TP5 – Segmentation par l'algorithme EM

Le TP4 de TAV vous a permis de comparer trois méthodes d'estimation des paramètres d'une ellipse :

- La première méthode, qui utilise la définition géométrique d'une ellipse (« méthode du jardinier »), consiste à optimiser une fonction de moindres carrés non linéaires par tirages aléatoires. Cette méthode est lente et donne des résultats relativement imprécis.
- La deuxième et la troisième méthodes, qui utilisent l'équation cartésienne d'une ellipse, consistent à optimiser une fonction de moindres carrés linéaires par des outils d'optimisation différentiable. Elles constituent deux variantes de la même méthode, selon que la contrainte sur les paramètres est linéaire ou non, et s'avèrent toutes les deux à peu près aussi rapides et aussi précises.

Au vu de ce bilan, vous êtes en droit de vous questionner sur l'intérêt réel de la première méthode d'estimation. Au travers d'un problème apparemment très proche, à savoir l'estimation simultanée des paramètres de deux ellipses, vous allez voir que ces méthodes peuvent être complémentaires.

Lancez le script `donnees_simulees.m`, qui affiche un nuage de n points P_i situés au voisinage de deux ellipses tirées aléatoirement. Si ces données étaient partitionnées, l'estimation des paramètres des deux ellipses ne poserait pas de difficulté, mais cela n'est justement pas le cas. On peut néanmoins modéliser la densité de probabilité des points par un mélange de deux lois, à hauteur de proportions π_1 et π_2 telles que $\pi_1 + \pi_2 = 1$:

$$f_p(P_i) \approx \frac{\pi_1}{\sigma_1\sqrt{2\pi}} \exp\left\{-\frac{e_1(P_i)^2}{2\,\sigma_1^2}\right\} + \frac{\pi_2}{\sigma_2\sqrt{2\pi}} \exp\left\{-\frac{e_2(P_i)^2}{2\,\sigma_2^2}\right\} \tag{1}$$

où :

- Les écarts sont définis par $e_1(P_i) = P_iF_{1,1} + P_iF_{2,1} - 2a_1$ et $e_2(P_i) = P_iF_{1,2} + P_iF_{2,2} - 2a_2$, où les points $F_{1,k}$ et $F_{2,k}$ désignent les deux foyers et a_k le grand axe de l'ellipse numéro k, $k = 1, 2$.
- La liste $p = (F_{1,1}, F_{2,1}, F_{1,2}, F_{2,2}, a_1, a_2, \sigma_1, \sigma_2, \pi_1)$ des neuf paramètres du modèle correspond à treize degrés de liberté. La proportion π_2 ne fait pas partie de cette liste car $\pi_2 = 1 - \pi_1$.
- L'égalité approchée $\approx$ signifie que les lois normales sont tronquées (cf. TP4 de TAV).

L'estimation par maximum de vraisemblance consiste à chercher la valeur $\widehat{p}$ qui maximise la log-vraisemblance :

$$\widehat{p} = \underset{p\in(\mathbb{R}^2)^4\times(\mathbb{R}^+)^4\times[0,1]}{\arg\max} \left\{\ln\left[\prod_{i=1}^{n} f_p(P_i)\right]\right\} = \underset{p\in(\mathbb{R}^2)^4\times(\mathbb{R}^+)^4\times[0,1]}{\arg\max} \left\{\sum_{i=1}^{n} \ln\left[f_p(P_i)\right]\right\} \tag{2}$$

Le problème (2) est à première vue très difficile à résoudre, car il s'agit d'optimisation non convexe dans $\mathbb{R}^{13}$, mais on peut le résoudre en combinant astucieusement les différentes méthodes d'estimation du TP4 de TAV. En effet, si la maximisation de la vraisemblance est une méthode d'estimation peu précise, elle permet d'estimer les paramètres de n'importe quelle loi, y compris d'un mélange de lois tel que (1). De plus, une fois le mélange optimal trouvé, on peut en déduire une partition des données en deux classes, en procédant comme suit :

$$\widehat{k}(P_i) = \underset{k=1,2}{\arg\max} \left\{\frac{\pi_k}{\sigma_k} \exp\left\{-\frac{e_k(P_i)^2}{2\,\sigma_k^2}\right\}\right\} \tag{3}$$

Lancez le script `exercice_0.m`, qui résout le problème (2) en trois étapes (pour faciliter la résolution, les paramètres $(\sigma_1, \sigma_2, \pi_1)$ sont fixés à leurs valeurs réelles) :

1. Maximisation de la vraisemblance des données par tirages aléatoires de $(F_{1,1}, F_{2,1}, F_{1,2}, F_{2,2}, a_1, a_2)$.
2. Partitionnement des données P_i en deux ensembles, comme indiqué en (3).
3. Pour chaque ensemble, estimation des paramètres de l'ellipse (cf. TP4 de TAV).

Lancez plusieurs exécutions de ce script : vous constatez que les résultats sont très variables.

Exercice 1 : algorithme EM (Espérance-Maximisation)

La deuxième figure affichée par le script `exercice_0.m` indique quel mélange de lois maximise la vraisemblance, parmi un ensemble fini de lois de mélange tirées aléatoirement. Or, la probabilité de « tomber pile » sur les paramètres optimaux $\widehat{p}$ est quasiment nulle. En revanche, les ellipses de la figure de droite sont généralement plus proches des données que les ellipses de la figure du milieu. Une idée consiste donc à *boucler*, c'est-à-dire à utiliser les ellipses de la figure de droite pour définir un nouveau mélange de lois, et à mettre à jour la partition.

L'algorithme EM, qui est très général, s'inspire de cette idée, à ceci près qu'il n'effectue pas une partition stricte des données. Les paramètres à estimer sont initialisés par le script `exercice_0.m`. L'algorithme EM répète en boucle les deux étapes suivantes :

- **Étape E** – Calcul de la probabilité d'appartenance $\mathcal{P}_{i,k}$ de la donnée P_i, $i \in [1,n]$, à la classe $k = 1, 2$:

$$\mathcal{P}_{i,k} = \frac{\frac{\pi_k}{\sigma_k} \exp\left\{-\frac{e_k(P_i)^2}{2\,\sigma_k^2}\right\}}{\frac{\pi_1}{\sigma_1} \exp\left\{-\frac{e_1(P_i)^2}{2\,\sigma_1^2}\right\} + \frac{\pi_2}{\sigma_2} \exp\left\{-\frac{e_2(P_i)^2}{2\,\sigma_2^2}\right\}}$$

- **Étape M** – Mise à jour des proportions du mélange $\pi_k = \frac{1}{n} \sum_{i=1}^{n} \mathcal{P}_{i,k}$, et estimation des paramètres des ellipses par résolution des systèmes de *moindres carrés pondérés* suivants, pour $k = 1, 2$:

$$\mathcal{P}_{i,k} \left(x_i^2\,\alpha + x_i\,y_i\,\beta + y_i^2\,\gamma + x_i\,\delta + y_i\,\epsilon + \phi\right) = 0\,, \qquad i \in [1,n]$$

Écrivez la fonction `probabilites` et une variante de la fonction `estimation`, de nom `estimation_ponderee`, de manière à ce que le script `exercice_1.m` mette en œuvre cet algorithme. Il est conseillé de résoudre le système de n équations $\alpha\,x_i^2 + \beta\,x_i\,y_i + \gamma\,y_i^2 + \delta\,x_i + \epsilon\,y_i + \phi = 0$, $i \in [1,n]$, sous la contrainte non linéaire $\|\mathbf{X}\| = 1$, où $\mathbf{X} = [\alpha, \beta, \gamma, \delta, \epsilon, \phi]^\top$, comme cela a été fat dans l'exercice 3 du TP4 de TAV.

Exercice 2 : segmentation par l'algorithme EM

Lancez le script `donnees_reelles.m`. Chaque point du nuage affiché correspond à la description statistique locale d'un pixel : l'abscisse est égale au niveau de gris moyen, l'ordonnée à sa variance. Une méthode de segmentation par classification (*classification non supervisée*) consiste à utiliser comme modèle, pour les couples constitués de la moyenne et de la variance du niveau de gris, un mélange de deux gaussiennes *bidimensionnelles* :

$$f_p(\mathbf{x}) = \frac{\pi_1}{2\pi\sqrt{\det\Sigma_1}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)\Sigma_1^{-1}(\mathbf{x}-\mu_1)^\top\right\} + \frac{\pi_2}{2\pi\sqrt{\det\Sigma_2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_2)\Sigma_2^{-1}(\mathbf{x}-\mu_2)^\top\right\} \quad (4)$$

où :

- $\mathbf{x} = [x, y]^\top$ est le vecteur des coordonnées (moyenne et variance) d'un point du nuage ;
- μ_k désigne la moyenne et Σ_k la matrice de variance/covariance de la loi normale numéro $k = 1, 2$.

Complétez le script `exercice_2.m` de manière à effectuer une segmentation « fond-forme » par une adaptation de l'algorithme EM, qui est très général, au mélange de gaussiennes (4). Remarquez que le script `exercice_2.m`, plutôt que de lancer le script `donnees_reelles.m`, lit les données enregistrées dans le fichier `donnees_reelles.mat` (`load donnees_reelles`). Cela évite de recalculer les données à chaque exécution.

Conseils de programmation :

- Comme dans l'exercice 1, commencez par estimer les paramètres du mélange de gaussiennes en maximisant la vraisemblance.

- À l'intérieur de la boucle, les matrices de variance/covariance Σ_k, pour $k = 1, 2$, sont symétriques, de même que leurs inverses $\Sigma_k^{-1} = \begin{bmatrix} a_k & b_k \\ b_k & c_k \end{bmatrix}$. La loi (4) se réécrit donc :

$$f_p(\mathbf{x}) = \sum_{k=1,2} \frac{\pi_k}{2\pi\sqrt{\det\Sigma_k}} \exp\left\{-\frac{a_k(x-x_{\mu_k})^2 + 2\,b_k(x-x_{\mu_k})(y-y_{\mu_k}) + c_k(y-y_{\mu_k})^2}{2}\right\}$$