Analyse et Reconnaissance d'Images

James L. Crowley

M2R IVR Premier Semestre 2007/2008

Séance 8 16 décembre 2007

Discrimination Linéaire

Sources Bibliographiques :

Viola, P. and Jones, Rapid object detection using a boosted cascade of simple features. In IEEE Conference on Computer Vision and Pattern Recognition., IEEE CVPR 2001.

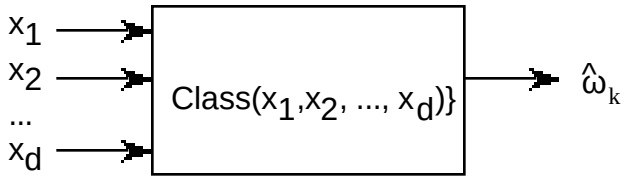
Notations

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 de Séance 6)

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 E $\{T_k\}$ et
i, j tel que $i \neq j$: $T_i \cap T_j =$

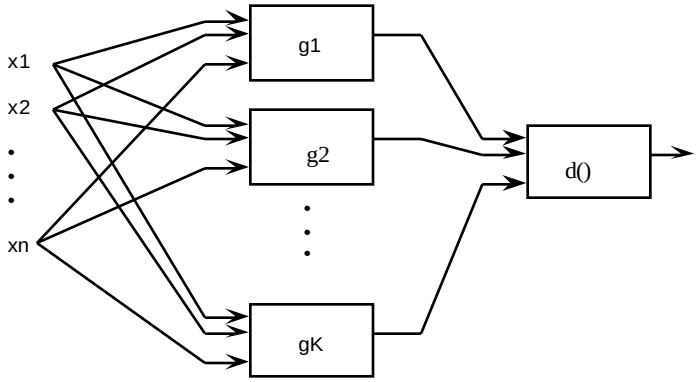
La classification est un processus d'estimation de l'appartenance d'un événement E à une des classes T_k fondée sur les caractéristiques de l'événement, $\vec{X}$.



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



Dans le cas générale D > 1 et 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 :

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

Cecit peut être traduit dans une forme cannonique :

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

avec une terme 2^{ieme} ordre : $B_k = \frac{1}{2} C_k^{-1}$

une terme 1^{iere} ordre : $\vec{d}_k = C_k^{-1}\vec{\mu}_k$

Constant : $b_k = -\frac{1}{2}(\vec{\mu}_k{}^T C_k^{-1}\vec{\mu}_k) - \frac{1}{2} \text{Log}\{\det(C_k)\} + \text{Log}\{p(\omega_k)\}$

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".

Classification Linéaire Bayesienne.
Le cas des variances blanches ("Matched Filter").

Si $\forall i,j\ \sigma_{ij}^2 = \sigma^2$ dans ce cas. $\det(C) = (\sigma^2)^n$ et $C_k^{-1} = \frac{1}{\sigma^2} I$

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

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

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 ?

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

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

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

Mais, si tous les message ont le même probabilité :

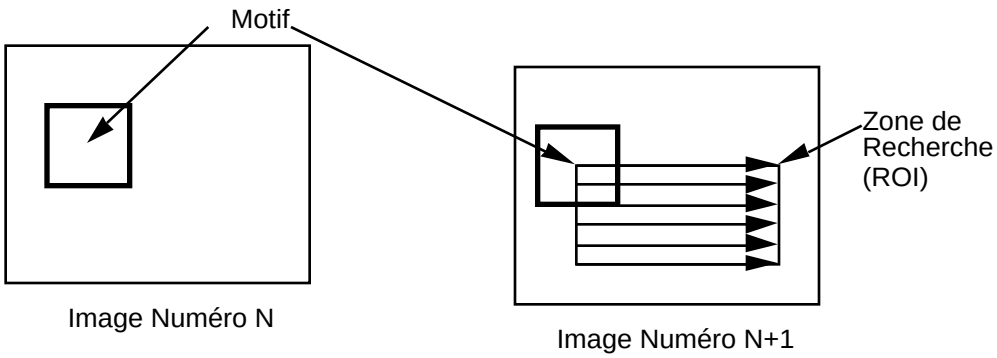
$$b_k = -\frac{1}{2}(\vec{\mu}_k^T \vec{\mu}_k) = -\frac{1}{2} \|\vec{\mu}_k\|^2$$

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

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 : $\forall d \quad \sigma_{dd}^2 = \sigma^2$
les variances des pixels sont indépendantes : $\forall i,j \quad i \neq j \; \sigma_{ij}^2 = 0$ 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.

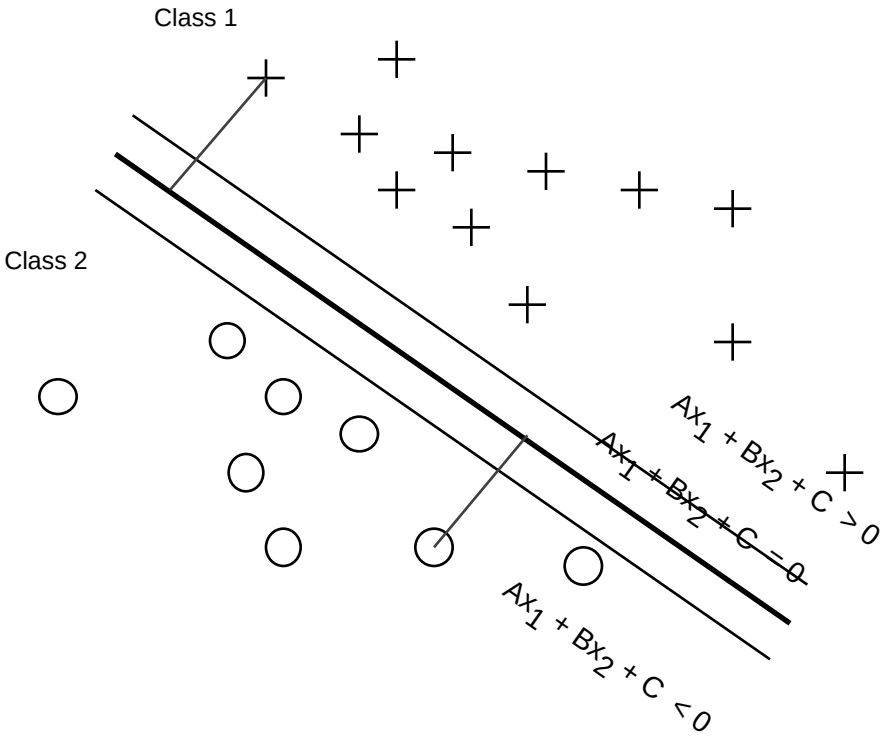
Quelques Faits sur les Hyperplans

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

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

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

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

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

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

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

Le Minimum est trouvé quand :

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

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

Discrimination Linéaire MultiClasse

Pour le cas ou K > 2 Nous allons chercher k vecteurs W_k tel que

$$\omega_k = \arg\max_k \{ W_k^T \vec{X} + b \}$$

Pour chaque $\vec{X}$ On définit un K dimensionnel vecteur d'indication, $\vec{Y}$.

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

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

$$y_k \triangleq \begin{cases} 1 & \text{si } k = T_k \\ -1 & \text{sinon} \end{cases}$$

Pour l'ensembles des M exemple $\{\vec{Y}_m, \vec{X}_m\}$. on compose des matrices **X** et **Y**

$$\mathbf{X} = X_m^d = (\vec{X}_1, \vec{X}_2, ..., \vec{X}_M)$$

$$\mathbf{Y} = Y_m^k = (\vec{Y}_1, \vec{Y}_2, ..., \vec{Y}_M)$$

X a D lignes et M colonnes. Y a K lignes et M colones.

Soit K=2, D=3, M = 4.

$$X_m^d = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \qquad Y_m^k = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{Y}^T = \left(\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \right) \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}$$

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

$$\omega_k = \arg\max_k \{ W_k^T \vec{X} + b \}$$

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".

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

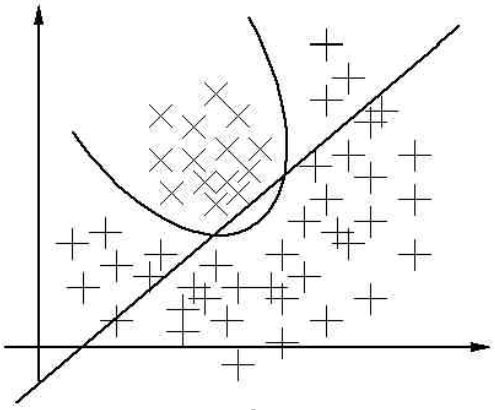
Le "qualité" d'un perceptron est donnée par la distribution des, par exemple, par un histogramme des marges géométrique.

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 pour D >>1, 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 :

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

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

Les noyaux sont utilisé avec les méthodes linéaire tel que Regression ou les Perceptrons pour faire les système efficace.

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 avec $a_m = 1$. Initialiser S = M. i = 1
(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_m^d et une vecteur Y_m pour calculer le I^{ieme} fonction de discrimination avec le poid a_m.

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

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

On trouve une classfication, par exemple avec :

$$\mathbf{W_i} = (\mathbf{X}\,\mathbf{X}^T)^{-1}\,\mathbf{X}\,\mathbf{Y}^T$$

3) Pour chaque exemple en $\{X_m\}$, si la classification est en erreur, on accroitre son poid a_m

si $d\{g_n(\vec{X})\} \neq \omega_k$ alors error :
$a_m = a_m + 1$, S = S +1,

4) Normalise les poids a_m : $a_m := a_m / \sum_{m=1}^{M} a_m$

5) i = i+1; Repeter étape 2.

La Classification final est fait par une Vote des classifiers "boostés"

$$\omega_k = \arg\max_k \{ \sum_{i=1}^{I} W_{ik}^T \vec{X} + b_i \}$$

Détection des Visage par Ada-Boost.

Integral Images :

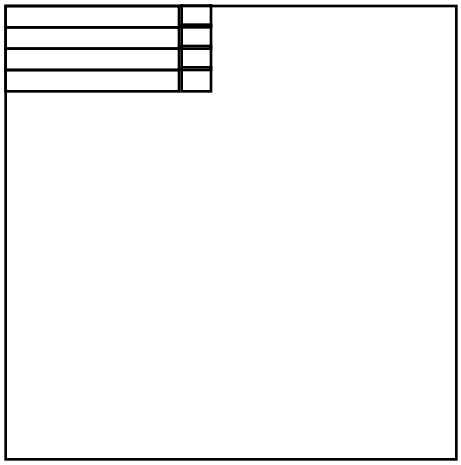
Soit une image P(i, j).

L'integral emage mn est :

$$S_{mn}(i, j) = \sum_{k=1}^{m} \sum_{l=1}^{n} P(i+k, j+l)$$

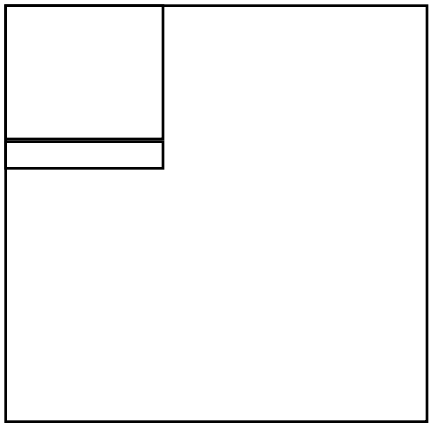
Il existe un algorithme rapide pour le calcul somme des voisinage de tous les tailles.

2) Pour chaque m, calculer $R_m(i,j) = \sum_{k=0}^{m} P(i+m, j) = R_{m-1}(i, j) + P(i+m, j)$



$R_m(i,j) = R_{m-1}(i, j) + P(i+m, j)$

3) pour chaque $S_{mn}(i,j) = \sum_{m=0}^{M} \sum_{n=0}^{N} P(i+m, j+n) = S_{n-1}(i, j) + R(i, j+n)$



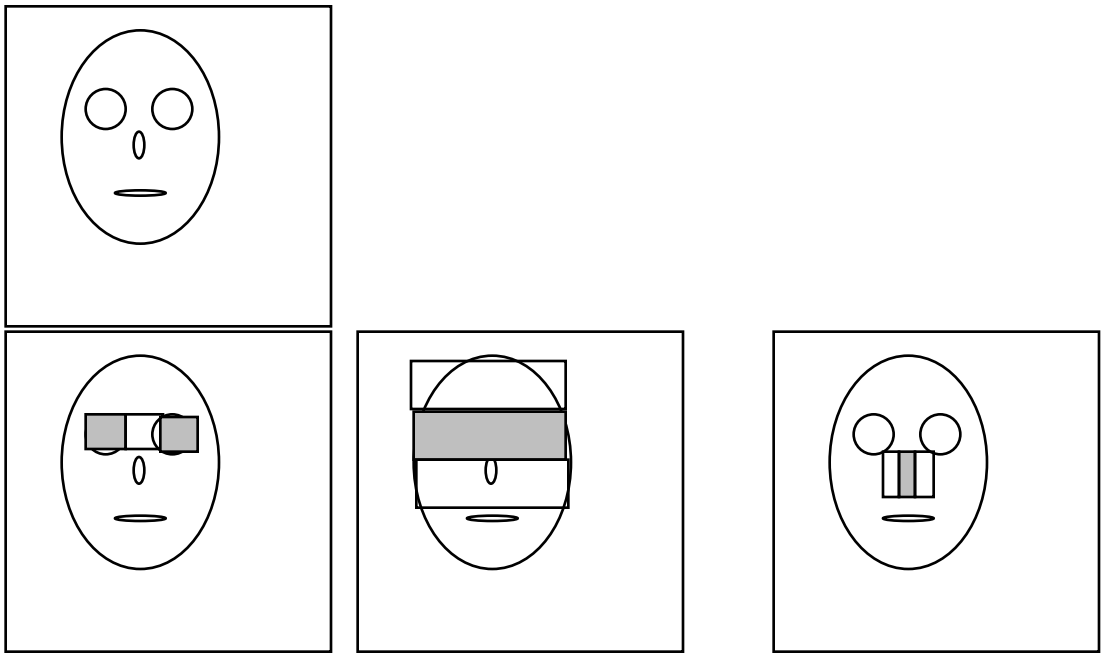
Les differences des Integrals Images donnes les characteristiques

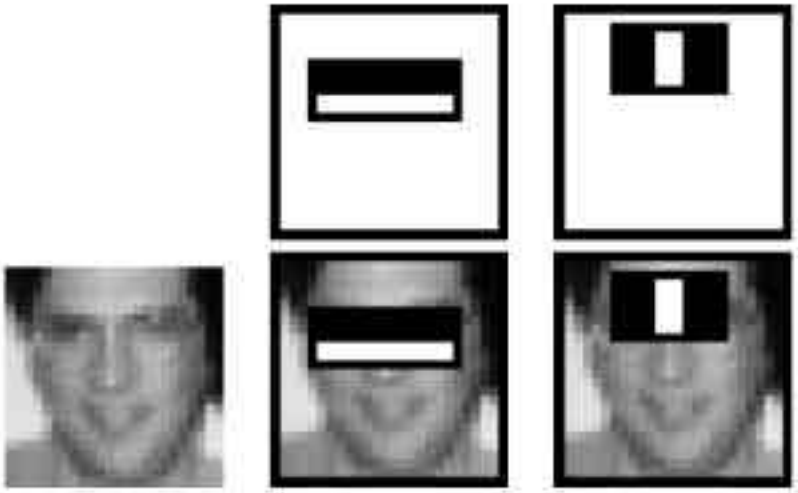
Ceci donne une grande quantité (D) de differences de fenetres uniformes : (les ondellettes de Haar).

Chaque ondelette est une hyperplan dans l'Espace a D dimensions.

On utilise Boosting pour determiner les plans les les plus discriminant.
Exemples :

Exemples

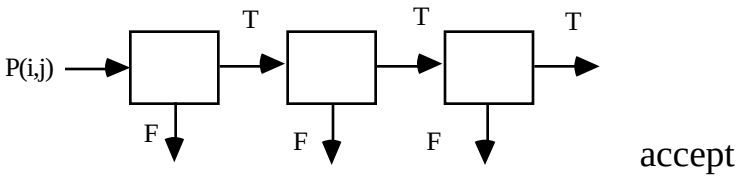




Exemple extrait du Viola Jones 2001

Détection par cascade de classifications.

Une cascade est appris par boosting.



Reject

Exemple : 38 etages et 6000 caracteristiques

Le detecteur est appliqué à une gamme de positions, i, j, est à une emsemble de echelles, S.