Non-Negative Matrix Factorization

Paul-Henri Icher, Nathan Leclercq, Samuel Carpentier

April 12, 2024

1 Introduction

La Factorisation en Matrices Non Négatives (NMF, pour Non-negative Matrix Factorization) est une technique de réduction de dimensionnalité et d'analyse de données.
Le but de cette approche est d'approximer une matrice par deux autres dont toutes les entrées sont non-négatives.
Cette propriété de non-négativité permet une bonne interprétabilité de la réduction, chaque composantes étant "positives" on n'a pas de calques "négatifs". Par exemple, si on travaille sur des données images, chaque classe correspond à un assemblage fini de composantes. Ce fait est illustré dans la suite du rapport. Chaque composante peut donc, en théorie, être plus facilement interprétable que pour une PCA (chaque composante peut être vu comme un élément simple et significatif et on peut la raccrocher facilement au contexte original).

La technique de NMF est utilisée dans de nombreux domaines comme par exemple :

- L'analyse de texte pour la détection de sujets, l'identification de motifs et de thèmes récurrents dans de grands ensembles de documents, facilitant ainsi la classification et le résumé automatique de textes. On illustrera cela dans la suite de ce document en comparant PCA et NMF pour la classification de texte.
- On peut aussi l'utiliser dans le traitement d'images, ou nous décomposons les données en parties constitutives, ouvrant la voie à des applications telles que la reconnaissance faciale, la segmentation d'images, et l'amélioration de la qualité d'image. Nous réalisons une expérimentation dans la suite de ce document à partir du dataset MNIST.
- On trouve aussi des application de cette algorithme en bioinformatique, dans l'analyse d'expressions génétiques, aidant les chercheurs à découvrir des patterns biologiques significatifs dans des données génomiques complexes [Brunet et al., 2004].

2 Description de l'algorithme

2.1 Description

NMF est appliquée pour décomposer une matrice V de dimensions m×n en le produit de deux matrices W et H, où W est une matrice de dimensions m×k et H une matrice de dimensions k×n, avec k le nombre de composantes ou de caractéristiques cachées. On veut minimiser la différence entre V et le produit WH, sous la contrainte que W et H ne contiennent aucune valeur négative.

2.2 Pseudo-code

- Initialisation : On choisit des valeurs initiales pour W et H.
- Répéter jusqu'à convergence :
  - Mise à jour de H : Ajustez H en minimisant l'erreur de reconstruction avec W fixé.
  - Mise à jour de W : Ajustez W de manière similaire avec H fixé.

Critère d'arrêt : L'algorithme s'arrête lorsque la diminution de l'erreur de reconstruction entre deux itérations successives est inférieure à un seuil prédéfini ou après un nombre maximal d'itérations.

2.3 Initialisation

2 options : un choix aléatoire (risque de se perdre dans un minimum local) ou basé sur les données (approche k-means ou svd pour avoir un départ plus proche de la valeur optimale). Comme les matrices doivent être non-négatives, on préferera NND SVD, une variante de SVD respectant la contrainte de non-négativité.

L'initialisation aléatoire est la plus simple mais potentiellement la moins efficace. Computationnellement parlant, c'est la plus directe. La SVD peut accélérer la convergence en fournissant un point de départ "informé", mais nécessite une modification pour garantir la non-négativité. Comme les matrices sont

des matrices triangulaires, se baser sur les valeurs singulières semble être un bon choix d'initialisation. La NND SVD, en fournissant directement des matrices non-négatives, est souvent considérée comme offrant un bon équilibre entre simplicité et efficacité de convergence. Même argument que pour la SVD, mais sans nécessité de bricolage.

2.4 Choix du β-loss

Le choix du β-loss influence la mesure de distance ou de divergence utilisée pour calculer l'erreur de reconstruction entre V et WH. Le choix dépend souvent du type de données et de l'application :

= 2 : Correspond à l'erreur quadratique, adaptée aux données de comptage ou aux mesures de similarité euclidienne.

$$D(V||WH) = \frac{1}{2}\sum_{ij}(V_{ij} - (WH)_{ij})^2$$

→ cette notation est ressurie aux divergences...

= 1 : Produit une divergence de Kullback-Leibler, utile pour les données probabilistes ou les distributions.

$$D(V||WH) = \sum_{ij}\left(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}\right)$$

Ici c'est bon mais formule fausse

=0 (Itakura-Saito divergence) : Particulièrement utile pour les données audio ou pour les applications en traitement du signal où les différences relatives sont plus importantes que les différences absolues.

$$D(V||WH) = \sum_{ij}\left(\frac{V_{ij}}{(WH)_{ij}} - \log\frac{V_{ij}}{(WH)_{ij}} - 1\right)$$

Valeurs intermédiaires : Des valeurs de entre 1 et 2 peuvent être explorées pour des compromis entre les caractéristiques de ces deux mesures.

Dans ces formules, V_{ij} représente les éléments de la matrice originale V, et $(WH)_{ij}$ les éléments du produit des matrices factorisées WW et HH. Ces différentes mesures de distance ou de divergence sont choisies en fonction de l'application et du type de données à traiter.

L'erreur quadratique est généralement adaptée aux données où la précision de chaque point de donnée est importante. La divergence de Kullback-Leibler convient mieux aux données probabilistes ou aux distributions. La divergence Itakura-Saito est souvent utilisée pour les données de signal ou audio.

2.5 Temps de calcul

Dépends de la dimension des matrices (forcément), du nombre de composantes ciblé (car plus de possibilités) et des critères de convergence (plus c'est petit et plus on doit faire d'itérations).

à mathématiser

3 Application à un premier dataset : Minst.

Nous avons utilisé le datasets MINST composé d'images de chiffres en nuance de gris comprises entre 0 et 9 qui nous permet de comparer la méthode de réduction de dimension NMF à PCA.
Nous avons tout d'abord essayé de visualiser les composantes extraites par les deux algorithmes et leurs utilités sur le premier chiffre du dataset. Puis nous avons étendu l'expérimentation sur les 1000 premières données pour comparer les performances de reconstruction des images à partir des composantes extraites des données originels ainsi qu'une analyse de durées d'exécution de chaque algorithmes.

3.1 Extraction des composantes

Pour cette première visualisation nous initialisons la matrice de NMF aléatoirement ('random') et nous définissons la dimension de la décomposition à 50. Dans la suite du document nous feront exploration des meilleurs paramètres algorithmes pour cette tâche.

NMF est un modèle qui permet de décomposer notre matrice V en 2 matrices W (coefficients) et H (composantes)

Notons que W est de forme 1000x50 et H 50x784 :

- 1000 : Le nombre d'images originels (= V.shape[0])
- 50 : Le nombre de caractéristiques extraites des images originels (Liaison entre W et H)
- 784 : Une image représentant la caractéristiques

hum...

H contient l'ensemble des 50 caractéristiques sous la formes d'images 28x28. (Visualisation sur la figure 1.) W contient 1000 colonnes représentant l'ensemble des poids à appliquer sur les composantes pour recréer une approximation de l'image originel.

3.2 Reconstruction des données

Le processus pour recréer une approximation de l'image originel se fait en mélangeant une association de formes parmi les composantes (Et leur intensité en fonction des poids nécessaires stocké dans la matrice W)

est-ce ainsi que vous expliquerez la reconstruction pour PCA

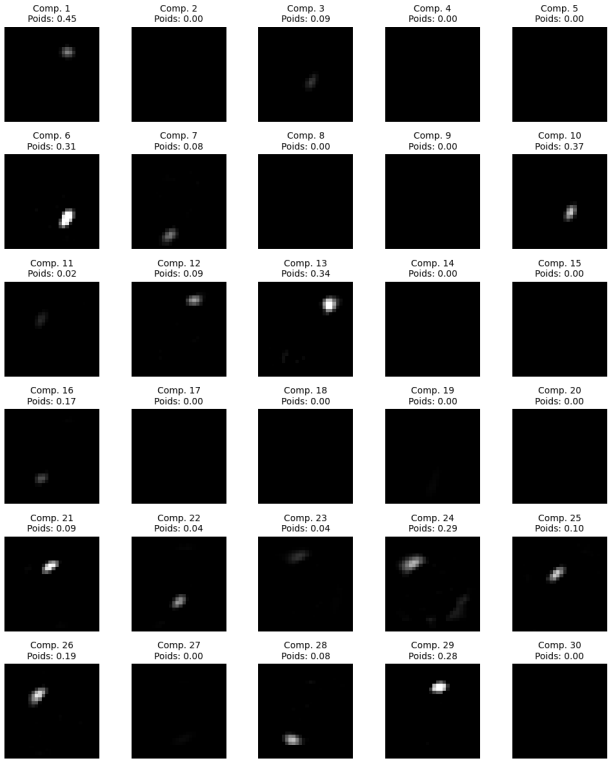

Figure 2: Extrait de visualisation des poids appliqués sur les composantes afin de reconstruire la première image de MINST : Chiffre 5

NMF est très intéressante car les composantes sont non-négatives, permettant une interprétation en termes de contribution additive des caractéristiques permettant ainsi une détection des patterns ou de features sous-jacentes dans les données.

3.2.1 Comparaison entre PCA et NMF

Dans notre expérimentation nous avons aussi essayé d'extraire les composantes apprises par une PCA (Fig 3). Cependant nous pouvons voir que même si les composantes ressemblent vaguement aux "ombres" des chiffres, elles sont beaucoup moins humainement interprétables que les composantes extraites par la NMF car elle représentent les variations majeurs entre les chiffres.

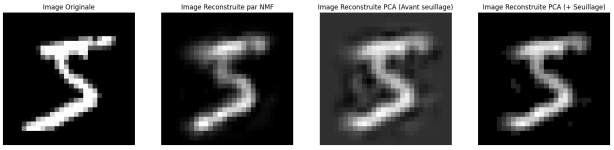

Figure 4: Comparaison de reconstruction de la première image de MINST en ayant utilisé NMF et PCA

L'association des différentes composantes selon les poids prévu dans W permettent donc d'obtenir une reconstruction approximé la première image du dataset. Les deux algorithmes aproxime le 5, cependant on peut voir pour la PCA que le fond de l'image reconstruite est un peu grisé, ce qui nous éloigne de notre image d'origine. La plage des pixels sur l'image d'origine étant compris entre 0 et 1, pour luter contre le phénomène nous avons donc décidé de définir un seuil à 0.1. (Fig 4) Ce qui supprime cette couleur grisâtre et améliorera possiblement la qualité de la reconstruction.

3.3 Hyperparamètre tunning et comparaison des performances

Afin d'optimiser les algorithmes nous utilisons une gridsearch en faisant varier les hyperparamètres tel que :

- Le nombre de composantes à extraire des données
- Pour NMF nous faisons varier l'initialisation des matrices
- Pour NMF, le choix du beta (Cité dans 2.4)

Pour comparer la qualité des reconstruction nous utilisons la moyenne des MSE(Mean Squared Error) entre les images reconstruites et les images d'origines.

Notons que pour notre cas nous ne pouvons pas utiliser le solver 'cd' avec kullback-leiber. Nous utiliseront donc le solver 'mu'. Nous ne pouvons pas non plus utiliser un beta <= 0 quand la matrice d'origine (V) contient des zeros, donc nous n'utiliseront pas itakura-saito. Aussi, le solveur de mise à jour multiplicative ('mu') ne peut pas mettre à jour les zéros présents dans l'initialisation, et conduit donc à de moins bons résultats lorsqu'il est utilisé conjointement avec init='nndsvd'. Nous n'utiliseront donc pas cet initialisation et préféreront nndsvda et nndsvdar à la place. Nous fixons aussi le nombre d'itérations max pour NMF à 200 afin de limiter le temps d'exécution excessif. Nous pourrions augmenter cette valeur pour essayer d'atteindre la convergence.

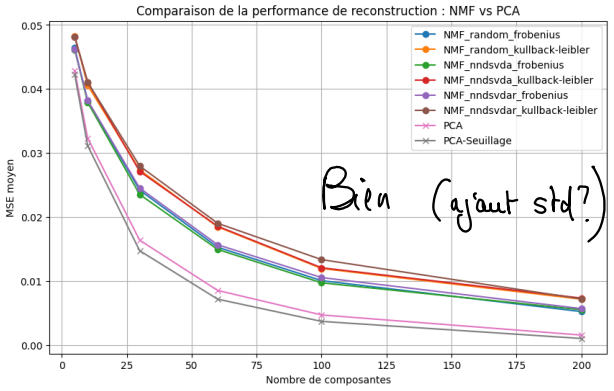

Figure 5: Comparaison de performances utilisant le MSE moyen en fonction du nombre de composantes extraites par la PCA, NMF et leurs variantes

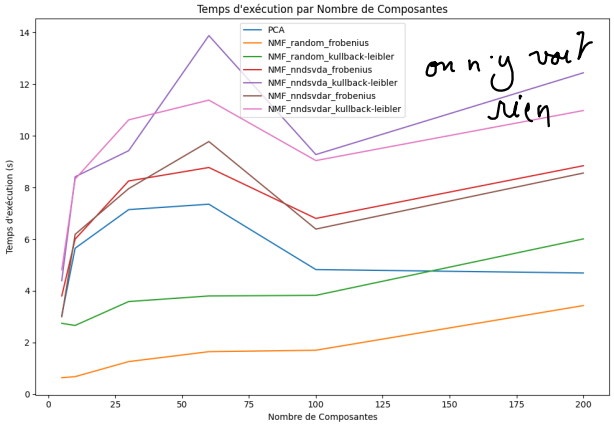

Figure 6: Comparaison des temps d'exécution des algorithmes PCA, NMF et leur variantes en fonction du nombre de composantes extraites.

PCA avec seuillage semble être la meilleure approximation moyenne des images d'origine. L'algorithme semble globalement meilleure que NMF pour la reconstruction d'images à partir des composantes extraites. Notons que nous voyons que l'approximation des chiffres est meilleure en fonction de l'augmentation du nombre de composantes extraites.

ne soyez pas la diplomatie

Ces résultats sont intéressant mais pas très utile car en générale les méthodes de réduction de dimension ne sont souvent pas utilisées seules mais comme prétraitement pour d'autres techniques d'analyse de données ou d'apprentissage automatique.
Sans buts nous pouvons seulement tirer comme conclusion que les caractéristiques extraites d'une NMF sont plus interprétables humainement qu'une PCA.

3.4 Durées d'exécution

Afin de faire une comparaison des rapidités d'exécutions nous avons enregistré les temps d'exécutions pour chaque algorithmes

PCA tend à être assez rapide par rapport à NMF en raison de sa simplicité algorithmique. PCA implique principalement des opérations de décomposition en valeurs singulières qui sont bien optimisées et moins coûteuses en termes de calcul comparativement aux itérations nécessaires pour converger vers une solution en NMF.

La figure 6 nous montre bien que les différentes initialisations et fonctions de coût en NMF (random, nndsvda, nndsvdar; frobenius, kullback-leibler) influencent le temps de convergence.

Selon les courbes l'algorithme le plus rapide semble être le nmf_random_frobenius cependant ces résultats sont à relativiser car nos algorithmes (surtout NMF) sont limité à un nombre d'itérations maximales ce qui peux modifier le graphique et nos interprétation de celui-ci.

4 Application à un second dataset : données textuelles.

5 Conclusion

→ confrontez vous au problème : comment on update W et H ?

→ essayez de trouver une interprétation de vos composantes

→ discuter le choix du nb de composante ou de loss par contre les solvers ont peu d'intérêt ici.

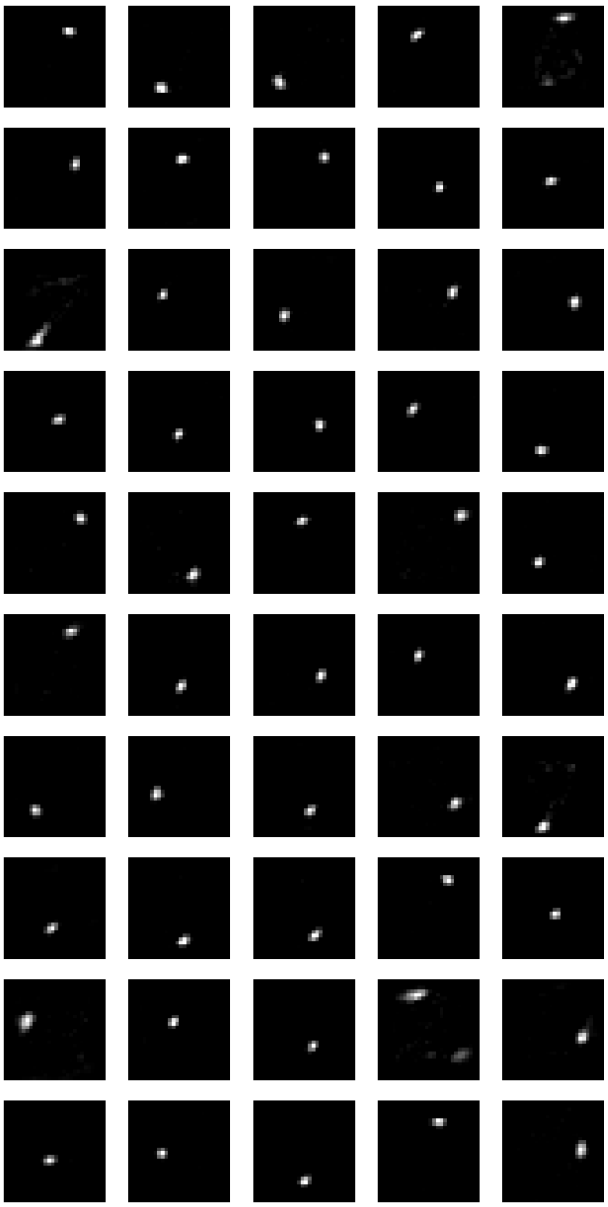

Figure 1: 50 composantes extraite par la NMF à partir d'un dataset de 1000 chiffres

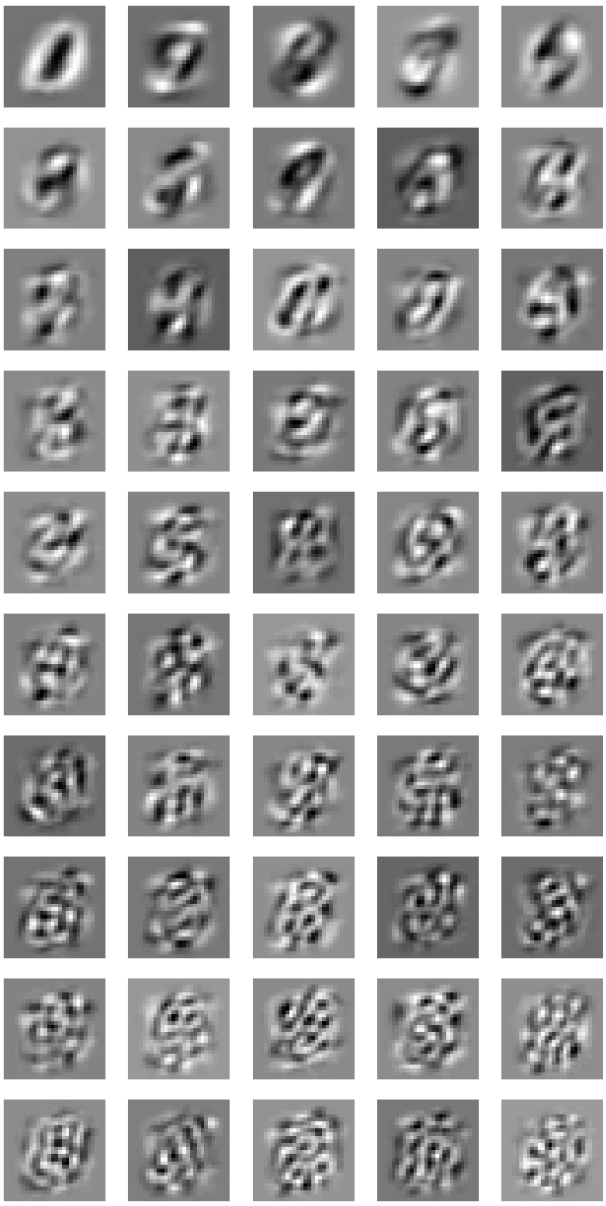

Figure 3: 50 composantes extraites par PCA sur un dataset de 1000 images