MINES Paris
http://moodle.psl.eu *2024-2025*

Décompositions en matrices de faible rang

Ce mini-projet du cours de traitement du signal est à effectuer seul ou en binôme. Dans le cadre de ce mini-projet, nous travaillerons avec une base de morceaux très simplifiée contenant quelques morceaux libres de droits (extraits de la bibliothèque d'audios libres de droits de YouTube), disponible dans les fichiers attachés au projet. Vous pourrez vous appuyer sur les fichiers python pour votre implémentation. Les livrables du mini-projet sont le code et un rapport concis contenant les réponses aux questions et les figures demandées. Le rendu s'effectuera sur Moodle. La date limite de rendu est le 25 avril, aucun rendu ne sera possible après cette date.

1 Séparation de Sources Audio par NMF

Le problème de la séparation de sources audio vise à isoler des sources sonores individuelles à partir d'un mélange de sons. Une technique puissante pour y parvenir est la *factorisation matricielle non-négative* ou *Non-negative Matrix Factorization (NMF)* en anglais. L'objectif de ce mini-projet est d'utiliser cette technique afin d'extraire une piste de violon dans un enregistrement audio mixé.

La NMF est une technique d'algèbre linéaire qui permet de factoriser une matrice non négative V en deux matrices **non négatives** W et H, c'est-à-dire à coefficients positifs. Dans le cadre de signaux audio, nous nous intéressons plus précisément à la factorisation de la matrice représentant le spectrogramme du morceau que nous cherchons à démixer i.e le module de la transformée de Fourier fenêtrée du morceaux. Étant donné une matrice $V \in \mathbb{R}_+^{m\times n}$, la NMF l'approxime par :

$$V \approx W \cdot H$$

où :
- W est la matrice dite de base de taille $m \times k$, représentant les motifs spectraux (par exemple, les profils de fréquence des instruments).
- H est la matrice d'activation de taille $k \times n$, représentant les gains variables dans le temps des motifs spectraux.
- k est le nombre de composantes

On remarque que la NMF réduit la dimensionnalité des données en les représentant avec moins de composantes, puisque $k \ll m$ et $k \ll n$. La matrice W peut être vue comme un dictionnaire qui contient les motifs musicaux de base du morceau, et la matrice H comme une matrice qui indique à quel instant du morceau ces motifs musicaux sont joués.

D'un point de vue pratique, la NMF est calculée en résolvant le problème d'optimisation (non convexe) suivant :

$$\hat{W}, \hat{H} := \mathrm{argmin}_{W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{k \times n}} \|S - WH\|_F^2 \tag{1}$$

sous les contraintes de non-négativité

$$W \geq 0, \quad H \geq 0.$$

Dans la première partie de ce mini-projet, nous nous focalisons sur l'utilisation de la NMF dans le cadre d'un problème de traitement du signal. On pourra donc employer directement l'implémentation de la NMF de la librarie scipy, sans chercher à recoder l'algorithme. Pour isoler la j-ème piste au sein du morceau, on procède de la manière suivante :

— On isole le spectrogramme correspondant à la j-ème piste en effectuant le produit matriciel de la j-ème colonne de W et de la j-ème ligne de H.
— On multiplie le spectrogramme par la phase de la transformée de Fourier fenêtrée du morceau d'origine pour récupérer la transformée de Fourier fenêtrée de la piste.
— On inverse enfin la transformée de Fourier fenêtrée de la piste pour récupérer la piste elle-même.

Question 1: Implémenter le calcul du spectrogramme et la reconstruction du morceau d'origine à partir des amplitudes et des phases du spectrogramme.

Question 2: En supposant que le signal considéré ne contient que $k = 3$ sources, soit exactement le nombre d'instruments du morceau, appliquer la NMF pour isoler trois pistes musicales. Commenter le résultat. Pour mesurer la qualité de la reconstruction, on pourra utiliser la métrique du Signal-to-Reconstruction Error (SRE), définie par :

$$\mathrm{SRE} = 10 \log_{10} \left(\frac{\|S\|_2}{\|S - \hat{S}\|_2} \right),$$

où S est le signal original et $\hat{S}$ la reconstruction du signal.

Question 3: Appliquer la NMF avec $k = 30$ au morceau d'origine. Arrive-t-on maintenant à reconstruire correctement le morceau de musique ?

Question 4: L'étape précédente nous a permis de décomposer le morceau d'origine en $k = 30$ pistes. Proposer une approche basée sur une analyse du spectre de Fourier des différents motifs identifiés dans la matrice W qui permet de déterminer si la piste appartient ou non au violon. Utiliser cette approche pour extraire la piste de violon du morceau.

Question 5: (Bonus) En pratique, rien ne garantit qu'un motif musical de la matrice de base ne contient qu'un unique instrument. Proposer et implémenter une approche de filtrage permettant de mieux isoler le violon sur la piste obtenue.

2 Etude et résolution numérique du problème de NMF

Dans cette partie, pour $S \in \mathbb{R}_+^{m\times n}$, on s'intéresse au problème d'optimisation

$$\begin{array}{cc} \displaystyle\min_{W \in \mathbb{R}^{m\times k},\, H \in \mathbb{R}^{k\times n}} & f(W,H) = \|S - WH\|_F^2 \\ \text{tel que} & W, H \geq 0 \end{array} \tag{2}$$

où $\|M\|_F = \sqrt{\mathrm{Tr}(M^T M)} = \sqrt{\sum_{i,j} m_{ij}^2}$ est la norme de Frobenius de $M \in \mathbb{R}^{m\times n}$ et $W, H \geq 0$ signifie que tous les coefficients de W et H sont positifs ou nuls.

Question 6:

(a) Calculer les gradients partiels de la fonction f. On considèrera le produit scalaire associé à la norme de Frobenius $\langle M_1, M_2\rangle_F = \mathrm{Tr}(M_1^T M_2)$.

(b) Montrer que f est convexe par rapport à X pour W fixé et convexe par rapport à W pour X fixé.

(c) Justifier qu'en revanche f n'est pas convexe. On considèrera le cas $k = 1$, avec les variables $w = W \in \mathbb{R}^m$ et $h = H^T \in \mathbb{R}^n$, et on admettra le résultat dans le cas général.

(d) Justifier qu'il n'y a pas unicité de la solution au problème (2). Montrer qu'il en existe au moins une. (On pourra montrer que l'on peut supposer sans perte de généralité que chaque colonne de W est de norme inférieure ou égale à 1, puis se ramener à H borné, pour conclure.)

Question 7: Une méthode habituellement utilisée pour résoudre ce problème est un algorithme de descente de gradient projeté alternée, comme le suivant.

Algorithme A-PGD (Alternating Projected Gradient Descent)
Inputs : S, W0, H0, maxIter, Tol

i $\leftarrow$ 0, $W \leftarrow W0$, $H \leftarrow H0$

while($\|\nabla_W f(W,H)\| \leq$ Tol) and ($\|\nabla_H f(W,H)\| \leq$ Tol) and ($i \leq$ maxIter)
  $i \leftarrow i{+}1$
  $\alpha_W \leftarrow \mathrm{argmin}_{\alpha\geq 0} f(W - \alpha\nabla_W f(W,H), H)$
  $W \leftarrow P_1(W - \alpha_W \nabla_W f(W,H))$
  $\alpha_H \leftarrow \mathrm{argmin}_{\alpha\geq 0} f(W, H - \alpha\nabla_H f(W,H))$
  $H \leftarrow P_2(H - \alpha_H \nabla_H f(W,H))$

Outputs : W, H

P_1 et P_2 sont les projections sur $\mathbb{R}_+^{m\times k}$ et $\mathbb{R}_+^{k\times n}$ respectivement. Elles garantissent les contraintes $W, H \geq 0$. La descente alternée, suivant d'abord W à H fixée puis suivant H à W fixée, s'inspire du fait que $W \mapsto f(W,H)$ et $H \mapsto f(H,W)$ sont des fonctions convexes.

(a) Déterminer les expressions analytiques des variables $\alpha_W \geq 0$ et $\alpha_H \geq 0$ introduites dans l'algorithme.

(b) Déterminer une formule explicite pour P_1 et P_2.

(c) Implémenter cet algorithme et le tester.

(d) Comparer les résultats de cet algorithme avec :
— ceux fournis par la fonction de la libraire `scikit-learn`
— ceux obtenus par une librairie standard d'optimisation (`scipy.minimize` ou `casadi`, par exemple)

Commenter et anlyser ces résultats.

Comme f est non-convexe, il n'y a a priori pas de raison que l'algorithme A-PGD converge vers un point stationnaire de f, et encore moins vers un minimiseur global.

Néanmoins, dans le cas où l'on cherche à minimiser une fonction Ψ (a priori non-convexe) de la forme

$$\Psi(W,H) = f(W,H) + g(W) + h(H) \tag{3}$$

et sous couvert de certaines hypothèses [1] sur f, g et h , il a été démontré que l'algorithme suivant converge vers un point stationnaire de Ψ.

Algorithme PALM (Proximal Alternating Linearized Minimization)
Inputs : S, W0, H0, $\gamma_W > 1$, $\gamma_H > 1$, maxIter, Tol

i $\leftarrow 0$, $W \leftarrow W0$, $H \leftarrow H0$, $Wp \leftarrow (\mathrm{Tol}+1)^{-1}W0$, $Hp \leftarrow (\mathrm{Tol}+1)^{-1}W0$

while $\left(\frac{\|W-Wp\|}{\|Wp\|} + \frac{\|H-Hp\|}{\|Hp\|} \leq \mathrm{Tol}\right)$ and $(i \leq \mathrm{maxIter})$
$i \leftarrow i{+}1$
$Wp \leftarrow W$
$c \leftarrow \gamma_W L_W(H)$
$W \leftarrow \mathrm{Prox}_{g/c}\left(W - \frac{1}{c}\nabla_W f(W,H)\right)$
$Hp \leftarrow H$
$d \leftarrow \gamma_H L_H(W)$
$H \leftarrow \mathrm{Prox}_{h/d}\left(H - \frac{1}{d}\nabla_H f(W,H)\right)$

Outputs : W, H

Les variables $L_W(H)$ et $L_H(W)$ utilisées dans cet algorithme sont les constantes de Lipschitz des gradients partiels $W \mapsto \nabla_W f(W,H)$ (pour H fixé) et $H \mapsto \nabla_W f(W,H)$ (pour W fixé), qui sont supposées exister. On note que les opérateurs proximaux ne sont ici pas uniquement déterminés, si le problème n'est pas convexe.

Question 8:

(a) Déterminer [2] les constantes de Lipschitz L_W et L_H.

(b) On considère $g = \delta_{W\geq 0}$ et $h = \delta_{H\geq 0}$, où δ_S est la fonction indicatrice associée à un ensemble S définie comme

$$\delta_S(x) = \begin{cases} 0 & \text{if } x \in S \\ +\infty & \text{otherwise} \end{cases} \tag{4}$$

Montrer que Prox_{δ_S} est la projection sur l'ensemble S.

1. Les fonctions peuvent notamment être non-convexes, sous réserve de vérifier une propriété dite de Kurdyka-Lojasiewicz, voir [1].

2. On rappelle ou admet que la norme de Frobenius est sous-multiplicative, c'est-à-dire que $\|AB\|_F \leq \|A\|_F\|B\|_F$, pour toutes matrices A, B de dimensions appropriées.

(c) Comparer dans ce cas l'algorithme PALM correspondant (sans l'implémenter) à l'algorithme A-PGD précédent. Que peut-on en conclure ?

Afin d'arbitrer entre les différentes solutions possibles du problème de NMF, on considère un problème pénalisant la norme des matrices W et H. Dans ce qui suit, on s'intéresse à deux normes différentes.

Question 8: On considère

$$g : W \mapsto \delta_{W\geq 0}(W) + \lambda_W \|\mathrm{vec}(W)\|_1 \text{ et } h : H \mapsto \delta_{H\geq 0}(H) + \lambda_H \|\mathrm{vec}(H)\|_1 \quad (5)$$

avec $\lambda_W, \lambda_H \geq 0$ deux paramètres de pénalité et $\mathrm{vec}(A)$ la forme vectorisée d'une matrice A.

(a) Déterminer $\mathrm{Prox}_{\mu g}$ (resp. $\mathrm{Prox}_{\mu h}$) pour tout $\mu \geq 0$.

(b) Implémenter l'algorithme PALM.

(c) Comparer les solutions obtenues à celle du problème précédent, selon les valeurs choisies de λ_W et λ_H. Commenter leur effet, et celui de la norme $\|\cdot\|_1$ dans la fonction objectif.

Question 9: On considère maintenant

$$g = \delta_{W\geq 0} + \delta_{\|W\|_0 \leq p_W} \text{ et } h = \delta_{H\geq 0} + \delta_{\|H\|_0 \leq p_H} \quad (6)$$

avec p_W, p_H deux entiers naturels et $\|M\|_0$ le nombre de composantes non-nulles[3] d'une matrice M.

(a) Pour une matrice donnée A, on définit la projection suivante

$$T_p(A) = \mathrm{argmin}\left\{\|A - M\|_F^2 \mid \|M\|_0 \leq p\right\}$$

Montrer que $T_p(A)$ est telle que $(T_p(A))_{ij} = A_{ij}$ pour les p indices (i, j) correspondants aux p plus grandes valeurs (en valeur absolue) de A, et $(T_p(A))_{ij} = 0$ sinon.

(b) On définit $\mathcal{I}^+(A) = \{(i, j) \mid A_{ij} > 0\}$ et $\mathcal{I}^-(A) = \{(i, j) \mid A_{ij} \leq 0\}$. Montrer que

$$\mathrm{Prox}_g(A) = \mathrm{argmin}\left\{\sum_{i\in\mathcal{I}^+(A)} (A - M)_{ij}^2 + \sum_{i\in\mathcal{I}^+(A)} A_{ij}^2 \mid M \geq 0\,,\ \|M\|_0 \leq p\right\}$$

puis que

$$\mathrm{Prox}_g(A) = \mathrm{argmin}\left\{\|A - \max\{M, 0\}\|_F^2 \mid \|M\|_0 \leq p\right\}$$

où le maximum est appliqué à chaque composante. Déduire de la question précédente une expression de Prox_g.

(c) Implémenter l'algorithme PALM pour ce nouveau problème.

(d) Constater l'effet des paramètres p_W et p_H. Comparer les solutions obtenues à celles des problèmes précédents et conclure sur l'impact des différents choix de fonction objectif.

3. On observera qu'une telle fonction n'est pas une norme, contrairement à ce que la notation $\|\cdot\|_0$ qui est usuellement employée pourrait faire croire.

Références

[1] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1) :459–494, 2014.