

**Séance 4**  
**Détermination des paramètres d'un mélange de lois**  
**Algorithme EM**

---

Au cours de ce TP on demande de programmer l'algorithme EM et de tester son comportement sur des données sauvegardées dans le fichier Données.mat. Ces données représentent une population de 1000 individus décrits par deux paramètres.

On rappelle les formules de ré-estimation des paramètres du modèle:

$$P(j) = \frac{1}{N} \sum_{i=1}^N P(j/x_i, \Theta^g)$$

$$\mu_j = \frac{\sum_{i=1}^N x_i P(j/x_i, \Theta^g)}{\sum_{i=1}^N P(j/x_i, \Theta^g)}$$

$$\Sigma_j = \frac{\sum_{i=1}^N P(j/x_i, \Theta^g)(x_i - \mu_j)(x_i - \mu_j)^t}{\sum_{i=1}^N P(j/x_i, \Theta^g)}$$

**1- Programmation de l'algorithme EM**

On demande de programmer les formules de ré-estimation de l'algorithme EM pour un modèle diagonal. On s'aidera des fonctions suivantes disponibles dans la boîte à outils Netlab :

**modele = gmm(dimension, nb\_composantes,'diag')**

Crée un modèle diagonal. Pour affecter à un modèle les paramètres souhaités, utiliser les instructions suivantes dans le cas d'un modèle diagonal à 3 composantes en 2 dimensions:

melange.centres = [1 1; 2 2; 3 3];  
 melange.priors = [0.333, 0.333 0.333];  
 melange.covars = [0.5 0.5; 0.5 0.5; 0.5 0.5];

**[post, vrais]=gmmpost(modele, donnees);**

calcule la vraisemblance de chaque individu pour chaque composante, soit :

$$f(x/j) = \frac{1}{(2\pi)^{p/2} \prod_i \sigma_{ji}} \exp\left(-\sum_{i=1}^p \frac{(x_i - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

*(expression pour un modèle diagonal)*

ainsi que les probabilités à posteriori de chaque composante pour chaque individus, soit :

$$P(j/x) = \frac{f(x/j)P(j)}{\sum_j f(x/j)P(j)}$$

**prob = gmmprob(mix, x) ;**

Calcule la vraisemblance du modèle pour chaque individus, soit :

$$f(x) = \sum_j f(x/j)P(j)$$

## 2- Etude du comportement de l'algorithme

Etudier le comportement de l'algorithme sur les données en 2 dimensions sauvegardées dans le fichier Donnees.mat. On chargera ces données et les visualisera par les instructions :

```
>>load Donnees ;
>>plot(donnees(:,1),donnees(:,2),'yo');
```

2.1- A partir du modèle initial ci-dessous, relever l'évolution du critère log-vraisemblance au cours des itérations de l'algorithme. Combien d'itérations sont nécessaires pour que ce critères ait un accroissement inférieur à 1% entre 2 itérations successives ?

```
melange.centres = [1 1; 2 2; 3 3];
melange.priors = [0.333, 0.333 0.333];
melange.covars = [0.5 0.5; 0.5 0.5; 0.5 0.5];
```

2.2- Etudier le comportement de l'algorithme en fonction des conditions initiales. Pour cela on comparera les résultats obtenus pour 3 apprentissages différents menés à partir des conditions initiales suivantes :

```
.centres = [1 1; 2 2; 3 3];
.priors = [0.333, 0.333 0.333];
.covars = [0.5 0.5; 0.5 0.5; 0.5 0.5];
```

```
.centres = [1 1; 1 1; 1 1];
.priors = [0.0, 0.5, 0.5];
.covars = [1 1 ; 1 1 ; 1 1];
```

```
.centres = [5 5 ; 3 4; 1 2];
.priors = [0.2, 0.4, 0.4];
.covars = [2 2; 1 1; 1 2];
```