Projet sur article : Multivariate Temporal Dictionary Learning for EEG

Hugo Dalla-Torre, Virginie Loison

March 2021

1 Introduction

Nous avons travaillé sur l'article *Multivariate Dictionary learning for EEG*, par Barthelemy et al, datant de 2013. [2].

Les éléctro-encéphalogrammes (EEG) sont fréquemment utilisés dans le domaine médical. En effet, ils bénéficient d'une bonne résolution temporelle, et d'une latence très faible. Néanmoins, les EEG sont difficilement étudiables sous leur forme brute, car peu lisibles, et peu interprétables (voir figure 1). Il est donc nécessaire de les traduire dans des données plus compactes, et plus interprétables, en utilisant par exemple des dictionnaires.

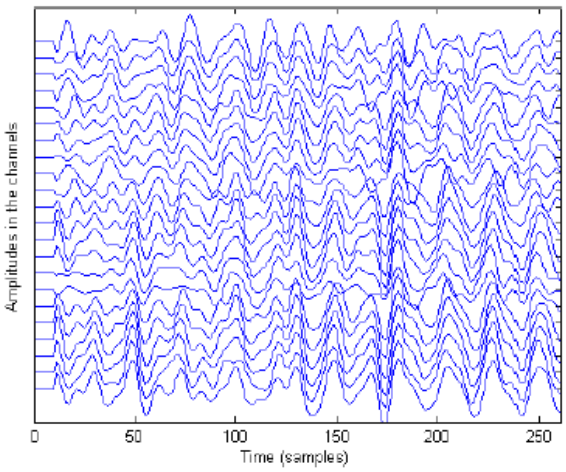

Figure 1: Example of EEG

Les dictionnaires classiques (Fourier, ondelettes) ne sont pas un bon choix de représentation ici, car ils ne sont pas adaptés a la structure particulière des EEG. L'approche de dictionary learning, qui est data-driven, est donc un bon choix aux yeux des auteurs.

Les auteurs proposent d'utiliser une approche multivariée pour le dictionary learning. Leur approche permet de tenir compte de deux spécificités des EEG :

- Modélisation spatiale : prise en compte des liens entre les électrodes. Les signaux des électrodes ne sont pas identiques, mais sont liés. Il est donc cohérent de traiter les canaux d'un même bloc, et non indépendamment. Ce point est assuré par le modèle multivarié spatial.
- Modélisation temporelle : shift-invariance. Ce point est assuré par l'usage d'un modèle convolutionnel.

2 Modèles et algorithmes utilisés dans l'article

2.1 Modèle multivarié

2.2 Dictionary learning

Pour le dictionary learning, les auteurs ont suivi une approche classique : alterner des phases de sparse coding et de mise à jour du dictionnaire.

2.2.1 Sparse coding - Multivariate orthogonal matching pursuit (M-OMP)

Pour le sparse coding, les auteurs ont choisi d'utiliser une adaptation multivariée de l'algorithme Orthogonal Matching Pursuit (OMP). L'OMP est un algorithme glouton : à chaque itération, il sélectionne l'atome qui permet la plus forte diminution de l'erreur quadratique. Dans la version multivariée convolutionnelle, en plus de l'atome, l'algorithme sélectionne à chaque itération le shift à appliquer à l'atome. Le détail de l'algorithme est expliqué dans [1], et le pseudo-code est représenté en figure 2

Algorithm 2 : $x = \text{Multivariate\_OMP}\ (y, \Psi)$

initialization : $k = 1,\ \epsilon^0 = y,$ dictionary $D^0 = \varnothing$
repeat
  for $l \leftarrow 1, L$ **do**
    Correlation : $C_l^k(\tau) \leftarrow \sum_{u=1}^{V} \Gamma\{\epsilon^{k-1}[u], \psi_l[u]\}(\tau)$
  end for
  Selection : $(l_{max}^k, \tau_{max}^k) \leftarrow arg\ max_{l,\tau}\ \mid C_l^k(\tau) \mid$
  Active Dictionary : $D^k \leftarrow D^{k-1} \cup \psi_{l_{max}^k}(t - \tau_{max}^k)$
  Active Coefficients : $x^k \leftarrow arg\ min_x \| y \rfloor - D^k \rfloor x \|^2$
  Residue : $\epsilon^k \leftarrow y - D^k x^k$
  $k \leftarrow k + 1$
until stopping criterion

Figure 2: Pseudo-code du sparse coding par M-OMP

2.2.2 Mise à jour du dictionnaire

L'étape de mise à jour du dictionnaire se fait en fonction du dictionnaire précédent et du sparse coding issu de l'étape présentée au dessus. A chaque atome est appliqué une étape de descente de gradient stochastique de second ordre dit de Levenberg-Marquardt, combinaison de la descente de gradient stochastique et de la méthode de Gauss-Newton [4]. La formule de mise à jour du noyau ψ_l est donnée ci-dessous:

$$\psi_l^{i+1}(t) = \psi_l^i(t) + (H^{i+1} + \lambda^{i+1} I) \sum_{\tau \in \sigma_l} x_{l,\tau}^{i+1} \epsilon^i(t + \tau)$$

où H^{i+1}, λ^{i+1}, $x_{l,\tau}^{i+1}$ and $\epsilon^i(t + \tau)$ sont respectivement la Hessienne de ψ_l^{i+1}, le learning rate, l'activation du noyau l correspondant au décalage temporel τ et le signal résiduel issue de l'itération précédente. Le calcul de la hessienne est accélérée grâce à la parcimonie des coefficients d'activation x qui permet d'ignorer les corrélations entrainées par la superposition d'un noyau avec lui même 3 fois ou plus. La parcimonie ne permet pas d'ignorer la corrélation entraînée par la superposition d'un noyau avec lui même, ainsi celle ci est majorée pour accélérer les calculs.

3 Implémentation

Nous avons choisi de réimplémenter l'expérience 1 de l'article. Le but de cette expérience est de faire du dictionary learning sur des EEG de patients, enregistrés alors qu'ils effectuaient des tâches motrices simples.

3.1 Dataset

L'article a utilisé le dataset 2.a de la compétition BCI IV [3]. Ce dataset contient des sessions EEG enregistrées sur 9 patients. Deux sessions ont été enregistrées pour chaque patient : la première a été mise dans l'ensemble d'entraînement, l'autre dans l'ensemble de test. Le but de la compétition était de faire de la classification, mais les auteurs ont utilisé ce dataset pour faire du dictionary learning. Les EEG ont été enregistrés à une fréquence de 250 Hz.

3.2 Preprocessing

Les auteurs de l'article ont choisi de filtrer les signaux, pour ne garder que les fréquences comprises entre 8Hz et 30Hz. C'est en effet dans cet intervalle que se situent les ondes de l'activité motrice. Nous avons fait de même. Nous pensons que les auteurs ont également fait du débruitage sur les signaux, mais il n'en est pas question dans l'article.

Dans l'article, du dictionary learning est fait sur l'ensemble du signal d'entraînement du premier patient. Ce signal est très gros, environ 627 000 échantillons x 22 channels. Ne disposant pas de la puissance de calcul nécessaire pour traiter ce signal, nous avons appliqué la méthode sur deux sous-échantillons :

- Première implémentation : sur les 500 premiers échantillons, et les 5 premiers channels. Ce dataset "jouet" permet de bien visualiser l'action de l'algorithme. En effet il comprend un motif ressemblant à un battement de coeur, ainsi la reconstruction se fait autour et permet une évaluation à l'oeil de la qualité de reconstruction.
- Seconde implémentation : sur un sous-échantillon plus uniforme (échantillons 100000 à 100500), et sur les 5 premiers channels. Ce sous-échantillon nous a semblé plus proche d'un signal EEG sans outlier, nous avons donc voulu tester l'algorithme dessus.

Pour quantifier la performance de reconstruction de l'algorithme, nous avons utilisé la même métrique que l'article : le taux de reconstruction :

$$\rho = 1 - \frac{||\epsilon||}{||y||}$$

où ϵ est le résiduel (signal d'origine - signal reconstruit), et y est le signal d'origine.

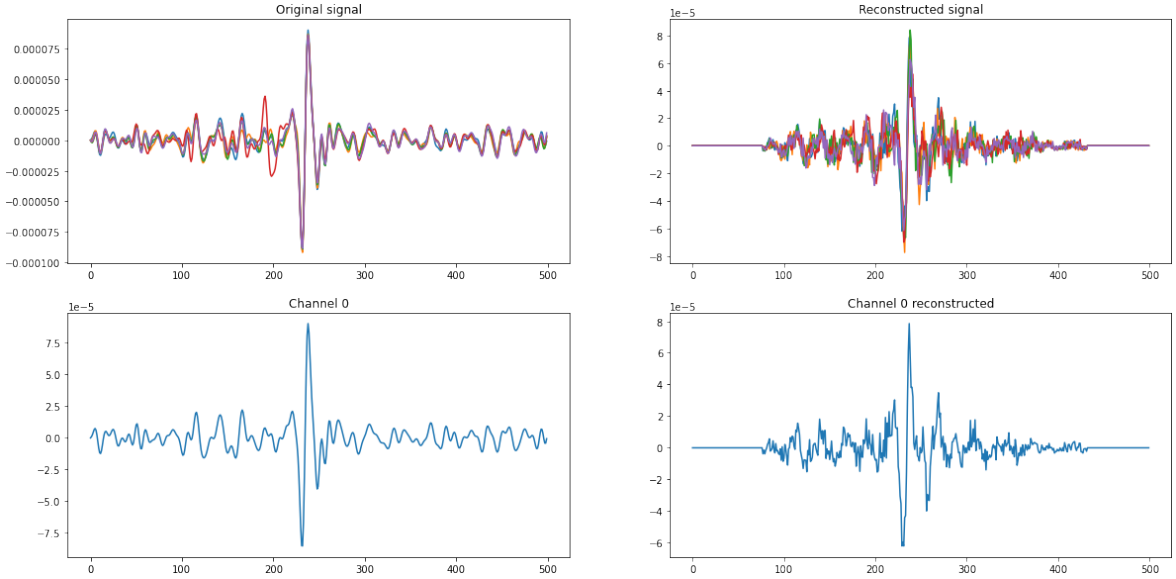

Figure 3: Multivariate dictionary learning reconstruction of the toy signal

3.3 Première implémentation

Le signal jouet est extrait comme expliqué ci-dessus et l'algorithme d'apprentissage du dictionnaire est testé sur cette version réduite du signal d'origine. La figure 3 montre le signal et sa recomposition, avec une focalisation sur un des channels.

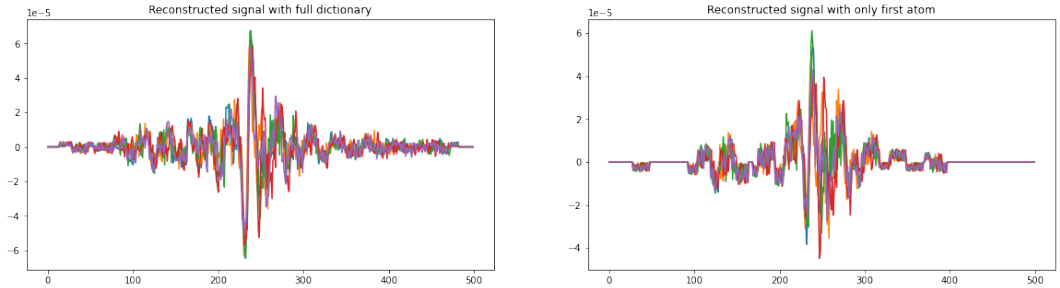

Figure 4: Partial reconstruction with only the most used atom

La reconstruction est faite à partir d'un dictionnaire composé de 5 noyaux de tailles de support 20,60,150,230 et 350. Ce choix est complètement arbitraire et permet de capturer les différentes fréquences du signal. On voit clairement sur la visualisation des activations (Figure 5) que le premier noyau, au support le plus restreint, est majoritairement utilisé pour la reconstruction. D'ailleurs, en Figure 4, lorsque le signal est reconstruit uniquement avec ce premier atome, on atteint un signal quasi similaire à celui de la reconstruction complète obtenu en Figure 3.

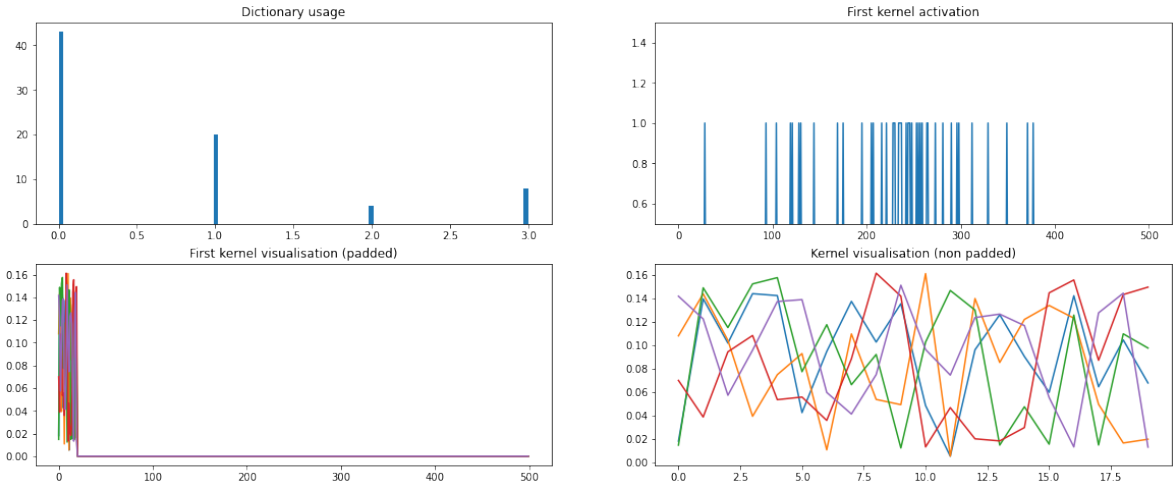

Figure 5: Dictionary usage and most used kernel visualisation

La reconstruction complète et partielle entraînent des taux de reconstruction de respectivement 37% et 20%.

3.4 Deuxième implémentation

Le sous-échantillon que nous avons étudié ici, ainsi que sa reconstruction, sont représentés figure 6. Par rapport au précédent, on remarque qu'il n'a pas de pic.

Après ajustement des paramètres, voici ceux que nous avons retenus. Le dictionnaire est composé de 30 kernels, de taille (10, 5). Une largeur de 10 semble faible, mais la performance est nettement moins bonne quand on augmente la largeur.Pour le sparse coding, nous avons autorisé 50 coordonnées non nulles. L'algorithme a tourné pendant 100 itérations. Nous avons obtenu un taux de reconstruction de 0.44.

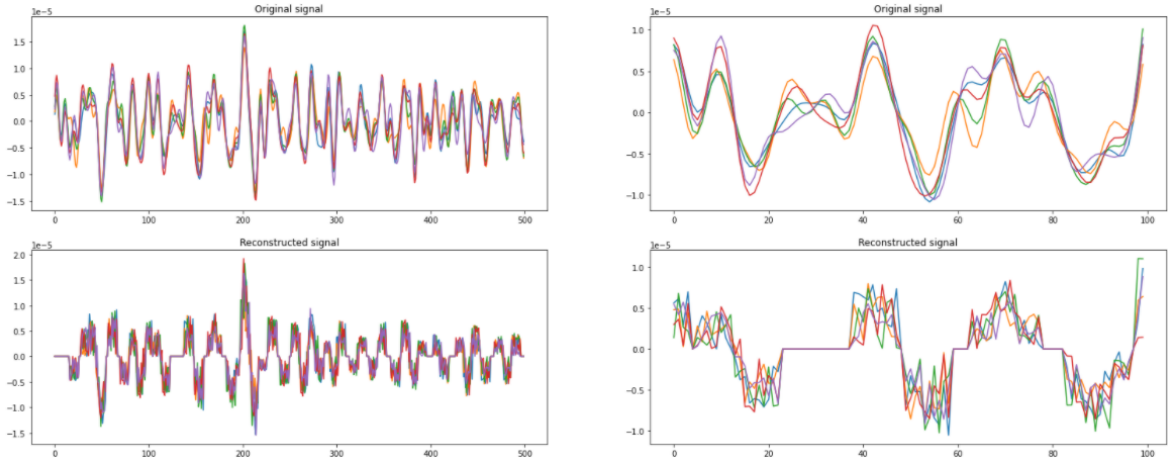

Figure 6: Deuxième signal, et sa reconstruction. Première colonne : signal complet. Deuxième colonne : zoom.

On remarque que l'algorithme parvient à reproduire l'allure générale du signal. En revanche, le signal reconstruit est beaucoup moins lisse que le signal d'origine. Il pourrait être intéressant d'introduire une pénalité qui force les atomes du dictionnaire à être relativement lisses.

4 Conclusion et commentaires

L'algorithme proposé dans cet article pour le traitement d'EEG semble particulièrement pertinent : il tient compte des spécificités spatiales et temporelles des EEG. On peut noter néanmoins que les expériences ne sont pas reproductibles en l'état : des informations manquent. On notera par exemple que pour la mise à jour de la taille des kernels, et pour la mise à jour du learning rate, seules des heuristiques floues sont données. D'ailleurs, concernant le learning rate, nous avons obtenu de meilleures performances en ne suivant pas l'heuristique de l'article.

Nous avons néanmoins réussi à avoir des résultats sur deux échantillons extraits d'enregistrements d'EEG. Notre algorithme a pu reconstruire l'allure générale des signaux, et obtenir des taux de reconstruction corrects. Mais nous avons constaté deux bémols. Premièrement, les signaux reconstruits sont beaucoup moins lisses que les signaux d'origine. Cela peut sûrement être résolu, en introduisant par exemple une pénalité sur les kernels du dictionnaire. Deuxièmement, les atomes du dictionnaire appris ne nous semblent pas vraiment interprétables.

Si cette méthode semble prometteuse et ingénieuse, elle semble finalement plutôt fastidieuse comparée à d'autres méthodes classiques de traitement des EEG, comme l'usage de spectrogrammes.

References

[1] Quentin Barthelemy, Anthony Larue, Aurélien Mayoue, David Mercier, and Jérôme I. Mars. Shift & 2d rotation invariant sparse coding for multivariate signals. 60(4):1597–1611.

[2] Quentin Barthélemy, Cédric Gouy-Pailler, Yoann Isaac, Antoine Souloumiac, Anthony Larue, and Jérôme I. Mars. Multivariate temporal dictionary learning for EEG.

[3] C Brunner, R Leeb, G R Muller-Putz, and A Schlogl. BCI competition 2008 – graz data set a. page 6.

[4] K. Madsen, H. B. Nielsen, and O. Tingleff. IMM OPTIMIZATION WITH CONSTRAINTS 2 nd edition , march 2004.