Institut Mines-Télécom

TP Factorisations en matrices positives

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

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 :

$$(\mathbf{W}^\star, \mathbf{H}^\star) = \text{argmin}_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} \sum_{f=1}^{F}\sum_{t=1}^{T} d_\beta(x_{ft}\|\hat{x}_{ft}), \tag{1}$$

où x_{ft} est un coefficient de $\mathbf{X} \in \mathbb{R}_+^{F\times T}$, c'est-à-dire la matrice de données *positive*, et $\mathbf{W} \in \mathbb{R}_+^{F\times R}$ et $\mathbf{H} \in \mathbb{R}_+^{R\times T}$ sont les matrices de facteurs *positifs* inconnus. Nous définissons également $\hat{x}_{ft} = \sum_r w_{fr}h_{rt}$. La fonction de coût que nous minimisons s'appelle la β-divergence, qui est définie comme suit :

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

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 :

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

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

où $\circ$ désigne la multiplication coefficient par coefficient et / et $\vdots$ 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.

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

Contexte académique } sans modifications
Voir page 3

— pour chaque note r, fabriquer un spectre harmonique (non négatif) de la forme $\mathbf{W}_r(f) = \sum_{k=1}^{K_r} a_k w(f - kF_{0,r})$ 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 $a_k = exp(-k/K_r)$). K_r est fixé à l'aide du critère de Shannon. En déduire la matrice spectrale $\mathbf{W}_s$.
— pour les activations, on utilisera une fonction $h_r(t) = \sum_p b_{r,p} h(t - p\Delta T)$ 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 $\mathbf{X}_s$. Afin d'éviter que les indéterminations de la NMF ne conduisent à des problèmes numériques, à la fin de chaque itération on renormalisera les colonnes de $\mathbf{W}$ et on multipliera les lignes de $\mathbf{H}$ par les facteurs adéquats afin de conserver le produit $\mathbf{WH}$ inchangé. Les matrices $\mathbf{W}$ et $\mathbf{H}$ seront initialisées avec des valeurs aléatoires.
— 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 ?)
— 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 sinusoïdes 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 Matlab ou la fonction Python fourni(e) `build_signal` 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.

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