TP2 – Segmentation par l'algorithme EM

Le TP4 de TAV vous a permis de comparer deux 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 méthode, qui utilise l'équation cartésienne d'une ellipse, consiste à optimiser une fonction de moindres carrés linéaires par des outils d'optimisation différentiable. Trois variantes de cette méthode ont été testées, qui s'avèrent toutes les trois 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 sont en fait complémentaires.

Lancez le script `donnees.m`, qui affiche un nuage de n points P_i issus de la superposition de deux ellipses tirées aléatoirement. Si ces données étaient partitionnées, l'estimation des paramètres des deux ellipses ne poserait pas de difficulté particulière, 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 $\pi_1 + \pi_2 = 1$:

$$f_p(P_i) \approx \frac{\pi_1}{\sqrt{2\pi}\,\sigma_1}\,\exp\left\{-\frac{E_1(P_i)^2}{2\,\sigma_1^2}\right\} + \frac{\pi_2}{\sqrt{2\pi}\,\sigma_2}\,\exp\left\{-\frac{E_2(P_i)^2}{2\,\sigma_2^2}\right\}, \qquad i \in [1,n] \tag{1}$$

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 désigne 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 paramètres du modèle contient neuf paramètres correspondant 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 $\hat{p}$ qui maximise la log-vraisemblance :

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

Le problème (2) semble très difficile à résoudre, car il s'agit d'optimisation non convexe dans $\mathbb{R}^{13}$, mais il est possible de le résoudre en combinant astucieusement les deux méthodes du TP4 de TAV.

Exercice 1 : résolution du problème (2) en trois étapes

Comme nous l'avons déjà dit, il suffirait de partitionner correctement les données pour que la résolution du problème (2) devienne immédiate. Or, comme les n points P_i ne sont pas linéairement séparables, une méthode de partitionnement (*clustering*, cf. TP4 de TIM) ne serait d'aucun secours. En revanche, 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). Qui plus est, une fois le mélange optimal trouvé, on peut en déduire une partition des données en deux classes, en procédant ainsi (le facteur $1/\sqrt{2\pi}$ a été omis) :

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

En vous inspirant de ce principe, et en supposant pour simplifier que les paramètres $(\sigma_1, \sigma_2, \pi_1)$ sont fixés à leurs valeurs réelles, complétez le script `exercice_1.m` de manière à résoudre le problème (2) en trois étapes :

1. Maximiser la vraisemblance des données en tirant des valeurs aléatoires de $(F_{1,1}, F_{2,1}, F_{1,2}, F_{2,2}, a_1, a_2)$.
2. Partitionner les données P_i en deux ensembles, comme indiqué en (3).
3. Pour chaque ensemble, appeler la fonction `estimation_ellipse` du TP4 de TAV.

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

Exercice 2 : algorithme EM (Espérance-Maximisation)

L'observation des résultats du script `exercice_1.m` suggère une idée toute simple pour améliorer les résultats. En effet, la figure du milieu indique quel mélange de lois maximise la vraisemblance, parmi un ensemble fini de lois de mélange tirées aléatoirement. La probabilité de « tomber pile » sur les paramètres optimaux $\hat{p}$ 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. L'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 des données.

L'algorithme EM s'inspire de cette idée, à ceci près qu'il n'effectue pas une partition stricte des données. Il répète en boucle les deux étapes suivantes (les paramètres à estimer sont initialisés par tirages aléatoires) :

- **Étape E** – Calculer 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** – Mettre à jour les proportions du mélange $\pi_k = 1/n \sum_{i=1}^{n} \mathcal{P}_{i,k}$ et les paramètres des ellipses, en résolvant aux moindres carrés le système des équations suivantes, où $i \in [1, n]$ et $k = 1, 2$:

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

Complétez le script `exercice_2.m` de façon à mettre en œuvre cette méthode.

Exercice 3 : segmentation par l'algorithme EM

Lancez le script `donnees_reelles.m`. Chaque point du nuage 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 très simple de segmentation par *classification semi-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_3.m` en adaptant l'algorithme EM au mélange de gaussiennes (4), qui effectue une segmentation « fond-forme », et en suivant les conseils suivants :

- Dans l'étape initiale de maximisation de la log-vraisemblance par tirages aléatoires, les matrices Σ_1 et Σ_2 sont supposées égales à $\sigma^2\,\mathbf{I}_2$, où $\mathbf{I}_2$ est la matrice identité dans $\mathbb{R}^2$ et σ est une valeur fixée. Dans ces conditions, la loi (4) se réécrit :

$$f_p(\mathbf{x}) = \frac{\pi_1}{2\pi\,\sigma^2} \exp\left\{-\frac{(x-x_{\mu_1})^2 + (y-y_{\mu_1})^2}{2\,\sigma^2}\right\} + \frac{\pi_2}{2\pi\,\sigma^2} \exp\left\{-\frac{(x-x_{\mu_2})^2 + (y-y_{\mu_2})^2}{2\,\sigma^2}\right\}$$

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

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

Question facultative. Testez le script `exercice_3.m` sur l'image `image.bmp` du TP1. Comme il y a plus de deux classes dans cette image, le résultat est forcément décevant. Faites une copie du script `exercice_3.m`, de nom `exercice_3_bis.m`, de manière à généraliser cette méthode de segmentation par classification semi-supervisée à un nombre de classes N quelconque (en l'occurrence, $N = 6$ pour `image.bmp`).