Systèmes Intelligents : Raisonnement et Reconnaissance

James L. Crowley

Deuxième Année ENSIMAG Deuxième Sémestre 2008/2009

Séance 11 6 mai 2009

Discrimination Linéaire

Sources Bibliographique :

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

Notations

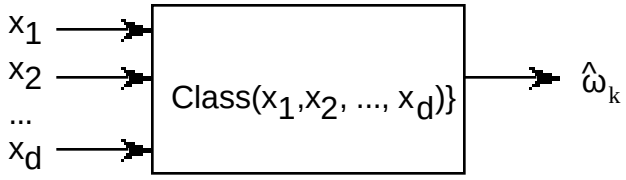
x	Une variable
X	Une valeur aléatoire (non-prévisible).
N	Le nombre de valeurs possibles pour x ou X
$\vec{x}$	Un vecteur de D variables
$\vec{X}$	Un vecteur aléatoire (non-prévisible).
D	Nombre de dimensions de $\vec{x}$ ou $\vec{X}$
T_k	La classe k
k	Indice d'une classe
K	Nombre de classes
M	Nombre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

$E_m : (\vec{X}_m, Y_m)$ 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 $\vec{X}$.
Soit K classes d'événements $\{T_k\} = \{T_1, T_2, ..., T_K\}$ avec une classe T_k $\{T_k\}$

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



$\hat{\omega}_k$ est la proposition que (E T_k): $\hat{\omega}_k = d\{g(\vec{X})\}$.

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

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

Dans les séances précédentes, nous avons développé

La règle de décision : $\hat{\omega}_k = \underset{\omega_k}{\text{arg-max}}\{ p(\omega_k \mid \vec{X})\}$

en utilisant le règle de Bayes : $p(\omega_k \mid \vec{X}) = \dfrac{p(\vec{X} \mid \omega_k)p(\omega_k)}{p(\vec{X})}$

La règle de décision est : $\hat{\omega}_k = \underset{\omega_k}{\text{arg-max}}\{ p(\vec{X} \mid \omega_k) p(\omega_k)\} = \underset{\omega_k}{\text{arg-max}}\{ g_k(\vec{X})\}$

en utilisant la loi Normale :

$$p(\vec{X} \mid \omega_k) = \mathcal{N}(\vec{X} ; \vec{\mu}_k , C_k) = \frac{1}{(2\pi)^{\frac{D}{2}} \det(C_k)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X} - \vec{\mu}_k)^T C_k^{-1} (\vec{X} - \vec{\mu}_k)}$$

on obtient :

$$g_k(\vec{x}) = -\frac{1}{2}\ \mathrm{Log}\{\det(C_k)\} - \frac{1}{2}(\vec{X} - \vec{\mu}_k)^T C_k^{-1}(\vec{X} - \vec{\mu}_k) + \mathrm{Log}\{p(\omega_k)\}$$

Forme canonique de la fonction de discrimination :

$$g_k(\vec{X}) = \vec{X}^T D_k \vec{X} + \vec{W}_k^T \vec{X} + B_k.$$

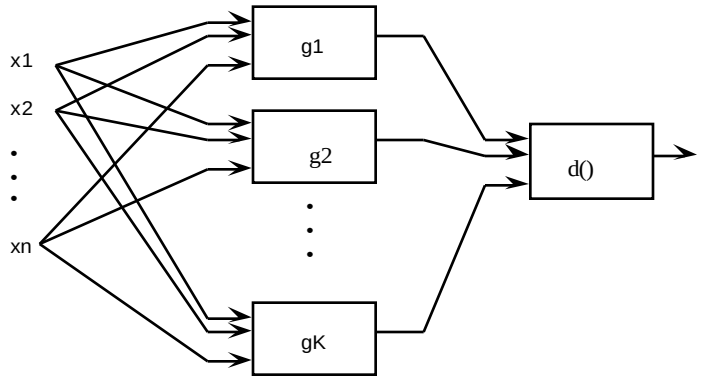
avec

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

$$\vec{W}_k = C_k^{-1}\vec{\mu}_k$$

$$B_k = -\frac{1}{2}(\vec{\mu}_k^T C_k^{-1}\vec{\mu}_k)\ -\frac{1}{2}\ \mathrm{Log}\{\det(C_k)\} + \mathrm{Log}\{p(\omega_k)\}$$

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



La Classification Linéaire

Nous allons regarder des méthodes linéaires pour concevoir les $g_k(\vec{X})$.
Rappeler que chaque observation et corrompus 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

$$\vec{X} = \vec{x} + \vec{B}_i + \vec{B}_o$$

Le bruit d'observation : $\vec{B}_o$, et typiquement Normale.

Cas simple : si le bruit d'observation domine.

En Général le Bruit d'observation et Normale.

Si $\vec{B}_o >> \vec{B}_i$ alors pour toutes les classes : $\forall j, i \; C_i \approx C_j$ et $D_i \approx D_j$

Dans ce cas, Les fonction de discrimination devient linéaire

$$g_k(\vec{X}) = \vec{W}_k^T \vec{X} + B_k.$$

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

Si la variation interclasse domine

Dan beaucoup de cas réel, la variation $\vec{B}_i$ domine $\vec{B}_o$ et $\vec{B}_i$ n'est pas normal.

$\vec{B}_i >> \vec{B}_o$. et $\vec{B}_i \neq \mathcal{N}(\vec{X}\,;\vec{\mu}\,,C)$

Que faire ?

Nous allons regarder plusieurs techniques pour accommoder les variations $\vec{B}_i$ en utilisant les fonctions de discrimination linéaire.

1) Boosting : Classification par comité de classifiers 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.

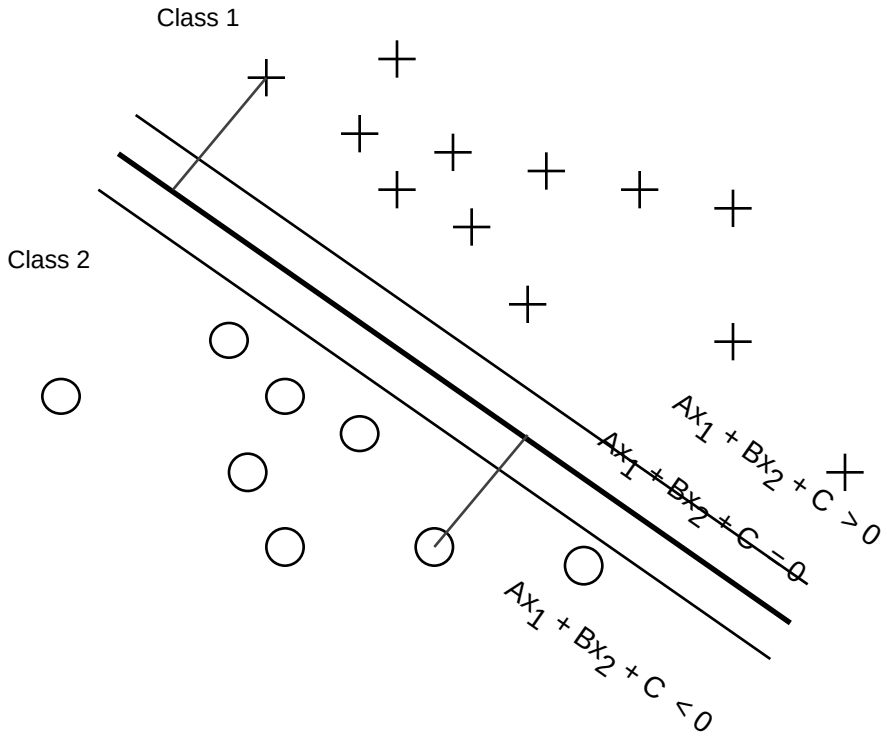
Nous allons considérer le cas d'une "détecteur" pour les événements d'une classe.

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 $\{\vec{X}_m, Y_m\}$

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( )= sgn().



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

$$w_1x_1 + w_2x_2 + ... + w_Dx_D + B = 0$$

En forme de vecteur : $\vec{W}^T \vec{X} + B = 0$

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

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

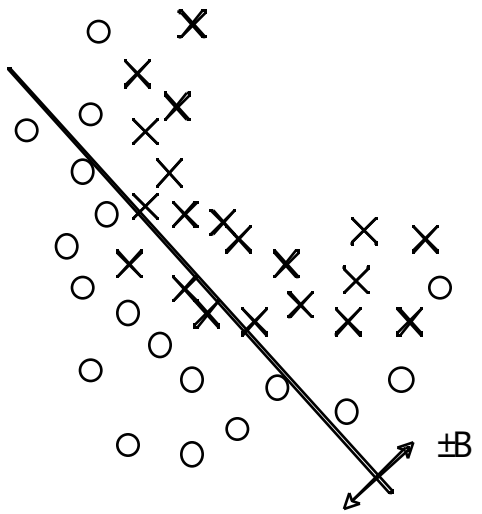
$B = -\vec{W}^T \vec{X}$ est la distance perpendiculaire à l'origine.

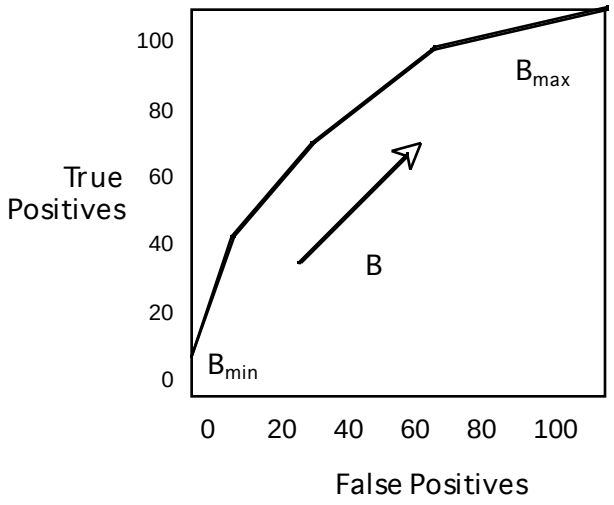
si $\| \vec{W} \| \neq 1$ alors utilise $\vec{W}' = \frac{\vec{W}}{\|\vec{W}\|}$, et $B' = \frac{B}{\|\vec{W}\|}$

Dans ce cas, on peut dire que $Y = \vec{W}^T \vec{X} + B$

projet les caractéristiques d'un événement sur une norme, $\vec{W}$.

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". (Reciever Operating Characteristics)
Un détecteur est caractérisé par son ROC (et non pas par la réponse a 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.

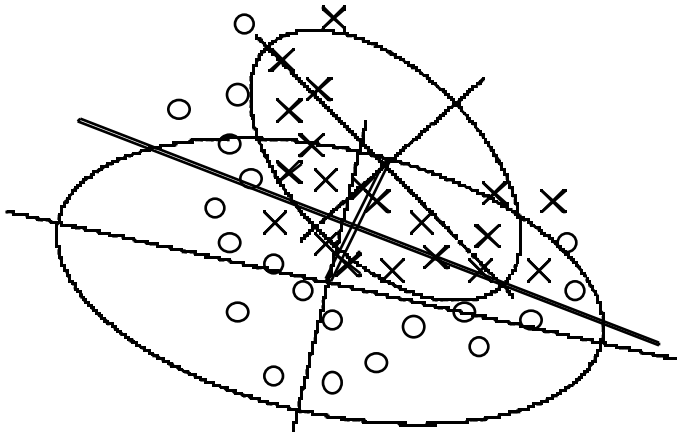
Soit $g_1(\vec{X}) = \vec{W}_1^T \vec{X} + B_1$. et $g_2(\vec{X}) = \vec{W}_2^T \vec{X} + B_2$.

ou :

$$\vec{W}_k = C_k^{-1}\vec{\mu}_k$$
$$B_k = -\frac{1}{2}(\vec{\mu}_k^T C_k^{-1}\vec{\mu}_k) - \frac{1}{2}\,\mathrm{Log}\{\det(C_k)\} + \mathrm{Log}\{p(\omega_k)\}$$

La frontière entre les deux est :

$$g_1(\vec{X}) - g_2(\vec{X}) = 0$$
$$(\vec{W}_1^T - \vec{W}_2^T)\vec{X} + B_1 - B_2 = 0$$
$$(C_1^{-1}\vec{\mu}_1 - C_2^{-1}\vec{\mu}_2) + B_1 - B_2 = 0$$



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

Donc, pour la frontiere entre les classe : $\vec{W}_{12}^T = (C_1^{-1}\vec{\mu}_1 - C_2^{-1}\vec{\mu}_2)$ et $B_{12} = B_1 - B_2$

Donc

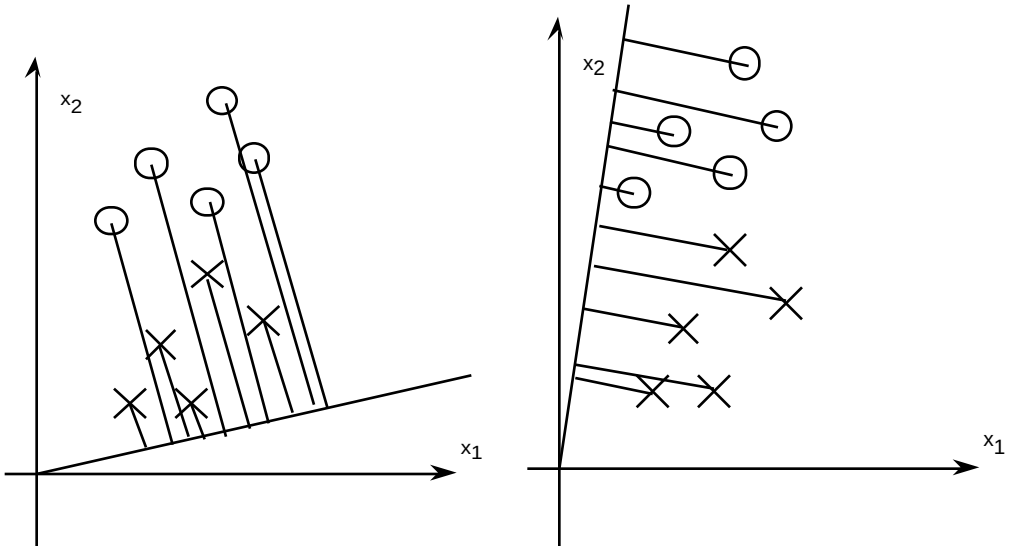
La discriminante linéaire de Fisher

Le principe de Fisher est de projeter le vecteur de caractéristique, $\vec{X}$ de D_X dimensions vers un espace $\vec{Z}$ de D_Z par une transformation linéaire F choisit tel quel

1) $D_Z << D_X$ et
2) Les exemples des classes A_k sont séparés.

$$\vec{z} = F^T \vec{x}$$

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



La discriminabilité des classes dépend de la direction de $\vec{F}$

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 $\vec{X}_{1m}$ et $\vec{X}_{2m}$
Dans ce cas, D_Z= 2–1 = 1. z est un scalaire
Pour chaque exemple :

$$z_{km} = \vec{F}^T \vec{X}_{km}$$

Nota que F est une projection telle que $\| F \| = 1$
Les moyennes des exemples pour chaque classe sont

$$\vec{\mu}_k = E\{\vec{X}_{km}\} = \frac{1}{M_k} \sum_{m=1}^{M_k} \vec{X}_{km}$$

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

$$\tilde{\mu}_k = E\{Z_{km}\} = \frac{1}{M_k} \sum_{m=1}^{M_k} z_{km} = \frac{1}{M_k} \sum_{m=1}^{M_k} F^T \vec{X}_{km} = F^T \vec{\mu}_k$$

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

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

La dispersion ("Scatter") pour une ensemble $\{\vec{X}_{km}\}$ d'exemples et pour une classe k est une **matrice** $\mathbf{S}_k$.

$$\mathbf{S}_k = M_k \mathbf{C}_k = \sum_{m=1}^{M_k} (\vec{X}_{km} - \vec{\mu}_k)(\vec{X}_{km} - \vec{\mu}_k)^T$$

La transformation F projet le vecteur $\vec{X}$ 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.

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

Fisher cherche la transformation F^T tel quel

$$F = \arg\max_{F} \{ \frac{||F^T(\vec{\mu}_1 - \vec{\mu}_2)||^2}{\tilde{s}_1 + \tilde{s}_2} \}$$

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

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

La moyenne de chaque classe est

$$\vec{\mu}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} \vec{X}_{km}$$

La moyenne de TOUS les exemples est

$$\vec{\mu} = \frac{1}{M} \sum_{k=1}^{K} M_k \vec{\mu}_k = \frac{1}{M} (M_1 \vec{\mu}_1 + M_2 \vec{\mu}_2)$$

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

$$s_B = (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$$

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

$$S_W = \sum_{k=1}^{K} S_k = \sum_{k=1}^{K} M_k \mathbf{C}_k = S_1 + S_2$$

La meilleure transformation F est celle que

$$F = \arg\max_{F} \{ \frac{|| F^T S_B F||}{|| F^T S_w F||} \}$$

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

$$S_B = \lambda S_w F.$$

donc

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

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

$$F = S_w^{-1} S_B = S_w^{-1}(\vec{\mu}_1 - \vec{\mu}_2)$$

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 :

$$F^T \vec{X} + b_o = 0 \qquad \text{où } F = C^{-1}(\vec{\mu}_1 - \vec{\mu}_2)$$

et b_o est un constant

Note que avec l'approche Baysienne, nous avons trouvé :

$$\vec{W}_{12}^{\,T} X + B_{12} = 0$$

ou $\vec{W}_{12}^{\,T} = (C_1^{-1}\vec{\mu}_1 - C_2^{-1}\vec{\mu}_2)$
et $B_{12} = B_1 - B_2$

Estimation par Moindres de Carrées

Soit M exemples $\{y_m, \vec{X}_m\}$ tel que y=+1 pour les événements de T_1,
et y = –1 pour les événements de T_2

On cherche $y = g(\vec{X}) = \vec{W}^T \vec{X}$

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

Pour l'ensemble des M exemples $\{y_m, \vec{X}_m\}$. on compose le matrice **X** et une vecteur **Y**

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

$\mathbf{Y} = (y_1, y_2, ..., y_M)^T$ (taille M lignes).

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

Le Minimum est trouvé quand :

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

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

et E T_1 si $\hat{Y} = \vec{W}^T \vec{X} > 0$

on peut ajouter un biais : E T_1 si $\vec{W}^T \vec{X} + B \geq 0$

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 classifierslinéaire 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 classifier.

Résumé de l'algorithme de Boosting

Soit une ensemble des exemples $\{Y_m, \vec{X}_m\}$

1) Initialiser un vecteur de M coefficients $w_m = 1$

(w_m/S serait le poids de chaque exemple, $\{Y_m, \vec{X}_m\}$. ou S est la somme des poids.

2) Pour l'ensemble des M exemples $\{Y_m, \vec{X}_m\}$. on compose le matrice **X** et une vecteur $\vec{Y}$ pour calculer le N^{ieme} fonction de discrimination.

$$S = \sum_{m=1}^{M} w_m$$

$m,\ a_m = w_m/S$

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

$\vec{Y} = (a_1 Y_1, a_2 Y_2, ..., a_M Y_M)^T$ (taille M lignes).

On trouve une classfication, par exemple avec :

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

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

3) Pour chaque exemple en $\{X_m\}$, si $\hat{W}^T \vec{X}_m - Y_m \leq 0$ alors error :
$w_m = w_m + 1$,

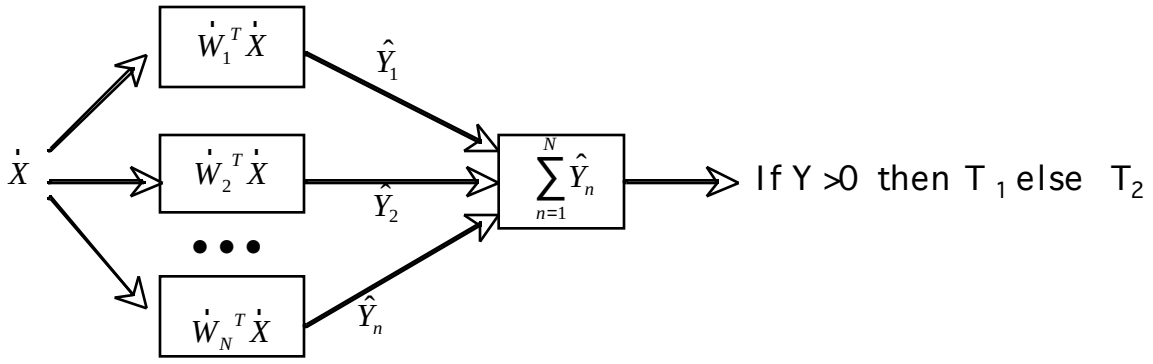
4) Repeter étape 2 pour classifier N+1.

Détection par une Comité de Classifiers :

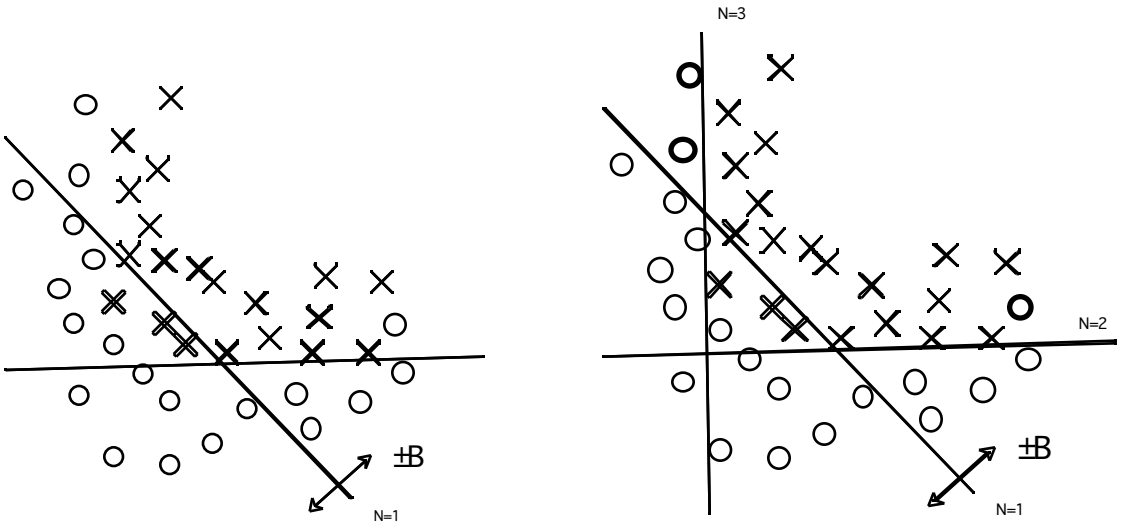
Avec un comité de classifiers, la décision sont faites par vote

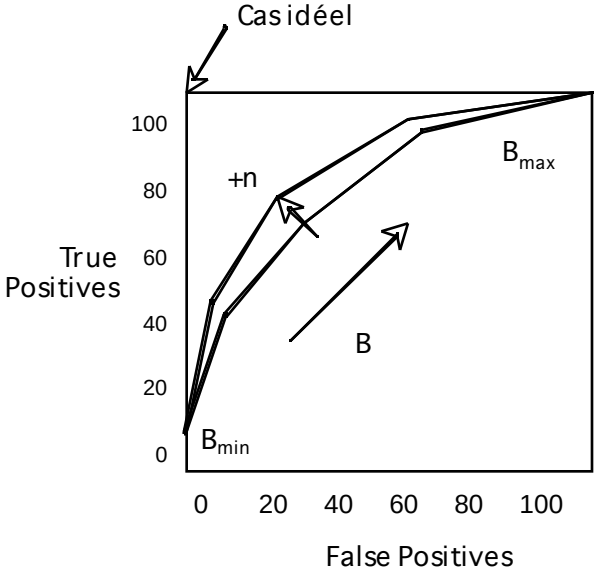
Pour N détecteurs : $\hat{Y} = \sum_{n=1}^{N} \hat{W}_n^T X$

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.




Cas idéel
100
80
60
40
20
0
True Positives
+n
B_{max}
B
B_{min}
0 20 40 60 80 100
False Positives

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". γ, 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 :

$$y_m(\vec{W}^T \vec{X}_m + b) = \begin{cases} 1 & \text{Si la classification et correcte} \\ -1 & \text{Si la classification est en erreur} \end{cases}$$

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

Algorithme :

$\vec{W}_o \leftarrow 0$; $b_o \leftarrow 0$; $i = 0$;
$R \leftarrow \max \{ \| \vec{X}_m \| \}$
REPEAT
  FOR m = 1 TO M DO
    IF $y_m(\vec{W}_i^T \vec{X}_m + b_i) \le 0$ THEN
      $\vec{W}_{i+1} \leftarrow \vec{W}_i + \eta\, y_m \vec{X}_m$;
      $b_{i+1} \leftarrow b_i + \eta\, y_m R^2$;
      $i \leftarrow i + 1$;
    END IF
  END FOR
UNTIL no mistakes in FOR loop.

La marge pour chaque exemple est :

$$\gamma_m = y_m(\vec{W}_i^T \vec{X}_m + b_i)$$

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

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

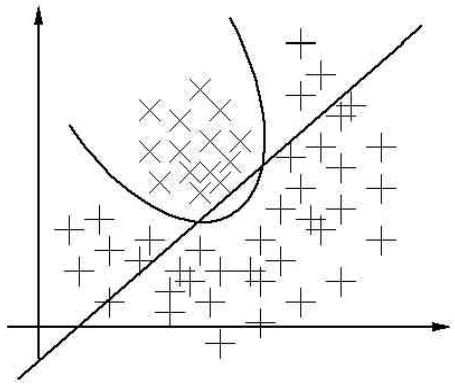
Le "qualité" d'un perceptron est donnée par la distribution des, par exemple, par un histogramme des marges géométrique.

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

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 ou 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 :

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

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

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

$\vec{X} = (x_1, x_2)$ $K(\vec{X}) = \vec{W} = (x_1, x_2, x_1^2, x_1x_2, x_2^2)$

$g(K(\vec{X})) = g(\vec{W}) = aw_1 + b\,w_2 + c\,w_3 + d\,w_4 + e\,w_5$

$= ax_1 + b\,x_2 + c\,x_1^2 + d\,x_1x_2 + e\,x_2^2$

Autres Noyaux populaires :

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