



TP Factorisations en matrices positives

Roland Badeau

d'après les sujets écrits par U. Simsekli, B. David et P. Magron



Contexte académique } **sans modifications**

Voir page 3

Master Sciences et Technologies - Parcours ATIAM - Module TSM



Les fichiers nécessaires à ce TP sont disponibles sur le Moodle ATIAM. Vous pourrez charger un fichier son sous Matlab en tapant la commande `[x,Fs] = wavread('file.wav')`. Pour l'écouter, vous pourrez taper `soundsc(x,Fs)`. Un modèle de notebook Python `NMF_TP.ipynb` est aussi fourni.

1 Factorisation en matrices positives avec la β -divergence

Dans ce sujet de travaux pratiques, nous traiterons de la factorisation en matrices positives (NMF) avec la β -divergence. Le problème que nous cherchons à résoudre est énoncé comme suit :

$$(\mathbf{W}^*, \mathbf{H}^*) = \operatorname{argmin}_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} \sum_{i=1}^I \sum_{j=1}^J d_{\beta}(x_{ij} \| \hat{x}_{ij}), \quad (1)$$

où x_{ij} est un coefficient de $X \in \mathbb{R}_+^{I \times J}$, c'est-à-dire la matrice de données *positive*, et $\mathbf{W} \in \mathbb{R}_+^{I \times K}$ et $\mathbf{H} \in \mathbb{R}_+^{K \times J}$ sont les matrices de facteurs *positifs* inconnus. Nous définissons également $\hat{x}_{ij} = \sum_k w_{ik} h_{kj}$. La fonction de coût que nous minimisons s'appelle la β -divergence, qui est définie comme suit :

$$d_{\beta}(x \| \hat{x}) = \frac{x^{\beta}}{\beta(\beta-1)} - \frac{x \hat{x}^{\beta-1}}{\beta-1} + \frac{\hat{x}^{\beta}}{\beta}. \quad (2)$$

Lorsque $\beta = 2$, on obtient la distance Euclidienne au carré (EUC), lorsque $\beta = 1$, on obtient la divergence de Kullback-Leibler (KL), lorsque $\beta = 0$ on obtient la divergence d'Itakura-Saito (IS).

L'un des algorithmes les plus populaires pour la NMF est constitué des règles de mise à jour multiplicatives (MUR). L'algorithme MUR comporte les règles de mise à jour suivantes :

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{(X \circ \hat{X}^{\beta-2}) \mathbf{H}^{\top}}{\hat{X}^{\beta-1} \mathbf{H}^{\top}} \quad (3)$$

$$\mathbf{H} \leftarrow \mathbf{H} \circ \frac{\mathbf{W}^{\top} (X \circ \hat{X}^{\beta-2})}{\mathbf{W}^{\top} \hat{X}^{\beta-1}}, \quad (4)$$

où \circ désigne la multiplication coefficient par coefficient et $/$ et \div désignent la division coefficient par coefficient. En suivant la technique que nous avons utilisée dans le cours, dérivez l'algorithme MUR par vous-même.

2 Variantes autour d'un exemple simple

Il s'agit ici de programmer la NMF telle que décrite ci-dessus. Pour cela, on s'appuiera sur le script à compléter `test_nmf.m` qui gère la partie affichage.

2.1 Construction de l'exemple simple

Nous proposons de synthétiser la NMF d'un exemple musical simple : un DO, suivi d'un MI, suivi d'un accord DO/MI, ce que nous noterons DO-MI-DO/MI.

Cet exemple est obtenu en construisant la matrice \mathbf{W} de la base spectrale et la matrice \mathbf{H} des activations de manière synthétique. Aucun son n'est donc calculé ici. Pour cela on pourra par exemple :

- utiliser la fréquence d'échantillonnage $F_e = 8000$ Hz, l'ordre de TFD : $N_{\text{fft}} = 512$.
- calculer les fréquences fondamentales F_0 , du DO et du MI placés sous le LA 440.



- pour chaque note r , fabriquer un spectre harmonique (non négatif) de la forme $\mathbf{W}_r(f) = \sum_{k=1}^K a_k w(f - kF_{0,r})$ où w est une fenêtre usuelle (Hann de largeur 80 Hz par exemple) et a_k décrit l'amplitude des raies (par exemple décroissante, de la forme $a_k = \exp(-k/K)$). K est fixé à l'aide du critère de Shannon. En déduire la matrice spectrale \mathbf{W}_s .
- pour les activations, on utilisera une fonction du type $h_r(t) = \sum_p b_{r,p} h(t - p\Delta T)$ où $h(t) = e^{-t/\tau}$ pour $t \in [0, \Delta T]$ avec ΔT de l'ordre de 0.5 secondes et $h(t) = 0$ ailleurs. b_p vaut 0 ou 1. En déduire la matrice \mathbf{H}_s . Attention, pour se mettre dans des conditions réalistes, on supposera que la représentation est obtenue avec des trames de longueur $N = N_{\text{fft}}$ et un recouvrement de 75%, soit un taux d'échantillonnage de $F_e/(N_{\text{fft}}/4)$ pour les activations temporelles. τ sera pris de l'ordre de $\Delta T/3$.
- construire alors la matrice temps-fréquence $\mathbf{X}_s = \mathbf{W}_s \mathbf{H}_s$.

2.2 NMF classique

- Utiliser les règles de mise à jour multiplicatives pour factoriser en produit de matrices non-négatives la matrice \mathbf{X}_s . Les matrices \mathbf{W} et \mathbf{H} seront initialisées avec des valeurs aléatoires (en respectant toutefois les ordres de grandeur et la normalisation).
- Tracer le graphe de la fonction de coût en fonction de l'itération. Que constatez-vous ?

2.3 Variantes

Testez votre algorithme en faisant varier les données et les paramètres :

- en répétant avec plusieurs tirages d'initialisations (quelles différences constatez-vous sur les résultats ?)
- avec ajout de bruit (attention à respecter la non-négativité, utiliser `rand`)
- avec différentes valeurs de β (0, 1 et 2 typiquement)
- avec plus de notes et des indéterminations du type DO/MI, DO/SOL, DO/MI/SOL
- en fabriquant des notes synthétiques (formes d'onde) par somme de sinusoides et en calculant la TFCT du signal temporel obtenu.

3 Transcription automatique à l'aide de la NMF semi-supervisée

Les notes isolées d'un piano vous sont fournies ainsi qu'un morceau réalisé à l'aide du même instrument. La NMF semi-supervisée consiste à définir la matrice spectrale \mathbf{W} à l'aide des notes isolées (fournies sous la forme de `xxx.wav` où `xxx` est le numéro MIDI, 21-108 pour la tessiture du piano), en calculant simplement une estimation de la densité spectrale de puissance associée. La seule mise à jour restante est donc celle de \mathbf{H} . On travaillera sur des signaux sous-échantillonnés à 22 kHz.

Le script fourni `build_signal.m` permet, à l'aide d'une syntaxe simple, de fabriquer un morceau à partir de notes définies par leur numéro MIDI, leur temps d'attaque, leur durée et la vélocité associée.

1. Constituer la base \mathbf{W} correspondant aux 88 notes du piano
2. Fabriquer un signal test correspondant à la gamme C3-C4.
3. Programmer la NMF semi-supervisée à l'aide des résultats de la partie précédente et l'appliquer au signal test. Observer les activations temporelles obtenues.
4. Fabriquer une fonction de détection des attaques à appliquer sur les activations pour connaître les instants où les notes sont jouées.



Contexte académique } **sans modifications**

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif. Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : sitepedago@telecom-paristech.fr

