# NMF Factorisation par matrices non négatives


## Résumé

*Réduction de dimension par factorisation d'une matrice creuse sous contrainte de non négativité des facteurs. Contrairement à l'ACP les facteurs ne sont pas orthogonaux et ne permettent pas de représentation mais, réduisant la dimension ils permettent classifications non supervisées et modèles de prévision. Description sommaire des nombreuses options de ces algorithmes principalement conçues pour l'analyse des très grandes matrices du e-commerce,* text mining...

*Retour au plan du cours.*


## 1 Introduction

Le principe de la factorisation $\mathbf{X} = \mathbf{UV}'$ d'une matrice est largement utilisé en analyse en composantes principales qui utilise la décomposition en valeurs singulières de la matrice $\mathbf{X}$ (SVD) pour construire des facteurs orthogonaux deux à deux. Paatero et Tapper (1994)[5] puis Lee et Seung (1999)[4] ont proposé une autre décomposition sans contrainte d'orthogonalité mais avec celle de non négativité des matrices des facteurs afin d'en simplifier l'interprétation et sur la base d'une motivation "neuronale" : les neurones ne fonctionnent que de façon additive, pas soustractive. Cette technique a depuis été depuis largement utilisé dans de très nombreux domaines : imagerie, reconnaissance de formes, fouille de textes, systèmes de recommandations, génomique, avec pour objectif d'étudier la structure des très grandes matrices creuses. La bibliographie s'est donc largement développée autour de ce thème en proposant différentes versions de l'algorithme avec différentes initialisations et contraintes, par exemple de parcimonie, dont certaines parallélisables, et tout un ensemble d'applications.

La NMF est donc une technique de réduction de dimension adaptée aux matrices creuses contenant des données positives, par exemple des occurrences ou dénombrements de mots, de pannes... La méthode est donc plus adaptée à certaines situations que la SVD mais cela a un prix ; la complexité algorithmique de la SVD est polynomiale de l'ordre du produit $n \times p$ des dimensions de la matrice. La complexité de la NMF est un problème *nondeterministic polynomial - NP* ; l'existence d'un algortihme de complexité polynomiale est inconnue. En revanche, il existe des approches itératives efficaces mais convergeant vers une solution locale sauf dans des cas très spécifiques (Donoho et Stodden, 2003)[2] ; contrairement à la SVD qui conduit à une solution unique (vecteurs propres et valeurs propres d'une matrice).

Par ailleurs, les facteurs non orthogonaux ne permettent pas de représentation comme en ACP mais sont utilisés comme base d'une classification non supervisée ou préalable à une modélisation pour de l'apprentissage supervisée.

Lee et Seung (1999) illustre cette méthode sur la classification d'un corpus de 30991 articles de l'encyclopédie Grolier. Plutôt que de classer ces articles par thèmes choisis *a priori*, ils sont classés sur la base d'un vocabulaire de 15276 mots. chaque article se décompose (coefficients positifs), en principe parcimonieusement, sur des "facteurs" ou thèmes, eux-mêmes définis chacun par un sous-ensemble petit, jugé "pertinent", de ces mots. En traitement d'images, un corpus se classifie à partir de facteurs ou motifs élémentaires d'images, en génomique par rapport à des "métagènes". L'approche non supervisée est ainsi susceptible de révéler des structures cachées ou des tendances sans *a priori*. Par ailleurs, les facteurs de décomposition n'étant pas orthogonaux, des superpositions apparaissent : des même mots participants à plusieurs thèmes, des gènes à plusieurs fonctions...

## 2 NMF : méthode et implémentations

La description présente de la méthode de NMF ne se veut pas exhaustive ; elle est axée sur l'implémentation réalisée dans le package éponyme par Gaujoux et Seoighe (2010)[3] afin d'en préciser les options et critères mis en œuvre.

### 2.1 Principes

Soit $\mathbf{X}$ une matrice $(n \times p)$ ne contenant que des valeurs non négatives et sans ligne ou colonne ne comportant que des 0 ; $r$ un entier choisi relativement petit devant $n$ et $p$.

La factorisation non-négative de la matrice $\mathbf{X}$ est la recherche de deux matrices $\mathbf{W}_{n\times r}$ et $\mathbf{H}_{r\times p}$ ne contenant que des valeurs positives ou nulles et dont le produit approche $\mathbf{X}$.

$$\mathbf{X} \approx \mathbf{WH}.$$

Le choix du *rang* de factorisation $r << min(n,p)$ assure une réduction drastique de dimension et donc des représentations parcimonieuses. Évidemment, la qualité d'approximation dépend de la parcimonie de la matrice initiale.

La factorisation est résolue par la recherche d'un optimum local du problème d'optimisation :

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \left[L(\mathbf{X},\mathbf{WH}) + P(\mathbf{W},\mathbf{H})\right].$$

$L$ est une fonction perte mesurant la qualité d'approximation et $P$ une fonction de pénalisation optionnelle ; $L$ est généralement soit un critère de moindres carrés (LS ou norme de Frobenius des matrices ou "norme trace"), soit la divergence de Kullback-Leibler (KL) ; $P$ est une pénalisation optionnelle de régularisation utilisée pour forcer les propriétés recherchées des matrices $\mathbf{W}$ et $\mathbf{H}$, par exemple, la parcimonie des matrices ou la régularité des solutions dans le cas de données spectrales.

$$\begin{aligned} LS : L(\mathbf{A},\mathbf{B}) &= \text{tr}\left((\mathbf{A}-\mathbf{B})(\mathbf{A}-\mathbf{B})'\right) = \sum_{i,j}(a_{i,j}-b_{i,j})^2, \\ KL : L(\mathbf{A},\mathbf{B}) &= KL(\mathbf{A}||\mathbf{B}) = \sum_{i,j} a_{i,j}\log(\frac{a_{i,j}}{b_{i,j}}) - a_{i,j} + b_{i,j}. \end{aligned}$$

Dans la librairie NMF de R, construite surtout pour des applications en génomiques, les variables (*features*) sont en ligne et les individus / échantillons (*samples*) sont en colonnes. Ceci n'a pas d'importance lorsque le critère des moindres carrés est utilisé (LS), la résolution est invariante par transposition mais a du sens avec la divergence de Kullback-Leibler qui introduit une dissymétrie entre lignes et colonnes.

**N.B.** Non seulement la solution est locale car la fonction objectif n'est pas convexe en $\mathbf{W}$ et $\mathbf{H}$ mais en plus la solution n'est pas unique. Toute matrice $\mathbf{D}_{r\times r}$ non négative et inversible fournit des solutions équivalentes en terme d'ajustement :

$$\mathbf{X} \approx \mathbf{WDD^{-1}H}.$$

Une fois la factorisation construite il est ensuite facile d'utiliser ces matrices $\mathbf{W}$ et $\mathbf{H}$ pour construire des classifications (CAH, $k$-means), représentations (ACP, MDS), et prévisions à l'aide d'une des nombreuses méthodes d'apprentissage.

## 2.2 Algorithmes

De nombreuses variantes algorithmiques ou sur la forme des pénalisations ont été publiées et implémentées généralement en Matlab, parfois en C, quelques unes spécifiques en R ; Berry et al. (2007)[1] proposent un tour d'horizon de certaines tandis que Gaujoux et Seoighe (2010) en ont implémentées dans R pour rendre facilement possible la comparaison des résultats. Trois familles d'algorithmes sont généralement citées :

- Standard NMF algorithm with multiplicative update,
- Alternate Least Square (ALS) algorithm,
- Descente du gradient.

Chacun de ces algorithmes peut par ailleurs être initialisé de différentes façons :

- plusieurs initialisations aléatoires de $\mathbf{W}$ et $\mathbf{H}$, le meilleur ajustement est conservé,
- non-negative double singular value decomposition (NNSVD),
- une classification ($k$-means) des lignes ou des colonnes,
- parts positives de matrices issues d'une analyse en composantes indépendantes (ACI),
- ...

Entre le choix de la fonction objectif : fonction perte (LS ou KL) et l'éventuelle pénalisation ($L^1$, $L^2$, régularité), le choix de l'algorithme ou d'une de ses variantes, le choix de l'initialisation... cela fait beaucoup d'options à comparer, tester. Comme toujours avec une nouvelle méthode et la pression de publication, de très nombreuses variants apparaissent avant qu'une sélection "naturelle" n'opère pour aboutir à des choix plus efficaces et consensuels d'options en fonction du type de données traitées.

Berry et al. (2007)[1] décrivent très brièvement les principes de ces différents algorithmes et commentent leurs propriétés : convergence, complexité.

L'algorithme initial de Lee et Seung (1999)[4] (*Multiplicative update algorithms*) peut converger vers un point stationnaire pas nécessairement minima

local, voire un point de la frontière même pas point stationnaire. Ces cas sont heureusement rares en pratique mais la convergence est considérée comme lente, demandant plus d'itérations que ses concurrents alors que chaque itération nécessite de nombreux calculs ($O(n^3)$). Les algorithmes de descente du gradient posent des questions délicates concernant le choix des deux pas de descente. La dernière famille d'algorithme : moindres carrés alternés (ALS), exploite le fait que si le problème n'est pas convexe en à la fois $\mathbf{W}$ et $\mathbf{H}$, il l'est soit en $\mathbf{W}$ soit en $\mathbf{H}$. Il suit le principe ci-dessous et possède de bonnes propriétés (convergence, complexité).

---

ALGORITHME 1 : ALS

$\mathbf{W}$ =*random*$(n, r)$
***for*** $i = 1$ à *Maxiter* ***do***
*Résoudre en* $\mathbf{H}$ *:* $\mathbf{W}'\mathbf{W}\mathbf{H} = \mathbf{W}'\mathbf{X}$
*Mettre à* 0 *les termes négatifs de* $\mathbf{H}$
*Résoudre en* $\mathbf{W}$ *:* $\mathbf{H}\mathbf{H}'\mathbf{W}' = \mathbf{H}\mathbf{X}'$
*Mettre à* 0 *les termes négatifs de* $\mathbf{W}$
***end for***

---

L'un des inconvénients du (*Multiplicative update algorithms*) originel est que si un élément des matrices $\mathbf{W}$ ou $\mathbf{H}$ prend la valeur 0, il reste à cette valeur, n'explorant ainsi pas de solutions alternatives. L'ALS est lui plus souple en permettant d'échapper à de mauvaises solutions locales.

La librairie `NMF` de R implémente 11 méthodes ; 9 sont basées sur l'algorithme initial de Lee et Seung (1999)[4] (*Multiplicative update algorithms*) avec différentes options de perte (LS, KL) et de pénalisation ou d'arrêt, deux sont basées sur les moindres carrés alternés (ALS) avec contrainte de parcimonie sur les lignes ou les colonnes. Systématiquement, l'option est offerte, et encouragée, de lancer plusieurs exécutions à partir de plusieurs initialisations aléatoires pour sélectionner les options "optimales" puis, une fois les choix opérés, pour retenir la meilleure parmi un ensemble d'exécutions.

## 2.3 Critères de choix

Les auteurs proposent différents critères pour aider aux choix des méthodes, algorithmes et paramètres, notamment celui du rang $r$ de factorisation, pouvant intervenir au cours d'une étude. Ceux-ci sont illustrés dans la section suivante sur un jeu de données publiques. Un premier tableau (1) fournit des :

- résidus, part de variance expliquée, indice de parcimonie (*sparseness*), pour évaluer la qualité de l'ajustement,
- coefficient de corrélation cophénétique, pureté, entropie ou silhouette pour évaluer la "stabilité" sur plusieurs exécutions.

L'évaluation de la "stabilité" de plusieurs exécutions de NMF repose sur des critères (silhouette, consensus, corrélation cophénétique) issues des méthodes de classification non supervisée. Pour adapter ces critères à la NMF, la notion de classe d'une observation (resp. d'une variable) est remplacée par la recherche du facteur, ou élément de la base (colonne de $\mathbf{W}$ resp. de $\mathbf{H}$), pour laquelle l'observation (resp. la variable) a obtenu la plus forte contribution.

Comme pour le choix d'une dimension, d'un nombre de classes, seules des heuristiques sont proposées dans la littérature pour le difficile choix de $r$ pour lequel il n'y a pas de critère nettement tranché. C'est finalement l'interprétation, biologique ou autre, qui oriente le choix en sous main, ou encore ci-dessous la relative stabilité d'une classification non-supervisée.

## 2.4 Graphiques

La librairie `NMF` propose tout un ensemble de graphiques intégrant chacun une pléthore d'options qu'il serait fastidieux de décrire exhaustivement ; se reporter à la documentation en ligne et à l'article de référence (Gaujoux et Seoighe, 2010)[3]. Leur présentation est largement inspirée des habitudes de la bioinformatique qui mettent en avant des graphes de type *heatmap*.

Des premiers graphiques, non représentés sur l'exemple mais qu'il est facile d'obtenir en exécutant le scénario, visualisent les valeurs (*heatmap*) positives ou nulles des matrices $\mathbf{W}$ (resp. $\mathbf{H}$) de la décomposition. Par défaut une classification ascendante hiérarchique (métrique euclidienne, *average linkage*) des lignes (resp. colonnes) est associée. Il est évidemment possible de modifier ces options par défaut ou d'intégrer les résultats d'une classification exécutée par ailleurs.

Des *consensus maps* sont proposées pour aider au choix de la méthode (figure 1) et au choix de la dimension (figure 3). Attention, ces graphiques sont construits sur les colonnes (variables ou *features*) de la matrice $\mathbf{X}$ et dépendent du choix initial de décomposer la matrice ou de sa transposée. Ces graphiques

montrent si, au cours de plusieurs exécutions de l'algorithme pour différentes méthodes ou pour différentes valeurs du rang $r$, les mêmes variables sont au mieux représentées par le même facteur. C'est donc une information sur la stabilité de l'optimisation obtenue par différentes initialisations.

Les mêmes indicateurs, que ceux présentés dans un tableau (1) pour le choix de la méthode, sont déclinés dans des graphiques (figure 2) avec le rang $r$ des matrices en abscisse.

Enfin, un dernier graphique (figure 4) trace une *heatmap* représentant les valeurs de la matrice initiale $\mathbf{X}$ dans laquelle les lignes et colonnes sont réorganisées par double classification ascendante hiérarchique. Ces classifications sont construites sur les matrices en utilisant par défaut la distance euclidienne et le critère de saut moyen.

TABLE 1 – Critères pour chacune des méthodes testées.

| Méthode | brunet | lee | snmf/l | snmf/r |
|---|---|---|---|---|
| sparseness basis | 0.42 | 0.38 | 0.39 | 0.38 |
| sparseness coef | 0.87 | 0.74 | 0.69 | 0.74 |
| silhouette coef | 0.88 | 0.73 | 0.78 | 0.82 |
| silhouette basis | 0.57 | 0.62 | 0.51 | 0.39 |
| residuals | 23.k | 5.4k | 5.6k | 5.6k |
| niter | 510 | 2000 | 380 | 460 |
| cophenetic | 0.90 | 0.97 | 1.00 | 1.00 |
| dispersion | 0.67 | 0.82 | 1.00 | 0.97 |
| silhouette consensus | 0.50 | 0.84 | 0.98 | 0.95 |

# 3 Exemple

## 3.1 Les données

L'illustration de la factorisation non négative d'une matrice utilise les données décrites dans le scénario explorant les spécificités d'un corpus de pourriels. Elles se présentent sous une forme classique en fouille de texte d'un tableau avec en lignes des messages et en colonnes des nombres ou taux d'occurrences de mots ou caractères spécifiques. La nature des données : matrice très creuse pouvant présenter des valeurs très disparates rend les techniques factorielles habituelles (ACP, AFCM) peu adaptées. Le principal objectif sur ces données est de prévoir le statut *spam* ou non *spam* d'un message en fonction de son contenu et c'est l'objet d'un autre scénario. Il s'agit, dans un premier temps de les décrire, par exemple, en représentant et classifiant les principaux mots clefs.

## 3.2 Choix de méthode, de rang

La méthode (critère et algorithme) "optimale" est choisie en consultant le tableau 1 et les graphiques de la figure method. Sur ces données, il n'est pas difficile de se déterminer pour une méthode de moindres carrés (`snmf/l`) convergeant plus rapidement et présentant des valeurs optimales (cophenetic, residuals,...) ainsi que la meilleure stabilité sur plusieurs exécutions.

Ce choix étant arrêté, les figures 2 et 3 conduisent de façon consensuelle au choix de $r = 5$ : corrélation cophénétique de 1 avant décroissance et meilleur graphique de consensus.

La dernière représentation nécessite évidemment d'être agrandie pour être mieux interprétée. Il est néanmoins facile d'identifier les principaux critères (nombre de lettre capitales ;...) regroupés dans une même classe et correspondant simultanément à une classes de pourriels. En revanche le mot-clef "georges", qui est le prénom du destinataire, est isolé et caractérise des courriels correctes. Une analyse plus fine permettrait d'identifier le rôle d'autres classes de mots.

# Références

[1] Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca et Robert J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis **52** (2007), n° 1, 155 – 173.

[2] David Donoho et Victoria Stodden, *When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts ?*, Advances in Neural Information Processing Systems 16 (S. Thrun, L.K. Saul et B. Schölkopf, réds.), MIT Press, 2004, p. 1141–1148.

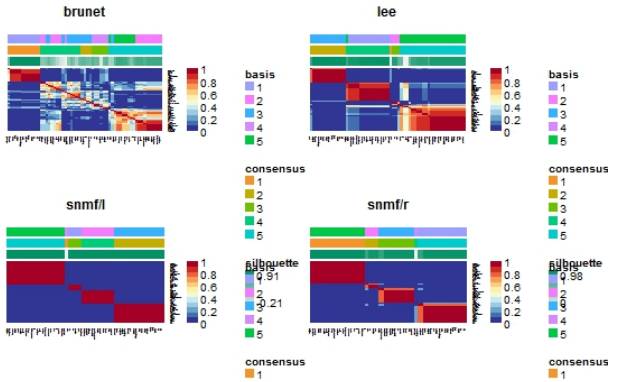


FIGURE 1 – *Spam : Matrice de "confusion" pour chaque méthode de la factorisation par NMF.*

[3] Renaud Gaujoux et Cathal Seoighe, *A flexible R package for nonnegativ matrix factorization*, BMC Bioinformatics **11** (2010), n° 1, 367, http //www.biomedcentral.com/1471-2105/11/367.

[4] D. Lee et S. Seung, *Learning the parts of objects by non-negative matri factorization*, Nature (1999).

[5] Pentti Paatero et Unto Tapper, *Positive matrix factorization : A non negative factor model with optimal utilization of error estimates of dat values*, Environmetrics **5** (1994), n° 2, 111–126.

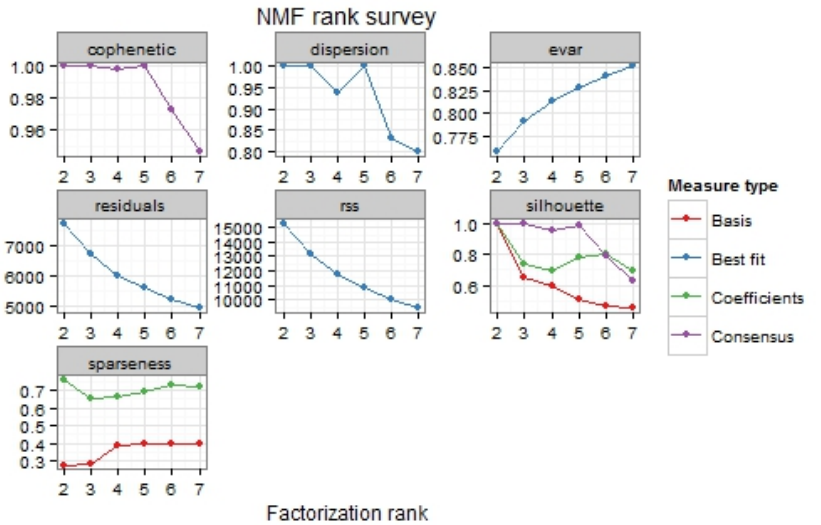


FIGURE 2 – *Spam : Évolution des différents critères en fonction du rang des matrices de la factorisation par NMF.*

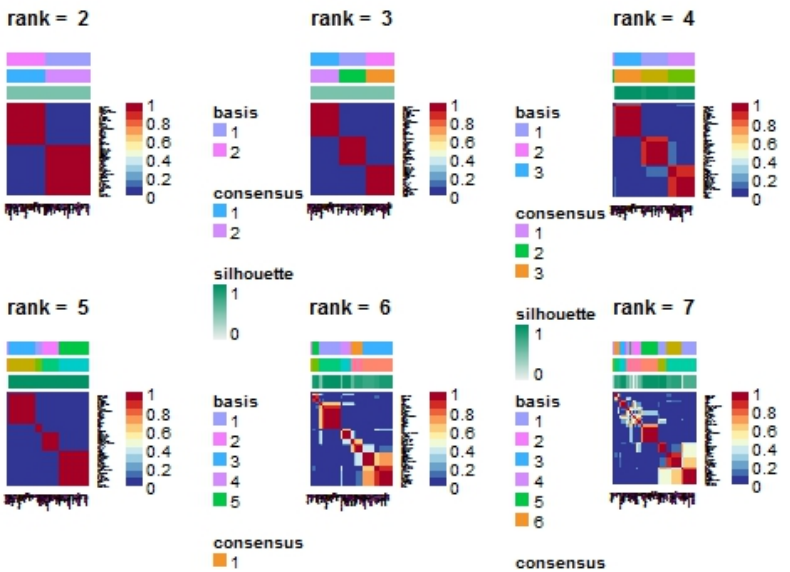


FIGURE 3 – *Spam : Matrice de "confusion" pour chaque valeur de rang des matrices de la factorisation par NMF.*

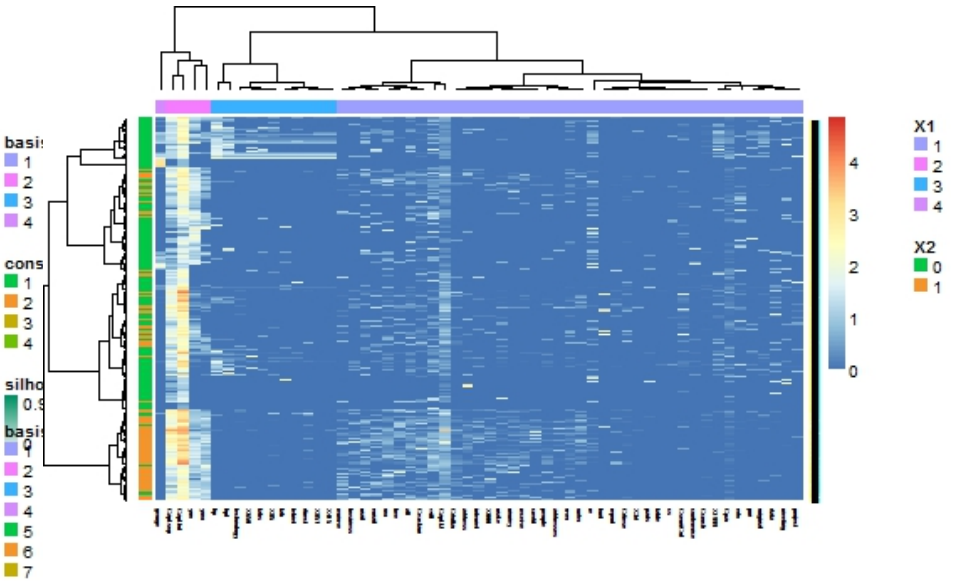


FIGURE 4 – *Spam : Double classification selon les facteurs de la factorisation par NMF et représentation de la matrice creuse initiale ; les message sont en lignes, les mots clefs en colonnes.*