Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3 2009/2010

Séance 12 18 janvier 2010

PCA, Mélange de Gaussiennes et L'Algorithme EM (Expectation-Maximization)

Sources Bibliographiques :

"Neural Networks for Pattern Recognition", C. M. Bishop, Oxford Univ. Press, 1995.

"Pattern Recognition and Scene Analysis", R. E. Duda and P. E. Hart, Wiley, 1973.

La Reconnaissance de Visage

Ayant détecté et normalisé les imagettes de visages en position et taille (séance 11), il est possible de construire un système de reconnaissance Baysienne.

Pour cela nous allons utiliser l'Analyse en Composants Principales (ACP ou PCA en Anglais) et une Mélange de Gaussiennes. (GMM pour Gaussian Mixture Model en Anglais)

Il nous faut une ensemble d'apprentissage. Soit M imagettes normalisés, $\{W_m\}$ composées de K ensembles $\{W_m^k\}$ de M_k imagettes pour K personnes.

$$\{W_m\} = \bigcup_k \{W_m^k\},\ M = \sum_k M_k$$

Notre problème :

Ayant un nouveaux imagette (normalisé) W, déterminer son identité, k.

$\omega_k \equiv$ W est une imagette du face de la personne k.

$$\omega_k = \underset{k}{\text{arg-max}} \{p(\omega_k | W)\}$$

Pour faire cette estimation, nous allons utiliser la règle de Bayes :

$$p(\omega_k | W) = \frac{p(W | \omega_k)p(\omega_k)}{p(W)}$$

Mais nous avons deux problèmes :

1) Quel vecteur de caractéristique, $\vec{X}$, doit en utiliser pour représenter W

2) Comment déterminer (estimer) $p(\omega_k | \vec{X})$ et $P(\vec{X})$.

Solutions proposeé

1) Quel vecteur de caracteristiques $\vec{X}$ d'utiliser ?

Pour le premier problème, nous allons utiliser une analyse en composants principales pour construire un "sous-espace" de l'espace de W.

Avec une espace de composant principales $\vec{\varphi}$

$\vec{X} = \langle W, \vec{\varphi} \rangle$

2) Comment representer $p(\omega_k \mid \vec{X})$ et $P(\vec{X})$?

Pour le 2ieme problème nous allons construire un mélange de Gaussiennes avec l'algorithme EM.

Mélange de Gaussians. (GMM)

Nous allons representer $P(\vec{X})$ par :

$$P(\vec{X}) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(\vec{X}; \vec{\mu}_n, \mathbf{C}_n)$$

et

$$P(\vec{X} \mid \omega_k) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(\vec{X}; \vec{\mu}_{kn}, \mathbf{C}_{kn})$$

L'Analyse en Composantes Principales

L'analyse en composant principales d'un signal est une méthode de déterminer un sous-espace "optimale" pour la reconstruction. Elle est parfois utilisée pour définir un espace de caractéristiques pour la reconnaissance.

Soit une ensemble de M imagettes normalisés en position et taille composé de N pixels : $W_m(i, j)$

$W_m(i, j)$ pour tel que $i \in [1, I], j \in [1, J]$,
Nous allons les exprimer en vecteur de N = I x J : $W_m(n)$

ou $n = j*I + i$: $W_m(n) = W_m(i, j)$

Il serait possible d'utiliser le les pixels directement comme caractéristiques. Par exemple, pour des imagettes de taille 32 x 32, $W_m(n)$ est une vecteur de de 1K caractérisitique.

Mais il est préferable d'avoir un espace de dimension D < K < M (K est le nombre de classes, M est le nombre d'exemplaires). Dans ce cas, nous pouvons déterminer une sous-espace de dimension réduite par l'ACP.

On cherche une base orthogonale $\vec{\varphi}(n) = \{\varphi_d(n)\}\ \forall\ d = 1, 2, \ldots D$
pour représenter les X(n). telles que D << M.

Imagette moyenne $\mu(n) = \frac{1}{M} \sum_{m=1}^{M} W_m(n)$

Imagette moyenne zéro $V_m(n) = W_m(n) - \mu(n)$

Base orthogonal $\vec{\varphi}(n) = \{\varphi_d(n)\}$ avec $D \leq M$.

Vecteur $x_d = \langle V(n), \varphi_d(n) \rangle$

Image reconstruite : $\hat{W}(n) = \mu(n) + \sum_{d=1}^{D} x_d \varphi_d(n)$

Image de résidu : $R(n) = W(n) - \hat{W}(n)$

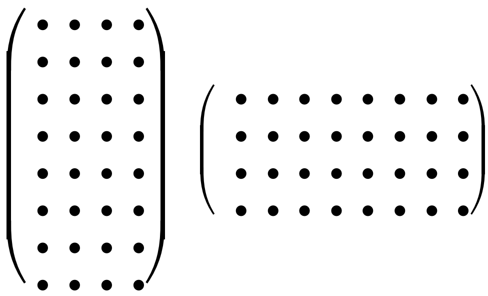
Energie de résidu : $\varepsilon_r^2 = \sum_{n=1}^{N} R(n)^2$

Les bases $\{\varphi_d(n)\}$ sont les vecteur propres du $\{V_m(n)\}$:
Sa covariance, **C**, est composée de N x N = N^2 termes

Soit le matrice **V** :

$$\mathbf{V} = \left[V_1(n)V_1(n)...V_m(N)\right] = \begin{bmatrix} V_1(1) & V_2(1) & ... & V_M(1) \\ V_1(2) & V_2(2) & ... & V_M(2) \\ .... & ... & ... & ... \\ V_1(n) & V_2(n) & ... & V_M(n) \end{bmatrix}$$

V est N par M. Chaque colonne de V est une image.

$\mathbf{C} \equiv V\,V^T =$ 

$$\mathbf{C} = E\{V_m(n)V_m(n)^T\}$$

Les termes de C sont les covariances pour des pairs de pixels i et j.

$$\sigma_{ij}^2 = \frac{1}{M} \sum_{m=1}^{M} (V_m(i) \cdot V_m(j))$$

Pour une imagette de taille N pixels, il y a N^2 termes dans la covariance.

Chaque coefficient, σ_{ij}^2, est une covariance de la pixel i et j pour l'ensemble de M images. Par exemple, pour une imagette de 32 x 32, le matrice $C= VV^T$ est de taille 1024 x 1024.

Les vecteurs propres de C sont des vecteurs orthogonales $\vec{\varphi}(n) = \{\varphi_m(n)\}$
tels que : $\varphi^T \mathbf{C} \varphi = \varphi^T \mathbf{V}\mathbf{V}^T \varphi = \Lambda$ où Λ_n est un matrice diagonal de N termes λ_n
des valeurs principales de C.

Chaque $\varphi_m(n)$ est un vecteur de direction. L'ensemble $\{\varphi_m(n)\}$ est orthogonal.

Une telle matrice de rotation est fournie par une d'analyse en composants principales. (Voir Recettes Numérique en C pour les procédures programmées).

$$(\varphi_m(n),\ \lambda_n) \leftarrow \text{PCA}(C).$$

Pour une imagette de 32 x 32, le matrice $C = V\,V^T$ est de taille $2^{10} \times 2^{10} = 2^{20}$.
Pour une image de 512 x 512, le matrice $C = V\,V^T$ est de taille $2^{18} \times 2^{18} = 2^{36}$.

Les plupart des procédures PCA exige que $N < 1024$.

Mais, il y a une astuce pour éviter le calcule d'une matrice de covariance de N x N coefficients.

Les algorithmes PCA fournissent les vecteurs propres $\{\varphi_m(n)\}$ dans l'ordre décroissante de λ_n.
Ors, pour une ensemble de M image M<<N , le rank de VV^T est de M.
Les premiers M vecteurs propres $\varphi_1(n)\ldots\varphi_m(n)$ sont les vecteurs propres de l'espace des images :

$C_m = V^TV$ est une MxM covariance pour les paires d'images.

C_m est de taille M^2, où M est le nombre d'images.
Les coefficients sont :

$$\sigma_{kl}^2 = \frac{1}{N}\sum_{n=1}^{N} V_k(n)V_l(n)$$

Les premiers M valeurs propres sont $\{R_m(n)\}$
Chaque coefficient est une produit de deux images !

$$V^TV = C_m = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$$

Pour M < 1024 on peut facilement calculer une matrice R de rotation tels que chaque colonne est un vecteur directeur orthogonal.

$$\mathbf{R}^T V^T V \mathbf{R} = \Lambda_m$$

On multiplie les deux cotés par R :

$$R R^T V^T V R = R \Lambda_m$$

On note que $R^T R = I$.

donc :

$$V^T V R = R \Lambda_m$$

Maintenant on multiplie par V

$$V V^T V R = V R \Lambda_m$$
$$= (VV^T)(V R) = (V R) \Lambda_m$$

Mais

$$\varphi^T (VV^T) \varphi = \Lambda_N$$

donc

$$(VV^T) \varphi = \varphi \Lambda_N$$

est le Λ_m est un MxM sous matrice de Λ_N

$\Lambda_m = \Lambda_N$ pour les premier m terms.

Donc

$$(VV^T) \varphi = \varphi \Lambda_n = (V R) \Lambda_m$$

pour les premier m terms

$$\varphi_m(n) = \sum_{l=1}^{M} V_l(n) R(l,m)$$

$$\begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{pmatrix}$$

Raison : Les mêmes images ont été utilisé pour φ et pour R.

Les premiers M vecteurs propres de $V^T V$ sont aussi les premiers M vecteurs propres de VV^T, le Rank de φ est le même que R, et λ_b sont les premiers M valeurs propre de λ_c et Donc, les vecteurs propres : $\varphi_d(n) = V\,R$ triés par λ_b

$$\varphi = VR$$

Chaque colonne est un vecteur dans une base orthogonale.

$$\vec{\varphi}(n) = \sum_{m-1}^{M} V_m(n)\, R(m,n)$$

$\vec{\varphi}(n)$ fournit une base orthogonal pour $W_m(n)$.

Une imagette (normalisée) peut être projetté sur $\vec{\varphi}(n)$

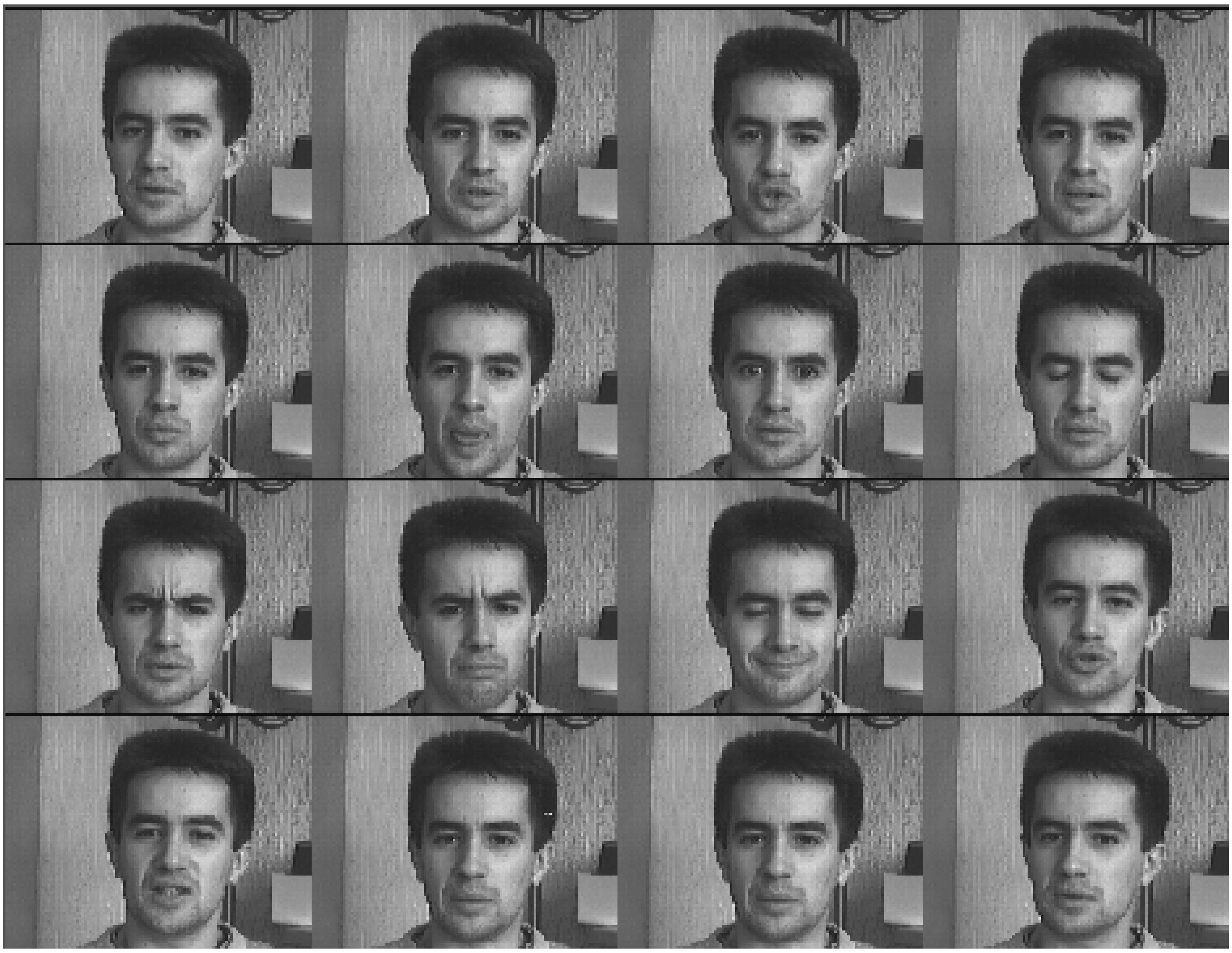
Nous pouvons même utiliser une sous ensemble de D composants.

$$x_d = \langle V, \varphi_d \rangle = \sum_{n=1}^{N} V(n)\varphi_d(n)$$

Les valeurs x_d sont un "code" qui représente V(n) pour la reconnaissance ou la reconstruction.

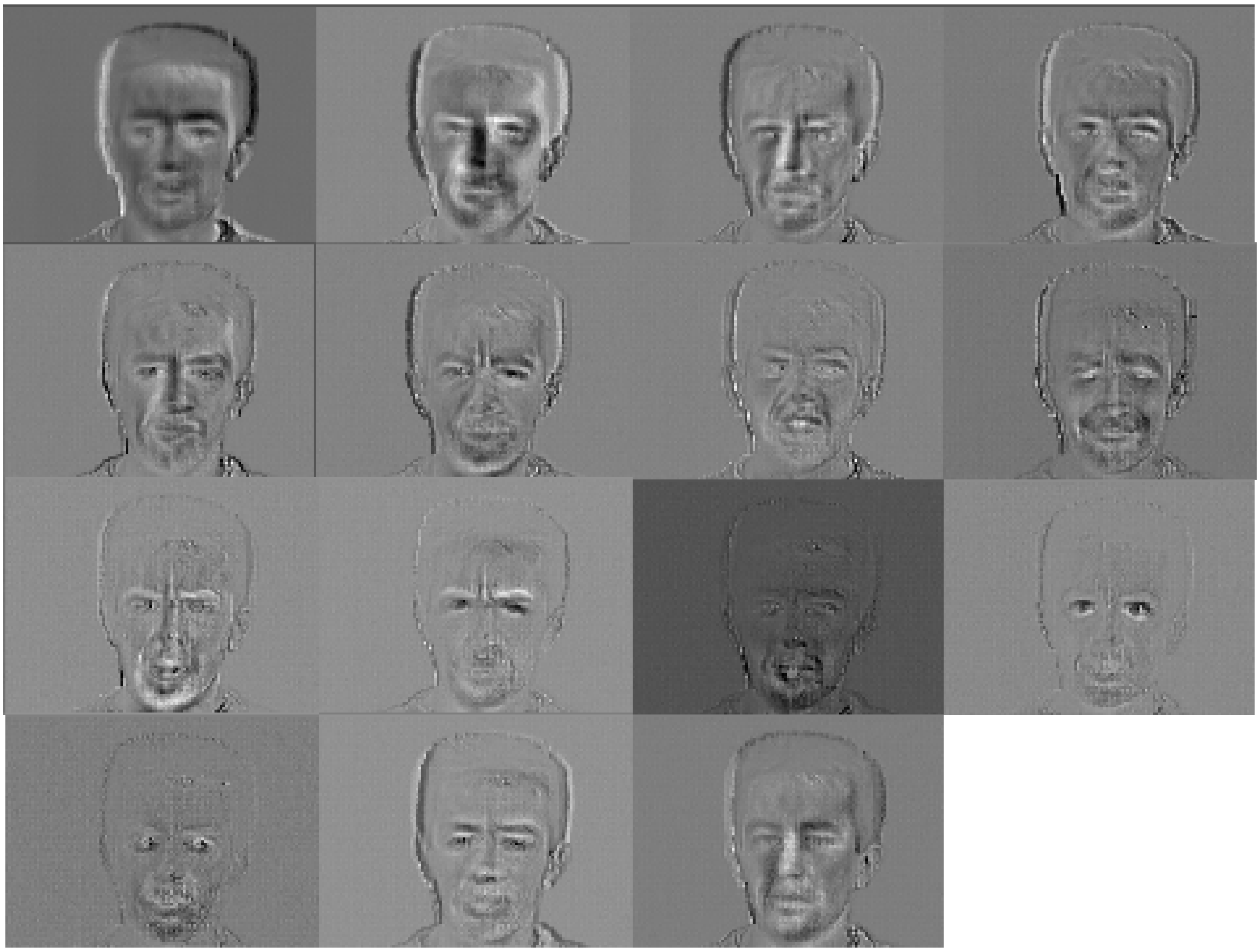
Exemple :

16 images prise au hasard dans une séquence de 2 minutes. (F. Bérard, 1995).

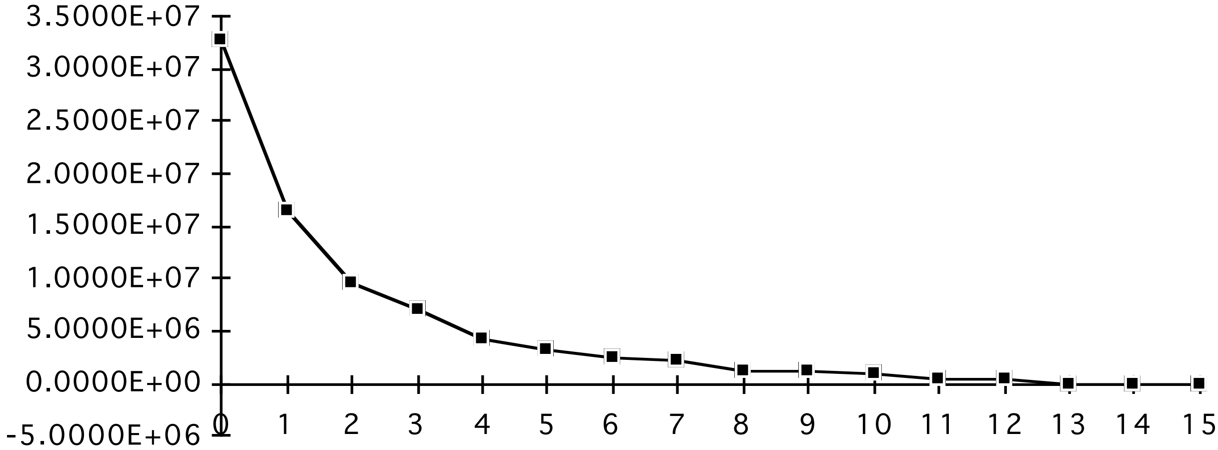


Average Image

Components Principales :



Eigen Values.



Reconstruction

Image Reconstructed image (120 bytes) Error Image.



Reconstruction (120 bytes)



Image Error

Mélange de Gaussiens.

Pour la reconnaissance, nous avons besoin :

$$p(\omega_k \mid \vec{X}) = \frac{p(\vec{X} \mid \omega_k)p(\omega_k)}{p(\vec{X})}$$

Si les imagettes de personne k sont issus d'une composition de "N" phénomènes, la densité $p(\vec{X} \mid \omega_k)$ prendra la forme d'une composition de N lois Normales. Pour les visages, il peut s'agit des variations dans l'angle de l'éclairage, l'angle de vue, ou les expressions.

Dans un tel cas, on peut approximer par une somme pondérée de densités Normales.

$$P(\vec{X}) = \sum_{n=1}^{N} \alpha_n N(\vec{X};\vec{\mu}_n, C_n) \quad \text{et} \quad P(\vec{X} \mid \omega_k) = \sum_{n}^{N} \alpha_n N(\vec{X};\vec{\mu}_{nk}, C_{nk})$$

De telle somme est connue par le terme "mélange de Gaussiens"
Pour chaque Gaussien il faut estimer trois paramètres :

$$\vec{v} = (\alpha_n, \vec{\mu}_n, C_n)$$

Pour une vecteur X de D dimensions
En total il y a $N(1+D+D^2)$ coefficient à estimer.

Pour simplifier, nous pouvons remplacer C_n par une matrice diagonal de variances : σ_{dn}^2

Ca donnerait : N(1+2D) parametres.

En tout cas, les parametres $\vec{v} = (\alpha_n, \vec{\mu}_n, C_n)$ sont les paramètres cachés des sources cachés des données.

L'algorithme d'Expectation Maximasation permettre d'estimer les parametres cachées.

Ebauche de l'Algorithme EM

On suppose chaque evennement E_m est issu d'un des N sources.
Nous allons construire une table de probabilités. h(m, n)

$h(m, n) = \Pr\{$l'événement E_m est issu de la source N$\}$

Les probabilités, $h(m, n)$ donne des facteurs de Mélange, α_n, ainsi que les μ_n,C_n.

Soit une ensemble ("training set") de M observations $\mathbf{T} = \{X_m\}$.
Fait une premiere estimation des paramètres, $\vec{\nu}^{(0)}$.
Ensuite l'algorithme vas les rafinner par altenance des étapes "E" et "M".

E: Faire une estimation des valeurs manquantes, $h(m, n)$, pour les événements.

$h(m, n)^{(i)} = p(h_m | X_1, X_2, ..., X_M, \vec{\nu}^{(i)})$ pour chaque terme "n".

$$h(m, n)^{(i)} = \frac{\alpha_n^{(i)} \mathcal{N}(X_m; \mu_n^{(i)}, \sigma_n^{(i)})}{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(X_m; \mu_j^{(i)}, \sigma_j^{(i)})}$$

M: Recalculer $\vec{\nu}^{(i+1)}$ avec $p(h_m | X_1, X_2, ..., X_M, \vec{\nu}^{(i)})$

$$S_n^{(i+1)} = \sum_{m=1}^{M} h(m,n)^{(i)}$$

$$\alpha_n^{(i+1)} = \frac{1}{M} S_n^{(i+1)} = \frac{1}{M} \sum_{m=1}^{M} h(m,n)^{(i)}$$

$$\mu_n^{(i+1)} = \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} X_m$$

$$\sigma^2{}_n^{(i+1)} = \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} (X_m - \mu_n^{(i+1)})^2$$

Pour la dérivation l'algorithme EM, il faut introduire le concept de "likelihood" (vraisemblance) .

L'Algorithme EM (Expectation-Maximization)

L'algorithme EM s'applique à l’estimation de données cachés.
Il est utilisé notamment pour l'estimation des Modèles de Markov Cachées (HMM's) et pour l'estimation des Mélanges de Gaussiens.

Pour un mélange de Gaussiens, les variables cachées sont les sources des événements. On suppose que chaque événement est produit par un des N sources.
Pour chaque événement E_m On définit la variable "cachée" h_m

h_m = N si l'événement E_m (avec caractéristique X_m) est issu de la source N.
Nous cherchons à estimer

$$P(\vec{X}) = \sum_{n}^{N} \alpha_n \mathrm{N}(\vec{X};\vec{\mu}_n,C_n) \quad \text{tels que} \quad \sum_{n=1}^{N} \alpha_n = 1.$$

ou

$$\vec{v} = (\alpha_n,\vec{\mu}_n,C_n)$$

Il faut estimer $\vec{v} = (\alpha_n,\vec{\mu}_n,C_n)$

Le likelihood de $\{X_m\}$ est $L(\vec{v} \mid \{X_m\}) = \prod_{m=1}^{M} P(X_m \mid \vec{v})$

Le Log-likelihood est

$$= \mathrm{Log}\,(L(\vec{v} \mid X_1, X_2, ..., X_M)) = Log\left(\prod_{m=1}^{M} P(X_m \mid \vec{v})\right)$$

$$= \sum_{m=1}^{M} \mathrm{Log}\,\{p(X_m \mid \vec{v}\,)\,)\}$$

$$= \sum_{m=1}^{M} \mathrm{Log}\left(\sum_{n=1}^{N} \alpha_n N(\vec{X};\vec{\mu}_n,C_n)\right)$$

Il faut $\hat{v} = \arg\!-\!\max_{v}\left\{L(\vec{v} \mid \{X_m\})\right\}$

Il n'y pas de solution analytique, parce qu'il n'y est pas une dérivé pour la logarithme de la somme.

Soit **T =** $\{X_m\} \cup h(m,n)$ le donnée Complèt.

On vas maximiser une mesure de qualité $Q(\vec{v}^{(i)})$ définit par une esperence conditionnel

$$Q(\vec{v}^{(i)}) = E\{\, l(\vec{v}^{(i)}) \mid \mathbf{T})\} = \text{Log}\ \{ \prod_{m=1}^{M} p(X_m | \overset{\varnothing}{v})\ \}$$

L'etape E

Pour chaque événement E_m avec caractéristique X_m,
On suppose qu'il manque l'information: h_m

$h_m = n$, la source de l'événement E_m.

On ne connaît pas h_m, mais on peut estimer les probabilités $P(h_m = n)$.
par une table h(m,n).

Pour chaque X_m, et son source caché h_m

$$p(h_m, X_m \mid \vec{v}) = p(h_m \mid X_m, \vec{v})\, p(X_m \mid \vec{v})$$

donc

$$p(h_m \mid X_m, \vec{v}) = \frac{p(h_m, X_m \mid \vec{v})}{p(X_m \mid \vec{v})}$$

d'où

$$p(h_m = n, X_m \mid \vec{v}) = \alpha_n \mathcal{N}(X_m; \mu_n, \sigma_n)$$

$$p(X_m \mid \vec{v}) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(x_m; \mu_n, \sigma_n)$$

Donc

$$p(h_m=n \mid X_1, X_2, ..., X_M, \vec{\nu}) = \frac{\alpha_n \mathcal{N}(X_m; \mu_n, \sigma_n)}{\sum_{j=1}^{N} \alpha_j \mathcal{N}(X_m; \mu_j, \sigma_j)}$$

Donc pour chaque itération (i) le premier étape E est :

$$h(m, n)^{(i)} = \frac{\alpha_n^{(i)} \mathcal{N}(X_m; \mu_n^{(i)}, \sigma_n^{(i)})}{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(X_m; \mu_j^{(i)}, \sigma_j^{(i)})}$$

L'etape M

Pour le deuxième étape, M, nous allons maximiser le "likelihood" Log{p($X_m \mid \vec{\nu}$)}
On défini la "Expected Complete Data Log Likelihood" pour **T**. = {X_m} U h(m,n)

$$\Delta Q = Q(\vec{\nu}^{(i)}) - Q(\vec{\nu}^{(i-1)})$$

Pour chaque cycle dans l'itération nous allons chercher :

$$\vec{\nu}^{(i+1)} = \underset{\nu}{\operatorname{argmax}} \{Q(\vec{\nu} \mid \vec{\nu}^{(i)})\}$$

Ce maximum est donné par :

$$S_n^{(i+1)} = \sum_{m=1}^{M} p(h_m=n \mid X_m, \vec{\nu}^{(i)})) = \sum_{m=1}^{M} h(m, n)$$

$$\alpha_n^{(i+1)} = \frac{1}{M} \sum_{m=1}^{M} P(h_m = n \mid X_m, \vec{\nu}^{(i)}) = \frac{1}{M} S_n^{(i+1)}$$

$$\mu_n^{(i+1)} = \frac{1}{S_n^{(i)}} \sum_{m=1}^{M} p(h_m=n \mid X_m, \vec{\nu}^{(i)}) X_m = \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,.n) X_m$$

$$\sigma^2{}_n^{(i+1)} = \frac{1}{S_n^{(i)}} \sum_{m=1}^{M} p(h_m=n \mid X_m, \vec{\nu}^{(i)}) \ (X_m - \mu_n^{(i+1)})^2.$$

$$= \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n) \ (X_m - \mu_n^{(i+1)})^2$$

Avec notre table de probabilités $h(m, n)$

$$h(m, n) = P(h_m = n \mid X_m, \vec{\nu}^{(i)})$$

E (Expectation) :

$$\boxed{h(m, n)^{(i)} := \frac{\alpha_n^{(i)} \mathcal{N}(X_m; \mu_n^{(i)}, \sigma_n^{(i)})}{\sum_{j=1}^{N} \alpha_j^{(i)} \mathcal{N}(X_m; \mu_j^{(i)}, \sigma_j^{(i)})}}$$

M: (Maximisation)

$$\boxed{S_n^{(i+1)} := \sum_{m=1}^{M} h(m, n)^{(i)}}$$

$$\boxed{\alpha_n^{(i+1)} := \frac{1}{M} S_n^{(i+1)}}$$

$$\boxed{\mu_n^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,.n)^{(i)} X_m}$$

$$\boxed{\sigma^2{}_n^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} (X_m - \mu_n^{(i+1)})^2}$$

Dans le cas Multivariate (D> 1) la covariance C est composée de σ_{jk}^2:

$$\boxed{\sigma^2{}_{jkn}^{(i+1)} := \frac{1}{S_n^{(i+1)}} \sum_{m=1}^{M} h(m,n)^{(i)} (X_{jm} - \mu_{jn}^{(i+1)})(X_{km} - \mu_{kn}^{(i+1)})}$$