleure séparation entre les classes.



On peut transformer une discrimination linéaire en discrimination quadratique par substitution de variables.

Il s'agit de projeter un vecteur avec  $D$  dimensions dans un espace en  $P > D$  dimensions avec un noyau,  $K()$ .

Par exemple :

$X = (x_1, x_2, \dots, x_D)$  peut être projeté sur un vecteur de  $P = \frac{D(D+1)}{2}$  dimension

$W = (x_1, x_2, \dots, x_D, x_1^2, x_1x_2, x_1x_3, \dots, x_{D-1}x_D, x_D^2)$

Ainsi, une fonction quadratique en  $D = 2$  dimensions est linéaire en  $P = 5$  dimensions

$$X = (x_1, x_2) \quad K(X) = W = (x_1, x_2, x_1^2, x_1x_2, x_2^2)$$

$$g(K(X)) = g(W) = aw_1 + bw_2 + cw_3 + dw_4 + ew_5 \\ = ax_1 + bx_2 + cx_1^2 + dx_1x_2 + ex_2^2$$

Autres Noyaux populaires :

$$K(X) = \mathcal{N}(X; \mu, \sigma^2)$$