Institut Mines-Télécom

TELECOM SudParis

INSTITUT POLYTECHNIQUE DE PARIS

Télécom SudParis - Diplôme d'Ingénieur

Méthodes statistiques pour la segmentation d'images - Partie 2

MAT5011: Modèles de Markov en Signal et Image

Réalisé pour le 5 novembre 2021

par

Lorenzo HERMEZ

Coordinateurs : Wojciech PIECZYNSKI & Clément FERNANDES

Table des matières

Introduction

Lors du TP précédent, la segmentation est rendue possible par la possibilité d'estimer, d'une part les paramètres du modèle, (p_1, p_2), grâce à la connaissance de l'image de classes mais surtout par le fait que l'on dispose de la connaissance des paramètres du bruit, $(m_1, \sigma_1, m_2, \sigma_2)$, puisqu'ils ont servi au bruitage de l'image de synthèse à segmenter...
La connaissance de ces paramètres est assurée grâce à notre bruitage artificiel mais ce n'est quasiment jamais le cas dans le pratique. Ainsi, l'estimation de ces paramètres est primordiale et c'est l'objectif de ce TP. C'est pour cela qu'il faut estimer les paramètres du modèle ainsi que ceux du bruit avant de faire la segmentation. Il va donc falloir estimer la loi du couple (X, Y) ce qui serait un travail classique d'inférence statistique si l'on disposait d'une observation (x, y), du couple (X, Y) :

— $p_i = \frac{1}{|S|} \sum_{s \in S} \mathbb{1}_{[x_s = \omega_i]}$

— $m_i = \frac{1}{\sum_{s \in S} \mathbb{1}_{[x_s = \omega_i]}} \sum_{s \in S} y_s \mathbb{1}_{[x_s = \omega_i]}$

— $\sigma_i^2 = \frac{1}{\sum_{s \in S} (\mathbb{1}_{[x_s = \omega_i]}) - 1} \sum_{s \in S} \mathbb{1}_{[x_s = \omega_i]} (y_s - m_i)^2$ (on divise par $\sum_{s \in S} (\mathbb{1}_{[x_s = \omega_i]}) - 1$ afin d'obtenir un estimateur non biaisé de σ_i^2)

La difficulté est, qu'ici, on n'observe que Y = y.
Il existe des méthodes permettant l'estimation des paramètres dans le cas de données manquantes. On s'intéresse à deux méthodes bien connues **EM** et **SEM**.

1 Algorithme EM

1.1 Théorie

Mise en place des outils mathématiques

Dans le cadre de l'algorithme Expectation-Maximisation, on considère des modèles statistiques où :

— Y sont les données observées

— X sont les données d'intérêt cachées

On se donne alors une famille de modèles pour décrire la loi jointe de (X, Y). On suppose que l'on dispose d'un paramètre $\theta \in \mathbb{R}^n$ et d'une densité de probabilité $(x, y) \mapsto p_\theta(x, y)$ (appelée vraissemblance jointe ou complète).

Cependant, comme on ne peut pas mesurer X, on ne peut pas directement maximiser $\theta \mapsto log(p_\theta(x, y))$. On a uniquement accès à Y et "donc" à $[\theta \mapsto log(p_\theta(y))]$.

Dans notre cas, le paramètre dans l'algorithme EM est $\theta = (p_1, p_2, m_1, m_2, \sigma_1^2, \sigma_2^2)$.

Comment calculer (ou estimer) le minimum de la log-vraissemblance

On souhaite calculer (ou estimer) $\theta_* \in \underset{\theta \in \mathbb{R}^n}{\text{Argmin}}\ log(p_\theta(y))$.

Cependant, il y a un **problème** : on ne peut pas calculer $p_\theta(y) = \int_{\mathbb{R}} p_\theta(x, y)dx$. Heureusement, comme nous sommes très futés, l'algorithme EM contourne cette difficulté en utilisant une quantité auxiliaire. En effet, on produit une suite d'estimateurs de θ, $(\theta^{(t)})_{t \leq 0}$ avec :

Algorithm 1 Algorithme EM

$\theta^{(0)}$ initialisé aléatoirement

for $t \geq 0$ (jusqu'à convergence) **do**

E-Step : Calculer $Q[\theta, \theta^{(t)}] = \mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[log(\mathbb{P}(x, y))]$ où $\mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[log(\mathbb{P}(x, y))] = \int_{\mathbb{R}} log(p_\theta(x, y))p_{\theta^{(t)}}(x|y)dx$

M-Step : Définir $\theta^{(t+1)} = \underset{\theta \in \mathbb{R}^n}{\text{Argmin}}\ Q[\theta, \theta^{(t)}]$

end for

Explicitons le calcul de la loi jointe :

$$\begin{aligned}\mathbb{P}(x, y) &= \prod_n \mathbb{P}(x_n)\mathbb{P}(y_n|x_n) \\ &= \prod_n (p_1 f_1(y_n; m_1, \sigma_1^2))^{\mathbb{1}_{\{x_n=\omega_1\}}} (p_2 f_2(y_n; m_2, \sigma_2^2))^{\mathbb{1}_{\{x_n=\omega_2\}}}\end{aligned}$$

ce qui nous donne :

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[log(\mathbb{P}(x,y))] &= \mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[\sum_n[\mathbb{1}_{\{x_n=\omega_1\}}(log(p_1)+log(f_1(y_n;m_1,\sigma_1^2)))\\
&+\mathbb{1}_{\{x_n=\omega_2\}}(log(p_2)+log(f_2(y_n;m_2,\sigma_2^2)))]]\\
&= \sum_n[\mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[\mathbb{1}_{\{x_n=\omega_1\}}](log(p_1)+log(f_1(y_n;m_1,\sigma_1^2)))\\
&+\mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[\mathbb{1}_{\{x_n=\omega_2\}}](log(p_2)+log(f_2(y_n;m_2,\sigma_2^2)))]\\
&= \sum_n[\mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})(log(p_1)+log(f_1(y_n;m_1,\sigma_1^2)))\\
&+\mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})(log(1-p_1)+log(f_2(y_n;m_2,\sigma_2^2)))]
\end{aligned}$$

Nous avons désormais l'expression de l'espérance pour tout $\theta^{(t)}$. Notons que j'ai remplacé, ici, p_2 par $1-p_1$ pour éviter de devoir dériver sous contraintes et ainsi engendrer des difficultés supplémentaires.

Nous avons calculer l'espèrance, il reste alors à dériver par rapport à chaque paramètre de $\theta^{(t)}$ et à annuler ces dérivées. Je détaillerai le calcul pour la dérivée par rapport à p_1 :
$\frac{\partial \mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[log(\mathbb{P}(x,y))]}{\partial p_1} = \sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})\frac{1}{p_1} - \sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})\frac{1}{1-p_1}$
d'où,

$$\begin{aligned}
\frac{\partial \mathbb{E}_{\mathbb{P}(x|y;\theta^{(t)})}[log(\mathbb{P}(x,y))]}{\partial p_1} = 0 \quad &\Leftrightarrow \frac{1}{p_1}\sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)}) = \frac{1}{1-p_1}\sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})\\
&\Leftrightarrow p_1 = \frac{\sum_n(\mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})}{\sum_n[\mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})+\mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})]}\\
&\Leftrightarrow p_1^{(t+1)} = \frac{1}{n}\sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)}) \quad \text{Ouf, ça nous rassure...}
\end{aligned}$$

De la même manière, on trouve :

$$\begin{aligned}
p_2^{(t+1)} &= 1 - p_1^{(t+1)}\frac{1}{n}\sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})\\
m_1^{(t+1)} &= \frac{\sum_n y_n\mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})}{\sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})}\\
m_2^{(t+1)} &= \frac{\sum_n y_n\mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})}{\sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})}\\
\sigma_1^{2(t+1)} &= \frac{\sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})(y_n-m_1^{(t+1)})^2}{\sum_n \mathbb{P}(x_n=\omega_1|y_n;\theta^{(t)})}\\
\sigma_2^{2(t+1)} &= \frac{\sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})(y_n-m_2^{(t+1)})^2}{\sum_n \mathbb{P}(x_n=\omega_2|y_n;\theta^{(t)})}
\end{aligned}$$

Une des propriétés les plus importantes de cet algorithme est qu'il augmente la vraissemblance à chaque itération. Cela permet d'assurer la convergence vers un maximum ; bien que la convergence vers le maximum global de la vraissemblance (ou de la log-vraissemblance, il s'agit du même combat) n'est pas assurée.

Les défauts de l'algorithme EM sont les mêmes que tous les algorithmes d'optimisation alternée :

— Convergence vers un minimum local.

— Dépend de l'initialisation.

1.2 Expérimentations

Montrons que l'algorithme EM dépend fortement de l'initialisation. Pour se faire, nous pouvons initialiser notre algorithme avec un paramètre θ^0 de plus en plus éloigné de la réalité. Comme nous avons la connaissance de $\theta^{\text{réel}}$, cela nous aide un peu. Voici les résultats de l'algorithme EM selon l'initialisation précisée dans le titre :

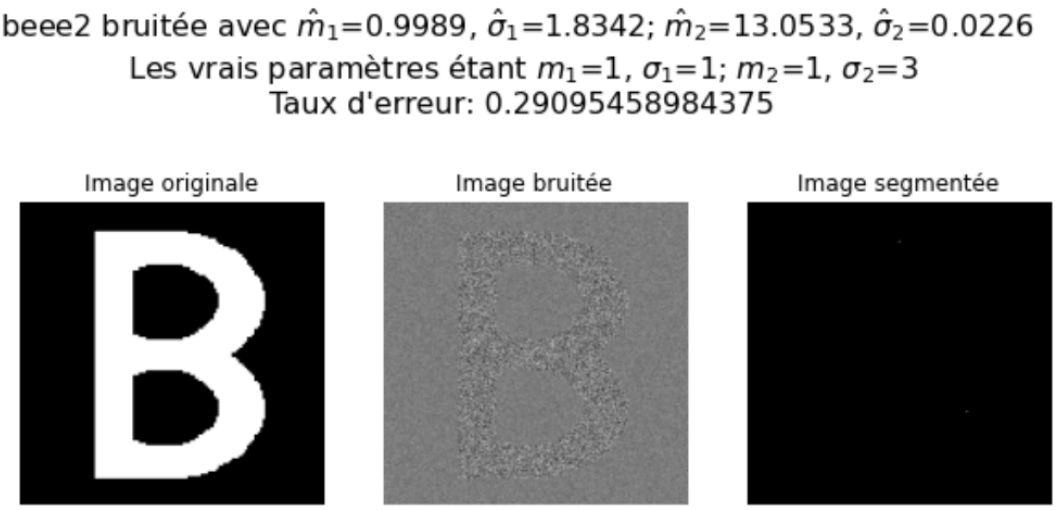

FIGURE 1 – Algorithme EM initialisé avec $m_1 = 1$, $\sigma_1 = 2$, $m_2 = 32$ et $\sigma_2 = 3$

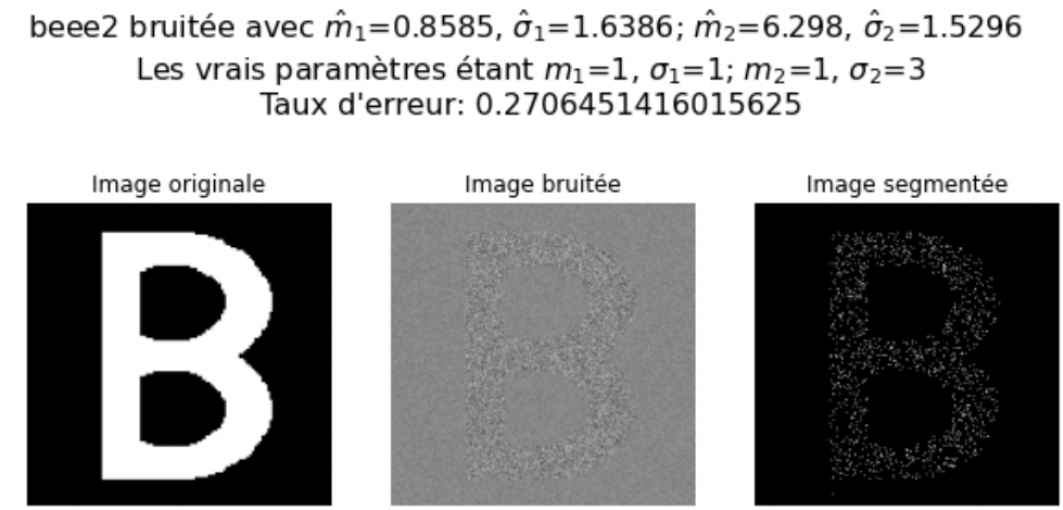

FIGURE 2 – Algorithme EM initialisé avec $m_1 = 3$, $\sigma_1 = 4$, $m_2 = 32$ et $\sigma_2 = 3$

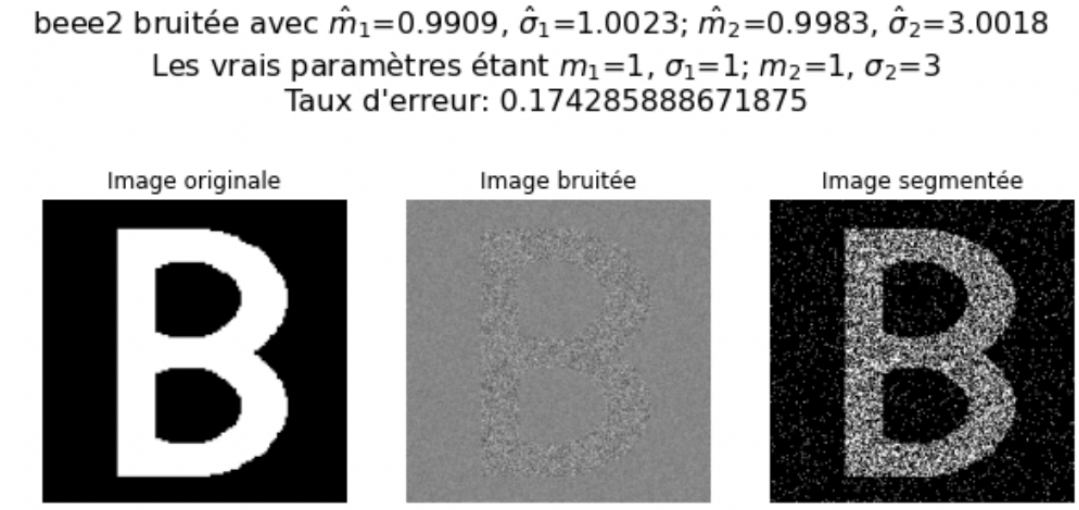

FIGURE 3 – Algorithme EM initialisé avec un K-Means pour avoir une initialisation proche de la réalité

On s'aperçoit assez facilement, ici, de la dépendance en l'initialisation de l'algorithme EM. Deux astuces simples permettent de contourner ce problème : répéter un grand nombre de fois l'algorithme en croisant les doigts pour tomber sur l'initialisation qui nous donne la convergence vers le maximum global de la vraissemblance. L'autre solution étant d'initialiser notre paramètre grâce à l'implémentation d'un algorithme K-Means nous permettant d'obtenir une estimation grossière de θ mais qui fonctionnera parfaitement

comme initialisation de l'algorithme EM. On peut également tracer l'évolution de nos paramètres en fonction des itérations de l'EM :

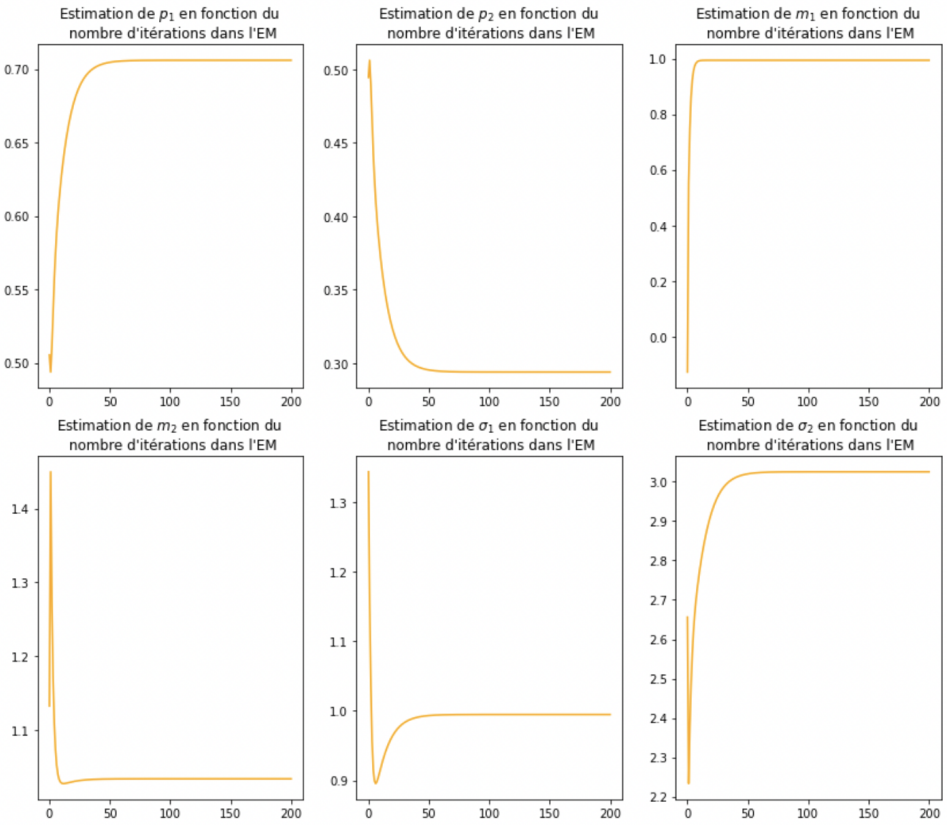

FIGURE 4 – Évolution des composantes de θ en fonction des itérations de l'EM

On remarque qu'une fois la phase d'initialisation passée, les valeurs des paramètres convergent rapidement vers les vraies valeurs des composantes de θ. On peut remarquer que l'augmentation de la vraissemblance à chaque itération nous permet d'obtenir des courbes *"lisses"*, ce qui ne sera pas le cas dans le Stochastic EM.
En initialisant de cette manière, on récupère une bonne estimation de θ en sortie de l'algorithme ce qui permet une segmentation presque aussi fidèle que dans le cas où l'on connaît les paramètres théoriques. On obtient les segmentations suivantes de l'image **beee2.bmp** bruitée par les 3 bruits :

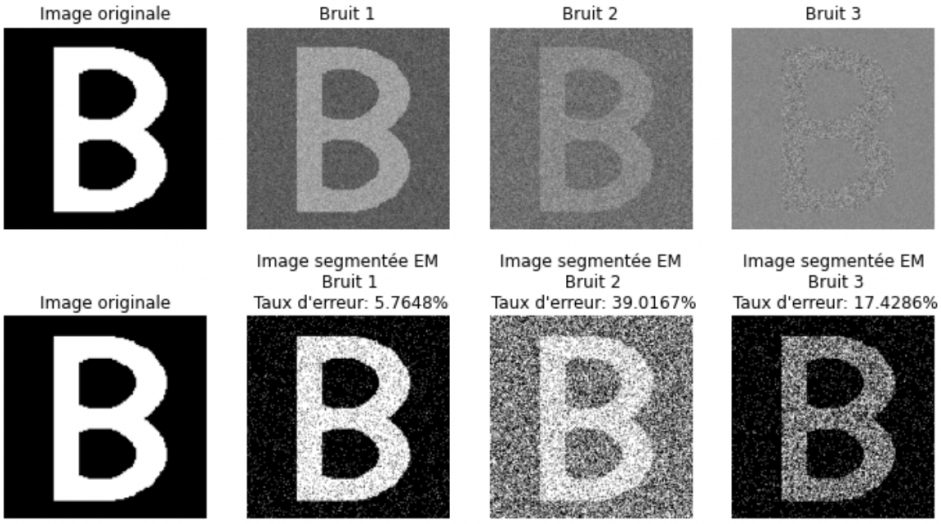

FIGURE 5 – Segmentation de l'image beee2.bmp bruitée par 3 bruits suite à l'estimation des paramètres par l'algorithme EM et suivant le critère du MPM

On fait de même pour nos 5 images choisies au début du premier TP et l'on regroupe les différents taux d'erreur dans un tableau :

Bruits	alfa2	beee2	country2	veau2	zebre2	Méthode utilisée
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(4,1^2)$	0.083	0.076	0.071	0.067	0.068	K-Means
	0.055	0.059	0.063	0.067	0.066	MPM
	0.055	0.06	0.063	0.067	0.066	EM + MPM
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(2,1^2)$	0.353	0.338	0.323	0.309	0.311	K-Means
	0.22	0.248	0.278	0.307	0.302	MPM
	0.418	0.382	0.369	0.309	0.333	EM + MPM
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(1,3^2)$	0.5	0.501	0.501	0.499	0.5	K-Means
	0.154	0.175	0.204	0.265	0.236	MPM
	0.152	0.176	0.205	0.265	0.236	EM + MPM

TABLE 1 – Tableau récapitulatif des taux d'erreur moyens (sur 100 itérations) entre la méthode du K-Means et le critère du MPM avec ou sans estimation de θ. Le meilleur taux d'erreur est coloré en vert et l'autre en rouge, ou en orange s'ils sont du même ordre de grandeur.

On remarque que dans les bruits n°1 et 3, les taux d'erreur sont sensiblement les mêmes que dans le cas où l'on connaît la valeur théorique de θ. Toutefois, l'algorithme

EM n'arrive pas à estimer les paramètres de notre modèle et le taux d'erreur augmente considérablement. Visualisons l'évolution de θ en fonction des itérations dans l'EM :

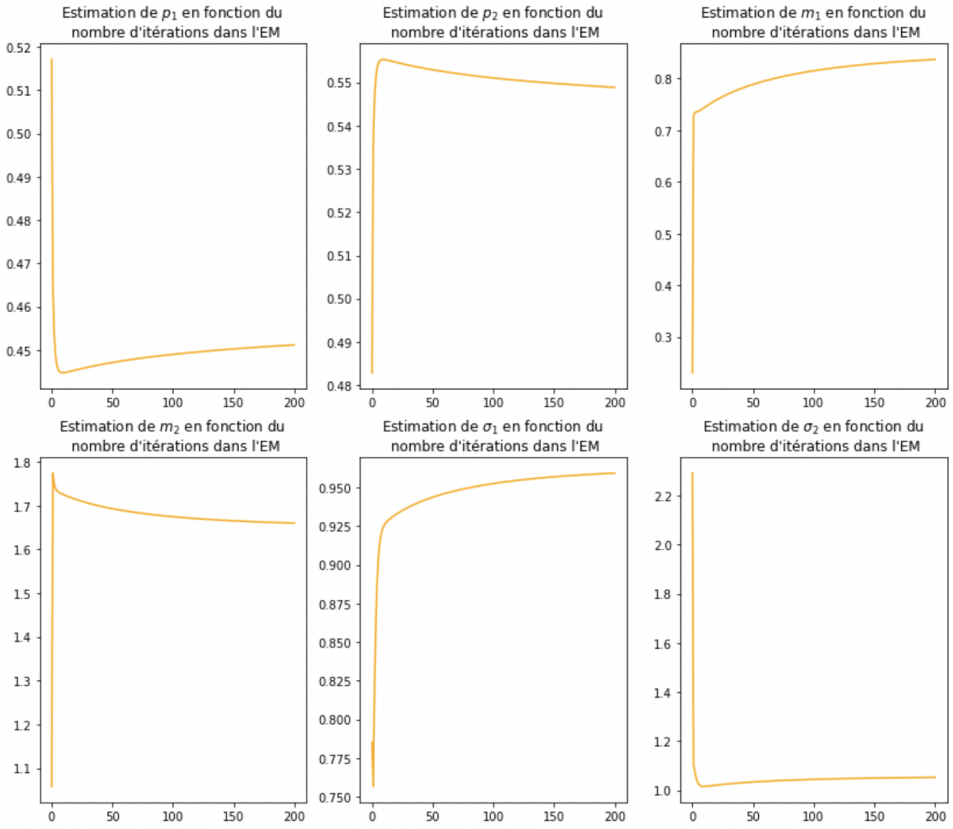

FIGURE 6 – Graphique représentant l'évolution de θ en fonction des itérations dans l'EM pour l'estimation des paramètres de l'image bruitée beee2.bmp par le bruit 2

On remarque que l'estimation ne se réalise pas correctement et cela empire lorsque l'on augmente le nombre d'itérations dans l'EM.

2 Algorithme SEM

L'algorithme SEM permet d'estimer le paramètre θ de manière stochastique en se donnant artificiellement la connaissance de X. Pour se faire, on réalise des tirages de X suivant la loi à posteriori afin d'utiliser les estimateurs empiriques des données complètes présentés en introduction. En utilisant des simulations de X, on s'affranchit du problème d'initialisation dans l'algorithme SEM. De la même manière que précédemment, on peut visualiser l'évolution stochastique des paramètres en fonction des itérations de l'algorithme et la segmentation obtenue :

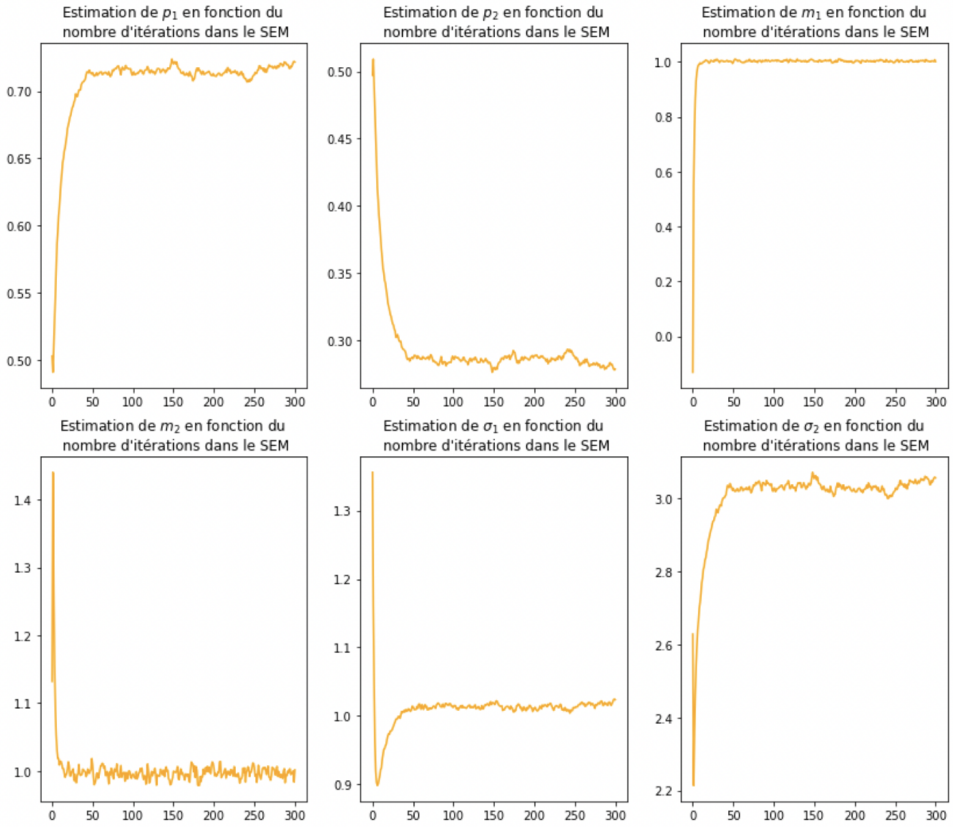

FIGURE 7 – Évolution des composantes de θ en fonction des itérations de le SEM

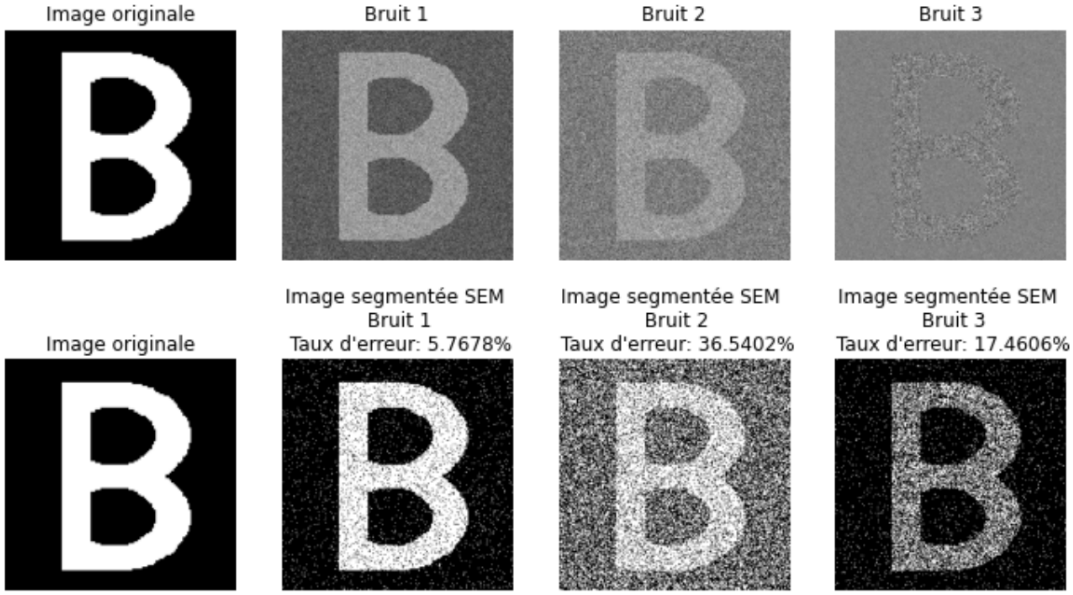

FIGURE 8 – Segmentation de l'image beee2.bmp bruitée par 3 bruits suite à l'estimation des paramètres par l'algorithme SEM et suivant le critère du MPM

Regroupons les taux d'erreurs moyens dans le précédent tableau :

Bruits	alfa2	beee2	country2	veau2	zebre2	Méthode utilisée
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(4,1^2)$	0.083	0.076	0.071	0.067	0.068	K-Means
	0.055	0.059	0.063	0.067	0.066	MPM
	0.055	0.06	0.063	0.067	0.066	EM + MPM
	0.055	0.058	0.063	0.066	0.065	SEM + MPM
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(2,1^2)$	0.353	0.338	0.323	0.309	0.311	K-Means
	0.22	0.248	0.278	0.307	0.302	MPM
	0.418	0.382	0.369	0.309	0.333	EM + MPM
	0.380	0.365	0.336	0.310	0.343	SEM + MPM
$\mathcal{N}(1,1^2)$ et $\mathcal{N}(1,3^2)$	0.5	0.501	0.501	0.499	0.5	K-Means
	0.154	0.175	0.204	0.265	0.236	MPM
	0.152	0.176	0.205	0.265	0.236	EM + MPM
	0.152	0.175	0.202	0.267	0.232	SEM + MPM

TABLE 2 – Tableau récapitulatif des taux d'erreur moyens (sur 100 itérations) entre la méthode du K-Means et le critère du MPM avec ou sans estimation de θ par 2 méthodes différentes. Le meilleur taux d'erreur est coloré en vert et l'autre en rouge, ou en orange s'ils sont du même ordre de grandeur.

On remarque comme précédemment que le bruit 2 est trop fort et ne permet pas la bonne estimation de θ. Cela pose donc un problème dans la segmentation de l'image et entraîne donc une hausse des erreurs dans la classification des pixels.

Conclusion

Nous avons mis en oeuvre deux méthodes différentes pour l'estimation des paramètres dans le cas où nous ne connaissons pas les paramètres théoriques de l'image et du bruit (ce qui est le cas 99.99% des cas en pratique). Toutefois, nous nous sommes limités la performance de la segmentation en considérant l'indépendance des pixels dans notre modèle. Les pixels des images sont dépendants de leurs voisins. On peut ainsi modéliser les images par des chaînes de Markov cachées ou des champs de Markov pour tenir compte de cette dépendance et c'est ce que nous ferons dans le prochain TP.