Systèmes Intelligents : Raisonnement et Reconnaissance

James L. Crowley

Deuxième Année ENSIMAG Deuxième Sémestre 2005/2006

Séance 11 19 avril 2006

Discrimination Linéarie

Sources Bibliographique :

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

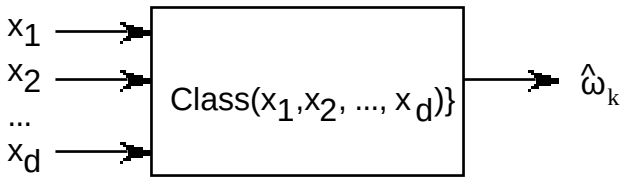
Notations

x	Une variable
X	Une valeur aléatoire (non-prévisible).
N	Le nombre de valeurs possible 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
$(y_m, \vec{X}_m)$	Une exemple d'événement.
y_m	Un variable d'Indication pour l'exemple $\vec{X}_m$ $y_m = \{-1, +1\}$
$\vec{Y}_m$	Un vecteur d'indication l'exemple $\vec{X}_m$ pour k classes
M	Nombre totale d'exemples de toutes les classes

Fonctions de Discrimination (rappel)

Soit les événements E décrivent par un vecteur de caractéristiques $\vec{X}$: $(E,\vec{X})$.
Soit K classes d'événements $\{T_k\} = \{T_1, T_2, ..., T_K\}$ avec une classe $T_k \quad \{T_k\}$

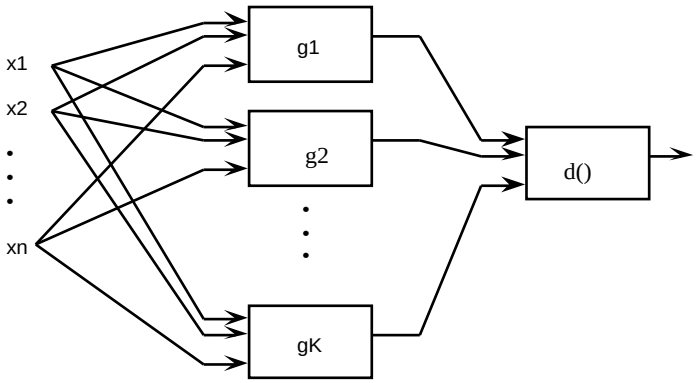
La classification est un processus d'estimation de l'appartenance d'un événement à une des classes T_k fondée sur les caractéristiques de l'événement, $\vec{X}$.



$\hat{\omega}_k$ est la proposition que $(E \quad T_k)$: $\hat{\omega}_k = d\{g(\vec{X})\}$.
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 cette forme, le classificateur est une machine qui calcule K fonctions $g_k(\vec{x})$ suivie d'une sélection de la décision.



Aujourd'hui nous allons regarder des méthodes linéaires pour concevoir les $g_k(\vec{X})$.

Il existe plusieurs techniques simples d'estimation de $g_k()$comme fonction linéaire. Certains sont très simples et très efficaces. Il ne repose pas sur l'hypothèse de bruit Gaussienne. On peut les combiner une classification linéaire avec la méthode de noyau pour réaliser les systèmes de classification dit "discriminative".

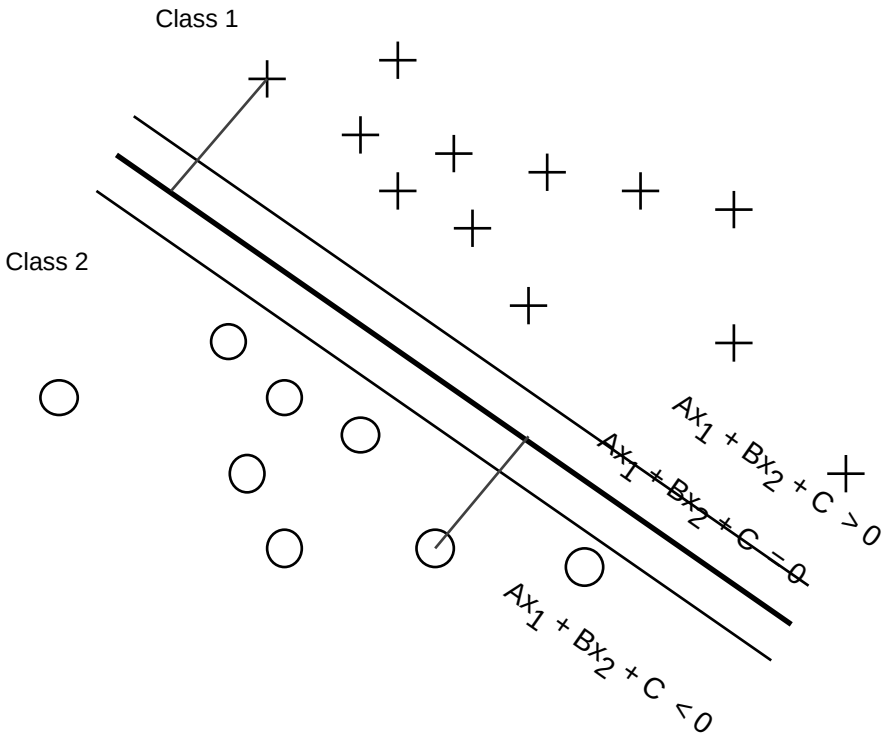
Rappelle de quelques faits sur les plans et les Hyperplans

Soit K = 2 (Deux classes)

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

Les y_m s'appel les variables d'indication.

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 points 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 $\vec{W} = \begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_D \end{pmatrix}$ 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, $\vec{W}$.

Le plus que y est grande, le plus que $\vec{X}$ est semblable à T_1.

Pour certaines opérations, il nous faut les coordonnées homogènes.

$\hat{X} = \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$ $\hat{W} = \begin{pmatrix} \vec{W} \\ b \end{pmatrix}$

Dans ce cas on peut écrire $y_m = \hat{W}^T \hat{X}_m = \hat{X}_m^T \hat{W} = w_1x_1 + w_2x_2 + ... + w_Dx_D + b$

Estimation par Moindres de Carrées

Discrimination Binaire : K=2

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 $y = g(\vec{X}) = \vec{W}^T \vec{X} + b = \hat{X}_m^T \hat{W}$ (en coordonnées Homogènes).

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

Le Minimum est trouvé quand :

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

Donc : $\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\hat{W}$ et donc $\hat{W} = \begin{pmatrix} \vec{W} \\ b \end{pmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{Y}$

Discrimination Linéaire MultiClasse

Pour le cas ou K > 2 On définit un K dimensionnel vecteur d'indication, $\vec{Y}$.

$$\vec{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{pmatrix}$$

Pour un exemple de la classe k, le k_{th} coefficient vaut 1, les autre –1.

$$y_k \triangleq \begin{cases} 1 & \text{E } T_k \\ 0 & \text{sinon} \end{cases}$$

(Les autres classes ne contribue pas au hyperplane).

Ensuite $\mathbf{Y} = (\vec{Y}_1, \vec{Y}_2, \ldots, \vec{Y}_M)$

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

W est une matrice composée de D lignes et K Colonnes.

Pour une d'événement $\vec{X}$, $\hat{X} = \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$ $\hat{W} = \begin{pmatrix} \vec{W} \\ b \end{pmatrix}$

$\vec{Y} = \hat{W}^T \hat{X} = w_1x_1 + w_2x_2 + \ldots + w_Dx_D + b$ est un vecteur de K valeurs de vraisemblance.

On cherche $\omega_k = \arg\max_{y_k} \{ \vec{Y} = \hat{W}^T \hat{X} \}$

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".

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

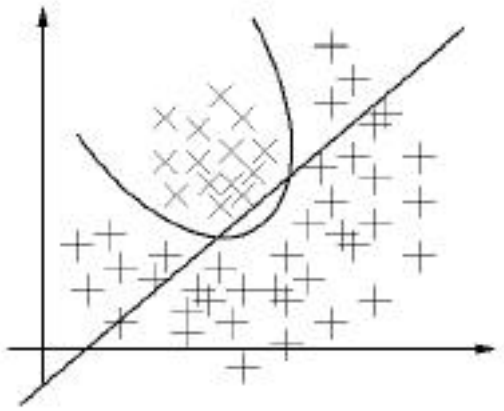
Le "qualité" d'un perceptron est donnée par la distribution des, par exemple, par un histogramme des marges géométrique.

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 sépararé par les plans.

Par exemple, si les covariances ne sont pas égale, 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 quadratic 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 :

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

$K(\vec{x}) = \ln(\vec{x})$

Boosting

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 classifiers 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 classifier.

Résumé de l'algorithme de Boosting quand K = 2

Soit deux classes T_1 et T_2. 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 $\{Y_m, \vec{X}_m\}$. on compose le matrice **X** et une vecteur $\vec{Y}$ pour calculer le N^{ieme} fonction de discrimination.

$\mathbf{X} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_M)$ (taille D lignes, et M colonnes)
$\vec{Y} = (Y_1, Y_2, ..., Y_M)^T$ (taille M lignes).

On trouve une classfication, par exemple avec :

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

3) Pour chaque exemple en $\{X_m\}$, si $d\{g_n(\vec{X})\} \neq \omega_k$ alors error :
$a_m = a_m + 1$, S = S +1

4) Repeter étape 2 pour classifier N+1.

La discriminante linéaire de Fisher

Dans beaucoup de domaines, il existe une multitude de caractéristiques utilisables pour la reconnaissance. Chaque caractéristique semble à apporter une contribution pour un cas ou dans une autre. Il semble souhaitable de les inclure dans le vecteur $\vec{x}$. Mais ceci induit une croissance exponentielle dans les nombres d'exemples nécessaires, M. En Anglais, on appelle ce problème "The Curse of Dimensionality".

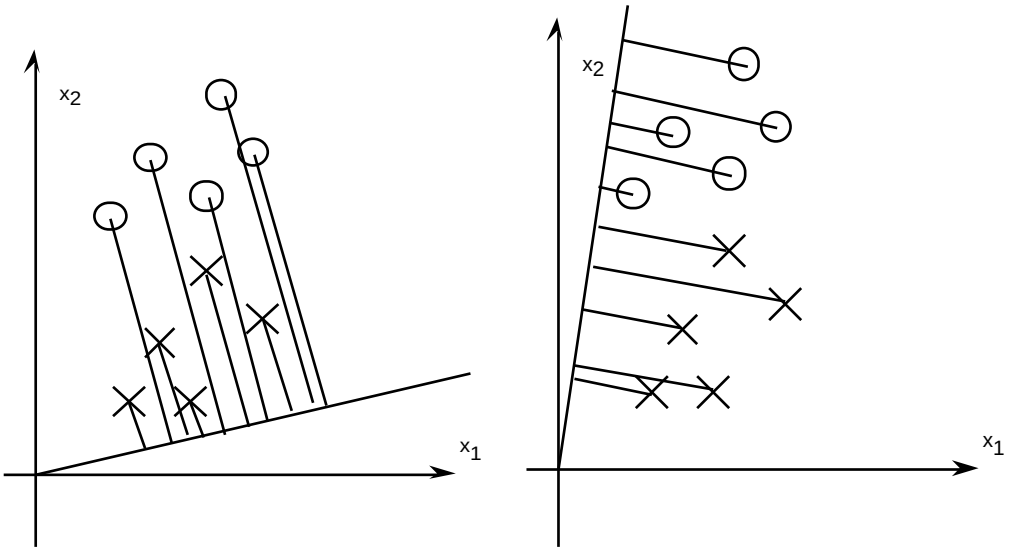
La technique de Fisher permet une réduction dans le nombre de dimensions, D avec une faible augmentation dans la probabilité d'erreur.

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

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

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

$$F = \underset{F}{\text{argmax}}\{ \frac{|| F^T S_B F||}{|| F^T S_w F||} \}$$

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

$$S_B F = \lambda S_w F.$$

donc

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

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

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

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 :

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

et b_o est un constant

Le Discriminant de Fisher avec K > 2

Comment généraliser en $D_z = K-1$ dimensions?

Pour le cas de K classes, la généralisation naturelle est avec K-1 fonctions de Fisher.
Il est supposé que $D \geq K$.

$$S_w = \sum_{k=1}^{K} S_k$$

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

et $$\vec{\mu}_k = \frac{1}{M_k} \sum \vec{X}_k$$

On peut définir une moyenne globale, $\vec{\mu}$ et une matrice de dispersion totale S_T comme :

$$\vec{\mu} = \frac{1}{M} \sum_{m=1}^{M} \vec{X}_m = \frac{1}{M} \sum_{k=1}^{K} M_k \, \mu_k$$

et $$S_T = \sum_{k=1}^{K} \sum_{m=1}^{M} (\vec{X}_{km} - \vec{\mu})(\vec{X}_{km} - \vec{\mu})^T$$

S_T est la dispersion "totale".

On peut definir la dispersion entre classe ("between class") comme

$$S_B = \sum_{k=1}^{K} M_k \, (\vec{\mu}_k - \vec{\mu})(\vec{\mu}_k - \vec{\mu})^T$$

Il est possible de demontrer que

$$S_T = S_w + S_B$$

Pour chaque classe, k, on obtient une transformation F_k dans la forme d'un vecteur à D dimensions.

$$z_k = F_k^T \vec{X}$$

Si on aligne les transformations dans une matrice de taille D x (K-1) on a

$$\vec{Z} = F^T \vec{X}$$

Ou $\vec{Z}$ est un vecteur de K-1 coefficients.

Par invariance des moments, on peut montrer que

$$\tilde{S}_w = F^T S_w F$$
$$\tilde{S}_B = F^T S_B F$$

Le Critère de Fisher est de maximiser

$$J(W) = \frac{\det(\tilde{S}_B)}{\det(\tilde{S}_w)} = \frac{\det(F^T S_B F)}{\det(F^T S_w F)}$$

La solution est rendue par une analyse en composant principales de $\tilde{S}_B$

$\tilde{S}_B F_k = \lambda_k \tilde{S}_w F_k$ Résoudre F_k telle que $(\tilde{S}_B - \lambda_k \tilde{S}_w) F_k = 0$

Les colons peuvent être calculé par une simple orthogonalisation par l'algorithme de Gram-Schmidt des vecteurs

$(\vec{\mu}_k - \vec{\mu})$ pour $k = 1, ..., K-1$.

On note que F n'est pas unique. Il existe une classe d'équivalence avec les rotations et multiplications par une constant.

Classification Linéaire Bayesienne.

Le cas des variances blanches ("Matched Filter").

Si i,j $\sigma_{ij}^2 = \sigma^2$ dans ce cas.

$$\det(C) = (\sigma^2)^n \quad \text{et} \quad C_k^{-1} = \frac{1}{\sigma^2} \quad I$$

Parce que les termes $\frac{n}{2}$ Log{2π}, $\frac{1}{\sigma^2}$ I et $(\sigma^2)^n$ sont indépendants de i et j,

$$g_k(\vec{x}) = -\frac{\|(\vec{x} - \vec{\mu}_k)\|^2}{2\sigma^2} + \text{Log}\{p(w_k)\}$$

Ce cas arrive quand les observations sont corrompues par un bruit additif blanc indépendant des classes et d'une puissance égales pour toutes les caractéristiques. Ce cas se rencontre dans les systèmes de réception des signaux hertziens, ainsi que pour la numérisation des images et des sons. En électronique, il est connu comme le cas de la réception "optimal" ("matched Filter")

Ceci est la forme d'un détecteur optimal étudié en théorie de la communication. Pour chaque classe, T_k, le vecteur $\vec{\mu}_k$ est utilisé comme "prototype" ou motif. Avec cette formule, C. Shannon a fait une révolution pour la communication hertzienne en 1946. Son résultat a résolus un problème posé depuis la naissance du télégraphe en 1840 : Combien de message peut-on placer sur un canal de communication ?

Si les caractéristiques suivent une densité Normale :

$$p(\vec{X} \mid \omega_k) = \mathcal{N}(\vec{X}\ ;\ \vec{\mu}_k, C) \quad \text{et } g_k(\vec{X}) = \text{Log}\{\ \mathcal{N}(\vec{X}\ ;\ \vec{\mu}_k, C) \cdot p(\omega_k)\ \}$$

La fonction de discrimination devient est une fonction quadratic

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

On peut réécrire $g_k(\vec{x})$ comme

$$g_k(\vec{X}) = \vec{X}^T \mathbf{B}\, \vec{X} + \vec{b}_k^{\,T} \vec{X} + b_{ko.}$$

Mais parce que $\mathbf{B} = \frac{1}{2}\ \mathbf{C}^{-1}$ est independent de k, on peut l'éliminer.

Le terme linéaire s'exprime : $\vec{b}_k = C^{-1}\ \vec{\mu}_k^{\,T}$

et le constant est $b_{ko} = -\frac{1}{2}(\vec{\mu}_k^{\,T} C^{-1} \vec{\mu}_k) + \mathrm{Log}\{p(\omega_k)\}$

Mais, si tous le message ont le même probabilité :

$$b_{ko} = -\frac{1}{2}(\vec{\mu}_k^{\,T} \vec{\mu}_k) = -\frac{1}{2}\ \|\vec{\mu}_k\|^2$$

Donc, on peut réduire $g_k(\vec{X}) = \vec{\mu}_k^{\,T} \vec{X} - \frac{1}{2}\ \|\vec{\mu}_k\|^2$

La problème de la communication est formalisé comme :
Comment chercher un message "M(t)" dans un signal S(t)?
Pour la communication hertzienne, les caractéristiques sont les canaux de bande passante, et le bruit est blanc. Il est indépendant du signale

Pour chaque message T_k, on prend une moyenne d'échanillon comme un "prototype". On normalise l'énergie (L'énergie ne dépend pas du message) pour former une prototype.

$$\vec{b}_k = b(t) = \frac{\vec{\mu}_k}{\|\vec{\mu}_k\|}$$

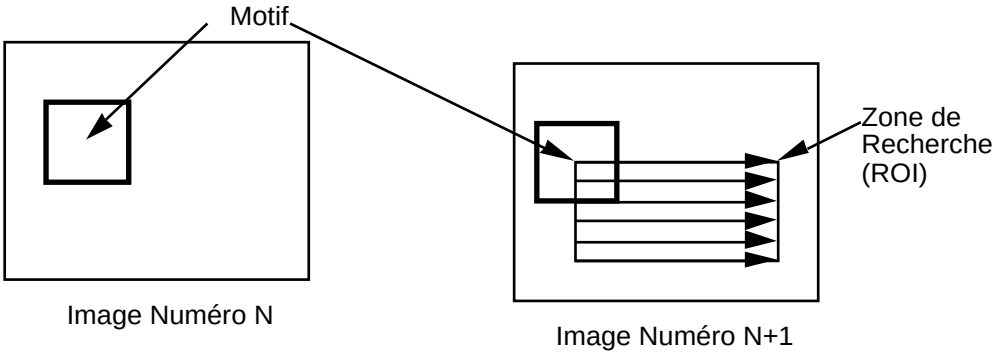
Ensuite, à chaque t_o pour chaque k, on cherche $\int b(t)\ S(t-t_o)\ dt$ < seuil. Si oui,

$$\omega_k = \arg\max_k \{\int b(t)\ S(t-t_o)\ dt\}$$

Exemple : Intercorrelation de motifs (NCC).

(Normalised Cross Correlation). Il s'agit d'une technique d'analyse d'image utilisée pour suivi de cible dans une séquence d'images.

Problème : Soit deux image $S_t(i,j)$ et $S_{t+1}(i,j)$.
Soit le voisinage (imagette) M(i, j) issu du $S_t(i,j)$ a position (i_o, j_o).
Retrouver sa position (i_1, j_1) dans l'image $S_{t+1}(i,j)$.



Les classes sont les imagettes de l'Image t+1. Ils sont égaux : $p(\omega_i) = p(\omega_j)$.
les variances sont égales : d $\sigma_{dd}^2 = \sigma^2$
les variances des pixels sont indépendantes : i,j $i \neq j$ $\sigma_{ij}^2 = 0$ Donc $C_k = \sigma^2 I$

Pour éviter les variations d'intensité, on normalise les imagettes.

Si les vecteurs $\vec{M}$ et $\vec{S}$ ont une longueur unitaire, le produit scalaire est un cosinus de l'angle entre les vecteurs.

$$M_u(m, n) = \frac{\vec{M}}{\|\vec{M}\|} = \frac{M(m, n)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} M(m, n)^2}$$

$$S_u(m, n) = \frac{\vec{S}}{\|\vec{S}\|} = \frac{S(i_1+m, j_1+n)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} S(i_1+m, j_1+n)^2}$$

On obtient un inter corrélation "normalisée" par l'énergie (NCC) :

$$NCC(i_1, j_1) = < \vec{M}_u, \vec{S}_u > = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{S(i+m,j+n)}{\|\vec{S}\|} \frac{M(m, n)}{\|\vec{M}\|}$$

Le NCC est le cosinus entre les vecteurs $\vec{M}$ et $\vec{S}$. Sa valeur est entre –1 et 1.