

Sujet 1 : Mise en œuvre de l'algorithme FastICA pour la séparation aveugle de sources

I. Génération d'un mélange artificiel de deux sources

1. En utilisant la fonction Matlab *rand*, générer deux sources aléatoires chacune contenant 2000 échantillons¹.
2. Faire le traitement nécessaire pour obtenir deux sources centrées et de puissances unitaires $s_1(t)$ et $s_2(t)$.
3. Mesurer le coefficient de corrélation entre les deux sources et commenter.
4. Représenter le nuage de points bidimensionnel associé aux deux sources. Si on sait par exemple que $s_1=1.5$, qu'est-ce qu'on peut dire sur la valeur probable de s_2 à partir de cette représentation ? Peut-on dire que les sources sont indépendantes ?
5. Mélanger les deux sources avec la matrice de mélange $A = \begin{pmatrix} 1 & 1 \\ 0.5 & 1 \end{pmatrix}$ pour obtenir le vecteur d'observations $x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$.
6. Mesurer le coefficient de corrélation entre les deux observations et représenter le nuage de points correspondant. Conclure sur l'indépendance des observations.

II. Blanchiment

Comme vous l'avez vu en cours, la première étape de l'algorithme FastICA consiste à blanchir les observations pour obtenir un vecteur d'observations blanchies contenant deux signaux non-correlés et de puissances unitaires, noté $z(t) = \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix}$ dans la suite.

1. En utilisant la décomposition en valeurs propres de la matrice de covariance de x , calculer la matrice de blanchiment M permettant de générer les observations blanchies.
2. Mesurer le coefficient de corrélation entre $z_1(t)$ et $z_2(t)$ et représenter le nuage de points associé. Conclure sur la non-corrélation et l'indépendance des observations blanchies.

III. Estimation des sources

La deuxième étape de l'algorithme FastICA standard (à kurtosis et à déflation) consiste à trouver un vecteur w_I tel que la valeur absolue du kurtosis du signal $y_I(t) = w_I^T z(t)$ soit maximum. $y_I(t)$ correspondra alors à une des sources estimées.

1. Rappeler la règle de mise à jour de w_I dans un algorithme du point fixe. Faut-il normaliser w_I après chaque mise à jour ? Pourquoi ?

¹ Ajouter la commande *rand('seed',I)* avant de générer les sources.

2. Ecrire un programme Matlab permettant l'optimisation du vecteur w_1 en utilisant un algorithme itératif et la règle de mise à jour ci-dessus. On initialisera w_1 avec des valeurs aléatoires gaussiennes en utilisant la fonction $randn$ et en fixant $seed=1$. A chaque itération, on calculera la valeur absolue du kurtosis de $y_1(t)$ et on stockera le résultat dans un vecteur.
3. Après la convergence, tracer l'évolution de la valeur absolue du kurtosis au fil des itérations. Conclure sur le nombre d'itérations nécessaires pour la convergence.
4. Tracer $y_1(t)$ et la comparer avec les sources originales $s_1(t)$ et $s_2(t)$. Conclusion ?

Pour calculer la deuxième source estimée $y_2(t) = w_2^T z(t)$, on peut utiliser la propriété d'orthonormalité des vecteurs w_i dans l'espace des signaux blanchis.

5. Expliquer pourquoi w_1 et w_2 sont orthonormales (orthogonales et de normes unitaires).

L'orthonormalité peut se traduire par les trois conditions suivantes : $w_1^T w_2 = 0$, $w_1^T w_1 = 1$, $w_2^T w_2 = 1$.

6. On définit $w_1 = \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix}$ et $w_2 = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix}$. Montrer que w_1 et w_2 vérifient les 3 conditions d'orthonormalité.

On peut donc estimer w_2 à partir de w_1 en calculant $\alpha = \arctan\left(\frac{-w_1[1]}{w_1[2]}\right)$.

7. Utiliser cette approche pour estimer w_2 , puis en déduire la deuxième source estimée $y_2(t)$. Tracer $y_2(t)$ et la comparer avec $s_1(t)$ et $s_2(t)$. Conclusion ?
8. Mesurer le coefficient de corrélation entre $y_1(t)$ et $y_2(t)$ et représenter le nuage de points associé. Conclure sur l'indépendance des sources estimées.

IV. Mesure de performances de l'algorithme

1. Rappeler les indéterminations de facteur d'échelle et de permutation dans les algorithmes de séparation de sources.

L'un des critères utilisés pour mesurer les performances des algorithmes de séparation de sources est le Rapport Signal sur Interférence (RSI). Dans le cas de deux sources et deux observations, ce critère peut être calculé de la manière suivante.

On normalise d'abord chacun des deux signaux sources (s_1 et s_2) et chacun des deux signaux estimés (y_1 et y_2) pour qu'ils aient tous une même puissance (généralement égale à 1). L'indétermination sur la valeur absolue du facteur d'échelle est ainsi levée mais il reste encore les indéterminations de signe et de permutation (à titre d'exemple, y_1 peut être une estimée de s_2 et y_2 peut être une estimée de $-s_1$). Ces indéterminations peuvent par exemple être levées en comparant les représentations graphiques de s_1 , s_2 , y_1 et y_2 . Notons par \hat{s}_1 la sortie estimée correspondant à la source s_1 (qui peut être l'un des signaux y_1 , y_2 , $-y_1$ ou $-y_2$) et par \hat{s}_2 la sortie estimée correspondant à la source s_2 . Le critère RSI s'écrit alors :

$$RSI = \frac{1}{2} \sum_{i=1}^2 10 \log_{10} \frac{E\{s_i^2\}}{E\{(s_i - \hat{s}_i)^2\}}$$

2. Calculer le RSI pour le test ci-dessus.

V. Tests avec d'autres signaux

1. Refaire le test en utilisant les sources gaussiennes (remplacer *rand* par *randn* pour la génération des sources originales). Expliquer brièvement pourquoi l'algorithme ne fonctionne pas dans ce cas.
2. Les fichiers *audio_mix1.wav* et *audio_mix2.wav* contiennent deux mélanges artificiels de deux signaux audio. Ecouter les deux signaux puis les séparer avec votre algorithme. Ecouter les signaux estimés. Conclusion ?
3. Les fichiers *image_mix1* et *image_mix2* contiennent deux mélanges de deux images. Appliquer votre algorithme à ces images. Comparer visuellement les images estimées et les mélanges. Conclusion ?

VI. Séparation de plusieurs sources

Dans cette dernière partie, on va mettre en œuvre la méthode FastICA symétrique à kurtosis qui permet la séparation de n sources à partir de n mélanges.

1. Rappeler les règles de mise à jour et d'orthogonalisation utilisées dans cette méthode.
2. Ecrire un programme permettant de séparer n sources à partir de n mélanges.
3. Créer 3 sources aléatoires uniformément distribuées, de moyennes nulles et de puissances unitaires. Mélanger les sources avec la matrice $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0.6 \\ 1 & 0.7 & 1 \end{pmatrix}$.

Séparer les sources avec votre programme. Comparer visuellement les sources séparées et les sources estimées et calculer le RSI. Conclusion ?

4. Les fichiers *melange1.wav*, ..., *melange9.wav* contiennent 9 mélanges de 9 sources audio. Ecouter, puis séparer les mélanges. Ecouter les sources estimées et commenter.

Sujet2 : Séparation aveugle de sources par l'analyse en composantes parcimonieuses

Remarque : les questions dont le numéro est suivi du signe † sont des questions théoriques, ne nécessitant pas l'utilisation de Matlab, qui doivent impérativement être résolues avant la séance de TP.

VII. Présentation du TP

Dans ce TP, on s'intéresse à une classe de méthodes de SAS exploitant l'existence des zones mono-sources dans les représentations temporel ou temps-fréquence des signaux. Ces méthodes n'exigent pas l'indépendance statistique des sources, contrairement aux méthodes basées sur l'analyse en composantes indépendantes telles que FastICA. Pour des raisons de simplicité, nous ne considérons dans ce TP que les versions temporelles de ces méthodes même si les versions temps-fréquence sont en principe plus intéressantes puisque les signaux ont une plus grande parcimonie dans le domaine temps-fréquence. Nous allons donc mettre en œuvre dans la suite les méthodes LI-TEMPROM et LI-TEMPCORR vues en cours.

Remarque : Les équations utilisées pour décrire les méthodes LI-TEMPROM et LI-TEMPCORR sont pratiquement les mêmes que celles utilisées par les variantes temps-fréquence de ces méthodes (LI-TIFROM et LI-TIFCORR) présentées en cours : il suffit juste de supprimer la pulsation ω dans ces équations.

VIII. Génération d'un mélange artificiel de deux sources

1. Le fichier *s.mat* contient deux sources aléatoires s_1 et s_2 . Charger le fichier et tracer chacune des sources. Conclure sur l'existence des zones temporelles mono-sources dans des mélanges de ces signaux (une zone mono-source est par définition une zone temporelle où seulement l'une des sources est non-nulle).
2. Mesurer le coefficient de corrélation entre les deux sources et conclure sur leur indépendance.
3. Mélanger les deux sources avec la matrice de mélange $A = \begin{pmatrix} 1 & 0.7 \\ 0.8 & 1 \end{pmatrix}$ pour obtenir le vecteur d'observations $x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$.

IX. Méthode LI-TEMPROM

La méthode LI-TEMPROM utilise les rapports d'observations pour détecter les zones mono-sources dans les signaux observés et en déduire une estimée de la matrice de mélange.

4. † Former le rapport $x_2(t)/x_1(t)$. Que devient ce rapport lorsque la source s_1 est nulle ? Que devient-il quand s_2 est nulle ? Que peut-on dire sur la variance de ce rapport dans une zone temporelle où l'une des sources est nulle ? Proposer alors une méthode pour détecter les zones mono-sources.

5. Découper les signaux x_i en zones de $M=1000$ échantillons (appelées *zones d'analyse*). Dans chaque zone, calculer la variance du rapport $x_2(t)/x_1(t)$ et décider si la zone d'analyse peut être qualifiée de mono-source ou non.
6. † Compte tenu de votre réponse à la question 4, préciser le lien entre le rapport $x_2(t)/x_1(t)$ dans les zones mono-sources et les colonnes de la matrice de mélange. Proposer alors une méthode pour estimer une colonne de la matrice de mélange (à un facteur d'échelle près) dans chaque zone mono-source.
7. Utiliser la méthode proposée pour calculer une estimée d'une colonne de la matrice de mélange (à un facteur d'échelle près) dans chaque zone mono-source détectée.

On dispose ainsi d'un ensemble de vecteurs, constitués d'une ou plusieurs estimées de chacune des colonnes de la matrice de mélange A . Il reste alors à déduire, à partir de cet ensemble, une unique estimation de chacune des colonnes de A .

8. Tracer les colonnes estimées dans la question 7 et en déduire une unique estimation de chacune des colonnes de A .
9. Calculer les sources estimées en utilisant $y=A^T x$. Appliquer les traitements nécessaires pour que les sources estimées soient de puissances unitaires. Comparer chacune des sources estimées avec des sources originales et commenter.
10. Répéter le test avec les zones d'analyse plus courtes (par exemple $M=200$ et $M=50$).
11. Les fichiers *audio_mix1_tp3.wav* et *audio_mix2_tp3.wav* contiennent deux mélanges artificiels de deux signaux audio. Ecouter les deux signaux puis les séparer avec votre algorithme. Ecouter les signaux estimés. Conclusion ?

IV. Méthode LI-TEMPCORR

La méthode LI-TEMPCORR remplace les critères mettant en jeu des rapports d'observations par des critères fondés sur la corrélation des signaux.

12. † Que devient la valeur absolue du coefficient de corrélation $\rho_{x_1x_2} = \frac{E\{x_1x_2\}}{\sqrt{E\{x_1^2\}E\{x_2^2\}}}$ dans les zones mono-sources ? Proposer une méthode pour détecter les zones mono-sources à partir de ce critère.
13. † Quel est le lien entre la valeur du paramètre $C = \frac{E\{x_1x_2\}}{E\{x_1^2\}}$ dans les zones mono-sources et les colonnes de la matrice de mélange ? Proposer alors une méthode pour estimer une colonne de la matrice de mélange dans chaque zone mono-source.
14. Utiliser cette méthode pour séparer les sources artificielles et audio.

V. Conclusion

15. A votre avis, et compte tenu de vos tests, quels sont les principaux inconvénients et avantages de ces approches comparées à la méthode FastICA ?
16. Entre les deux méthodes LI-TEMPPROM et LI-TEMPCORR laquelle vous paraît la plus intéressante ? Pourquoi ?