

CR 09 Hidden Markov models for time series analysis

Lab Session – **EM algorithm in mixture model, with application to image segmentation**

S. Derrode

L'idée de cette séance de TP est de segmenter une image, c'est à dire d'affecter à chacun de ses pixels un numéro de classe, parmi $1, \dots, K$. Dans une premier, nous travaillerons sur une image de cible à deux classes ($K=2$). Pour être en mesure de mesurer le taux d'erreur, nous bruyterons nous-même l'image au préalable. Ensuite, en faisant mine de ne pas connaître les paramètres du bruit, nous essaierons de les estimer à l'aide de l'algorithme itératif EM. Enfin, sur la base des paramètres estimés, nous appliquerons la décision bayésienne et finirons par comparer l'image originale avec l'image segmentée pour estimer le taux d'erreur de classification.

Le TP consiste à compléter certaines fonctions d'un programme écrit en Python, et ensuite à examiner les courbes de convergence de l'algorithme EM en fonction des paramètres de bruitage de l'image.

Préambule

Téléchargez et décompressez le fichier zip disponible à l'adresse

<http://perso.ec-lyon.fr/derrode.stephane/Teaching.php>

Celui contient

- Cet énoncé.
- Dans le sous-répertoire « sources », l'image d'une cible noire et blanche en deux dimensions : 64x64 et 128x128 (ci-contre).
- Un programme pour ajouter du bruit à l'image de cible : « BruitagelImage.py »
- Le squelette du programme principal « LabSession_MM_EM.py », ainsi que le fichier « func.py » contenant des fonctions partiellement codées, à compléter durant la session.

**Bruitage d'une image**

- Exécutez le programme « BruitagelImage.py » et observez les images générées par le programme dans le répertoire « results ».
- Analyser le code source pour en comprendre les étapes. Jouez avec les paramètres du bruit et observez l'impact sur les images du répertoire « results ».
- **Question** : Dans le mélange, comment sont fixées les probabilités a priori ? Modifier le programme pour en afficher les valeurs.

Segmentation d'images : implémentation des algorithmes

Analyser le squelette de programme du fichier « LabSession_MM_EM.py ». Celui-ci fait appel à des fonctions qui sont toutes décrites dans le fichier « func.py ». Dans ce dernier fichier, deux fonctions sont vides : à vous de les compléter !

Travail à effectuer :

- Examinez la structure du programme. Vous verrez 5 phases :
 1. La lecture des images.
 2. L'initialisation des paramètres du mélange.
 3. L'estimation des paramètres du mélange par l'algorithme EM.
 4. La classification de l'image selon la décision bayésienne.
 5. Le dessin des courbes.

A chaque itération, nous prenons soin de sauvegarder les paramètres, ainsi que l'erreur de classification, pour dessiner leur évolution lors de la 5^{ème} phase.

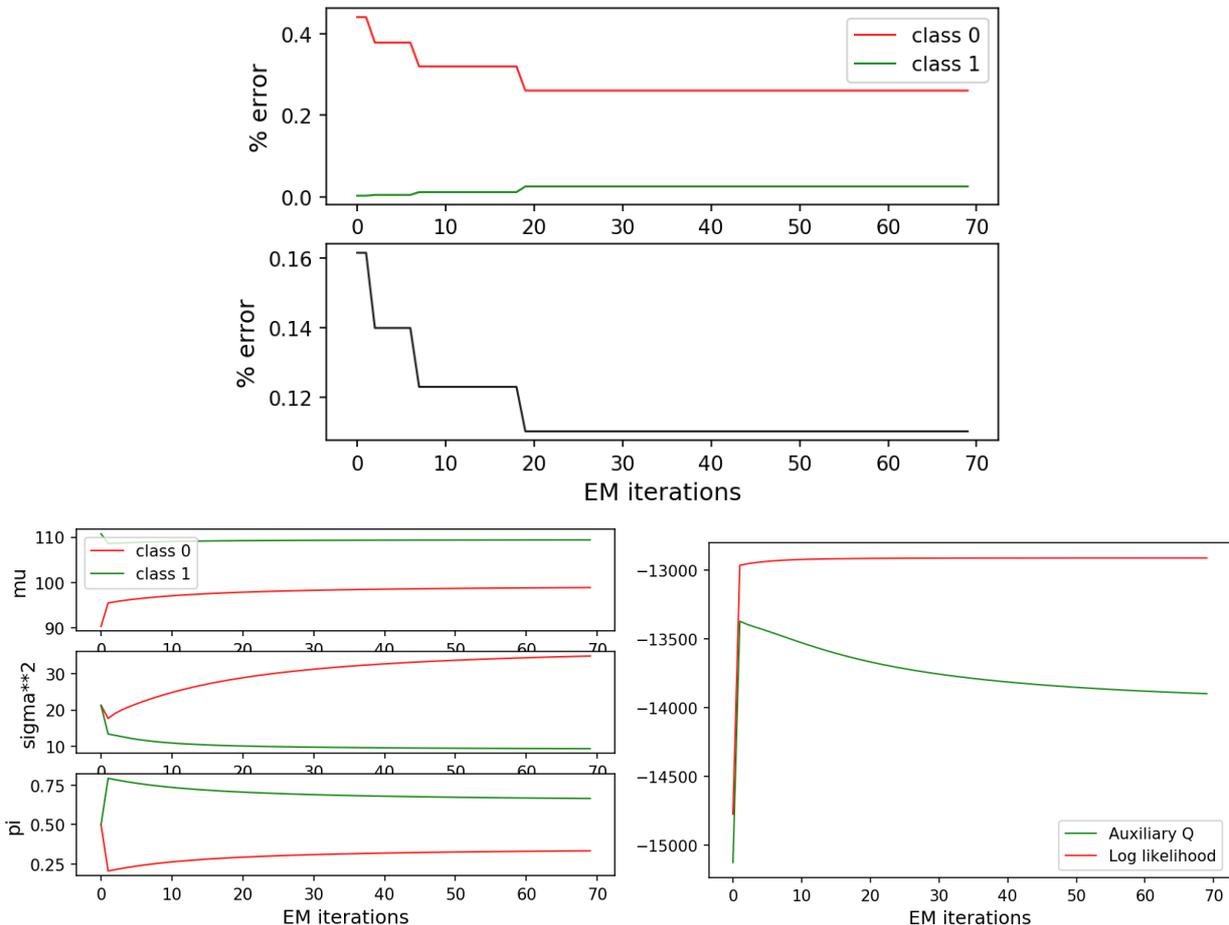
- Complétez la fonction d'initialisation des paramètres avec un algorithme à définir.
- Complétez la fonction qui itère EM avec les formules de re-estimation vues en cours.
- Si vous avez le temps, ajoutez une fonction de calcul de la vraisemblance et de la fonction auxiliaire Q , à toutes les itérations de EM. En vous inspirant des fonctions « DrawCurvesParam.py » et « DrawCurvesError.py », ajoutez votre propre fonction pour sauvegarder les nouvelles figures sur le disque dur.

Truc :

- Travaillez avec la petite image 64x64, sinon vos tentatives vont prendre du temps !
- Réduisez à 3 ou 4 le nombre d'itérations de EM, le temps de réaliser vos tests lors de la phase de dev.

Il faut entre 100 et 150 itérations pour obtenir une bonne estimation des paramètres du mélange prévu dans le programme de bruitage de l'image. Comme vous pouvez le constater dans le programme principal.

Voici les allures des courbes que vous devriez obtenir, après 130 itérations de EM sur le mélange défini dans « BruitageImage.py ».



$$\xi_1 = 0.261,$$

$$\xi_2 = 0.025$$

$$\xi = 0.11$$