Eléments de Correction : La Segmentation bayésienne

I) Une première idée des enjeux du problème

Code des fonctions :

```
def bruit_gauss2(X,cl1,cl2,m1,sig1,m2,sig2):
    return (X == cl1) * np.random.normal(m1, sig1, X.shape) + (X == cl2) *
np.random.normal(m2, sig2, X.shape)
```

```
def classif_gauss2(Y,cl1,cl2,m1,sig1,m2,sig2):
    return np.where((norm.pdf(Y, m1, sig1)) > (norm.pdf(Y, m2, sig2)), cl1,
cl2)
```

```
def taux_erreur(A,B):
    return np.count_nonzero(A!=B)/A.size
```

Dans la Figure 1 nous affichons le signal `signal.npy` original, le signal `signal.npy` bruité par un bruit gaussien indépendant et la segmentation supervisée du signal bruité par le critère du maximum de vraisemblance. Supervisée signifie que les paramètres originaux du bruit sont connus et utilisés (ce qui n'est pas le cas, en général, dans les applications réelles).

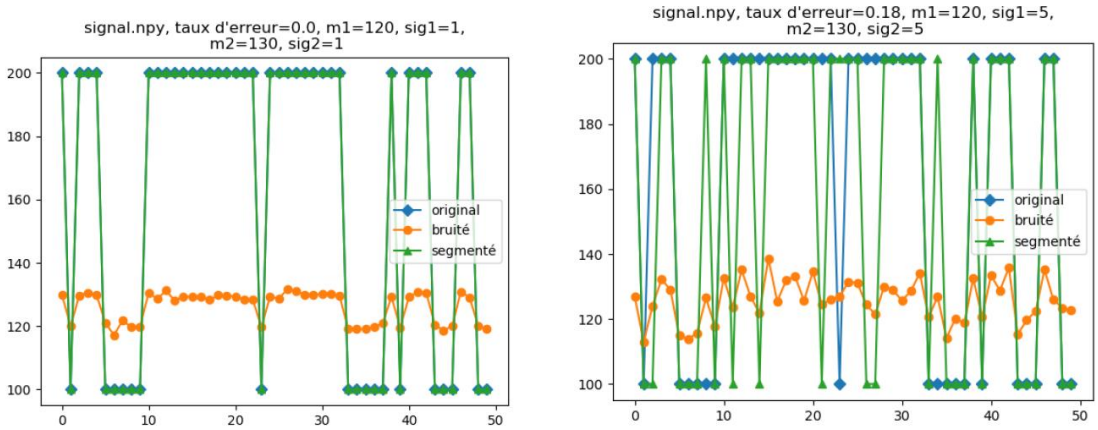


Figure 1: Exemples de bruitages et segmentations d'un signal unidimensionnel.

Remarque: Par commodité de représentation graphique et informatique nous avons choisi des réels pour les classes cachées ainsi que pour les observations. Il ne faut cependant pas

confondre l'espace des réalisations des variables cachées et l'espace des variables observées qui sont, en général, distincts

La Figure 2 illustre l'évolution du taux d'erreur moyen calculé sur T simulations lorsque T augmente. On constate une stabilisation du taux d'erreur moyen lorsque T devient grand (au moins 500 itérations semblent nécessaires). Pour des paramètres de bruit donnés, le taux d'erreur est aléatoire, il dépend du signal bruité observé qui est une variable aléatoire. Ainsi pour estimer la moyenne statistique des taux d'erreurs (ce que nous appelons erreur moyenne), nous pouvons, en vertu de la loi des grands nombres, calculer la moyenne empirique d'une population de taux d'erreur. La loi des grands nombres nous assure que plus la population est grande plus nous nous approcherons du vrai (au sens statistique) taux d'erreur moyen.

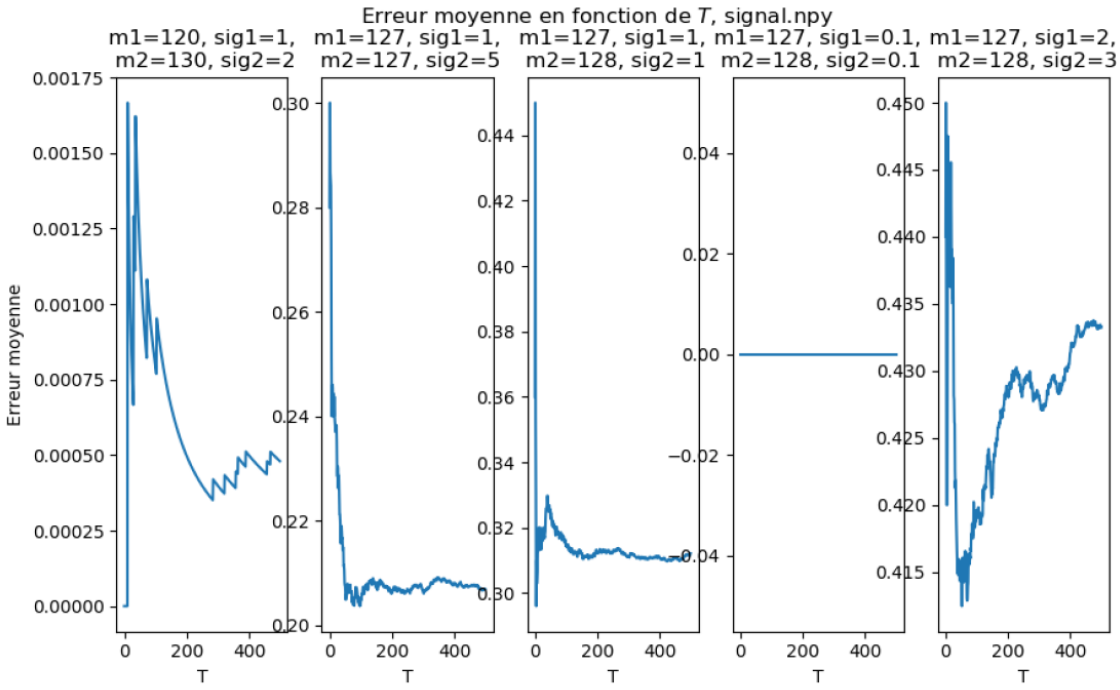


Figure 2: Taux d'erreur moyen en fonction de T pour les 5 bruits proposés sur `signal.npy`.

Le Tableau 1 donne les taux d'erreur moyens sur 500 simulations pour les 6 signaux bruités avec les 5 bruits. On qualifie de fort un bruit qui va beaucoup dégrader le signal. La segmentation du signal va donc être complexe et les stratégies de restauration vont commettre un nombre plus important d'erreurs que pour un bruit faible.
Une analyse du Tableau 1 semble indiquer que le Bruit 5 est le plus fort, suivi du Bruit 3, du Bruit 2, du Bruit 1 et du Bruit 4. Ces deux derniers bruits peuvent être qualifiés de très faibles car le classifieur du maximum de vraisemblance ne commet presque aucune erreur sur ces deux bruits, quel que soit le signal. Il semble difficile de conjecturer des règles toujours valables sur les propriétés du bruit en fonction des valeurs des moyennes et des écarts-types avec seulement ces 5 exemples. Cependant, nous pouvons nous servir des représentations graphiques des densités gaussiennes qui caractérisent notre bruit. Le critère du maximum de vraisemblance est directement lié à ces densités : pour une valeur des observations donnée nous choisissons la courbe qui prend la plus grande valeur. Il vient que le taux d'erreur peut être lié à une aire sous la courbe. En effet, en regardant la Figure 3 qui illustre deux densités

gaussiennes, si la vraie classe d'une observation correspond à la courbe orange, le classifieur va se tromper partout où la courbe orange sera sous la courbe bleue. Et inversement, lorsqu'il faut choisir la courbe bleue. Un calcul d'aire sous la courbe nous permet ensuite de déduire ce taux d'erreur théorique.

	Bruit 1	Bruit 2	Bruit 3	Bruit 4	Bruit 5
`signal.npy`	< 0.01	0.21	0.32	0	0.43
`signal1.npy`	< 0.01	0.18	0.31	0	0.39
`signal2.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal3.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal4.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal5.npy`	< 0.01	0.18	0.31	< 0.01	0.38

Table 1: Taux d'erreur moyens pour les différents bruits et les différents signaux (arrondis à 10^{-2}).

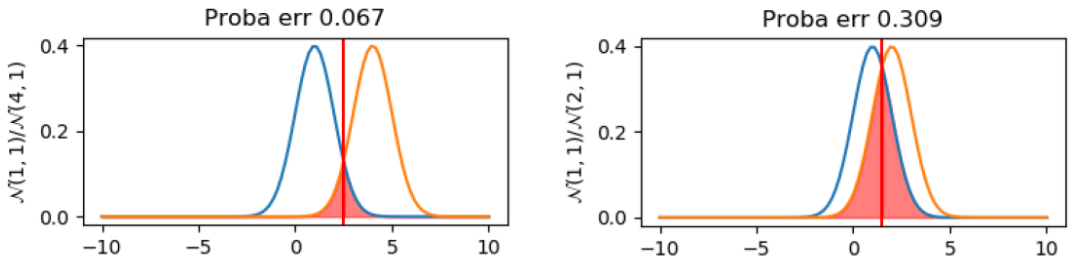


Figure 3: Taux d'erreur du classifieur de maximum de vraisemblance : lecture graphique.

II) Apport des méthodes bayésiennes de segmentation

Code des fonctions :

```
def calc_probaprio2(X,cl1,cl2):
    return np.sum(X==cl1)/X.size, np.sum(X==cl2)/X.size
```

```
def MAP_MPM2(Y,cl1,cl2,p1,p2,m1,sig1,m2,sig2):
    return np.where((p1*norm.pdf(Y, m1, sig1)) > (p2*norm.pdf(Y, m2,
sig2)), cl1, cl2)
```

```
def simul2(n,cl1,cl2,p1,p2):
    X = np.random.binomial(1, p1, n)
    X[X==1]=cl1
    X[X==0]=cl2
    return X
```

Dans le Tableau 1, nous donnons, pour tous les bruits et tous les signaux fournis, les taux d'erreur moyens du classifieur MAP. Nous indiquons les gains obtenus par rapport à la segmentation MV effectuée dans le compte-rendu précédent.

Remarque : Dans le cas aveugle considéré (les variables aléatoires en chacun des sites sont considérées indépendantes), la stratégie MAP est équivalente à la stratégie MPM.

Il est notable que les taux d'erreurs ne sont meilleurs que pour le signal `signal.npy`. C'est de plus le seul signal à posséder une distribution a priori qui dévie de la loi équiprobable, à savoir : $p(\omega_1) = 0.36\ et\ p(\omega_2) = 0.64$. Dans la suite nous détaillons le rôle de la loi a priori sur le taux de d'erreur.

	Bruit 1	Bruit 2	Bruit 3	Bruit 4	Bruit 5
`signal.npy`	< 0.01	$0.20\ (-1)$	$0.28\ (-4)$	0	$0.36\ (-7)$
`signal1.npy`	< 0.01	0.18	0.31	0	0.39
`signal2.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal3.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal4.npy`	< 0.01	0.18	0.31	< 0.01	0.38
`signal5.npy`	< 0.01	0.18	0.31	< 0.01	0.38

Table 1: Classification MAP/MPM. Taux d'erreur moyens (sur 500 itérations) pour les différents bruits et les différents signaux (arrondis à 10^{-2}) et comparaison avec les taux d'erreur du MV (compte-rendu 1). Une cellule orange indique un taux d'erreur similaire entre MAP et MV. Une cellule verte indique un meilleur taux d'erreur du MAP face au MV. Le gain (en points) peut aussi être indiqué.

Le Tableau 2 récapitule la segmentation de 5 signaux générés (tirages indépendants avec des lois a priori variables) avec les méthodes MAP/MPM et MV.

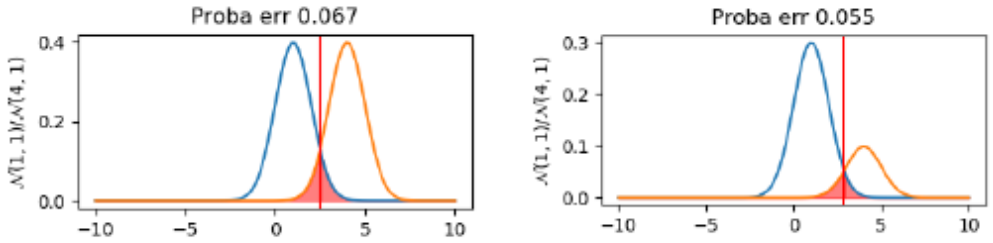
Remarque: Il faut toujours faire attention à simuler des signaux assez grands et de prendre les taux d'erreur moyens sur un grand nombre d'itérations afin de pouvoir correctement évaluer ce taux d'erreur.

La première constatation que l'on peut faire en analysant le Tableau 2 est que le classifieur du MAP/MPM est toujours meilleur ou équivalent au classifieur MV. Les cas les plus favorables au MAP semblent être les cas où la loi a priori dévie le plus de la loi équiprobable. Les cas d'égalité ont lieu pour une loi a priori qui est équiprobable. Cette dernière remarque est directement visible en comparant les équations des deux classifieurs : si $p(\omega_1) = p(\omega_2)$ (loi équiprobable) alors les équations définissant le MAP/MPM deviennent équivalentes à celles définissant le MV. Le classifieur MAP est donc capable de prendre en compte l'information a priori pour améliorer les segmentations.

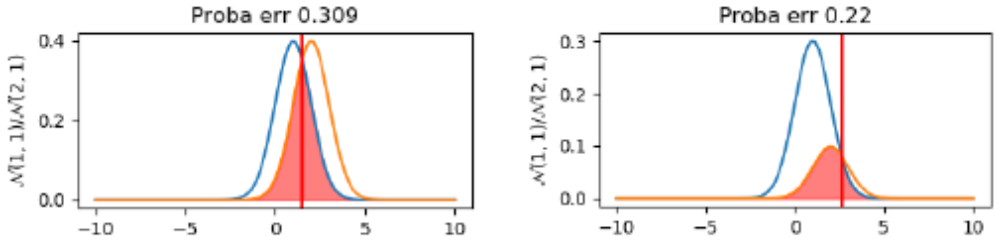
	Bruit 1		Bruit 2		Bruit 3		Bruit 4		Bruit 5	
	MAP	MV	MAP	MV	MAP	MV	MAP	MV	MAP	MV
$p(\omega_1) = 0.01$	< 0.01	< 0.01	< 0.01	0.28	< 0.01	0.31	< 0.01	< 0.01	< 0.01	0.46
$p(\omega_1) = 0.13$	< 0.01	< 0.01	0.13	0.26	0.13	0.31	< 0.01	< 0.01	< 0.13	0.50
$p(\omega_1) = 0.25$	< 0.01	< 0.01	0.20	0.23	0.23	0.31	< 0.01	< 0.01	< 0.25	0.46
$p(\omega_1) = 0.37$	< 0.01	< 0.01	0.20	0.21	0.28	0.31	< 0.01	< 0.01	0.36	0.42
$p(\omega_1) = 0.50$	< 0.01	< 0.01	0.18	0.18	0.31	0.31	< 0.01	< 0.01	0.38	0.38

Table 2: Classification MAP/MPM et classification MV. Taux d'erreur moyens (sur 500 itérations) pour les différents bruits et différents signaux simulés (arrondis à 10^{-2}, signaux de longueur 10000). À partir de $p(\omega_1)$, on a $p(\omega_2) = 1 - p(\omega_1)$. Si les résultats des deux classifieurs sont comparables, les deux cellules concernées sont colorées en orange, sinon le meilleur taux d'erreur est coloré en vert et l'autre en rouge.

A l'instar de ce qui a été fait dans la première partie, pour bien comprendre les taux d'erreur obtenus, nous pouvons à nouveau nous fier à une lecture graphique. Il s'agit cette fois, d'après le critère du MAP/MPM, de maximiser densité de la loi a posteriori dont la densité est proportionnelle à la densité de la vraisemblance conditionnelle multipliée par la probabilité a priori. La densité de la loi a posteriori peut donc être tracée et l'erreur facilement visualisée (en plus de pouvoir être analytiquement calculée). C'est ce que nous voyons sur la Figure 1 : l'apport de la loi a priori permet de réduire le taux d'erreur entre une classification MV et une classification MAP/MPM sur les deux exemples proposés.



(a) *Gauche* : les densités de vraisemblance conditionnelle et le taux d'erreur associé à une stratégie MV. *Droite* : les densités *a posteriori* et le taux d'erreur associé à une stratégie MAP/MPM.



(b) *Gauche* : les densités de vraisemblance conditionnelle et le taux d'erreur associé à une stratégie MV. *Droite* : les densités *a posteriori* et le taux d'erreur associé à une stratégie MAP/MPM.

Figure 1: Deux exemples de l'apport de la loi *a priori* sur le taux d'erreur. Dans les deux cas, la probabilité *a priori* de la classe bleue est trois fois plus grande que la probabilité *a priori* de la classe orange. La ligne rouge verticale indique le seuil théorique de décision.