Factorisation bayésienne de matrices pour le filtrage collaboratif

Olivier Gouvert

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique et Télécommunication

Présentée et soutenue par :

M. OLIVIER GOUVERT

le jeudi 19 décembre 2019

Titre :

Factorisation bayésienne de matrices pour le filtrage collaboratif

---

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

M. CEDRIC FEVOTTE

M. THOMAS OBERLIN

Rapporteurs :

M. FRANCOIS CARON, UNIVERSITY OF OXFORD

M. PATRICK GALLINARI, UNIVERSITE PARIS 6

Membre(s) du jury :

M. JEAN-MICHEL MARIN, UNIVERSITE DE MONTPELLIER, Président

M. CEDRIC FEVOTTE, CNRS TOULOUSE, Membre

Mme JOSIANE MOTHE, UNIVERSITE TOULOUSE 2, Membre

M. ROMAIN HENNEQUIN, DEEZER, Membre

M. THOMAS OBERLIN, TOULOUSE INP, Membre

Morty : We should listen to one random song a day, you know ? We'd end up hearing more songs we didn't like, but we'd discover a lot more that we did.
Rick : That is an interesting concept. You know, it makes me wonder if there's an algorithmic expression that could achieve the ideal ratio. Pfft ! Listen to me, trying to calculate happiness over here.
Morty : Hoo, if anyone could, Rick.

Rick and Morty - Rest and Ricklaxation

Remerciements

Cette section de remerciements est essentielle à mes yeux car ce manuscrit n'aurait pas pu exister sans le soutien d'un grand nombre de personnes. Hélas, je ne suis pas sûr de trouver les bons mots pour vous remercier tous comme il se doit. J'espère que vous saurez me le pardonner.

Tout d'abord, je voudrais remercier mes deux directeurs de thèse Cédric et Thomas. Merci de m'avoir fait confiance pour ce sujet de thèse, j'ai énormément appris à vos côtés, autant sur l'aspect scientifique qu'humain. Merci pour votre bienveillance et votre gentillesse, c'était un réel plaisir de travailler avec vous durant ces trois années. Je vous souhaite beaucoup de réussites pour la suite, et merci encore !

Je voudrais ensuite remercier tous les membres de mon jury pour leur investissement. Merci à François Caron et Patrick Gallinari d'avoir rapporté ce manuscrit de thèse. Merci à Romain Hennequin, Jean-Michel Marin et Josiane Mothe d'avoir examiné mon travail.

Louis, je sais que les remerciements sont ta section préférée des manuscrits de thèse, je vais donc essayer d'être à la hauteur de tes espérances. Merci pour ton soutien constant dans les bons et mauvais moments, on a pu tout partager lors cette longue aventure, toujours sans jugement et en ne souhaitant à l'autre que la réussite. Merci aussi de ton aide pour tous les périples administratifs qui ont jalonnés cette thèse. On aura bien travaillé, rigolé et râlé tous les deux.

Un gigantesque merci à toute l'équipe SC, ce fut un plaisir de vous côtoyer. Merci à Marie et Nicolas, vous êtes les garants du fabuleux état d'esprit qui règne dans l'équipe. Un très grand merci à Étienne et Adrien qui m'ont accompagné du premier au dernier jour de ma thèse. Merci aux anciens doctorants : Pierre-Antoine, Yanna et Vinicius, vous avez été un exemple pour nous tous (et, il faut bien l'avouer, un peu une source de stress lorsqu'est venu notre tour de soutenir). Merci à tous les autres doctorants qui sont arrivés ensuite : Maxime, Claire, Camille, Pierre-Hugo (j'espère ne pas t'avoir trop traumatisé, on a fait de belles œuvres d'art ensemble), Asma et Vinicius Second. Vous avez maintenant la responsabilité d'accueillir comme il se doit les futurs doctorants de l'équipe. Merci à tous

les autres membres de l'équipe : Emmanuel, Alberto, Dylan, Sixin, Paul, Mouna, Baha, Tatsumi, Yassine et Dana. Merci aussi à tout le personnel de l'ENSEEIHT, de l'EDMITT, de l'UPS, et d'ELIOR bien entendu.

Je voudrais aussi remercier Pierre Chainais qui m'a transmis cette offre de thèse, et tous ceux qui m'ont amené à me lancer dans le monde de la recherche.

Je voudrais maintenant profiter de cette section pour remercier tous mes amis qui m'ont accompagné durant ces trois années. Tout d'abord, un très grand merci à tous ceux qui ont été mes colocataires. En particulier, je voudrais remercier Jean, Ianis et Charles, vous êtes géniaux (même si vous pourriez faire un peu mieux la vaisselle...). Un grand merci aux membres de ma ligue MPG : Mazz, Amestoy, CDO, Berger et Ggette. Vous m'avez appris à ne jamais reculer face à l'adversité, même face aux plus profondes injustices. Merci à mes amis toulousains : Amiar (qui reste fidèle malgré mon évidente mauvaise foi), Barbara (et les invitations à partager le cassoulet familial), mon équipe de quiz (qui continue à écouter mes réponses fausses), Marianne, Robin, Elene, Roch, etc. Merci aussi à la bande de Pals, Claire, Charlotte, Juliette, Yoyo, Hugo, Anna, et tous les autres. Un très grand merci à Quitterie qui m'a aidé à me lancer dans cette thèse. Un autre très grand merci à Julie.

Enfin, je voudrais remercier toute ma famille pour leurs encouragements. En particulier, merci à mes parents et beaux-parents d'avoir fait le déplacement pour ma soutenance de thèse et d'avoir organisé un superbe pot qui a été la conclusion parfaite à ses trois années. Merci aussi au petit Maël pour ton regard neuf et les conseils que tu as pu me donner.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Notations et acronymes

Notations génériques

a	Scalaire a
$\mathbf{a}$	Vecteur colonne $\mathbf{a}$
a_i	i-ème coefficient du vecteur $\mathbf{a}$
$\mathbf{A}$	Matrice $\mathbf{A}$
a_{ij}	Coefficient (i, j) de la matrice $\mathbf{A}$
$(.)^{\mathrm{T}}$	Transposée
$\mathbf{AB}$	Produit matriciel
$\mathbf{A} \odot \mathbf{B}$	Produit de Hadamard
$\mathrm{diag}(\mathbf{a})$	Matrice diagonale fondée sur $\mathbf{a}$
$\|\|\mathbf{A}\|\|_F$	Norme de Frobenius de $\mathbf{A}$
$\mathbf{0}_K$	Vecteur nul de taille K

Probabilités

$x \sim p(\theta)$	x suit la loi $p(\theta)$
$p(x; \theta)$	Densité associée à la loi p paramétrée par θ
$\mathbb{E}(x)$	Espérance de x
$\mathbb{E}_q(x)$	Espérance de x sous la loi q
$\mathrm{var}(x)$	Variance de x
$\mathcal{H}(q)$	Entropie de la distribution q
δ_a	Loi Dirac localisée en a

La Table A résume les distributions de probabilité usuelles utilisées dans cette thèse.

Fonctions

$\Gamma(.)$	Fonction gamma
$\Psi(.)$	Fonction digamma
$\mathbb{1}[.]$	Fonction indicatrice

Ensembles

$\mathbb{R}$	Ensemble des nombres réels
$\mathbb{R}_+$	Ensemble des nombres réels positifs
(a, b)	Intervalle ouvert
$[a, b]$	Intervalle fermé
$\mathbb{R}^K$	Ensemble des vecteurs de taille K
$\mathbb{N}$	Ensemble des entiers naturels
$\mathbb{N}^*$	Ensemble des entiers naturels strictement positifs

Acronymes

BePo	Bernoulli-Poisson
CAVI	*Coordinate ascent VI*
CF	*Collaborative filtering*
dcPF	*Discrete compound Poisson factorization*
EDM	*Exponential dispersion model*
ELBO	*Evidence lower-bound*
EM	*Expectation-maximization*
IS	Itakura-Saito
KL	Kullback-Leibler
LDA	*Latent Dirichlet allocation*
MAR	*Missing-at-random*
McF	*Matrix co-factorization*
MCMC	*Markov chain Monte Carlo*
MF	*Matrix factorization*
MM	*Majorization-Minimization*
MNAR	*Missing-not-at-random*
NBF	*Negative binomial factorization*
NDCG	*Normalized discounted cumulative gain*
NMF	*Non-negative matrix factorization*
OrdNMF	*Ordinal NMF*
PF	*Poisson factorization*
PPC	*Posterior predictive check*
S-coNMF	*Scale-free non-negative matrix co-factorization*
VBEM	*Variational Bayes EM*
VI	*Variational inference*
WMF	*Weighted matrix factorization*

Tableau A. – Distributions usuelles.

Loi	**Notation**	**Support**	**Paramètres**	**Densité**	**Statistiques**
Normale	$x \sim \mathcal{N}(\mu, \sigma^2)$	$x \in \mathbb{R}$	$\mu \in \mathbb{R}$, $\sigma > 0$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$\mathbb{E}(x) = \mu$, $\text{var}(x) = \sigma^2$
Gamma	$x \sim \text{Gamma}(\alpha, \beta)$	$x \in \mathbb{R}_+$	$\alpha > 0$, $\beta > 0$	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$	$\mathbb{E}(x) = \alpha\beta^{-1}$, $\text{var}(x) = \alpha\beta^{-2}$, $\mathbb{E}(\log x) = \Psi(\alpha) - \log(\beta)$
IG	$x \sim \text{IG}(\alpha, \beta)$	$x \in \mathbb{R}_+$	$\alpha > 0$, $\beta > 0$	$\frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{x}\right)^{\alpha+1} \exp\left(-\frac{\beta}{x}\right)$	$\mathbb{E}(x) = \frac{\beta}{\alpha - 1}$ (pour $\alpha > 1$)
Poisson	$x \sim \text{Poisson}(\lambda)$	$x \in \mathbb{N}$	$\lambda > 0$	$\frac{\lambda^x e^{-\lambda}}{x!}$	$\mathbb{E}(x) = \lambda$, $\text{var}(x) = \lambda$
ZTP	$x \sim \text{ZTP}(\lambda)$	$x \in \mathbb{N}^*$	$\lambda > 0$	$\frac{1}{1-e^{-\lambda}} \frac{\lambda^x e^{-\lambda}}{x!}$	$\mathbb{E}(x) = \frac{\lambda}{1-e^{-\lambda}}$
NB	$x \sim \text{NB}(\alpha, p)$	$x \in \mathbb{N}$	$\alpha > 0$, $p \in (0,1)$	$\frac{\Gamma(x+\alpha)}{x!\Gamma(\alpha)} p^x (1-p)^\alpha$	$\mathbb{E}(x) = \frac{\alpha p}{1-p}$, $\text{var}(x) = \frac{\alpha p}{(1-p)^2}$
Bernoulli	$x \sim \text{Bern}(p)$	$x \in \{0, 1\}$	$p \in [0,1]$	$p^x (1-p)^{1-x}$	$\mathbb{E}(x) = p$, $\text{var}(x) = p(1-p)$
Binomiale	$x \sim \text{Bin}(n, p)$	$x \in \{0, \dots, V\}$	$n \in \mathbb{N}$, $p \in [0,1]$	$\binom{n}{x} p^x (1-p)^{n-x}$	$\mathbb{E}(x) = np$, $\text{var}(x) = np(1-p)$
Mult.	$x \sim \text{Mult}\left(n, \frac{\lambda_k}{\lambda}\right)$	$\mathbf{c} \in \mathbb{N}^K$	$n \in \mathbb{N}$, $\lambda_k > 0$, $\lambda = \sum_k \lambda_k$	$\frac{n!}{\lambda^n} \prod_k \frac{\lambda_k^{c_k}}{c_k!}$	$\mathbb{E}(c_k) = n\frac{\lambda_k}{\lambda}$

Introduction

Ces quinze dernières années, les systèmes de recommandation ont fait l'objet de nombreuses recherches. Ces systèmes ont pour but de recommander aux utilisateurs de plateformes des contenus qu'ils pourraient apprécier. Ils facilitent ainsi la navigation des utilisateurs parmi de larges catalogues de produits. De tels systèmes ont notamment été mis en place pour le commerce (Amazon), la musique (Deezer, Spotify) ou l'audiovisuel (Netflix, Youtube). Ils sont devenus un enjeu majeur pour ces plateformes, leur permettant de fidéliser leurs clients par le biais de recommandations personnalisées. Par exemple, des sites de streaming musical comme Deezer ou Spotify proposent à leurs utilisateurs des playlists hebdomadaires personnalisées.

Les techniques dites de filtrage collaboratif (CF) permettent de faire de telles recommandations à partir des historiques de consommation des utilisateurs. Ces informations sont stockées dans une matrice où chaque coefficient correspond au retour (*feedback*) d'un utilisateur sur un article (*item*). Cette matrice est typiquement de grande dimension (des millions d'utilisateurs qui interagissent avec des millions d'articles) et extrêmement creuse (les utilisateurs n'ont interagi qu'avec une petite portion du catalogue). Les retours collectés par le système peuvent être de deux types : explicites ou implicites. Un retour explicite traduit directement l'intérêt d'un utilisateur, par le biais d'une note par exemple. Au contraire, un retour implicite donne une indication indirecte de ses préférences. Les données implicites sont plus faciles à collecter que les données explicites, mais sont aussi de moins bonne «qualité». Dans cette thèse, nous nous intéressons aux retours implicites, et plus particulièrement aux données de comptage. Ces données, qui correspondent par exemple aux nombres d'écoutes de chansons, sont très courantes en CF, mais sont difficiles à modéliser. Elles sont en effet généralement très bruitées et sur-dispersées, i.e., leur variance est supérieure à leur moyenne.

Le concours *Netflix Prize* [BL+07], lancé en 2006, a popularisé l'utilisation des méthodes de factorisation de matrices (MF) pour le CF. La MF vise à approximer la matrice des retours par le produit de deux matrices de plus petite taille, produisant ainsi une approxi-

mation de rang faible des données. Dans le cadre du CF, les deux matrices ainsi obtenues correspondent aux préférences des utilisateurs et aux attributs des articles. Cela signifie que chaque utilisateur et chaque article sont représentés par un vecteur de petite dimension correspondant à leur «profil». La force d'une interaction est alors mesurée par le produit scalaire du profil d'un utilisateur (ses préférences) avec celui d'un article (ses attributs). La factorisation en matrices non-négatives (NMF) impose une contrainte de non-négativité supplémentaire sur les facteurs de préférences et d'attributs. De telles contraintes favorisent une reconstruction dite par parties et permettent une meilleure représentation des données. L'estimation des facteurs latents permet alors de faire des prédictions afin d'établir des listes de recommandations aux utilisateurs.

Les approches de MF probabilistes supposent que les données observées sont la réalisation d'un processus aléatoire paramétré par une matrice de rang faible. Différents modèles de bruits peuvent alors être choisis pour s'adapter à la nature des données. Par ailleurs, dans cette thèse, nous nous adoptons un point de vue bayésien. Par conséquent, nous assignons des lois a priori aux facteurs latents, et cherchons à estimer la loi a posteriori de ces variables. Nous utilisons notamment des méthodes d'inférence variationnelle afin d'approximer cette distribution insoluble.

L'objectif de cette thèse est de proposer des modèles bayésiens de NMF permettant de modéliser les données de comptage sur-dispersées rencontrées en CF.

Structure du manuscrit

La suite du manuscrit est structurée de la manière suivante.

Le Chapitre 1 introduit les notions clés relatives aux systèmes de recommandation et quelques pré-requis sur les outils d'inférence bayésienne que nous utilisons. En particulier, la dernière section de ce chapitre présente la factorisation Poisson (PF) qui est une variante de la NMF adaptée aux données de comptage. Elle nous servira de référence tout au long de cette thèse. Nous présentons les propriétés importantes de cette méthode et montrons ses limites pour la modélisation de données sur-dispersées. Une façon de contourner ce problème est de pré-traiter les données et de les rendre binaire. Dans la suite de la thèse, nous proposons donc différentes méthodes probabilistes de NMF permettant soit de modéliser directement les données de comptage sur-dispersées, soit de minimiser l'impact du pré-traitement appliqué aux données.

Le Chapitre 2 présente la factorisation binomiale négative (NBF). Ce modèle est fondé

sur la loi binomiale négative qui est une extension naturelle de la loi Poisson permettant de modéliser les données sur-dispersées. En particulier, nous étudions une formulation de la NBF introduisant une variable latente dite d'exposition. Cette variable permet de prendre en compte les variations locales des données, mais son inférence s'avère être coûteuse pour les problèmes de CF.

Le Chapitre 3 présente une instance de la factorisation Poisson dite composée spécialement conçue pour traiter les données de comptage sur-dispersées (dcPF). Ce modèle hiérarchique propose de regrouper les données de comptage en un petit nombre de «sessions d'écoutes». Ainsi, la dcPF permet de rendre les données sur-dispersées «plus factorisables» au sens de la PF. Nous montrons que la dcPF est une extension naturelle de la PF qui correspond à ses cas limites. Contrairement à la NBF, ce modèle permet de traiter des matrices creuses de grande dimension.

Le Chapitre 4 présente une nouvelle façon de travailler avec les données implicites. Dans ce chapitre, les données implicites sont quantifiées de manière à les rendre plus facilement factorisables. Par conséquent, nous proposons un nouveau modèle de NMF probabiliste (OrdNMF) spécialement conçu pour ce type de données dites ordinales.

Le Chapitre 5 s'intéresse à un autre aspect du CF : le problème du démarrage à froid. Afin de résoudre ce problème, nous utilisons des informations supplémentaires sur les articles à recommander et obtenons ainsi des données multimodales. Nous proposons un modèle de co-factorisation de matrices afin de traiter conjointement les deux modalités liées aux articles.

Le chapitre de conclusion (page 125) revient sur les contributions de cette thèse, et propose une taxonomie des différents modèles développés tout au long de ce manuscrit. Différentes perspectives de notre travail sont discutées dans ce chapitre.

L'Annexe A détaille les dérivations des différents algorithmes CAVI développés dans cette thèse. Les algorithmes sont disponibles sur GitHub au lien suivant : `https://github.com/Oligou`.

Liste des publications

[GOF18a]
Gouvert, O., Oberlin, T., et Févotte, C. (2018). Matrix Co-Factorization for Cold-Start Recommendation. In *Proc. International Society for Music Information Retrieval (ISMIR)*.
Disponible en ligne : http://ismir2018.ircam.fr/doc/pdfs/142_Paper.pdf

[GOF19]
Gouvert, O., Oberlin, T., et Févotte, C. (2019). Recommendation from Raw Data with Adaptive Compound Poisson Factorization. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*.
Disponible en ligne : http://auai.org/uai2019/proceedings/papers/17.pdf

[GOF18b]
Gouvert, O., Oberlin, T., et Févotte, C. (2018). Negative Binomial Matrix Factorization for Recommender Systems. Workshop SPARS (résumé de deux pages expertisé par un comité de lecture, présentation poster).
Version longue disponible sur arXiv : https://arxiv.org/pdf/1801.01708.pdf

Chapitre 1.

Pré-requis

Contents

1.1. Systèmes de recommandation

Le but des systèmes de recommandation [AT05 ; RRS11] est de proposer à des «utilisateurs» des «articles» qu'ils sont susceptibles d'aimer. Selon le contexte, ces articles peuvent être des films (Netflix [GH16]), des vidéos (Youtube), des musiques (Spotify, Deezer), des

marchandises (Amazon), etc. On peut classer ces systèmes à partir du type de données utilisées pour établir les recommandations de la manière suivante [BS97].

Filtrage fondé sur le contenu. Les recommandations fondées sur le contenu (*content-based* en anglais) [BC92] sont construites à partir de données disponibles sur les articles. Par exemple, pour la recommandation musicale, on peut utiliser des informations haut niveau (*metadata*) comme l'artiste, l'année de parution, le genre de la chanson (rock, rap, folk) ; ou bas niveau avec des descripteurs audio [vDS13]. L'idée est alors de recommander aux utilisateurs des articles qui partagent des caractéristiques similaires à ceux qu'ils ont précédemment aimés. De la même manière, ces systèmes peuvent être fondés sur des données démographiques concernant les utilisateurs (âge, sexe, lieu de vie). L'une des principales limites de ce genre de filtrage est que les recommandations sont fortement dépendantes de la qualité des informations extérieures utilisées, qui peuvent être difficiles à collecter. Un exemple de filtrage fondé sur le contenu est le service de radio en ligne Pandora, qui fait appel à des experts pour labelliser chaque chanson de sa base de données.

Filtrage collaboratif. Le filtrage collaboratif (CF, *collaborative filtering* en anglais) [Gol+92 ; HKR00] exploite uniquement les retours (*feedbacks* en anglais) qu'ont donnés les utilisateurs sur les articles. Le CF n'utilise aucune information extérieure et peut donc explorer plus librement les relations qui lient les utilisateurs et les articles. Ce genre de filtrage est réputé pour être plus performant que le filtrage fondé sur le contenu [Sla11]. Néanmoins, le CF souffre du problème de démarrage à froid (*cold-start problem*) [Sch+02 ; Lam+08]. Ce terme signifie que le système doit disposer d'un historique de consommation suffisant avant de pouvoir faire des recommandations pour un utilisateur ou un article. De plus, les recommandations obtenues avec les techniques de CF sont difficiles à expliquer et à interpréter [DGG19].

Approches hybrides. Les approches hybrides [Bur02 ; Paz99] permettent de combiner à la fois le filtrage fondé sur le contenu et le filtrage collaboratif. L'introduction de données extérieures permet notamment de résoudre le problème de démarrage à froid. En effet, si un nouvel utilisateur arrive dans le système, ses données démographiques peuvent être exploitées pour lui proposer de nouveaux articles. De la même façon, un nouvel article pourra être recommandé auprès des utilisateurs à partir de leurs metadata, même sans interaction préalable.

Dans cette thèse, on s'intéressera à des approches de type filtrage collaboratif dans les

Chapitres 2, 3 et 4. Dans le Chapitre 5 on étudiera une approche hybride pour résoudre la démarrage à froid.

1.1.1. Filtrage collaboratif

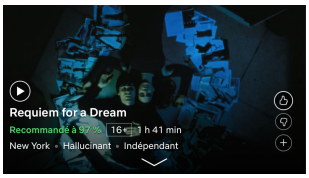
Le filtrage collaboratif est fondé uniquement sur l'historique de consommation des utilisateurs. On note U le nombre d'utilisateurs et I le nombre d'articles disponibles dans le catalogue. Les interactions passées entre les utilisateurs et les articles sont stockées dans une matrice $\mathbf{Y}$ de taille $U \times I$. Chaque coefficient y_{ui} de la matrice correspond au retour d'un utilisateur $u \in \{1, \dots, U\}$ sur un article (*item* en anglais) $i \in \{1, \dots, I\}$. Généralement, la matrice $\mathbf{Y}$ est de très grande dimension (des millions d'utilisateurs qui interagissent avec des millions d'articles). Cependant, cette matrice est généralement très creuse puisque les utilisateurs n'interagissent qu'avec un nombre limité d'articles du catalogue.

Il existe deux approches pour faire des recommandations en CF.

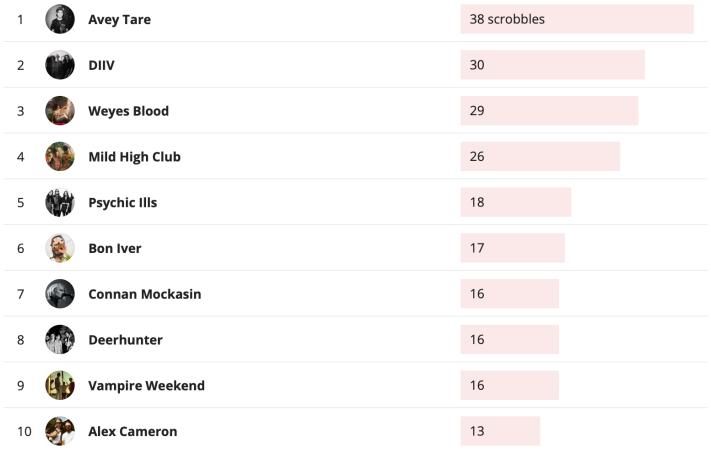
— La première approche est fondée sur la «mémoire» du système (*memory-based* en anglais). On utilise alors des heuristiques fondées sur $\mathbf{Y}$ qui permettent de faire des prédictions. Dans cette catégorie, on retrouve notamment les méthodes dites des plus proches voisins. Grossièrement, ces approches consistent à comparer les utilisateurs [Her+99] (ou les articles [Sar+01]) entre eux à l'aide de mesures de corrélation (en utilisant la corrélation de Pearson par exemple). Ainsi, pour chaque utilisateur, on pourra se baser sur l'historique de consommation de ses voisins pour lui proposer de nouveaux contenus.

— La seconde approche est fondée sur l'apprentissage d'un modèle (*model-based* en anglais). Cette approche se décompose en deux étapes [BHK98]. D'abord, on infère les paramètres du modèle à partir des observations passées (les retours des utilisateurs). Puis, on utilise ces paramètres pour faire des prédictions sur les prochains retours d'utilisateurs. Dans cette thèse on traitera des méthodes dites à facteurs latents. Elles regroupent entre autre : l'analyse sémantique latente probabiliste (PLSA, de l'anglais *probabilistic latent semantic analysis*) [Hof99], l'allocation de Dirichlet latente (LDA, de l'anglais *latent Dirichlet allocation*) [BNJ03], les méthodes de factorisation de matrices [SRJ05] et les réseaux de neurones [Lia+18]. L'un des enjeux majeurs de ces méthodes est la capacité des algorithmes à passer à l'échelle sur des grands jeux de données. Pour cela, on peut chercher à tirer profit de la parcimonie extrême de la matrice $\mathbf{Y}$ (de l'ordre de 1% des données observées).

(a) Allociné



(b) Netflix



(c) Last.fm

FIGURE 1.1. – Illustration de quelques retours explicites ou implicites.

1.1.2. Données explicites et implicites

Les données utilisées en CF correspondent aux retours que font les utilisateurs sur les articles avec lesquels ils interagissent. Ces retours peuvent être classés en deux catégories : les retours explicites et les retours implicites. La Figure 1.1 illustre quelques exemples de retours d'utilisateurs que nous détaillons dans la suite.

Retours explicites. Un retour est dit *explicite* lorsque l'utilisateur donne directement son avis sur un article. Le plus souvent ces retours se font au moyen d'un système de notation. Par exemple, sur le site Allociné un utilisateur peut noter un film sur une échelle d'une à cinq étoiles (voir Figure 1.1(a)), sur Netflix avec un système de *like*/*dislike* (voir Figure 1.1(b)). Ce type de retour est de haute qualité puisqu'il reflète explicitement l'intérêt des utilisateurs. Cependant, ils sont coûteux à collecter puisqu'ils requièrent une action spécifique de leur part.

Les données explicites ne sont que partiellement observées (puisque les utilisateurs n'ont noté qu'une portion des articles). Un retour explicite peut être soit positif (note élevée), négatif (note faible) ou bien manquant (pas de note associée). Le but du système de recommandation est alors de prédire les notes manquantes à partir de celles déjà observées. Cela correspond à un problème de complétion de matrice (voir Section 1.2.1). L'évaluation de ce genre de tâche repose essentiellement sur des métriques d'erreur de prédiction telles que l'erreur quadratique moyenne ou l'erreur absolue moyenne.

Retours implicites. Un retour est dit *implicite* s'il reflète indirectement l'intérêt d'un utilisateur pour un article. Les données implicites correspondent le plus souvent à des historiques de navigation, d'achats, etc. De fait, elles sont très faciles à collecter. Les retours implicites peuvent prendre peuvent prendre plusieurs formes.

— Les données de comptage, i.e., $y_{ui} \in \mathbb{N}$, représentent le nombre de fois où un utilisateur a effectué une action : nombre de clics sur une page web, nombre de fois où un utilisateur a écouté une chanson (voir Figure 1.1(c)). Ces données arrivent souvent par rafale et sont alors sur-dispersées (la variance des données est supérieure à la moyenne). Nous nous intéresserons spécifiquement à ce type de données dans les Chapitres 2, 3 et 4.

— Les données continues, i.e., $y_{ui} \in \mathbb{R}$ ou $\mathbb{R}_+$ peuvent représenter par exemple la valeur d'un panier d'achat ou la somme de dons fait à des associations [BE16].

— Les données binaires, i.e., $y_{ui} \in \{0, 1\}$, représentent si un utilisateur a interagi avec un utilisateur ou non. Elles sont aussi appelées données à une classe [Pan+08 ; Sin+10 ; PK13] puisque seuls les retours positifs sont collectés (une absence de retour peut à la fois signifier que l'utilisateur ne connaît pas ou n'aime pas l'article).

Contrairement aux données explicites, les données implicites sont entièrement observées. Elles sont aussi de moins bonne qualité et très bruitées [HKV08]. Par exemple, une écoute d'un utilisateur sur une chanson ne traduit pas forcément une préférence, l'utilisateur peut être déçu ou bien ne pas être présent lorsque la chanson est jouée. Au contraire, un très haut nombre d'écoutes ne traduit pas forcément une plus forte préférence, l'utilisateur peut préférer une chanson qu'il écoute modérément mais régulièrement (correspondant à un classique intemporel) à une chanson qu'il écoute sans arrêt pendant une courte durée (correspondant à un succès éphémère).

Le but du système de recommandation va être ici de prédire les futures interactions des utilisateurs. Pour chaque utilisateur, les recommandations se présentent sous la forme d'une liste personnalisée d'articles. Sur les sites de streaming musical comme Spotify ou Deezer, on retrouve ce type de recommandation via des playlists personnalisées qui sont créées chaque semaine.

1.1.3. Évaluation des listes de recommandations

Pour chaque utilisateur, on propose une liste ordonnée de recommandations contenant L articles avec lesquels il n'a jamais interagi. Cette liste est construite à partir d'un score de prédiction s_{ui} calculé à partir des paramètres inférés du modèle choisi. Nous utilisons

la métrique NDCG (pour *normalized discounted cumulative gain* en anglais) [JK02] pour évaluer la qualité de cette liste de recommandations.

On définit, pour chaque utilisateur le DCG (*discounted cumulative gain*) comme :

$$\mathrm{DCG}_u = \sum_{l=1}^{L} \frac{\mathrm{rel}(u,l)}{\log_2(l+1)}, \tag{1.1}$$

où $\mathrm{rel}(u,l) \in \mathbb{R}$ correspond à la pertinence du l-ième article de la liste de recommandations (plus cette valeur est élevée, plus l'article est pertinent). Le numérateur récompense la présence d'articles pertinents dans la liste, tandis que le dénominateur pénalise leur position dans la liste. Plus un article pertinent est placé en fin de liste plus il sera pénalisé. Cela traduit le fait qu'un utilisateur portera plus d'attention au début qu'à la fin de la liste. On définit le DCG idéal (IDCG) comme le score de DCG qu'obtiendrait un oracle qui classe tous les articles les plus pertinents en tête de liste. Par conséquent, on peut définir la version normalisée du DCG :

$$\mathrm{NDCG}_u = \frac{\mathrm{DCG}_u}{\mathrm{IDCG}_u}. \tag{1.2}$$

Ainsi pour chaque utilisateur on a un score de NDCG qui appartient à $[0,1]$ (puisque $\mathrm{DCG}_u \leq \mathrm{IDCG}_u$), où $\mathrm{NDCG}_u = 1$ correspond à la liste parfaite. Le NDCG d'un modèle de recommandation est défini comme la moyenne des NDCG de tous les utilisateurs.

Il existe d'autres critères pour mesurer la qualité d'une liste de recommandations comme la précision et le rappel. Dans nos expériences ces métriques n'apportent pas d'information supplémentaire par rapport au NDCG, nous ne développerons donc pas ces métriques dans cette thèse. Le lecteur intéressé peut se référer à [Her+04] pour plus de détails sur l'évaluation dans les systèmes de recommandation.

Pertinence. En pratique, la pertinence d'un article $\mathrm{rel}(u,l)$ est mesurée via un ensemble de test $\mathbf{Y}^{\mathrm{test}}$ préalablement mis de côté. Soit $i(l)$ l'indice de l'article placé en l-ième position de la liste de recommandations, nous choisissons de travailler dans cette thèse avec une pertinence définie par :

$$\mathrm{rel}_s(u,l) = \mathbb{1}[y^{\mathrm{test}}_{ui(l)} > s], \tag{1.3}$$

où $s \geq 0$ est un seuillage. Nous avons donc ici $\mathrm{rel}_s(u,l) \in \{0,1\}$. Un article est donc considéré comme pertinent pour un utilisateur si sa valeur dans l'ensemble de test est strictement

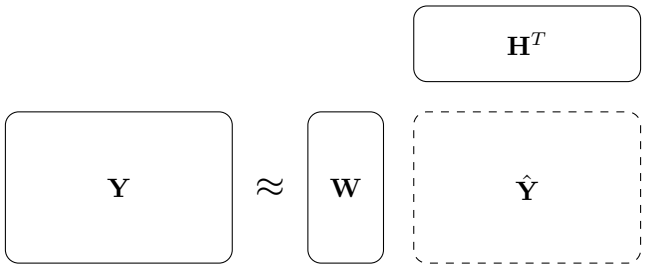


FIGURE 1.2. – Illustration de la factorisation de matrices.

supérieure à un certain seuil s. On note par NDCGs, le score de NDCG moyen calculé avec un seuil s. Par exemple, si les données collectées correspondent à des nombres d'écoutes de chanson, le NDCGs correspond au score NDCG où les articles sont jugés pertinents seulement lorsqu'ils ont été écoutés plus de s fois.

Autres critères d'évaluation. L'évaluation des systèmes de recommandation est un problème difficile [Her+04]. D'autres critères d'évaluation existent pour mesurer la diversité des recommandations [Can+11], l'exploration de la queue lourde des données, la sérendipité (fait de découvrir par hasard de nouveaux contenus), etc.

1.2. Factorisation de matrices

Parmi les méthodes à facteurs latents, la factorisation de matrices (MF, *matrix factorization* en anglais) est devenue très populaire en filtrage collaboratif à la suite du concours *Netflix Prize* [BL+07 ; BK07]. La MF consiste à chercher une approximation de rang faible $\hat{\mathbf{Y}}$ de la matrice d'observation $\mathbf{Y}$. Autrement dit, on s'intéresse à l'approximation suivante :

$$\mathbf{Y} \approx \mathbf{W}\mathbf{H}^T = \hat{\mathbf{Y}}, \tag{1.4}$$

où $K \ll \min(U, I)$ est le nombre de facteurs latents, $\mathbf{W} \in \mathbb{R}^{U \times K}$ est la matrice des préférences des utilisateurs et $\mathbf{H} \in \mathbb{R}^{I \times K}$ est la matrice des attributs des articles[1]. La Figure 1.2 représente cette approximation. Par conséquent, la MF suppose que chaque utilisateur $u \in \{1, \ldots, U\}$ est représenté par un vecteur de préférences $\mathbf{w}_u \in \mathbb{R}^K$ (qui correspond à une ligne de $\mathbf{W}$), et que chaque article $i \in \{1, \ldots, I\}$ est représenté par un vecteur d'attributs $\mathbf{h}_i \in \mathbb{R}^K$ (qui correspond aux lignes de $\mathbf{H}$) [KBV09]. Ces deux vecteurs appartiennent

1. Dans le cadre de l'apprentissage de dictionnaire, on dit que la matrice $\mathbf{W}$ correspond au dictionnaire et est composé d'atomes, et que la matrice $\mathbf{H}$ correspond aux activations.

au même espace latent $\mathbb{R}^K$. De plus, la force d'une interaction entre un utilisateur et un article est mesurée par le produit scalaire : $\langle \mathbf{w}_u, \mathbf{h}_i \rangle = \sum_{k=1}^{K} w_{uk} h_{ik}$, où w_{uk} et h_{ik} sont les coefficients respectifs des matrices $\mathbf{W}$ et $\mathbf{H}$. Une fois l'approximation calculée, la matrice $\hat{\mathbf{Y}}$ permet de faire des prédictions et des recommandations aux utilisateurs.

1.2.1. Complétion de matrices

Les premiers travaux de CF se sont focalisés sur le traitement de données explicites. Dans ce cas, le but du système de recommandation est de compléter les valeurs manquantes de la matrice $\mathbf{Y}$ [CR09 ; CP10]. Les prédictions de font alors à l'aide de l'approximation de rang faible $\hat{\mathbf{Y}}$.

Lorsque $\mathbf{Y}$ est une matrice pleine (dont tous les coefficients sont observés), la décomposition en valeurs singulières (SVD, de l'anglais *singular value decomposition*) permet d'obtenir la meilleure approximation de rang faible de $\mathbf{Y}$ au sens de la norme de Frobenius. Pour cela, il suffit de sélectionner les K plus grandes valeurs singulières de la décomposition. Cependant, dans le cadre du CF appliqué à des données explicites, la matrice $\mathbf{Y}$ n'est que partiellement observée. On ne peut donc plus appliquer cette méthode pour obtenir l'approximation. Pour pallier ce problème, les auteurs de [Sar+00] proposent de compléter artificiellement les valeurs manquantes de la matrice $\mathbf{Y}$ avant d'appliquer la SVD, mais ces méthodes sont coûteuses et sujettes au sur-apprentissage.

Une autre approche proposée notamment dans [SRJ05 ; RS05] propose de travailler uniquement sur les valeurs observées de la matrice $\mathbf{Y}$. On définit par $\mathcal{O} = \{(u, i) \text{ tel que } y_{ui} \text{ est observée}\}$ l'ensemble des données observées. Ainsi, le problème de complétion de matrice peut être formulé comme un problème d'optimisation :

$$\min_{\mathbf{W},\mathbf{H}} \sum_{(u,i)\in\mathcal{O}} \Big(y_{ui} - [\mathbf{W}\mathbf{H}^T]_{ui}\Big)^2 + \gamma(\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2). \tag{1.5}$$

où γ est un paramètre de régularisation et $\|.\|_F$ est la norme de Frobenius. Les termes $\|\mathbf{W}\|_F^2$ et $\|\mathbf{H}\|_F^2$ permettent de régulariser les matrices $\mathbf{W}$ et $\mathbf{H}$, afin d'éviter le sur-apprentissage. La fonction de coût, qui est non-convexe par rapport aux variables $\mathbf{W}$ et $\mathbf{H}$, peut être par exemple minimisée par une descente de gradient (les méthodes de second ordre ne convenant pas en général aux problèmes de grande dimension). Ce problème d'optimisation peut aussi être interprété comme l'estimation au sens du maximum de vraisemblance d'un modèle à bruit additif gaussien. Ainsi, des modèles bayésiens ont aussi été développés à partir de cette observation [SM07].

Hypothèses sur les données manquantes. Soit $\mathbf{M}$ la matrice indicatrice de taille $U \times I$ qui correspond au motif d'observation des données, i.e., $m_{ui} = \mathbb{1}[(u,i) \in \mathcal{O}]$. Dans la majeure partie des cas, on fait l'hypothèse que le motif d'absence $\mathbf{M}$ est déterministe. Cependant des approches récentes ont proposé de considérer le motif $\mathbf{M}$ comme la réalisation d'une variable aléatoire [LR14]. Différentes hypothèses ont été développées sur le processus générant le motif $\mathbf{M}$. Les données manquantes peuvent l'être au hasard (MAR, *missing-at-random* en anglais), ou pas au hasard (MNAR, *missing-not-at-random* en anglais). Dans le deuxième cas, le motif d'absence $\mathbf{M}$ contient alors de l'information et doit être pris en compte lors de l'inférence des paramètres d'intérêt. Nous invitons le lecteur intéressé à se référer à l'article [Sea+13] pour une discussion sur la signification de ces hypothèses. Des modèles de recommandation sous hypothèses MNAR ont été récemment proposés [Mar+07 ; MZ09 ; HHG14]. L'inférence se fait alors à la fois à partir de la matrice des retours partiellement observée et sur la matrice indicatrice complètement observée $\mathbf{M}$.

1.2.2. Factorisation pondérée de matrices (WMF)

Plusieurs travaux ont tenté d'adapter les méthodes MF aux données implicites. La matrice $\mathbf{Y}$ est alors totalement observée et est très creuse. Les modèles de MF classiques ont tendance à surestimer l'importance des zéros de $\mathbf{Y}$. De ce fait, plusieurs approches ont proposé de pondérer l'influence des valeurs nulles [Pan+08 ; Cai+10 ; PK13]. En particulier, la factorisation pondérée de matrices (WMF, de l'anglais *weighted matrix factorization*) [HKV08] consiste à minimiser la fonction de coût suivante :

$$C(\mathbf{W},\mathbf{H}) = \sum_{ui} \omega_{ui} \left(m_{ui} - [\mathbf{W}\mathbf{H}^T]_{ui}\right)^2 + \gamma(\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2), \tag{1.6}$$

où $m_{ui} = \mathbb{1}[y_{ui} > 0]$, et $\omega_{ui} \geq 0$ sont des coefficients de pondération. Dans [HKV08], les auteurs proposent de définir les coefficients de pondération en fonction des données implicites, i.e., $\omega_{ui} = f(y_{ui})$ où f est une fonction croissante prédéfinie. Par exemple, elle peut être choisie de la forme $f(y_{ui}) = 1 + a\log(1 + \varepsilon^{-1}y_{ui})$ où a et ε sont deux scalaires. Dans ce modèle, les retours implicites ne sont donc pas directement modélisés mais servent de pondération au modèle de factorisation. Ils correspondent à une confiance associée au retour binaire m_{ui} (qui correspond au fait qu'un article soit consommé ou non) et non pas directement à un intérêt pour l'article. Cette fonction de coût peut être minimisée à l'aide d'une méthode des moindres carrés alternés. Le pré-calcul de certaines statistiques permet d'obtenir un algorithme passant à l'échelle sur des données de grande dimension, et qui est parallélisable [Zha+06]. Contrairement à ces travaux, nous développerons dans la suite de

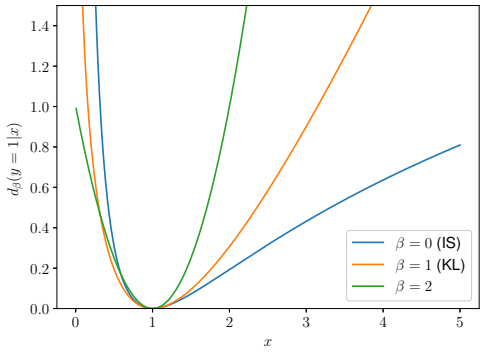


FIGURE 1.3. – β-divergence pour $\beta \in \{0, 1, 2\}$.

la thèse des modèles probabilistes permettant de modéliser directement les données brutes.

1.2.3. Factorisation en matrices non-négatives (NMF)

La factorisation en matrices non-négatives (NMF, *non-negative matrix factorization* en anglais) est une méthode de MF avec des contraintes de non-négativité [2] supplémentaires sur les matrices $\mathbf{W}$ et $\mathbf{H}$ [LS99 ; LS01]. Ces contraintes permettent d'imposer une approximation dite constructive de la matrice $\mathbf{Y}$. En effet, un coefficient y_{ui} est approximé par la somme de K valeurs positives $\sum_{k=1}^{K} w_{uk}h_{ik}$. Ainsi, la NMF est grandement appréciée pour l'interprétation rendue possible des matrices $\mathbf{W}$ et $\mathbf{H}$ (voir l'article [LS99] qui utilise la NMF pour décomposer des images de visages). Le problème d'optimisation lié à la NMF peut se formaliser de la sorte :

$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{Y}|\mathbf{W}\mathbf{H}^T), \text{ s.c. } \mathbf{W} \geq 0, \mathbf{H} \geq 0, \tag{1.7}$$

où $\mathbf{W} \geq 0$ exprime le fait que chaque coefficient de la matrice $\mathbf{W}$ est positif, et D est une divergence mesurant la dissemblance entre $\mathbf{Y}$ et $\mathbf{WH}$, telle que $D(\mathbf{Y}|\mathbf{W}\mathbf{H}^T) = \sum_{ui} d(y_{ui}|[\mathbf{W}\mathbf{H}^T]_{ui})$. Contrairement à une distance, une divergence peut ne pas être symétrique. La divergence entre deux scalaires est toujours positive, i.e., $d(y|x) \geq 0$ et est nulle uniquement en $x = y$.

2. On préférera utiliser le terme non-négativité à positivité pour éviter la confusion avec la notion de matrices positives utilisée en algèbre.

Cas de la β-divergence. Plusieurs choix de divergences ont été étudiés dans la littérature pour mesurer l'erreur de reconstruction, dont n-otamment la β-divergence [FI11] qui est une famille de divergences continue paramétrée par un scalaire $\beta \in \mathbb{R}$. La β-divergence entre deux scalaires y et x est définie comme suit :

$$d_\beta(y|x) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(y^\beta + (\beta-1)x^\beta - \beta y x^{\beta-1}\right), & \text{pour } \beta \in \mathbb{R}\setminus\{0,1\}, \\ \frac{y}{x} - \log\frac{y}{x} - 1, & \text{pour } \beta = 0, \\ y\log\frac{y}{x} - y + x, & \text{pour } \beta = 1. \end{cases} \tag{1.8}$$

La β-divergence regroupe plusieurs cas particuliers qui sont couramment utilisés dans la littérature : l'erreur quadratique pour $\beta = 2$; la divergence de Kullback-Leibler généralisée (KL) pour $\beta = 1$ qui est associée à la loi Poisson (voir Section 1.4) ; la divergence d'Itakura-Saito (IS) pour $\beta = 0$ qui est souvent utilisée en audio [FBD09]. Ces trois cas particuliers de la β-divergence sont illustrés dans la Figure 1.3.

Une propriété importante de la β-divergence est son comportement par changement d'échelle. Soit $\omega > 0$, la β-divergence respecte :

$$d_\beta(\omega y|\omega x) = \omega^\beta d_\beta(y|x). \tag{1.9}$$

Cela permet d'avoir une indication sur le paramètre β à choisir pour un problème donné. En effet, la β-divergence pénalise plus fortement les grandes valeurs si $\beta > 0$, et les faibles valeurs si $\beta < 0$. Si $\beta = 0$ (IS), la divergence est invariante par changement d'échelle et pénalise de la même façon les fortes et faibles valeurs. C'est une propriété recherchée en audio où l'on travaille avec des spectrogrammes de puissance. La distribution Tweedie [Twe84] donne une interprétation probabiliste à la β-divergence. En effet, la log-vraisemblance de la distribution Tweedie correspond à la β-divergence à un signe près [YC12 ; TF13].

Dans le reste de la thèse, nous nous intéresserons exclusivement à des problèmes de NMF. Nous adopterons un point de vue probabiliste dans les Chapitres 2, 3 et 4, et un point de vue optimisation dans le Chapitre 5.

1.3. Inférence bayésienne

Dans cette section, nous détaillons les outils d'inférence bayésienne que nous utiliserons tout au long de cette thèse (Chapitres 2, 3 et 4). En particulier, nous nous intéressons à l'inférence variationnelle [Jor+99 ; BKM17].

Soit $\mathbf{y} = \{y_1, \dots, y_N\}$ un ensemble de données observées que l'on suppose tirées de la distribution jointe $p(\mathbf{y}, \mathbf{z}) = p(\mathbf{y}|\mathbf{z})p(\mathbf{z})$, où $\mathbf{z} = \{z_1, \dots, z_M\}$ est un ensemble de variables latentes, $p(\mathbf{z})$ est la loi a priori assignée à ces variables, et $p(\mathbf{y}|\mathbf{z})$ est la vraisemblance associée aux données. L'inférence bayésienne repose sur l'analyse de la distribution a posteriori définie par $p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{y},\mathbf{z})}{p(\mathbf{y})}$, où $p(\mathbf{y}) = \int_{\mathbf{z}} p(\mathbf{y}, \mathbf{z})$ est appelée l'évidence. Souvent, l'évidence est trop coûteuse voire impossible à calculer, ce qui rend la loi a posteriori insoluble. L'estimation de cette loi nécessite alors des approximations.

Il existe deux paradigmes majeurs pour approximer cette distribution : les méthodes de Monte-Carlo par chaînes de Markov (MCMC, de l'anglais *Markov chain Monte Carlo*) et l'inférence variationnelle (VI, de l'anglais *variational inference*).

1.3.1. Méthodes de Monte-Carlo par chaînes de Markov

L'idée des méthodes MCMC est d'approcher la distribution a posteriori par un échantillonnage, sous la forme d'une somme de lois de Dirac :

$$p(\mathbf{z}|\mathbf{y}) \approx \frac{1}{J} \sum_{j=1}^{J} \delta_{\mathbf{z}^{(j)}}(\mathbf{z}), \tag{1.10}$$

où $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(J)}\}$ est un ensemble de J échantillons de $p(\mathbf{z}|\mathbf{y})$.

La génération de ces échantillons se fait à l'aide d'une chaîne de Markov dont la distribution stationnaire est la distribution d'intérêt (i.e., la loi a posteriori). Plusieurs schémas de chaînes de Markov ont été étudiées dans la littérature. En particulier, l'échantillonnage de Gibbs consiste à tirer séquentiellement chaque variable aléatoire z_i selon la loi conditionnelle $p(z_i|\mathbf{y}, \mathbf{z}_{-i})$, où $\mathbf{z}_{-i} = \mathbf{z} \setminus \{z_i\}$. Dans cette thèse, nous travaillons exclusivement avec des modèles augmentés qui nous permettent d'obtenir les lois conditionnelles. Ainsi, des échantillonneurs de Gibbs pourraient être développés pour chaque modèle présenté.

Les méthodes MCMC fournissent un échantillonnage de la distribution a posteriori $p(\mathbf{z}|\mathbf{y})$ qui est asymptotiquement exact. Cependant, ces méthodes sont réputées pour converger lentement et leur convergence est difficile à évaluer.

1.3.2. Inférence variationnelle

Principe de l'inférence variationnelle. L'inférence variationnelle (VI) [BKM17] consiste à approximer au sens de la divergence KL la loi a posteriori insoluble $p(\mathbf{z}|\mathbf{y})$ par une distribution variationnelle $q(\mathbf{z})$ plus simple. La distribution variationnelle est recherchée parmi

une famille $\mathcal{F}$ qui contrôle la complexité de l'approximation. Ainsi, le problème d'inférence devient un problème d'optimisation défini par :

$$\min_{q \in \mathcal{F}} \mathrm{KL}(q(\mathbf{z}) | p(\mathbf{z}|\mathbf{y})) \tag{1.11}$$

où la divergence KL est donnée par $\mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y})) = \mathbb{E}_q(\log q(\mathbf{z})) - \mathbb{E}_q(\log p(\mathbf{z}|\mathbf{y}))$. La divergence KL est positive, i.e., $\mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y})) \geq 0$, et nulle lorsque $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{y})$.

Il est important de noter que la VI n'est pas une méthode asymptotiquement exacte. En effet, elle ne permet pas d'atteindre la distribution a posteriori $p(\mathbf{z}|\mathbf{y})$ (sauf si $p(\mathbf{z}|\mathbf{y}) \in \mathcal{F}$). Le choix de la famille $\mathcal{F}$ est donc très important puisqu'il régit la qualité de l'approximation.

Borne inférieure de l'évidence. La minimisation de la divergence KL est impossible en pratique puisqu'elle nécessite le calcul de l'évidence $p(\mathbf{y})$. Pour contourner ce problème, on peut réécrire la divergence comme suit [CM07] :

$$\begin{aligned} \mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y})) &= \mathbb{E}_q(\log q(\mathbf{z})) - \mathbb{E}_q\left(\log \frac{p(\mathbf{y},\mathbf{z})}{p(\mathbf{y})}\right) && (1.12) \\ &= \mathbb{E}_q(\log q(\mathbf{z})) - \mathbb{E}_q(\log p(\mathbf{y},\mathbf{z})) + \log p(\mathbf{y}). && (1.13) \end{aligned}$$

De ce fait, on obtient la décomposition de l'évidence suivante :

$$\begin{aligned} \log p(\mathbf{y}) &= \mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y})) + \mathrm{ELBO}(q), && (1.14) \\ \text{où } \mathrm{ELBO}(q) &= \mathbb{E}_q(\log p(\mathbf{y},\mathbf{z})) - \mathbb{E}_q(\log q(\mathbf{z})), && (1.15) \end{aligned}$$

avec $\mathcal{H}(q) = -\mathbb{E}_q(\log q(\mathbf{z}))$ l'entropie de la distribution variationnelle. La divergence KL étant positive, nous obtenons une borne inférieure de l'évidence (ELBO, pour l'anglais *evidence lower-bound*), i.e., $p(\mathbf{y}) \geq \mathrm{ELBO}(q)$. D'après l'Éq. (1.14), minimiser la divergence KL équivaut donc à maximiser la ELBO, ce qui est plus simple en pratique. Le problème d'optimisation devient donc :

$$\max_{q \in \mathcal{F}} \mathrm{ELBO}(q). \tag{1.16}$$

1.3.3. Algorithme CAVI

Famille du champ moyen. Un choix commun pour le choix de $\mathcal{F}$ est la famille du champ moyen (*mean-field family* en anglais). Dans cette famille, on suppose que la distribution est

entièrement factorisable par rapport à chaque variable latente z_i, i.e.,

$$q(\mathbf{z}) = \prod_i q_i(z_i). \tag{1.17}$$

Cela signifie que les variables $\{z_1, \ldots, z_M\}$ sont indépendantes sous la distribution q. On note $q_{-i} = \prod_{j \neq i} q_j$ la distribution associée à l'ensemble de variables $\mathbf{z}_{-i}$.

La famille du champ moyen est couramment utilisée en VI car elle permet d'obtenir des algorithmes très simples à mettre en place. D'autres choix de familles plus complexes permettent d'obtenir de meilleures approximations de la loi a posteriori. Dans cette thèse, nous nous intéressons exclusivement à l'hypothèse de champ moyen.

Algorithme d'inférence variationnelle avec optimisation par bloc. Le problème d'estimation de la loi a posteriori a donc été réécrit sous la forme d'un problème d'optimisation :

$$\max_q \text{ELBO}(q), \text{ avec } q(\mathbf{z}) = \prod_i q_i(z_i). \tag{1.18}$$

Les algorithmes de VI avec optimisation par bloc (CAVI, *coordinate ascent VI* en anglais) consistent à optimiser séquentiellement la ELBO par rapport à chaque composante q_i, en supposant q_{-i} fixée, i.e.,

$$\max_{q_i} \text{ELBO}(q_i), \forall i \in \{1, \ldots, M\}. \tag{1.19}$$

Ainsi, à chaque itération de l'algorithme la ELBO croît. Dans cette thèse, nous choisissons de stopper nos algorithmes CAVI lorsque la différence relative de la ELBO entre deux itérations consécutives passe sous un seuil τ, i.e.,

$$\frac{\text{ELBO}(q^{(t)}) - \text{ELBO}(q^{(t-1)})}{\text{ELBO}(q^{(t-1)})} < \tau, \tag{1.20}$$

où $q^{(t)}$ correspond à la distribution variationnelle à l'itération t.

L'utilisation conjointe de l'hypothèse de champ moyen et de l'optimisation par bloc permet d'obtenir des règles de mise à jour analytiques pour les distributions q_i. En effet, la ELBO

par rapport à la loi q_i peut être réécrite sous la forme :

$$\begin{aligned}\text{ELBO}(q_i) &= \mathbb{E}_{q_i}(\mathbb{E}_{q_{-i}}(\log p(\mathbf{y},\mathbf{z}))) - \mathbb{E}_{q_i}(\log q_i(z_i)) - \mathbb{E}_{q_{-i}}(\log q_{-i}(\mathbf{z}_{-i})) && (1.21)\\ &= \mathbb{E}_{q_i}\left(\log \frac{e^{\mathbb{E}_{q_{-i}}(\log p(\mathbf{y},\mathbf{z}))}}{q_i(z_i)}\right) + cste && (1.22)\\ &= -\,\text{KL}(q_i(z_i)|d_i(z_i)) + cste, && (1.23)\end{aligned}$$

où $d_i(z_i) \propto e^{\mathbb{E}_{q_{-i}}(\log p(\mathbf{y},\mathbf{z}))}$ est une distribution. Maximiser la ELBO par rapport à q_i est alors équivalent à minimiser la divergence KL entre q_i et d_i. Sans contrainte supplémentaire sur la famille de distributions $\mathcal{F}$ (nous avons seulement imposé l'hypothèse de champ moyen), ce minimum est atteint en $q_i = d_i$. La solution du problème d'optimisation est donc donnée par :

$$\begin{aligned}q(z_i) &\propto \exp\left(\mathbb{E}_{q_{-i}}(\log p(\mathbf{y},\mathbf{z}))\right) && (1.24)\\ \text{ou } q(z_i) &\propto \exp\left(\mathbb{E}_{q_{-i}}(\log p(z_i|\mathbf{y},\mathbf{z}_{-i}))\right). && (1.25)\end{aligned}$$

Ainsi, pour obtenir la distribution variationnelle q_i il suffit de calculer l'espérance sous q_{-i} de la distribution jointe $p(\mathbf{y},\mathbf{z})$ ou de la distribution conditionnelle $p(z_i|\mathbf{y},\mathbf{z}_{-i})$. Comme mentionné précédemment, dans cette thèse, nous travaillons exclusivement avec des modèles augmentés où les distributions conditionnelles sont connues.

Inférence variationnelle stochastique. L'inférence variationnelle étant un problème d'optimisation sur les paramètres de distributions, des algorithmes de gradient stochastique peuvent être implémentés. Ces algorithmes proposent de traiter les données par paquets (*batch* en anglais), ce qui peut être particulièrement intéressant sur les gros jeux de données. Nous invitons le lecteur intéressé à se référer à l'article [Hof+13] pour le développement de tels algorithmes pour la VI. Dans cette thèse, nous développons uniquement des algorithmes CAVI. Cependant, des algorithmes stochastiques pourraient être facilement dérivés pour chaque modèle proposé.

1.3.4. Estimation des paramètres et algorithme EM

Dans les sections précédentes, nous avons vu comment obtenir une approximation de la loi a posteriori des variables latentes $\mathbf{z}$. Une autre tâche d'intérêt est de pouvoir apprendre conjointement les paramètres Φ qui contrôlent la distribution des données.

L'estimateur du maximum de vraisemblance marginalisé permet d'obtenir une estimée de

ces paramètres. Il consiste à résoudre le problème d'optimisation suivant :

$$\max_{\Phi} \log p(\mathbf{y}; \Phi). \tag{1.26}$$

Afin de maximiser cette quantité, nous utilisons un algorithme d'espérance-maximisation (EM, *expectation-maximization* en anglais) fondé sur l'égalité présentée précédemment en Éq. (1.14) :

$$\log p(\mathbf{y}; \Phi) = \mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y}; \Phi)) + \mathrm{ELBO}(q; \Phi) \tag{1.27}$$

$$= \underbrace{\mathrm{KL}(q(\mathbf{z})|p(\mathbf{z}|\mathbf{y}; \Phi))}_{\text{étape E}} + \underbrace{\mathbb{E}_q(\log p(\mathbf{y}, \mathbf{z}; \Phi))}_{\text{étape M}} + cste, \tag{1.28}$$

où *cste* est une constante par rapport aux paramètres du modèle Φ.

Les algorithmes EM consistent à alterner entre deux étapes : l'étape d'espérance (E) et l'étape de maximisation (M).

— L'étape E consiste à réduire l'écart entre la distribution q et la loi a posteriori $p(\mathbf{z}|\mathbf{y}; \tilde{\Phi})$, où $\tilde{\Phi}$ est la valeur courante des paramètres Φ. Comme nous l'avons vu précédemment, cet écart, mesuré par la divergence KL, est nul pour $\tilde{q}(\mathbf{z}) = p(\mathbf{z}|\mathbf{y}; \tilde{\Phi})$. On peut alors calculer l'espérance :

$$Q(\Phi|\tilde{\Phi}) = \mathbb{E}_{\tilde{q}}(\log p(\mathbf{y}, \mathbf{z}; \Phi)) \tag{1.29}$$

$$= \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}; \tilde{\Phi}) \log p(\mathbf{y}, \mathbf{z}; \Phi) \mathrm{d}\mathbf{z}. \tag{1.30}$$

— L'étape M consiste à maximiser l'espérance calculée dans l'étape précédente, i.e.,

$$\max_{\Phi} Q(\Phi|\tilde{\Phi}). \tag{1.31}$$

L'algorithme EM suppose que l'on ait accès à la loi a posteriori $p(\mathbf{z}|\mathbf{y})$. Lorsque cela n'est pas le cas, on peut utiliser une approximation de cette loi, fondée soit sur les méthodes MCMC (on parle d'algorithme MCEM pour *Monte Carlo EM*), soit sur les méthodes variationnelles (on parle alors d'algorithme VBEM pour *variational Bayes EM*).

L'algorithme VBEM, que nous utilisons tout au long de cette thèse, revient donc à optimiser alternativement le terme $\mathrm{ELBO}(q; \Phi)$ par rapport à q (étape E) puis par rapport à Φ (étape M). Cela nous permet d'obtenir à la fois une approximation de la loi a posteriori et une estimation des paramètres Φ. L'algorithme VBEM est résumé en Algorithme 1.

Algorithme 1 : Algorithme VBEM

Données : Données $\mathbf{y}$
Résultat : Distribution variationnelle q et paramètre Φ

Initialisation aléatoire de q et Φ;
répéter
  pour chaque $i \in \{1, \dots, M\}$ **faire**
    $q_i(z_i) \propto \exp\left(\mathbb{E}_{q_{-i}}(\log p(\mathbf{y}, \mathbf{z}))\right)$;
  fin
  $\Phi = \operatorname{argmax}_\Phi \mathbb{E}_q(\log p(\mathbf{y}, \mathbf{z}; \Phi))$;
  Calculer ELBO(q, Φ);
jusqu'à *ELBO converge*;

1.4. Factorisation Poisson (PF)

La factorisation Poisson (PF, pour *Poisson factorization* en anglais) [Can04 ; BJ06 ; Cem09 ; Ma+11 ; GHB15] est une méthode de NMF probabiliste permettant de modéliser des données de comptage. Tout au long de cette thèse, nous proposerons des extensions de la PF, qui nous servira donc de méthode référence.

La PF suppose que chaque observation est tirée d'une distribution Poisson :

$$y_{ui} \sim \text{Poisson}([\mathbf{W}\mathbf{H}^T]_{ui}), \tag{1.32}$$

avec $y_{ui} \in \mathbb{N}$. Les matrices $\mathbf{W}$ et $\mathbf{H}$ sont supposées non-négatives, ce qui implique que le produit matriciel $\mathbf{W}\mathbf{H}^T$ est lui aussi non-négatif (hypothèse nécessaire pour le paramètre d'une distribution Poisson).

La PF est particulièrement adaptée aux cas où les données collectées dans $\mathbf{Y}$ correspondent à des données de comptage. En effet, la distribution Poisson permet de décrire le nombre d'événements se produisant lors d'un intervalle de temps fixé. Comme nous l'avons vu précédemment, ces données sont particulièrement courantes en CF (par exemple des nombres d'écoutes, des nombres de cliques sur une page web, etc.).

1.4.1. Divergence associée

La divergence associée à la distribution Poisson est la divergence de KL généralisée. En effet, la log-vraisemblance d'une observation peut s'écrire sous la forme :

$$-\log p(y_{ui}|\mathbf{W},\mathbf{H}) = -y_{ui}\log[\mathbf{W}\mathbf{H}^T]_{ui} + [\mathbf{W}\mathbf{H}^T]_{ui} + \log y_{ui}! \tag{1.33}$$

$$= \mathrm{KL}(y_{ui}|[\mathbf{W}\mathbf{H}^T]_{ui}) + cste, \tag{1.34}$$

où *cste* est une constante par rapport aux variables $\mathbf{W}$ et $\mathbf{H}$.

1.4.2. Modèle augmenté

En utilisant la propriété de superposition de la loi Poisson [Cem09 ; GHB15], le modèle PF peut être augmenté de la façon suivante (modèle graphique donné en Figure 1.4) :

$$c_{uik} \sim \mathrm{Poisson}(w_{uk}h_{ik}), \tag{1.35}$$

$$y_{ui} = \sum_k c_{uik}. \tag{1.36}$$

L'introduction de ces variables latentes permet notamment de simplifier les règles de mises à jour liées aux matrices $\mathbf{W}$ et $\mathbf{H}$. En effet, chaque colonne de $\mathbf{W}$ et $\mathbf{H}$ sont indépendantes sachant ces variables latentes additionnelles.

La distribution conditionnelle de la variable latente $\mathbf{c}_{ui} = (c_{ui1},\ldots,c_{uiK})^T$ est donnée par :

$$\mathbf{c}_{ui}|y_{ui},\mathbf{W},\mathbf{H} \sim \mathrm{Mult}(y_{ui},\boldsymbol{\phi}_{ui}), \tag{1.37}$$

où Mult correspond à la distribution multinomiale et $\boldsymbol{\phi}_{ui}$ est un vecteur de probabilités de taille K ayant pour coefficients $\phi_{uik} = \frac{w_{uk}h_{ik}}{[\mathbf{W}\mathbf{H}^T]_{ui}}$. Si $y_{ui} = 0$, alors on a $\mathbf{c}_{ui} = \mathbf{0}_K$ où $\mathbf{0}_K$ est le vecteur nul de taille K. De ce fait, la variable latente $\mathbf{C}$ est partiellement connue et n'a besoin d'être estimée que pour les valeurs non nulles de la matrice $\mathbf{Y}$ seulement. Son coût computationnel est d'autant plus faible que la matrice $\mathbf{Y}$ est creuse, ce qui est le cas en pratique dans les problèmes de CF.

1.4.3. Notion de budget

On peut introduire une variable de budget définie par $b_u = \sum_i y_{ui}$ et qui correspond au nombre total d'interactions d'un utilisateur. Le modèle PF peut être réécrit sous la forme [GHB15] :

$$b_u \sim \text{Poisson}\left(\sum_k w_{uk}\left(\sum_i h_{ik}\right)\right), \tag{1.38}$$

$$(y_{u1}, \ldots, y_{uI})^T | b_u \sim \text{Mult}\left(b_u, \boldsymbol{\phi}_u\right), \tag{1.39}$$

où $\boldsymbol{\phi}_u$ est un vecteur de taille I ayant pour coefficients $\phi_{ui} = \frac{[\mathbf{W}\mathbf{H}^T]_{ui}}{\sum_i [\mathbf{W}\mathbf{H}^T]_{ui}}$. Dans cette réécriture, on tire d'abord le budget b_u associé à un utilisateur. Ensuite, ce budget est réparti entre les différents articles en fonction des préférences de l'utilisateur et des attributs de chaque article. De la même façon, on peut réécrire le modèle PF en introduisant une variable de budget pour chaque article à répartir entre les différents utilisateurs. De ce fait, la PF modélise conjointement les budgets associés aux utilisateurs et aux articles. La PF pondère naturellement l'effet des zéros de $\mathbf{Y}$ en limitant le budget des utilisateurs et des articles [GHB15].

Remarque. La formulation présentée ci-dessus permet de faire un rapprochement avec les modèles d'appartenance partagée (*mixed membership models* en anglais) [Zho17] et notamment avec la LDA [BNJ03]. En effet, si l'on impose les contraintes : $\sum_i h_{ik} = 1$, $\sum_k w_{uk} = 1$, alors l'Éq. 1.39 correspond au modèle de LDA avec $\phi_{ui} = [\mathbf{W}\mathbf{H}^T]_{ui}$.

1.4.4. Activité et popularité

Les extensions bayésiennes de la PF [Can04 ; BJ06 ; Cem09 ; Ma+11 ; GHB15] imposent généralement un a priori gamma sur chaque coefficient de la matrice $\mathbf{W}$ et/ou de la matrice $\mathbf{H}$:

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta_u^W), \tag{1.40}$$

$$h_{ik} \sim \text{Gamma}(\alpha^H, \beta_i^H), \tag{1.41}$$

où α^W et α^H sont des paramètres de forme et $\boldsymbol{\beta}^W = (\beta_1^W, \ldots, \beta_U^W)$ et $\boldsymbol{\beta}^H = (\beta_1^H, \ldots, \beta_I^W)$ sont des paramètres d'intensité. L'espérance des coefficients des matrices est donnée par : $\mathbb{E}(w_{uk}) = \alpha^W / \beta_u^W$ et $\mathbb{E}(h_{ik}) = \alpha^H / \beta_i^H$. Les lois gamma sont connues pour induire de la par-

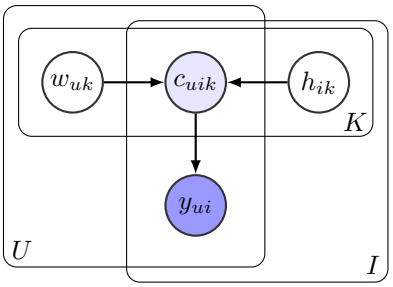


FIGURE 1.4. – Représentation graphique du modèle augmenté PF. En bleu foncé : les variables observées ; en bleu clair : les variables partiellement observées.

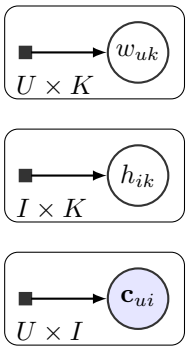


FIGURE 1.5. – Représentation graphique de la distribution variationnelle q.

cimonie lorsque le paramètre de forme est plus petit que 1. C'est une propriété intéressante puisqu'elle implique que chaque utilisateur et chaque article ne soit représenté que par un nombre réduit de facteurs. De plus, on obtient un modèle entièrement conjugué en utilisant la formulation présentée en Éq. (1.35)-(1.36). La conjugaison du modèle permet d'utiliser des algorithmes itératifs très faciles à mettre en place (voir le paragraphe suivant).

Les paramètres d'intensité peuvent être liés à l'activité des utilisateurs et à la popularité des articles respectivement. En effet, on peut écrire pour l'espérance de la variable de budget : $\mathbb{E}(b_u) \propto 1/\beta_u^W$. Dans cette thèse, nous considérons $\boldsymbol{\beta}^W$ et $\boldsymbol{\beta}^H$ comme des paramètres que l'on estime par maximum de vraisemblance à la manière de [Cem09]. Des approches hiérarchiques ont aussi été proposées où des a priori sur ces variables sont imposés. Dans [GHB15], les paramètres d'intensité ont un a priori gamma, alors que dans [SWZ16], ce sont les paramètres d'échelle (inverse du paramètre d'intensité) qui en ont un.

Le paramètre de forme de l'a priori gamma peut lui aussi être estimé. Dans [Cem09], les auteurs proposent un estimateur de maximum de vraisemblance et font appel à un algorithme de Newton Raphson. Des approches bayésiennes ont aussi été considérées, notamment dans [ZCC15] et font appel à des astuces d'augmentations de modèle dont nous nous servirons dans le Chapitre 3. Pour notre part, les paramètres de forme seront considérés comme des hyper-paramètres que nous ne chercherons pas à estimer.

Tableau 1.1. – Expression des distributions variationnelles pour le modèle PF.

Variable	Distribution q
$\mathbf{C}$	$q(\mathbf{c}_{ui}) = \text{Mult}\left(\mathbf{c}_{ui}; y_{ui},\ \tilde{\boldsymbol{\phi}}_{ui}\right)$
$\mathbf{W}$	$q(w_{uk}) = \text{Gamma}\left(w_{uk}; \tilde{\alpha}^W_{uk},\ \tilde{\beta}^W_{uk}\right)$
$\mathbf{H}$	$q(h_{ik}) = \text{Gamma}\left(h_{ik}; \tilde{\alpha}^H_{ik},\ \tilde{\beta}^H_{ik}\right)$

1.4.5. Inférence variationnelle

Nous nous intéressons dans cette section à l'estimation de la distribution a posteriori $p(\mathbf{W}, \mathbf{H}|\mathbf{Y})$ qui nous permet de faire des prédictions et des recommandations aux utilisateurs. Pour cela, nous travaillons avec le modèle augmenté illustré en Figure 1.4 :

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta^W_u),\ h_{ik} \sim \text{Gamma}(\alpha^H, \beta^H_i), \tag{1.42}$$

$$c_{uik}|\mathbf{W}, \mathbf{H} \sim \text{Poisson}(w_{uk} h_{ik}), \tag{1.43}$$

$$y_{ui} = \sum_k c_{uik}. \tag{1.44}$$

On note $\mathbf{Z} = \{\mathbf{C}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes du modèle augmenté et $\Phi = \{\boldsymbol{\beta}^W, \boldsymbol{\beta}^H\}$ l'ensemble des paramètres. Les paramètres de forme α^W et α^H sont considérés ici comme des hyper-paramètres. La distribution a posteriori $p(\mathbf{Z}|\mathbf{Y})$ étant insoluble, on utilise l'inférence variationnelle pour l'approximer (voir Section 1.3.2).

Famille du champ moyen. On choisit la famille $\mathcal{F}$ comme étant la famille de champ moyen (voir Section 1.3.3), où q est supposée être entièrement factorisable par rapport à chaque variable du modèle. Ainsi, on obtient l'expression de la distribution q suivante :

$$q(\mathbf{Z}) = \prod_{ui} q(\mathbf{c}_{ui}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \tag{1.45}$$

Le modèle graphique de la distribution variationnelle est donné en Figure 1.5.

Espérance prédictive a posteriori. La distribution prédictive a posteriori $p(\mathbf{Y}^*|\mathbf{Y})$ correspond à la distribution de nouvelles données $\mathbf{Y}^*$ sachant que des données $\mathbf{Y}$ ont été

observées. Elle peut se réécrire en utilisant les variables latentes $\mathbf{W}$ et $\mathbf{H}$ comme :

$$p(\mathbf{Y}^*|\mathbf{Y}) = \int_{\mathbf{W},\mathbf{H}} p(\mathbf{Y}^*|\mathbf{W},\mathbf{H})p(\mathbf{W},\mathbf{H}|\mathbf{Y})\mathrm{d}\mathbf{W}\mathrm{d}\mathbf{H}. \tag{1.46}$$

Cette expression fait notamment apparaître la loi a posteriori insoluble $p(\mathbf{W},\mathbf{H}|\mathbf{Y})$ que nous cherchons à approximer. Dans le but de faire des recommandations aux utilisateurs, nous nous intéressons à l'espérance de cette loi $\mathbb{E}(\mathbf{Y}^*|\mathbf{Y})$, appelée espérance prédictive a posteriori. Cette espérance peut être approximée en utilisant l'approximation variationnelle, en effet :

$$\mathbb{E}(\mathbf{Y}^*|\mathbf{Y}) = \int_{\mathbf{W},\mathbf{H}} \underbrace{\mathbb{E}(\mathbf{Y}^*|\mathbf{W},\mathbf{H})}_{=\mathbf{W}\mathbf{H}^T} \underbrace{p(\mathbf{W},\mathbf{H}|\mathbf{Y})}_{\approx q(\mathbf{W})q(\mathbf{H})} \mathrm{d}\mathbf{W}\mathrm{d}\mathbf{H} \tag{1.47}$$

$$\approx \mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T). \tag{1.48}$$

Ainsi, nous construisons les listes personnalisées de recommandations à partir du score défini par : $s_{ui} = [\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T)]_{ui}$.

Algorithme de VI avec optimisation par bloc. Comme présenté en Section 1.3.3, on utilise un algorithme avec optimisation par bloc pour maximiser la ELBO. Couplé avec l'hypothèse de champ moyen, on obtient des règles de mise à jour simples puisque le modèle augmenté est entièrement conjugué. La forme de la distribution variationnelle est résumée dans le Tableau 1.1. Les règles de mise à jour des paramètres variationnels sont présentées dans l'Algorithme 2 et le détail des dérivations est donné en Annexe A.1. À noter que lorsque $q(x) = \mathrm{Gamma}(\alpha,\beta)$, on a $\mathbb{E}_q(x) = \alpha/\beta$ et $\mathbb{E}_q(\log x) = \Psi(\alpha) - \log(\beta)$, où Ψ est la fonction digamma.

Optimisation des paramètres. Dans ce paragraphe, nous développons les règles de mise à jour liées à l'apprentissage des paramètres d'intensité $\boldsymbol{\beta}^W$ et $\boldsymbol{\beta}^H$. Comme décrit dans la Section 1.3.4, on cherche à maximiser la quantité :

$$\mathrm{ELBO}(q,\Phi) = \mathbb{E}_q(\log p(\mathbf{W};\boldsymbol{\beta}^W)) + \mathbb{E}_q(\log p(\mathbf{H};\boldsymbol{\beta}^H)) + cste, \tag{1.49}$$

où *cste* est une constante par rapport aux paramètres $\boldsymbol{\beta}^W$ et $\boldsymbol{\beta}^H$. Le problème est séparable et symétrique pour les deux paramètres. Ainsi, pour $\boldsymbol{\beta}^W$, on cherche à maximiser :

$$\mathbb{E}_q(\log p(\mathbf{W};\boldsymbol{\beta}^W)) = \sum_{uk} \left(-\beta_u^W \mathbb{E}_q(w_{uk}) - \alpha^W \log \beta_u^W\right) + cste. \tag{1.50}$$

Algorithme 2 : Algorithme CAVI pour la PF

Données : Matrice d'observation $\mathbf{Y}$
Résultat : Distribution variationnelle q

Initialisation aléatoire des paramètres variationnels ;
répéter
  pour chaque *couple* (u, i) *tel que* $y_{ui} > 0$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$;
    $\mathbb{E}_q(c_{uik}) = y_{ui} \dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
  fin
  pour chaque *utilisateur* $u \in \{1, \dots, U\}$ **faire**
    $\tilde{\alpha}^W_{uk} = \alpha^W + \sum_i \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^W_{uk} = \beta^W_u + \sum_i \mathbb{E}_q(h_{ik})$;
  fin
  pour chaque *article* $i \in \{1, \dots, I\}$ **faire**
    $\tilde{\alpha}^H_{ik} = \alpha^H + \sum_u \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^H_{ik} = \beta^H_i + \sum_u \mathbb{E}_q(w_{uk})$;
  fin
  Mise à jour des paramètres : $\beta^W_u = \dfrac{K\alpha^W}{\sum_k \mathbb{E}_q(w_{uk})}$ et $\beta^H_i = \dfrac{K\alpha^H}{\sum_k \mathbb{E}_q(w_{ik})}$;
  Calculer ELBO(q, Φ)
jusqu'à *ELBO converge*;

Ce qui nous donne les règles de mises à jour suivantes : $\beta^W_u = \dfrac{\alpha^W}{\sum_k \mathbb{E}_q(w_{uk})/K}$. De la même façon, on obtient $\beta^H_i = \dfrac{\alpha^H}{\sum_k \mathbb{E}_q(w_{ik})/K}$. L'algorithme de PF complet est résumé en Algorithme 2.

1.4.6. Exemple de recommandation

Jeu de données. Dans ce paragraphe, nous nous intéressons à un exemple de recommandation musicale obtenue avec la PF. Nous utilisons le jeu de données Taste Profile (TP) [Ber+11] qui contient le nombre de fois où des utilisateurs ont écouté des chansons. Nous pré-traitons les données comme dans [Lia+16]. Nous sélectionnons un sous-ensemble des données, et ne gardons que les utilisateurs et les chansons qui ont au moins 20 interactions. Nous obtenons une matrice $\mathbf{Y}$ de dimensions $U = 16\ 301$ et $I = 12\ 118$.

Nous divisons le jeu de données $\mathbf{Y}$ en un ensemble d'entraînement $\mathbf{Y}^{\text{train}}$ contenant 80%

des valeurs non nulles du jeu de données original et un ensemble de test $\mathbf{Y}^{\text{test}}$ contenant les 20% restants (ces valeurs sont mises à zéro dans l'ensemble d'entraînement). Le modèle est entraîné sur l'ensemble $\mathbf{Y}^{\text{train}}$. Le score de prédiction défini plus haut nous permet de créer des listes de recommandations de 100 chansons pour chaque utilisateur. Ces listes sont ensuite évaluées sur l'ensemble de test selon la métrique NDCG définie en Section 1.1.3.

Méthodes comparées. On compare dans cette section deux versions de la PF. La première est entraînée sur les données brutes $\mathbf{Y}^{\text{train}} \in \mathbb{N}^{U \times I}$, on utilisera l'acronyme PFbrut. La seconde est entraînée sur une version pré-traitée des données où n'est conservée que l'information de parcimonie, i.e., si l'article (ici une chanson) a été écouté au moins une fois. On obtient donc une version binaire des données [3] définie par $y_{ui}^{\text{bin}} = \mathbb{1}[y_{ui} > 0]$. On utilisera l'acronyme PFbin pour cette version de la PF.

Après comparaison sur la grille de recherche $\{0.1, 0.3, 1\}$, nous fixons les paramètres de forme à $\alpha^W = \alpha^H = 0.3$. Le nombre de facteurs latents est recherché sur la grille $K \in \{50, 150, 200, 500, 1000\}$. Pour chaque K fixé, nous exécutons l'algorithme PF (présenté en Algorithme 2) 5 fois avec des initialisations aléatoires différentes. La convergence de l'algorithme est établie lorsque la croissance relative de la ELBO passe sous le seuil $\tau = 10^{-5}$.

Résultats. La Figure 1.6 illustre les résultats de recommandation moyens (et l'écart-type sur 5 exécutions) obtenus avec PFbrut et PFbin par rapport au nombre de facteurs latents K. Les quatre figures correspondent à différents seuils $s \in \{0, 1, 2, 5\}$ qui définissent la pertinence d'une recommandation (voir Section 1.1.3). Plusieurs remarques peuvent être faites au regard de ces graphiques. Tout d'abord, le nombre de facteurs optimal pour PFbin semble être aux alentours de $K = 150$, alors qu'il est supérieur à $K = 500$ pour PFbrut (la courbe semble atteindre un palier à partir de cette valeur). Cela est en adéquation avec le fait que les données brutes contiennent plus d'information que les données binarisées.

Comparons maintenant les performances de PFbrut et PFbin sur le score de NDCG avec différents seuils. PFbin surpasse PFbrut pour le score NDCG0 (en haut à gauche de la Figure 1.6). Le surplus d'information (sur les valeurs non nulles de $\mathbf{Y}$) ne profite donc pas à PFbrut. Cependant, comme PFbin se concentre uniquement sur l'information de parcimonie, elle voit ses performances décroître avec l'augmentation du seuil s. PFbrut semble plus robuste à ce changement. Les résultats semblent être similaires entre PFbrut et PFbin pour $s = 2$, alors que pour $s = 5$ PFbrut surpasse PFbin.

3. Dans la suite de la thèse on utilisera le terme de «données binarisées» pour décrire ce pré-traitement.

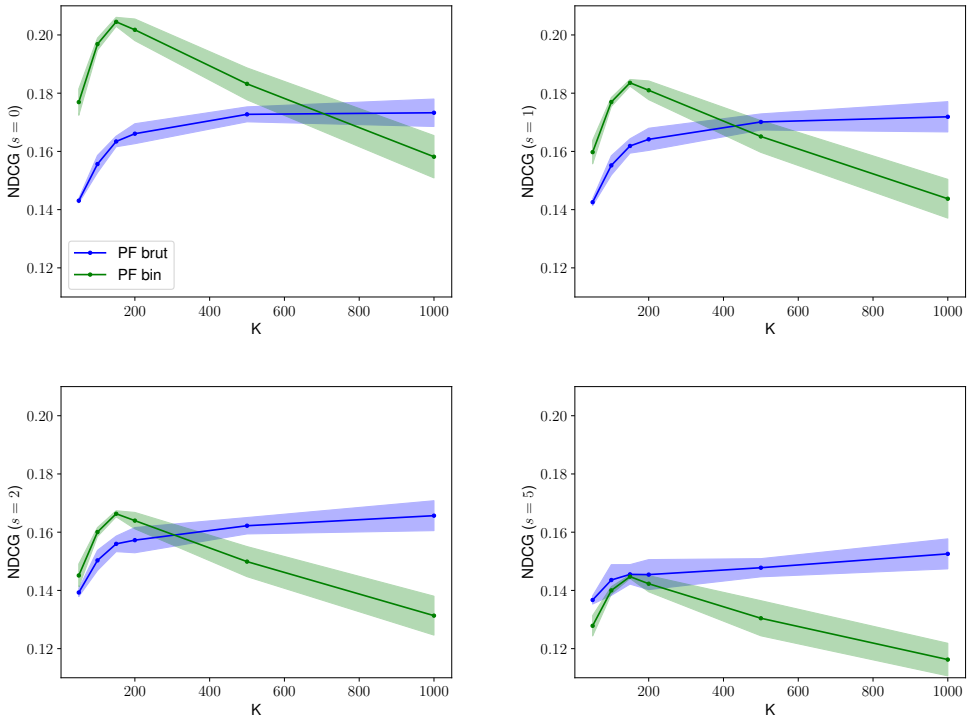


FIGURE 1.6. – Résultats de recommandation pour PFbrut et PFbin pour différents seuils de pertinence $s \in \{0, 1, 2, 5\}$.

Nombre de facteurs latents. Dans cette thèse, nous choisissons de fixer le nombre de facteurs latents et de comparer les résultats sur une grille pré-définie. On constate que la PF régularise l'influence des facteurs latents lorsque K est grand. Par exemple, pour PFbin avec $K = 200$, la Figure 1.8 illustre la valeur moyenne des colonnes de $\mathbb{E}_q(\mathbf{H})$. On constate qu'une cinquantaine de colonnes correspondent à des résidus. Cela confirme bien que le K optimal pour cette méthode (et ce choix de paramètres de forme) est atteint pour $K = 150$.

Représentation. La Tableau 1.2 illustre un exemple de trois facteurs latents inférés avec PFbin sur le jeu de données TP. Pour chacun des trois facteurs latents sélectionnés, nous affichons les artistes [4] qui ont les plus hautes valeurs $\mathbb{E}_q(h_{ik})$ associées. On constate bien que la PF regroupe des artistes similaires.

4. Nous avons choisi d'afficher ici le nom des artistes plutôt que le titre des chansons afin d'imposer plus de diversité dans la liste des artistes représentant les différents facteurs latents.

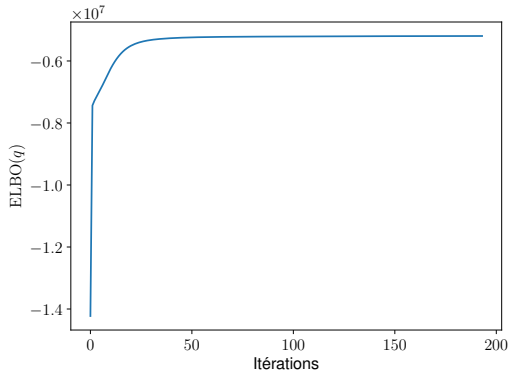


FIGURE 1.7. – Convergence de la ELBO pour PFbin avec $K = 200$.

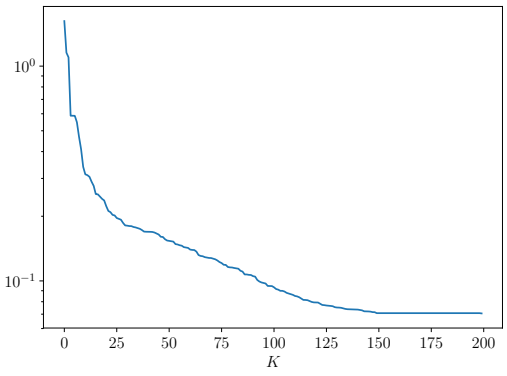


FIGURE 1.8. – Valeur moyennes des colonnes de $\mathbb{E}_q(\mathbf{H})$ pour PFbin avec $K = 200$.

Tableau 1.2. – Exemple de trois facteurs latents obtenus avec PFbin.

«Rock»	«Électro»	«Folk»
Pixies	Massive Attack	Beirut
Creedence Clearwater Revival	Thievery Corporation	Cat Power
The Rolling Stones	Röyksopp	José González
The Doors	Zero 7	Feist
Old Crow Medicine Show	Bonobo	Devendra Banhart

1.4.7. Modèles non paramétriques

Dans cette thèse, le nombre de facteurs latents est considéré comme un paramètre du modèle. Pour chaque expérience, on cherchera donc le nombre de facteurs latents K optimal sur une grille prédéfinie (voir Section 1.4.6). Cette recherche est coûteuse puisqu'elle nécessite d'exécuter l'algorithme plusieurs fois. Néanmoins, il existe des méthodes dites non paramétriques qui permettent d'inférer/régulariser le nombre de facteurs latents K. Cela se révèle particulièrement utile lorsque le nombre de communautés sous-jacentes est amené à croître avec les dimensions de $\mathbf{Y}$ [CF17].

Dans le cas où la vraisemblance est poissonnienne, un choix pratique est de construire le modèle non paramétrique à partir du processus gamma [Tit08 ; CTM14]. Un processus gamma permet de générer une collection infinie de poids positifs dont la somme est presque sûrement finie (lorsque la mesure de base du processus est finie). On retrouve beaucoup de modèles à facteurs latents qui utilisent ces processus dans la littérature [ZC15 ; Zho17 ; PC09].

En particulier, [Gop+14] propose une version non paramétrique de la PF où les préférences des utilisateurs suivent un processus gamma. Le modèle présenté repose sur la construction *stick-breaking* (en anglais) du processus de Dirichlet [Set94]. L'inférence du modèle se fait à l'aide de la VI. La distribution variationnelle est tronquée à partir d'un certain rang T, au delà duquel la distribution variationnelle est supposée égale à la loi a priori [KWV07]. Cela permet d'obtenir un nombre fini de paramètres à optimiser.

Bilan et perspectives. En conclusion de cette section, on peut établir qu'une étape de binarisation appliquée aux données dans le cas de la PF est bien efficace. Il permet d'obtenir un nombre de facteurs latents réduit et d'améliorer les performances sur les petites valeurs de seuil (et en particulier pour $s = 0$ qui est le cas le plus couramment étudié). Cependant, l'application de ce pré-traitement entraîne une perte d'information qui nuit à la qualité des recommandations pour des seuils $s \geq 2$.

Une des explications au fait que la PF ait du mal a traiter des données brutes réside dans le fait que la variance de la distribution Poisson est fixée et égale à son espérance, i.e., $\text{var}(y_{ui}) = \mathbb{E}(y_{ui})$. De ce fait, PFbrut a besoin de plus de facteurs latents pour expliquer la sur-dispersion des données, ce qui nuit aux capacités prédictives du modèle. Par conséquent, dans les Chapitres 2, 3 et 4 nous étudierons des moyens de contourner cette limitation de la PF dans le but de mieux prendre en compte la nature sur-dispersée des données.

Chapitre 2.

Factorisation binomiale négative

Ce chapitre est adapté du rapport technique [GOF18b].

Contents

2.1. Introduction

Les données implicites collectées dans les systèmes de recommandation sont souvent disponibles sous la forme de données de comptage. Ces données sont connues pour être parcimonieuses (peu de données observées) et sur-dispersées (de variance supérieure à leur moyenne) [HKV08 ; SWZ16]. Dans ce chapitre, nous prendrons l'exemple de données qui correspondent au nombre de fois où un utilisateur a écouté une chanson.

Comme nous l'avons vu en Section 1.4, la loi de Poisson est particulièrement adaptée

aux données de comptage. Cependant, la variance de cette loi est fixée et égale à son espérance, i.e., $\text{var}(y_{ui}) = \mathbb{E}(y_{ui})$, ce qui ne permet pas de modéliser la sur-dispersion présente dans les données. Ainsi la loi Poisson souffre d'un fort couplage entre la modélisation des valeurs nulles et de la sur-dispersion [BE16]. C'est pourquoi une étape de binarisation est souvent ajoutée avant d'appliquer la PF [GHB15] (voir Section 1.4.6). Ceci entraîne une perte d'information puisque le nombre d'écoutes ne peut plus être pris en compte par le modèle.

Dans ce chapitre, nous cherchons à construire un modèle qui permet de représenter les données de comptage sur-dispersées sans avoir recours à une étape de binarisation. Pour cela, nous utilisons la distribution binomiale négative (NB, pour *negative binomial* en anglais). Cette distribution est une extension naturelle de la loi Poisson qui possède un paramètre supplémentaire permettant de contrôler la variance. La loi NB a été largement utilisée dans des modèles de régression [Law87 ; GMS95 ; Hil11 ; Zho+12]. Contrairement à ces travaux, nous nous intéressons à l'utilisation de la loi NB dans le cadre de la NMF (nous utiliserons l'acronyme NBF pour *negative binomial matrix factorization*) et nous imposons que le modèle soit paramétré par sa moyenne.

Les contributions de ce chapitre sont les suivantes.

— Nous développons un modèle fondé sur la loi NB qui est paramétré par sa moyenne. Nous étudions en particulier sa formulation comme composition de lois Poisson et gamma et faisons le lien avec des modèles qui utilisent des variables dites d'exposition.

— Nous mettons en lumière la divergence associée à ce modèle probabiliste. Cette divergence, qui n'a pas été étudiée auparavant à notre connaissance, définit un nouveau problème de NMF.

— Nous développons un algorithme MM pour une approche fréquentiste du problème, et un algorithme CAVI pour une approche bayésienne.

— Nous testons ce modèle sur des tâches de recommandation et pointons ses limites, tout en proposant des perspectives d'amélioration.

Le reste du chapitre est organisé comme suit. Dans la Section 2.2 nous introduisons le modèle NBF ainsi que sa formulation comme composition de lois Poisson et gamma. De plus, nous établissons des connexions avec d'autres travaux existants. Dans la Section 2.3, nous étudions l'estimateur de maximum de vraisemblance et la divergence associée à notre modèle. Dans la Section 2.4, nous mettons en place un algorithme d'inférence variationnelle pour une version bayésienne du modèle. Dans la Section 2.5, nous testons la NBF sur des tâches de recommandation. Enfin, dans la Section 2.6, nous discutons des limites du modèle

proposé.

2.2. Description du modèle

2.2.1. Processus génératif & formulation Poisson-gamma

Pour chaque utilisateur $u \in \{1, ..., U\}$ et chaque article $i \in \{1, ..., I\}$, on suppose que le nombre d'écoutes y_{ui} est généré d'après le processus suivant :

$$y_{ui} \sim \text{NB}\left(\alpha, \frac{[\mathbf{WH}^T]_{ui}}{\alpha + [\mathbf{WH}^T]_{ui}}\right), \tag{2.1}$$

où $\text{NB}(\alpha, p)$ est la distribution NB paramétrée par un coefficient de dispersion (ou paramètre de forme) $\alpha \in \mathbb{R}_+$ et par un paramètre de probabilité $p \in (0, 1)$. Sa fonction de masse, illustrée en Figure 2.2, est donnée par :

$$\text{NB}(y; \alpha, p) = \frac{\Gamma(y + \alpha)}{y!\Gamma(\alpha)} p^y (1 - p)^\alpha. \tag{2.2}$$

Quand $\alpha \leq 1$, le mode de la loi NB se situe en 0. Quand α tend vers l'infini et que la moyenne est fixée, on retrouve la distribution Poisson.

Comme pour la PF et beaucoup d'autres modèles de factorisation de matrices paramétrés par la moyenne [TF13], l'espérance des observations est donnée par : $\mathbb{E}(y_{ui}) = [\mathbf{WH}^T]_{ui}$. Cela permet une compréhension intuitive du modèle. Contrairement à la distribution Poisson, la distribution NB possède un second paramètre qui permet d'ajouter de la variance au modèle. Dans la paramétrisation proposée en Éq. (2.1), la variance et le ratio variance/espérance des observations sont donnés par :

$$\text{var}(y_{ui}) = [\mathbf{WH}^T]_{ui} \left(1 + \frac{[\mathbf{WH}^T]_{ui}}{\alpha}\right), \tag{2.3}$$

$$\frac{\text{var}(y_{ui})}{\mathbb{E}(y_{ui})} = 1 + \frac{[\mathbf{WH}^T]_{ui}}{\alpha} \geq 1. \tag{2.4}$$

Remarque. Il est important de noter que, contrairement à la méthode du même nom introduite dans [Zho17], nous plaçons la factorisation $[\mathbf{WH}^T]_{ui}$ sur le paramètre de probabilité de la loi NB et non pas sur le paramètre de forme. Cette seconde approche sera plus longuement discutée dans le Chapitre 3.

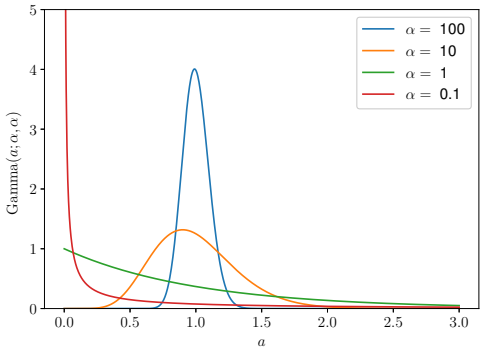


FIGURE 2.1. – Fonction de densité de la distribution gamma : $a \sim \text{Gamma}(\alpha, \alpha)$.

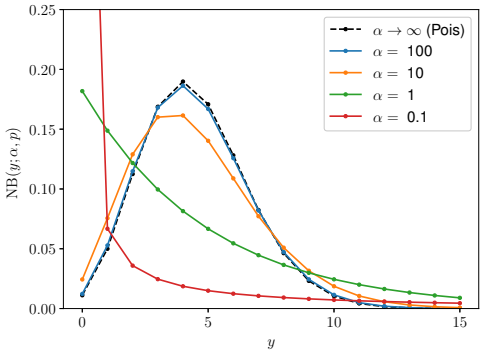


FIGURE 2.2. – Fonction de masse de la distribution NB : $y \sim \text{NB}(\alpha, p)$, tel que $\mathbb{E}(y) = 4.5$.

La distribution NB peut aussi être vue comme un mélange Poisson-gamma. En utilisant cette propriété, on peut écrire le modèle hiérarchique suivant, équivalent à Éq. (2.1) :

$$a_{ui} \sim \text{Gamma}(\alpha, \alpha), \tag{2.5}$$

$$y_{ui}|a_{ui} \sim \text{Poisson}(a_{ui}[\mathbf{W}\mathbf{H}^T]_{ui}). \tag{2.6}$$

On note $\mathbf{A}$ la matrice de taille $U \times I$ ayant pour coefficients $[\mathbf{A}]_{ui} = a_{ui}$. La variable $\mathbf{A}$ n'est contrainte que par sa loi a priori. La Figure 2.1 illustre la densité associée à cette loi gamma. Par construction, on a $\mathbb{E}[a_{ui}] = 1$ et $\text{var}(a_{ui}) = \alpha^{-1}$. Ainsi, la variable $\mathbf{A}$ peut être vue comme un bruit multiplicatif qui ajoute de la variance au modèle et qui modélise les variabilités locales.

Lien avec la factorisation Poisson robuste. D'autres travaux ont proposé des versions de la PF robustes aux données aberrantes. Par exemple, dans [FD15], les données sont supposées être générées d'après $y_{ui} \sim \text{Poisson}([\mathbf{W}\mathbf{H}^T]_{ui} + e_{ui})$, où e_{ui} est une variable aléatoire positive et parcimonieuse. Cette variable correspond à un bruit additif permettant de prendre en compte les valeurs aberrantes présentes dans les données. De par sa contrainte de positivité, elle ne permet cependant d'expliquer que les valeurs aberrantes. Au contraire, notre variable $a_{ui} \in \mathbb{R}_+$ peut à la fois expliquer les hautes et faibles valeurs.

2.2.2. Interprétation : la notion d'exposition

La matrice $\mathbf{A}$ décrit les variations locales qui ne peuvent pas être expliquées par le produit matriciel $\mathbf{WH}^T$. $\mathbf{A}$ peut atténuer ou accentuer les coefficients de $\mathbf{WH}^T$. Dans le domaine des systèmes de recommandation, $\mathbf{A}$ peut être interprétée comme une variable d'exposition [Lia+16]. Ici, nous avons $a_{ui} \in \mathbb{R}_+$, ce qui permet plusieurs interprétations possibles :

— Si $a_{ui} \ll 1$, l'utilisateur est sous-exposé à l'article. Cela peut être expliqué de diverses manières : l'utilisateur n'est pas au courant de la sortie d'une chanson, il n'habite pas dans un lieu où la chanson est populaire, etc.

— Si $a_{ui} \gg 1$, l'utilisateur est sur-exposé à l'article. Là aussi, cela peut s'expliquer de diverses manières : la chanson est un tube du moment et est largement diffusée à la radio, l'utilisateur a un comportement d'écoute compulsif et écoute sans arrêt cette chanson, etc.

— Si $a_{ui} \approx 1$, alors l'exposition n'affecte pas la préférence de l'utilisateur pour la chanson, qui est alors entièrement décrite par le produit $[\mathbf{WH}^T]_{ui}$.

La matrice $\mathbf{A}$ n'étant contrainte que par sa loi a priori $a_{ui} \sim \text{Gamma}(\alpha, \alpha)$, le paramètre α prend une grande importance puisqu'il détermine la variation possible autour de la moyenne $\mathbf{WH}^T$ ($\text{var}(a_{ui}) = \alpha^{-1}$). Si α est trop petit, le rôle de $\mathbf{A}$ devient prépondérant par rapport au produit $\mathbf{WH}^T$ et détruit le pouvoir prédictif du modèle.

Variable d'exposition dans des modèles Poisson. La NBF peut être vue comme une instance d'un modèle plus général décrit par :

$$\mathbf{A} \sim p(\mathbf{A}; \Theta), \tag{2.7}$$

$$y_{ui}|a_{ui} \sim \text{Poisson}(a_{ui}[\mathbf{WH}^T]_{ui}), \tag{2.8}$$

où $p(\mathbf{A}; \Theta)$ est une distribution paramétrée par Θ.

On trouve plusieurs exemples de tels modèles dans la littérature.

— PF. Si $\mathbf{A}$ est déterministe avec pour tout couple utilisateur/article $a_{ui} = 1$, on retrouve la PF [Can04 ; BJ06 ; Cem09 ; Ma+11 ; GHB15].

— Modèle Poisson avec excès de zéros (*zeroinflated Poisson* en anglais). Dans [Sim13], la variable d'exposition a_{ui} est tirée selon une loi Bernoulli : $a_{ui} \sim \mathcal{B}(\mu)$, et modélise le fait qu'un utilisateur connaisse un article ou non. Marginaliser cette variable permet

de retrouver une loi Poisson avec excès de zéros [Lam92] :

$$y_{ui} \sim (1-\mu)\delta_0 + \mu \,\text{Pois}([\mathbf{W}\mathbf{H}^T]_{ui}). \tag{2.9}$$

Dans notre cas, nous étudions un cas plus général où $a_{ui} \in \mathbb{R}^+$, ce qui permet de modéliser aussi les hautes valeurs ainsi qu'une interprétation plus souple. Dans [Sim13], un modèle hiérarchique plus sophistiqué est proposé pour la variable μ (qui devient une variable locale μ_{ui}), incluant des sources de données extérieures (comme des données géographiques ou des données issues de réseaux sociaux). De telles idées pourraient être incorporées dans notre cadre de travail.

— Structure de rang faible. Dans [BE17], les auteurs considèrent un modèle plus contraint où la variable d'exposition $\mathbf{A}$ possède, elle aussi, une structure de rang faible :

$$a_{ui} \sim \text{Poisson}([\mathbf{U}\mathbf{V}^T]_{ui}). \tag{2.10}$$

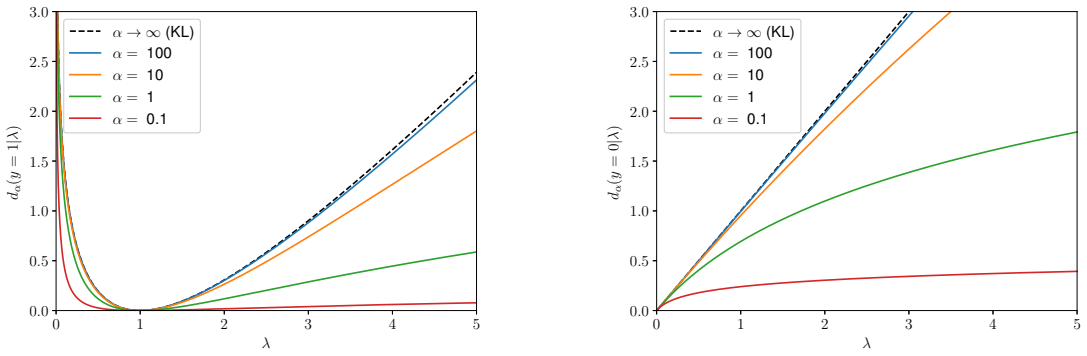
Cet article est une extension de [BE16] dont nous discuterons plus en détail dans le Chapitre 3.

— Graphe aléatoire. Dans [PK13], l'exposition est modélisée par un graphe aléatoire biparti. La moitié des articles non consommés est arbitrairement considérée comme correspondant à des retours manquants.

Variable d'exposition dans des modèles Gaussiens. Au-delà des modèles à vraisemblance poissonnienne, la notion d'exposition a aussi été introduite dans le cadre de modèles gaussiens. En particulier, dans [Lia+16], les auteurs proposent un modèle de factorisation de matrices à exposition :

$$y_{ui} \sim (1-\mu)\delta_0 + \mu \,\mathcal{N}([\mathbf{W}\mathbf{H}^T]_{ui}, \sigma^2), \tag{2.11}$$

où $y_{ui} \in \{0, 1\}$ sont des données implicites binaires (ou binarisées), $\mathbf{W} \in \mathbb{R}^{U\times K}$ et $\mathbf{H} \in \mathbb{R}^{I\times K}$. Les auteurs de ce papier soulignent le fait que le modèle WMF [HKV08] appliqué à des données binaires est un cas particulier de leur modèle. Un algorithme EM est développé pour inférer les paramètres du modèle. Contrairement à ce travail, nous choisissons de travailler avec la loi Poisson qui nous semble mieux adaptée aux données de comptage. De plus, nous n'appliquons pas de pré-traitement aux données.



(a) $y = 1$ (b) $y = 0$

FIGURE 2.3. – Divergence associée à la distribution NB paramétrée par sa moyenne $d_\alpha(y|\lambda)$.

Séparation aveugle de sources. Dans le contexte de la séparation aveugle de sources en traitement de signal audio, un modèle similaire à la NBF a été développé pour obtenir plus de flexibilité [YIG16 ; LBR17]. Ce modèle suppose que les coefficients y_{ui} de la transformée de Fourier à court-terme de chaque source suivent une loi t de Student. Les paramètres de cette distribution possèdent une structure NMF. On peut introduire dans ce modèle une variable latente a_{ui} et ainsi obtenir la formulation hiérarchique suivante :

$$a_{ui} \sim \mathrm{IG}(\alpha/2, \alpha/2) \tag{2.12}$$

$$y_{ui}|a_{ui} \sim \mathcal{N}(0, a_{ui}[\mathbf{W}\mathbf{H}^T]_{ui}). \tag{2.13}$$

La variable latente a_{ui} a un rôle équivalent à celui de notre variable d'exposition. Elle permet de rajouter de la variance au modèle afin d'obtenir une distribution marginale (loi t de Student) qui possède une queue plus lourde que la loi normale.

2.3. Estimation au sens du maximum de vraisemblance

2.3.1. Divergence associée à la distribution binomiale négative

L'estimateur du maximum de vraisemblance de la moyenne λ, associé à la distribution $y \sim \mathrm{NB}(\alpha, p)$ avec $p = \frac{\lambda}{\alpha+\lambda}$, est donné par la minimisation de la fonction de coût suivante :

$$\begin{aligned} C(\lambda) &= -\log \mathrm{NB}(y; \alpha, p) && (2.14)\\ &= -y \log p + \alpha \log(1-p) + cste && (2.15)\\ &= -y \log \lambda + (\alpha + y) \log(\alpha + \lambda) + cste, && (2.16) \end{aligned}$$

où *cste* est une constante par rapport à la variable λ.

Pour tout $\alpha > 0$ fixé, la divergence associée à la distribution NB paramétrée par sa moyenne peut ainsi s'écrire :

$$d_\alpha(y|\lambda) = \begin{cases} y \log \dfrac{y}{\lambda} - (\alpha + y) \log \dfrac{\alpha + y}{\alpha + \lambda}, & \text{pour } y > 0, \\ \alpha \log \left(1 + \dfrac{\lambda}{\alpha}\right), & \text{pour } y = 0. \end{cases} \tag{2.17}$$

On a bien $d_\alpha(y|\lambda) \geq 0$ et $d_\alpha(y|y) = 0$. Pour tout $\omega > 0$, on a le comportement à l'échelle suivant : $d_\alpha(\omega y|\omega\lambda) = \omega d_{\alpha/\omega}(y|\lambda)$. La Figure 2.3 illustre cette divergence pour $y = 1$ et $y = 0$ et pour différentes valeurs de α. À notre connaissance, cette divergence n'a pas de nom associé et ne correspond pas à un cas connu de la littérature. Comme attendu, la divergence généralisée de Kullback-Leibler (associée à la distribution Poisson) est un cas limite de notre divergence :

$$\lim_{\alpha \to +\infty} d_\alpha(y|\lambda) = \mathrm{KL}(y|\lambda). \tag{2.18}$$

La divergence associée au modèle 2.1 est donnée par : $D_\alpha(\mathbf{Y}|\mathbf{W}\mathbf{H}^T) = \sum_{ui} d_\alpha(y_{ui}|[\mathbf{W}\mathbf{H}^T]_{ui})$.

2.3.2. Algorithme de majoration-minimisation (MM)

L'estimateur du maximum de vraisemblance des paramètres $\mathbf{W}$ et $\mathbf{H}$ s'écrit donc comme la solution du problème d'optimisation :

$$\min_{\mathbf{W},\mathbf{H}} D_\alpha(\mathbf{Y}|\mathbf{W}\mathbf{H}^T), \text{ s.c. } \mathbf{W} \geq 0, \ \mathbf{H} \geq 0. \tag{2.19}$$

Cette équation définit un nouveau problème de NMF. Une approche standard pour minimiser $D_\alpha(\mathbf{Y}|\mathbf{W}\mathbf{H}^T)$ est la descente par blocs dans laquelle les matrices $\mathbf{W}$ et $\mathbf{H}$ sont mises à jour alternativement jusqu'à convergence vers un point stationnaire (qui peut ne pas être un minimum global compte tenu de la non convexité de la fonction de coût). Comme pour beaucoup de problèmes de NMF, notamment lors de l'utilisation d'une β-divergence [FI11], les mises à jour de $\mathbf{W}$ et $\mathbf{H}$ peuvent être obtenues en utilisant la majoration-minimisation (MM). Les rôles de $\mathbf{W}$ et $\mathbf{H}$ étant interchangeables par transposition ($\mathbf{Y} \approx \mathbf{W}\mathbf{H}^T$ est équivalent à $\mathbf{Y}^T \approx \mathbf{H}\mathbf{W}^T$), on se focalisera ici sur la mise à jour de la matrice $\mathbf{H}$ sachant $\mathbf{W}$.

Le principe des algorithmes MM est d'optimiser une majorante $G(\mathbf{H}|\tilde{\mathbf{H}})$ de la fonction de coût. Cette majorante est choisie de telle sorte qu'elle vérifie $G(\mathbf{H}|\tilde{\mathbf{H}}) \geq D_\alpha(\mathbf{Y}|\mathbf{W}\mathbf{H}^T)$ et $G(\tilde{\mathbf{H}}|\tilde{\mathbf{H}}) = D_\alpha(\mathbf{Y}|\mathbf{W}\tilde{\mathbf{H}}^T)$. Cela produit un algorithme de descente où la fonction de coût décroît à chaque itération [HL04].

En suivant le principe de l'approche de [FI11], la majorante peut être facilement construite en majorant séparément la partie convexe et la partie concave de $D_\alpha(\mathbf{Y}|\mathbf{W}\tilde{\mathbf{H}}^T)$:

$$d_\alpha(y_{ui}|[\mathbf{W}\mathbf{H}^T]_{ui}) = \underbrace{-y_{ui}\log([\mathbf{W}\mathbf{H}^T]_{ui})}_{\text{convexe}} + \underbrace{(\alpha + y_{ui})\log(\alpha + [\mathbf{W}\mathbf{H}^T]_{ui})}_{\text{concave}} + cste. \tag{2.20}$$

La partie convexe est majorée en utilisant une inégalité de type Jensen. La partie concave est majorée en utilisant l'inégalité de la tangente. Cette procédure entraîne la mise à jour multiplicative suivante pour $\mathbf{H}$:

$$h_{ik} = \tilde{h}_{ik} \frac{\sum_u \frac{y_{ui}}{[\mathbf{W}\tilde{\mathbf{H}}^T]_{ui}} w_{uk}}{\sum_u \frac{y_{ui}+\alpha}{[\mathbf{W}\tilde{\mathbf{H}}^T]_{ui}+\alpha} w_{uk}}. \tag{2.21}$$

Et, de la même façon, on obtient pour $\mathbf{W}$:

$$w_{uk} = \tilde{w}_{uk} \frac{\sum_i \frac{y_{ui}}{[\tilde{\mathbf{W}}\mathbf{H}^T]_{ui}} h_{ik}}{\sum_i \frac{y_{ui}+\alpha}{[\tilde{\mathbf{W}}\mathbf{H}^T]_{ui}+\alpha} h_{ik}}. \tag{2.22}$$

Ces mises à jour préservent la positivité de $\mathbf{W}$ et $\mathbf{H}$ tant que l'initialisation des matrices est elle aussi positive. Comme convenu, on retrouve les mises à jour multiplicatives liées à la divergence KL lorsque $\alpha \to +\infty$. Un autre moyen d'obtenir ces mises à jour est d'utiliser un algorithme EM basé sur la variable auxiliaire $\mathbf{C}$ introduite dans la Section 1.4.

2.4. Estimation bayésienne

2.4.1. Formulation bayésienne et modèle augmenté

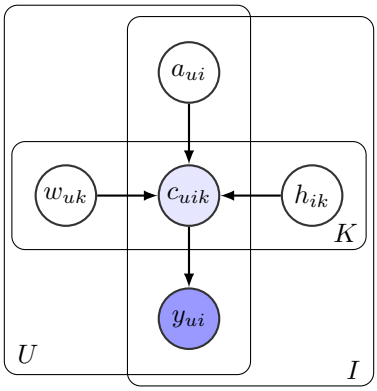


FIGURE 2.4. – Représentation graphique du modèle augmenté NBF. En bleu foncé : les variables observées ; en bleu clair : les variables partiellement observées.

Dans cette section, nous présentons une formulation bayésienne de la NBF. Des lois a priori gamma sont assignées aux variables $\mathbf{W}$ et $\mathbf{H}$ et le modèle est augmenté avec la variable latente $\mathbf{C}$ (voir Section 1.4) et avec la variable latente $\mathbf{A}$ (voir Section 2.4.1). Le processus génératif des observations, illustré en Figure 2.4 est donc le suivant :

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta^W), h_{ik} \sim \text{Gamma}(\alpha^H, \beta^H), \tag{2.23}$$

$$a_{ui} \sim \text{Gamma}(\alpha, \alpha), \tag{2.24}$$

$$c_{uik} | \mathbf{A}, \mathbf{W}, \mathbf{H} \sim \text{Poisson}(a_{ui} w_{uk} h_{ik}), \tag{2.25}$$

$$y_{ui} = \sum_k c_{uik}. \tag{2.26}$$

On note $\mathbf{Z} = \{\mathbf{C}, \mathbf{A}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes et $\Phi = \{\alpha, \alpha^W, \alpha^H, \beta^W, \beta^H\}$ l'ensemble des paramètres. Contrairement à la Section 1.4, on considère ici que les intensités sont les mêmes pour chaque utilisateur $\beta_u^W = \beta^W$ et pour chaque article $\beta_i^H = \beta^H$.

Tableau 2.1. – Expression des distributions variationnelles pour le modèle NBF.

Variable	Distribution q
$\mathbf{C}$	$q(\mathbf{c}_{ui}) = \text{Mult}\left(\mathbf{c}_{ui}; y_{ui},\ \tilde{\boldsymbol{\phi}}_{ui}\right)$
$\mathbf{A}$	$q(a_{ui}) = \text{Gamma}\left(a_{ui}; \tilde{\alpha}^A_{ui},\ \tilde{\beta}^A_{ui}\right)$
$\mathbf{W}$	$q(w_{uk}) = \text{Gamma}\left(w_{uk}; \tilde{\alpha}^W_{uk},\ \tilde{\beta}^W_{uk}\right)$
$\mathbf{H}$	$q(h_{ik}) = \text{Gamma}\left(h_{ik}; \tilde{\alpha}^H_{ik},\ \tilde{\beta}^H_{ik}\right)$

2.4.2. Inférence variationnelle

La distribution a posteriori $p(\mathbf{Z}|\mathbf{Y})$ étant insoluble, on utilise l'inférence variationnelle pour en obtenir une approximation. L'hypothèse du champ moyen donne ici :

$$q(\mathbf{Z}) = \prod_{ui} q(\mathbf{c}_{ui}) q(a_{ui}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \tag{2.27}$$

L'utilisation d'un algorithme CAVI et de l'hypothèse de champ moyen permettent d'obtenir les expressions analytiques des distributions variationnelles (décrites dans le Tableau 2.1). L'Algorithme 3 résume les mises à jour des paramètres variationnels. Le détail des dérivations est disponible en Annexe A.2.

Espérance prédictive a posteriori. De la même manière que pour la PF, nous pouvons utiliser l'approximation variationnelle pour obtenir une approximation de l'espérance prédictive a posteriori. Ainsi, nous obtenons pour le modèle NBF :

$$\mathbb{E}(\mathbf{Y}^*|\mathbf{Y}) \approx \mathbb{E}_q(\mathbf{W}) \mathbb{E}_q(\mathbf{H}^T). \tag{2.28}$$

Nous utilisons donc le score $s_{ui} = [\mathbb{E}_q(\mathbf{W}) \mathbb{E}_q(\mathbf{H}^T)]_{ui}$ pour construire les listes de recommandation de chaque utilisateur.

Paramètres d'intensité. On peut remarquer que nous avons une invariance d'échelle entre les paramètres β^W et β^H. En effet, pour $\lambda \in \mathbb{R}_+$ fixé, on pose $\bar{\beta}^W = \lambda \beta^W$, $\bar{\beta}^H = \lambda^{-1} \beta^H$, $\bar{\Phi} = \{\alpha, \alpha^W, \alpha^H, \bar{\beta}^W, \bar{\beta}^H\}$ et $\bar{q}(\mathbf{W}) = \text{Gamma}(\tilde{\alpha}^W_{uk}, \lambda \tilde{\beta}^W_{uk})$, $\bar{q}(\mathbf{H}) = \text{Gamma}(\tilde{\alpha}^H_{ik}, \lambda^{-1} \tilde{\beta}^H_{ik})$. On

peut montrer que :

$$p(\mathbf{Y}; \bar{\Phi}) = p(\mathbf{Y}; \Phi), \tag{2.29}$$

$$\text{ELBO}(\bar{q}, \bar{\Phi}) = \text{ELBO}(\bar{q}, \Phi). \tag{2.30}$$

Par conséquent, on considère le paramètre d'intensité β^W fixé et égal à α^W, de sorte que $\mathbb{E}(w_{uk}) = 1$. De plus, comme nous avons $\mathbb{E}(a_{ui}) = 1$, l'information d'échelle est uniquement portée par β^H. En optimisant la ELBO par rapport à cette variable on obtient la règle de mise à jour suivante :

$$\beta^H = \frac{\alpha^H}{\sum_{ik} \mathbb{E}_q(h_{ik})/IK}. \tag{2.31}$$

Algorithme 3 : Algorithme CAVI pour la NBF

Données : Matrice d'observation $\mathbf{Y}$
Résultat : Distribution variationnelle q

Initialisation aléatoire des paramètres variationnels et de β^W ;
Calculer : $\tilde{\alpha}^A_{ui} = \alpha + y_{ui}$;
répéter
  pour chaque *couple* (u, i) *tel que* $y_{ui} > 0$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$;
    $\mathbb{E}_q(c_{uik}) = y_{ui} \dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
  fin
  pour chaque *couple* (u, i) **faire**
    $\tilde{\beta}^A_{ui} = \alpha + \sum_k \langle w_{uk} \rangle_q \langle h_{ik} \rangle_q$;
  fin
  pour chaque *utilisateur* $u \in \{1, \ldots, U\}$ **faire**
    $\tilde{\alpha}^W_{uk} = \alpha^W + \sum_i \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^W_{uk} = \beta^W + \sum_i \langle a_{ui} \rangle_q \langle h_{ik} \rangle_q$;
  fin
  pour chaque *article* $i \in \{1, \ldots, I\}$ **faire**
    $\tilde{\alpha}^H_{ik} = \alpha^H + \sum_u \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^H_{ik} = \beta^H + \sum_u \langle a_{ui} \rangle_q \langle w_{uk} \rangle_q$;
  fin
  Mise à jour du paramètre d'intensité : $\beta^H = \dfrac{\alpha^H}{\sum_{ik} \mathbb{E}_q(h_{ik})/IK}$;
  Calculer $\text{ELBO}(q, \Phi)$;
jusqu'à *ELBO converge*;

Complexité algorithmique. Contrairement aux algorithmes utilisés pour la PF, les algorithmes pour la NBF ne passent pas à l'échelle (que ce soit l'algorithme MM ou CAVI). En effet, à chaque itération nous estimons les variables locales a_{ui} qui sont définies pour tout couple utilisateur/article. Ce calcul peut être prohibitif dans le cadre des systèmes de recommandation où l'on travaille avec de très grands jeux de données qui sont généralement très creux.

2.5. Résultats expérimentaux

2.5.1. Protocole expérimental

Jeu de données. Dans cette expérience, nous utilisons le jeu de données Taste Profile (TP) [Ber+11]. Ce jeu de données contient l'historique d'écoutes de chansons par des utilisateurs. Nous utilisons le même pré-traitement que [Lia+16]. Nous sélectionnons un sous-ensemble des données, et ne conservons que les utilisateurs qui ont écouté au moins 20 chansons différentes, et les chansons qui ont été écoutées par au moins 50 utilisateurs différents. Finalement, notre matrice d'observation contient $U = 1\ 502$ utilisateurs et $I = 786$ articles.

Méthode d'évaluation. Nous utilisons la même méthode d'évaluation que celle présentée en Section 1.4.6 pour la PF. Notre jeu de données est donc divisé en un ensemble d'entraînement $\mathbf{Y}^{\text{train}}$ contenant 80% des valeurs non nulles du jeu de données original et un ensemble de test $\mathbf{Y}^{\text{test}}$ contenant les 20% restants (ces valeurs sont mises à zéro dans l'ensemble d'entraînement). L'algorithme VI est entraîné sur l'ensemble d'entraînement et fournit une liste de recommandations de 100 nouvelles chansons à chaque utilisateur fondée sur le score de prédiction : $[\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T)]_{ui}$. La qualité de ces listes est ensuite évaluée sur l'ensemble de test à l'aide de la métrique NDCG présentée en Section 1.1.3.

Méthodes comparées. On compare la NBF avec la PF appliquée aux données brutes (PFbrut) et la PF appliquée aux données binarisées (PFbin). Les hyper-paramètres sont fixés $\alpha^W = \alpha^H = 1$ (cela correspond à une loi exponentielle). Tous les algorithmes sont exécutés 5 fois avec des initialisations aléatoires. Le critère de convergence de l'algorithme est fixé à $\tau = 10^{-6}$.

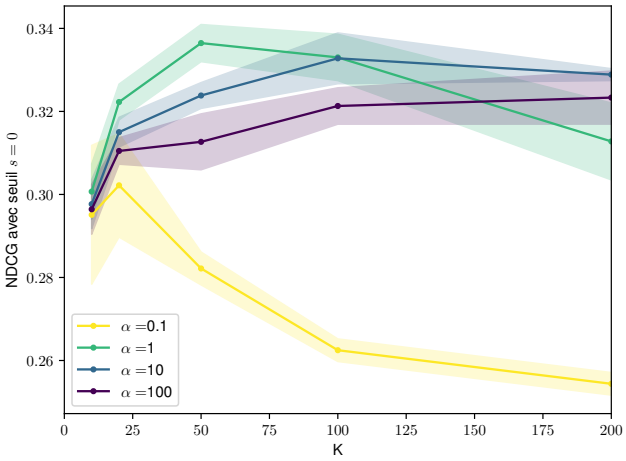


FIGURE 2.5. – Influence du paramètre α de la NBF.

2.5.2. Analyse des résultats

Influence du paramètre α. La Figure 2.5 représente l'influence du paramètre α et du nombre de facteurs latents K sur le comportement de la NBF pour des tâches de recommandation. L'axe des abscisses correspond au nombre de facteurs latents du modèle, testés parmi $K \in \{10, 20, 50, 100, 200\}$. L'axe des ordonnées correspond au score NDCG0 présenté en Section 1.1.3. Les quatre courbes affichées correspondent aux résultats de la NBF obtenus pour différentes valeurs de $\alpha \in \{0.1, 1, 10, 100\}$. On rappelle que lorsque α est très grand, la NBF tend vers la PF appliquée aux données brutes.

Premièrement, on constate que le nombre de facteurs latents optimal varie avec le paramètre α : plus α est petit, plus le K optimal est lui aussi petit. Comme nous l'avons vu précédemment, le paramètre α contrôle le degré de liberté de la variable $\mathbf{A}$. Lorsque α diminue, l'influence de la variable $\mathbf{A}$ augmente et permet d'expliquer de plus en plus les variations locales présentes dans les données. De ce fait, le terme de factorisation $\mathbf{WH}^T$ n'a besoin que d'un nombre de facteurs latents réduits pour modéliser la structure de $\mathbf{Y}$.

Néanmoins le choix du paramètre α semble être critique dans les performances de la NBF. En effet, si α est fixé à une valeur trop petite, le rôle de $\mathbf{A}$ va devenir prépondérant par rapport au produit matriciel $\mathbf{WH}^T$ dans l'inférence du modèle. La matrice $\mathbf{A}$ va absorber toute l'information présente dans les données et détruire le pouvoir prédictif du modèle porté par le produit matriciel $\mathbf{WH}^T$. On constate ce phénomène notamment pour la valeur

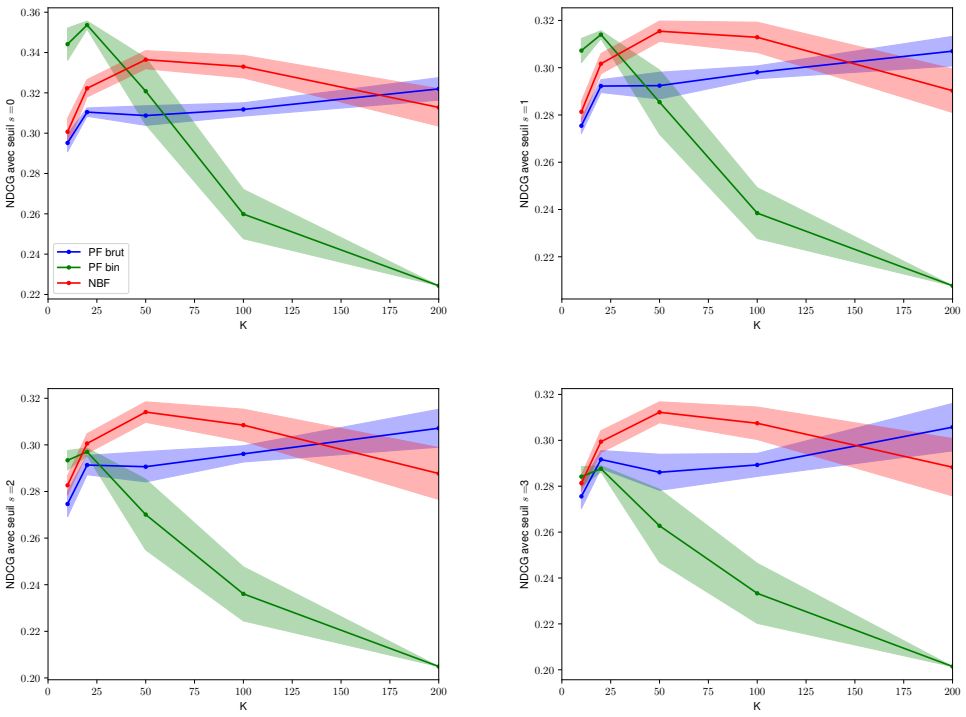


FIGURE 2.6. – Résultats de recommandation pour la NBF (avec $\alpha = 1$), PFbrut et PFbin pour différents seuils de pertinence $s \in \{0, 1, 2, 3\}$.

$\alpha = 0.1$ qui a un K optimal aux alentours de $K = 20$ mais qui a des faibles résultats de recommandation. Le paramètre α donnant les meilleurs résultats de recommandation (au sens du score NDCG0) semble être atteint pour $\alpha = 1$ et $K = 50$.

Nous avons tenté d'apprendre le paramètre α afin de diminuer le nombre de paramètres à régler dans le modèle. Pour cela, nous avons estimé α au sens du maximum de vraisemblance (nous avons une méthode de Newton-Raphson similaire à [Cem09] pour estimer le paramètre). Malheureusement, le paramètre inféré était de valeur très petite, ce qui place le modèle dans le cas pathologique présenté ci-dessus.

Résultats de recommandation. La Figure 2.6 présente les résultats de recommandation obtenus pour les modèles NBF (avec $\alpha = 1$), PFbrut et PFbin. Les quatre graphiques correspondent à différentes valeurs de seuil $s \in \{0, 1, 2, 3\}$ pour définir la pertinence d'une recommandation dans le score de NDCG (voir Section 1.1.3). L'ajout de variance par le biais du paramètre α dans la NBF permet de diminuer le nombre de facteurs latents nécessaires

pour faire des recommandations à partir de données brutes. PFbin reste toujours la méthode la plus performante pour le score NDCG0. Cependant, la NBF conserve la robustesse de PFbrut sur les seuils s plus élevés. À partir de $s = 1$, la NBF surpasse PFbin en terme de recommandations.

Ces résultats expérimentaux semblent donc être prometteurs. Néanmoins nous tenons à rappeler que la complexité algorithmique de la NBF ne nous permet pas de travailler sur des grands jeux de données. De plus, le paramètre α joue un rôle très important dans les performances du modèle. D'après notre expérience, il est difficile d'obtenir des résultats similaires sur des jeux de données plus larges.

2.6. Discussion

Dans ce chapitre, nous avons développé une première extension de la PF fondée sur la loi NB dans le but de mieux modéliser les données de comptage sur-dispersées. La NBF est paramétrée par sa moyenne et dispose d'un second paramètre permettant d'ajouter de la variance au modèle. Sa réécriture comme composition de lois Poisson et gamma permet de mettre en exergue une variable aléatoire d'exposition. Cette variable positive peut être interprétée comme un bruit multiplicatif expliquant les variabilités locales autour de la moyenne. La divergence associée à la NBF permet de définir un nouveau problème de NMF. Nous avons proposé deux approches pour résoudre le problème d'inférence. La première repose sur l'estimateur du maximum de vraisemblance pour lequel nous avons mis en place un algorithme MM. La seconde est une approche bayésienne où nous approximons la loi a posteriori à l'aide de la VI. Dans la dernière section de ce chapitre nous avons testé la NBF sur des tâches de recommandation à partir d'un jeu de données de petite taille.

Passage à l'échelle. La limite majeure de la NBF telle que nous l'avons présentée dans ce chapitre est que les deux algorithmes développés ne passent pas à l'échelle. En effet, ils reposent sur l'estimation des variables latentes d'exposition, d'une complexité algorithmique en $O(UI)$. Cela devient prohibitif lorsque la matrice $\mathbf{Y}$ est de très grande dimension. La NBF paraît donc inadaptée aux méthodes de filtrage collaboratif pour les systèmes de recommandation. Une alternative pouvant être envisagée pour réduire drastiquement le temps de calcul de l'algorithme VI serait d'imposer que la distribution variationnelle vérifie $q(a_{ui}) = \delta_1$ lorsque $y_{ui} = 0$. Ainsi, l'algorithme VI passerait à l'échelle avec le nombre de valeurs non nulles présentes dans la matrice $\mathbf{Y}$. Cette hypothèse supplémentaire supposerait néanmoins que chaque valeur nulle de $\mathbf{Y}$ provienne d'une loi Poisson et non plus d'une loi NB.

Rôle de la variable d'exposition. Même si l'interprétation de la variable d'exposition paraît claire, on observe parfois un comportement pathologique lors de l'inférence du modèle. En effet, cette variable n'est contrainte que par sa loi a priori paramétrée par le paramètre $\alpha > 0$, ce qui peut potentiellement lui procurer une grande liberté. Comme nous l'avons souligné plus tôt, plus le paramètre α est petit, plus la variable $\mathbf{A}$ peut expliquer les variations présentes dans la matrice $\mathbf{Y}$. Si le paramètre α est mal réglé (trop petit), la variable latente $\mathbf{A}$ va absorber des informations qui pourraient être utiles pour la recommandation (et qui devraient donc être capturées par le produit matriciel $\mathbf{WH}^T$). La répartition de l'information entre les paramètres $\mathbf{A}$ et $\mathbf{WH}^T$ paraît peu claire et très sensible au choix des paramètres. Un découplage des paramètres locaux et globaux semble nécessaire afin d'obtenir une réponse à ce problème. Le modèle présenté en Chapitre 3 répond notamment à ce point critique.

Perspectives en biologie. Au vu du problème de passage à l'échelle sur les très grandes données dont souffre la NBF, d'autres champs d'application que le CF sont à explorer. De ce fait, nous avons tenté d'utiliser l'algorithme MM (correspondant au cadre fréquentiste) sur d'autres tâches comme : la complétion de matrice, la séparation aveugle de source ou le traitement d'images hyper-spectrales. Malheureusement, nous n'avons pas observé de gain notable par rapport aux méthodes standards de NMF avec la divergence KL. Néanmoins, une perspective que nous n'avons pas encore exploité est d'appliquer la NBF à des données génomiques. En effet, dans ce domaine, la loi NB est souvent utilisée sur des matrices correspondant à l'expression de gène par des cellules [BF53 ; Dur16].

Chapitre 3.

Factorisation Poisson composée discrète

Ce chapitre est adapté de l'article [GOF19] publié à la conférence Uncertainty in Articial Intelligence *(UAI) en 2019.*

Contents

3.1. Introduction

Dans le Chapitre 2, nous avons présenté la factorisation NB (NBF, *negative binomial factorization* en anglais) sous la forme d'un mélange de lois Poisson et gamma. Cette extension de la PF permettait d'ajouter de la variance au modèle grâce l'introduction d'une variable d'exposition. Malheureusement, l'inférence de cette variable s'est avérée coûteuse et ne passait pas à l'échelle sur des jeux de données de grande dimension comme ceux utilisés en CF.

La factorisation Poisson composée (cPF, de l'anglais *compound Poisson factorization*) est une extension de la PF introduite dans [BE16]. Dans ce chapitre, nous nous intéressons à une instance de ce modèle appelée factorisation Poisson composée discrète (dcPF, *discrete compound Poisson factorization*) spécialement conçue pour les données de comptage sur-dispersées. Contrairement à la NBF, la dcPF préserve la propriété de passage à l'échelle de la PF. De plus, elle décorrèle le rôle des différents paramètres du modèle qui contrôlent la parcimonie et la sur-dispersion des données, permettant ainsi de mieux modéliser les données de comptage rencontrées en CF.

La dcPF est fondée sur la loi Poisson composée discrète (dcP, *discrete compound Poisson* en anglais) qui permet de compter des objets appartenant à des groupes. La loi dcP suppose que le nombre de groupes suit une loi Poisson, alors que le nombre d'objets par groupe suit une loi dite *élémentaire*. Cette distribution a notamment été utilisée pour la modélisation de groupes de populations [Ney39 ; Que49 ; Tho49]. Par exemple, dans [Que49], l'auteur étudie les populations de papillons. Il modélise le nombre d'espèces de papillon par une loi Poisson, tandis que le nombre de papillons par espèce est distribué selon une loi logarithmique.

En CF [BE16 ; BE17] et en analyse de documents [Zho17], les données arrivent souvent par rafale (le terme *bursty* est souvent employé en anglais [Kle03 ; SWZ16]), ce qui les rend sur-dispersées. Cela a amené certains auteurs à binariser les données avant d'appliquer leur modèle prédictif. Une autre approche consiste à modéliser la notion d'auto-excitation (*self-excitation* en anglais) pour expliquer les arrivées par rafale. En effet, l'auto-excitation traduit le fait qu'un événement «parent» puisse entraîner une salve d'événements «enfants». Par exemple, l'écoute d'une chanson peut en entraîner plusieurs autres ou bien l'emploi d'un mot peut conduire l'auteur d'un document à le ré-employer par la suite. Cette notion est aussi présente dans certains travaux sur les processus stochastiques poissoniens [Du+15 ; Hos+18 ; Kho+18 ; Zho17] où ce phénomène est modélisé temporellement. La loi cP nous

permet de modéliser l'auto-excitation via la loi élémentaire introduite dans le processus génératif des données.

La distribution Poisson composée (cP, *compound Poisson* en anglais) ne se limite pas au cas discret et peut aussi modéliser des données parcimonieuses continues. Dans [BE16 ; BE17], les auteurs considèrent aussi bien le choix de lois discrètes ou continues dans le contexte du CF. Cela peut par exemple permettre de modéliser le montant d'un panier d'achat sur un site web. Dans [YC12 ; SCY13], les auteurs s'intéressent à la distribution Tweedie qui est la distribution associée à la β-divergence (voir Section 1.2.3). En effet, pour $\beta \in (0, 1)$, la distribution Tweedie peut être représentée par une loi cP de loi élémentaire gamma. Cette représentation permet notamment d'apprendre le paramètre β.

Dans ce chapitre, nous présentons de nouvelles contributions pour la dcPF :

— Nous développons un cadre de travail unifié pour la dcPF. Nous étudions en particulier quatre distributions qui permettent de modéliser la notion d'auto-excitation et faisons le lien avec entre le choix de ces distributions et la combinatoire.

— Nous proposons une hypothèse simple et non contraignante sur le support de la distribution élémentaire afin de préserver la propriété de passage à l'échelle de la PF et d'obtenir des règles de mise à jour exactes pour l'estimation de la distribution a posteriori.

— Nous montrons que la PF appliquée aux données brutes et que la PF appliquée aux données binarisées sont deux cas limites de la dcPF. Cela fait de la dcPF une extension naturelle de la PF.

— Nous discutons du choix de la distribution élémentaire et en proposons une nouvelle dans le contexte des modèles Poisson composées, la loi NB translatée. Nous présentons une nouvelle méthodologie pour l'estimation des paramètres des distributions élémentaires et conduisons des expériences sur trois jeux de données différents.

Le reste du chapitre est organisé comme suit. Dans la Section 3.2, nous présentons quelques pré-requis sur la distribution cP et sur la famille exponentielle à dispersion qui nous permettent de motiver notre cadre de travail. Dans la Section 3.3, nous présentons le modèle dcPF et offrons une interprétation intuitive dans le contexte d'utilisateurs écoutant des chansons. Dans la Section 3.4, nous donnons l'exemple de quelques distributions élémentaires discrètes. Dans la Section 3.5, nous montrons que la dcPF est une extension de la PF et mettons en avant le choix d'un des paramètres du modèle. Dans la Section 3.6, nous proposons une inférence bayésienne du modèle qui passe à l'échelle et nous nous intéressons à l'estimation des paramètres. Dans la Section 3.7, nous conduisons des expériences de re-

commandations sur trois jeux de données différents. La Section 3.8 conclut ce chapitre et discute des perspectives.

3.2. Pré-requis

3.2.1. Distribution de Poisson composée

Une distribution de Poisson composée peut être représentée par le processus génératif suivant [JKK05] :

$$n \sim \text{Poisson}(\lambda), \tag{3.1}$$

$$x_l \sim \mathcal{E},\ \forall l \in \{1, \dots, n\}, \tag{3.2}$$

$$y = \sum_{l=1}^{n} x_l, \tag{3.3}$$

avec $y = 0$ si $n = 0$. $x_1, \dots, x_n$ sont n variables aléatoires indépendantes et identiquement distribuées. La distribution $\mathcal{E}$ associée à ces variables aléatoires est appelée *distribution élémentaire*. Dans la suite, on se restreint au cas où la distribution élémentaire est à support discret, i.e., $x_l \in \mathbb{N}$ pour tout $l \in \{1, \dots, n\}$, on parle alors de distribution Poisson composée discrète. On note $\text{Poisson}(\lambda) \bigvee \mathcal{E}$ la loi Poisson composée de distribution élémentaire $\mathcal{E}$ (notation reprise de [JKK05]). La fonction génératrice, définie par $G(z) = \mathbb{E}(z^y)$, associée à la variable aléatoire $y \sim \text{Poisson}(\lambda) \bigvee \mathcal{E}$ est donnée par $G(z) = \exp[\lambda(g(z) - 1)]$ où $g(z)$ est la fonction génératrice de la distribution élémentaire $\mathcal{E}$. On dit que la distribution associée à la variable y généralise la distribution élémentaire.

Représentation d'une loi Poisson composée. Les distributions Poisson composées discrètes représentent un grand nombre de distributions. En effet, on a le résultat suivant [Fel08] :

Théorème 3.1. *Toute distribution Poisson composée est une distribution infiniment divisible*[1]*. Réciproquement, toute distribution infiniment divisible à support discret peut s'écrire sous la forme d'une distribution de Poisson composée.*

Proposition 3.1 (Unicité)**.** $y \sim \text{Poisson}(\lambda) \bigvee \mathcal{E}$ *est équivalent à* $y \sim \text{Poisson}(\lambda(1-b)) \bigvee \mathcal{E}^b$

1. Une distribution est dite infiniment divisible si et seulement si pour tout entier n, sa fonction caractéristique ϕ peut s'écrire comme la puissance n-ième d'une fonction caractéristique ϕ_n, i.e., $\phi(t) = \phi_n(t)^n$.

où $\mathcal{E}^b$ est une distribution élémentaire modifiée définie par :

$$\mathcal{E}^b = \frac{1}{1-b}\left(\mathcal{E} - b\delta_0\right), \tag{3.4}$$

avec $b \in (-\infty, b_0]$, où b_0 est la masse en 0 de la distribution $\mathcal{E}$. Dans le cas où $b = b_0$, $\mathcal{E}^b$ correspond à la distribution $\mathcal{E}$ tronquée en zéro.

Démonstration. Soit $y \sim \text{Poisson}(\lambda) \bigvee \mathcal{E}$, la fonction génératrice associée peut se réécrire sous la forme :

$$G(z) = \exp\left[\lambda(g(z) - 1)\right] \tag{3.5}$$

$$= \exp\left[\lambda(1-b)\left(\frac{g(z)-b}{1-b} - 1\right)\right]. \tag{3.6}$$

La fonction génératrice $g^b(z) = \dfrac{g(z)-b}{1-b}$ correspond à la distribution élémentaire modifiée $\mathcal{E}^b$. □

Lemme 3.1. *La représentation d'une distribution infiniment divisible à support discret sous la forme d'une distribution de Poisson composée est unique si et seulement si la distribution élémentaire est à support dans $\mathbb{N}^*$.*

Sur-dispersion. Soit $y \sim \text{Poisson}(\lambda) \bigvee \mathcal{E}$ la distribution Poisson composée qui généralise la variable $x \sim \mathcal{E}$. L'espérance de la marginale est $\mathbb{E}(y) = \lambda\mathbb{E}(x)$ et, en utilisant le théorème de la variance totale, la variance est donnée par :

$$\text{var}(y) = \mathbb{E}(\text{var}(y|n)) + \text{var}(\mathbb{E}(y|n)) \tag{3.7}$$

$$= \lambda\left(\text{var}(x) + \mathbb{E}(x)^2\right) \tag{3.8}$$

$$= \lambda\mathbb{E}(x^2). \tag{3.9}$$

Le rapport variance/espérance d'une distribution Poisson composée est donc donné par $\frac{\text{var}(y)}{\mathbb{E}(y)} = \frac{\mathbb{E}(x^2)}{\mathbb{E}(x)} \geq 1$, puisque nous avons supposé que la distribution élémentaire est à support dans $\mathbb{N}$ (et donc $x^2 \geq x$). Par conséquent, la distribution Poisson composée est particulièrement adaptée aux données sur-dispersées.

Quelques exemples. Il existe dans la littérature beaucoup d'exemples de distributions Poisson composées définies avec des distributions élémentaires discrètes. Elles ont été lar-

Tableau 3.1. – Exemple de distributions Poisson composée.

Loi élémentaire	Loi marginale	Référence
Logarithmique	Binomiale négative	[Que49 ; Zho17]
Poisson (ou ZTP [2].)	Neyman de type A	[Ney39]
Poisson translatée	Thomas	[Tho49]
Géo. [3] (ou géo. translatée)	Pólya–Aeppli	[Pól30]
NB (ou NB tronquée en 0)	Pólya-Aeppli généralisée	[Ske52]
Bernoulli	Poisson	

gement étudiées dans le cadre de la modélisation de populations. Le Tableau 3.1 présente quelques lois élémentaires ainsi que la loi marginale de la loi dcP associée.

3.2.2. Famille exponentielle à dispersion discrète

Un élément central des distributions Poisson composées est le choix de la distribution élémentaire. Dans la suite de ce chapitre, on supposera que cette distribution appartient à la famille exponentielle à dispersion (EDM, *exponential dispersion model* en anglais) [Jør86 ; Jor87 ; Jor97], qui est une extension de la famille exponentielle. C'est un choix pratique dans le cadre des modèles composés [YC12 ; SCY13 ; BE16]. Comme évoqué précédemment, nous nous restreignons au cas où la distribution élémentaire est à support discret. La fonction de densité d'une EDM discrète peut s'écrire sous la forme :

$$p(x;\theta,\kappa) = \exp(x\theta - \kappa\psi(\theta))h(x,\kappa),\ \ x \in S_\kappa \tag{3.10}$$

où $\theta \in \Theta$ est le paramètre naturel, $\kappa > 0$ est le paramètre de dispersion, $\psi(\theta)$ est la fonction de log-partition, $h(x,\kappa)$ est la mesure de base, et S_κ est le support (qui dépend du paramètre de dispersion). On note $x \sim \mathrm{ED}(\theta,\kappa)$. L'espérance d'une EDM est donnée par $\mathbb{E}(x) = \kappa\psi'(\theta)$ et sa variance par $\mathrm{var}(x) = \kappa\psi''(\theta)$.

Une propriété qui nous sera utile par la suite est la propriété d'additivité des EDM [Jor87].

Proposition 3.2. *Soient n variables $x_1,\dots,x_n$ i.i.d. selon $x_l \sim \mathrm{ED}(\theta,\kappa_l)$ et $y = \sum_{l=1}^n x_l$, alors on a $y \sim \mathrm{ED}(\theta,\sum_{l=1}^n \kappa_l)$. En particulier, si $\kappa_l = \kappa$ pour tout $l \in \{1,\dots,n\}$ alors $y \sim \mathrm{ED}(\theta, n\kappa)$.*

2. *Zero-truncated Poisson* en anglais
3. Géo. fait référence à la loi géométrique.

3.3. Factorisation Poisson composée discrète

3.3.1. Modèle génératif

On considère le modèle proposé par [BE16]. Le modèle génératif des observations $\mathbf{Y}$ est donné par :

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta_u^W),\ h_{ik} \sim \text{Gamma}(\alpha^H, \beta_i^H), \tag{3.11}$$

$$n_{ui} \sim \text{Poisson}\left([\mathbf{W}\mathbf{H}^T]_{ui}\right), \tag{3.12}$$

$$x_{l,ui} \sim \text{ED}(\theta, \kappa),\ \forall l \in \{1, \dots, n_{ui}\}, \tag{3.13}$$

$$y_{ui} = \sum_{l=1}^{n_{ui}} x_{l,ui}. \tag{3.14}$$

On suppose ici que $x_{l,ui}$ est une variable aléatoire discrète et que son support est égal à $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. Ce choix n'est pas restrictif puisque, d'après le lemme 3.1, si le support est égal à $\mathbb{N}$ alors on peut se ramener au modèle présenté ci-dessus par un simple changement de variables. De plus, cette simple contrainte nous permet d'obtenir des propriétés qui seront détaillées dans la suite.

3.3.2. Interprétation : la notion de sessions d'écoutes

Dans cette section, on s'intéresse à un couple utilisateur/article fixé. Par souci de clarté, on omettra les indices $_{ui}$. Comparé à la PF, la dcPF introduit de nouvelles variables latentes n et $\{x_l\}_{l \in \{1,\dots,n\}}$. La variable latente n représente le nombre de sessions d'écoute d'un utilisateur associé à une chanson. Pour chaque chaque session, indexée par l, l'utilisateur écoute la chanson un nombre de fois $x_l \geq 1$. Cette variable latente x_l modélise le concept d'auto-excitation induit par une écoute de chanson [Zho17 ; Hos+18] : un utilisateur peut écouter une chanson, non pas parce qu'il aime cette chanson, mais en réponse à une précédente écoute (excitation). Par exemple, un utilisateur peut écouter sans arrêt une chanson sans que cela traduise une préférence sur le long-terme. Au sein d'une session, la première écoute d'un utilisateur reflète son intérêt pour la chanson, alors que les $x_l - 1$ écoutes suivantes ne sont que l'écho de la première écoute et traduisent une préférence sur le court-terme. Le nombre total d'écoutes y est alors l'agrégation de toutes les écoutes provenant des différentes sessions.

Par conséquent, le nombre total d'écoutes y peut être regroupé en un plus petit nombre de sessions d'écoute n qui sont plus à-même de représenter les préférences de l'utilisateur pour

la chanson sur le long-terme. La variable n peut donc être vue comme une façon de répartir les y écoutes en un plus petit nombre de sessions. Dans la suite, on notera $\mathbf{N} \in \mathbb{N}^{U \times I}$ la matrice des sessions d'écoute, avec pour coefficients $[\mathbf{N}]_{ui} = n_{ui}$.

3.3.3. Log-vraisemblance jointe

En utilisant la propriété d'additivité des EDM (Prop. 3.2), on peut facilement marginaliser les variables latentes $x_{l,ui}$, ce qui mène à : $y_{ui} \sim \mathrm{ED}(\theta, n_{ui}\kappa)$. Ainsi, la log-vraisemblance jointe des observations $\mathbf{Y}$ et des variables latentes $\mathbf{N}$, $\mathbf{W}$ et $\mathbf{H}$ est donnée par :

$$\log p(\mathbf{Y}, \mathbf{N}, \mathbf{W}, \mathbf{H}) = \underbrace{\log p(\mathbf{Y}|\mathbf{N}; \theta, \kappa)}_{\text{Mapping}} + \underbrace{\log p(\mathbf{N}|[\mathbf{W}\mathbf{H}^T])}_{\text{Structure PF}} + \underbrace{\log p(\mathbf{W}, \mathbf{H})}_{\text{Régularisation}}. \tag{3.15}$$

Cette log-vraisemblance peut être décomposée en trois termes : un *mapping* probabiliste entre les observations et le nombre de sessions, la structure PF sur la variable latente $\mathbf{N}$, un terme de régularisation induit par les a priori gamma sur $\mathbf{W}$ et $\mathbf{H}$. Contrairement à la PF, la factorisation est placée sur la variable latente $\mathbf{N}$ et non pas directement sur les observations $\mathbf{Y}$. Si l'on revient à notre interprétation, cela signifie que la variable latente $\mathbf{N}$ est plus à-même de donner des informations sur les préférences des utilisateurs que les observations brutes $\mathbf{Y}$. Le terme de *mapping* peut être vu comme une distorsion probabiliste appliquée aux données brutes, les rendant de ce fait «plus factorisables». Ce terme additionnel permet de travailler directement avec les données brutes et d'éviter tout pré-traitement (comme la binarisation des données) en laissant les données choisir la meilleure distorsion possible (au sens de la PF).

3.3.4. Propriétés

En imposant que les utilisateurs écoutent une chanson au moins une fois lors de chaque session, i.e., $x_{l,ui} \geq 1$, deux propriétés importantes peuvent être déduites.

Premièrement, nous avons l'équivalence entre les zéros de $\mathbf{Y}$ et ceux de $\mathbf{N}$, i.e., $y_{ui} = 0 \Leftrightarrow n_{ui} = 0$. En d'autres termes, le nombre d'écoutes est égal à zéro si et seulement si le nombre de sessions est égal à zéro. De ce fait, la variable latente $\mathbf{N}$ est partiellement observée et est aussi une matrice creuse puisqu'elle a les même zéros que ceux de $\mathbf{Y}$. Grâce à cela, la propriété de passage à l'échelle est préservée pour la dcPF. De plus, on a :

$$\mathbb{P}(y_{ui} = 0) = \mathbb{P}(n_{ui} = 0) = e^{-[\mathbf{W}\mathbf{H}^T]_{ui}}. \tag{3.16}$$

Tableau 3.2. – Exemple de quatre distributions élémentaires.

Distribution	θ	Θ	θ^{brut}	θ^{bin}	κ	$\psi(\theta)$	$h(x,\kappa)$
$x \sim \text{Log}(p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	1	$\log(-\log(1-e^{\theta}))$	$\frac{x!}{\kappa!}St_1(x,\kappa)$
$x \sim \text{ZTP}(p)$	$\log(p)$	$\mathbb{R}$	$-\infty$	$+\infty$	1	$\log(e^{e^{\theta}}-1)$	$\frac{x!}{\kappa!}St_2(x,\kappa)$
$x \sim \text{sGeo}(1-p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	1	$\log(\frac{e^{\theta}}{1-e^{\theta}})$	$\frac{x!}{\kappa!}St_3(x,\kappa)$
$x \sim \text{sNB}(a,p)$	$\log(p)$	$\mathbb{R}_-^*$	$-\infty$	0	$(1,a)^T$	$(\theta,-\log(1-e^{\theta}))^T$	$\frac{\Gamma(x-\kappa_1+\kappa_2)}{\Gamma(x-\kappa_1+1)\Gamma(\kappa_2)}$

On obtient donc le découplage suivant : les variables $\mathbf{W}$ et $\mathbf{H}$ sont les seules à contrôler la parcimonie de la matrice $\mathbf{Y}$, tandis que la distribution élémentaire et ses paramètres $\{\theta, \kappa\}$ se focalisent seulement sur la représentation des valeurs strictement positives.

La deuxième propriété qui nous intéresse ici est que $n_{ui} \leq y_{ui}$. En d'autres termes, le nombre de sessions est majoré par le nombre total d'écoutes. Ainsi, pour une observation y_{ui} donnée, la variable n_{ui} ne peut prendre qu'un nombre fini de valeurs : $n_{ui} \in \{0, ..., y_{ui}\}$ (en particulier, on a $y_{ui} = 1 \Leftrightarrow n_{ui} = 1$). Lors de l'inférence, cette propriété permet d'avoir une forme analytique pour la mise à jour de $\mathbf{N}$.

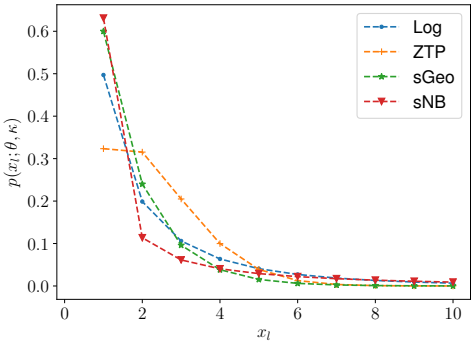
3.4. Exemples de distributions

Le choix de la distribution élémentaire est crucial dans le modèle présenté en Section 3.3.1. Comme expliqué précédemment, elle contrôle la distribution des valeurs non nulles. Dans cette section, on détaillera le choix de quatre distributions discrètes dont le support est $\mathbb{N}^*$. Les quatre distributions sont résumées dans le Tableau 3.2 et leur fonction de masse est affichée en Figure 3.1(a). Pour chaque distribution présentée, la distribution marginale y est affichée en Figure 3.1(b) et est comparée à la distribution Poisson (qui est une distribution Poisson composée de distributions Dirac). Par souci de clarté, les indices $_{ui}$ seront à nouveau omis.

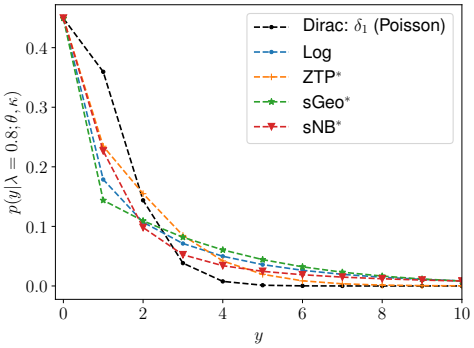
3.4.1. Distributions de Stirling

Dans cette section, on se focalise sur trois lois élémentaires particulières qui peuvent s'écrire sous la forme d'une EDM discrète avec un paramètre de dispersion $\kappa = 1$ (voir Tableau 3.2) :

— la loi logarithmique, notée $x_l \sim \text{Log}(p)$;



(a) Fonction de masse de quatre distributions élémentaires.



(b) Fonction de masse de la distribution marginale des observations. Les fonctions de masses marquées par un astérisque ne sont pas disponibles en forme analytique et sont représentées par un histogramme de valeurs simulées.

FIGURE 3.1. – Fonctions de masse de quatre distributions élémentaires (a) et des distributions marginales associées (b).

— la loi Poisson tronquée en zéro (ZTP), notée $x_l \sim \text{ZTP}(p)$;

— la loi géométrique translatée, notée $x_l \sim \text{sGeo}(1-p)$.

Ces trois lois sont appelées lois de Stirling car elles font intervenir les nombres de Stirling de première, deuxième et troisième espèce respectivement et sont détaillés dans les sections suivantes.

Loi logarithmique

La fonction de masse de la loi logarithmique avec $p \in (0,1)$ est donnée par :

$$\text{Log}(x;p) = \frac{1}{-\log(1-p)}\frac{p^x}{x},\ x \in \mathbb{N}^*. \tag{3.17}$$

Soit $y = \sum_{l=1}^{n} x_l$, où $x_l \sim \text{Log}(p)$ alors $y \sim \text{SumLog}(n,p)$, aussi appelée distribution de Stirling de première espèce. Sa fonction de masse est donnée par :

$$\text{SumLog}(y;n,p) = \frac{n!}{y!}St_1(y,n)\frac{p^y}{(-\log(1-p))^n},\ y \in \{n,\dots,+\infty\}, \tag{3.18}$$

où St_1 correspond au nombre de Stirling non signé de première espèce. Son écriture sous la forme d'une EDM est décrite dans le Tableau 3.2.

Si $n \sim \text{Poisson}(\lambda)$, alors la distribution marginale de y est une loi binomiale négative [Que49] :

$$y \sim \text{NB}\left(r, p\right), \text{avec } r = \frac{\lambda}{-\log(1-p)}. \tag{3.19}$$

Contrairement au Chapitre 2, le paramètre de probabilité p est fixé et le paramètre d'intérêt λ, qui possède une structure de rang-faible, est placé sur le paramètre de forme. Cela correspond au modèle introduit dans [Zho17] à un changement de variable près.

La distribution conditionnelle de la variable latente n est donnée par :

$$n|y \sim \text{CRT}\left(y, r\right), \tag{3.20}$$

où CRT est la distribution du nombre de tables dans un processus de restaurant chinois (CRP, *chinese restaurant process* en anglais) [ZC15]. L'espérance de cette loi conditionnelle est connue sous forme analytique : $\mathbb{E}(n|y) = r\left(\Psi(y+r) - \Psi(r)\right)$.

Loi Poisson tronquée en zéro

La fonction de masse de la loi Poisson tronquée en zéro avec $p \in \mathbb{R}_+$ est donnée par :

$$\text{ZTP}(x; p) = \frac{1}{e^p - 1}\frac{p^x}{x!}, \ x \in \mathbb{N}^*. \tag{3.21}$$

Soit $y = \sum_{l=1}^{n} x_l$, où $x_l \sim \text{ZTP}(p)$ alors $y \sim \text{SumZTP}(n, p)$, aussi appelée distribution de Stirling de deuxième espèce. Sa fonction de masse est donnée par [SV06] :

$$\text{SumZTP}(y; n, p) = \frac{n!}{y!} St_2(y, n) \frac{p^y}{(e^p - 1)^n}, y \in \{n, \dots, +\infty\}, \tag{3.22}$$

où St_2 correspond au nombre de Stirling non signé de seconde espèce. Son écriture sous la forme d'une EDM est décrite dans le Tableau 3.2.

La distribution marginale de y est une distribution de Neyman de type A [Ney39]. En effet, d'après la propriété 3.1, $y \sim \text{Poisson}(\lambda) \bigvee \text{ZTP}(p)$ est équivalent à $y \sim \text{Poisson}(\frac{\lambda}{1-e^{-p}}) \bigvee \text{Poisson}(p)$ qui correspond à la loi de Neyman de type A.

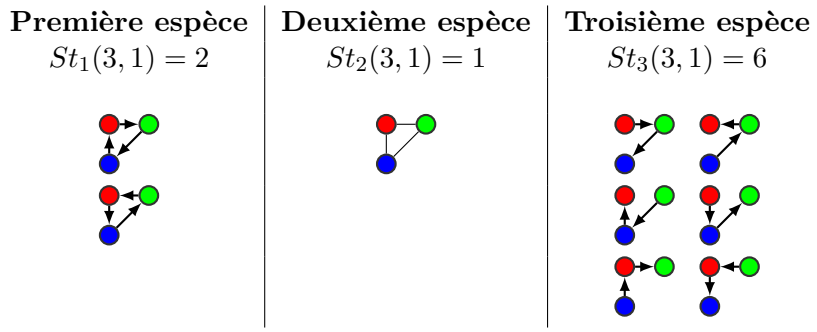


FIGURE 3.2. – Illustration des nombres de Stirling des trois espèces pour $y = 3$ et $n = 1$.

Loi géométrique translatée

La fonction de masse de la loi géométrique translatée, notée $x \sim \text{sGeo}(1-p)$ avec $p \in (0,1)$, et qui correspond à $x - 1 \sim \text{Geo}(1-p)$, est donnée par :

$$\text{sGeo}(x; 1-p) = p^{x-1}(1-p), \ \ x \in \mathbb{N}^*. \tag{3.23}$$

Soit $y = \sum_{l=1}^{n} x_l$ où $x_l \sim \text{sGeo}(1-p)$, alors on a $y - n \sim \text{NB}(n,p)$ (dans le cas où le paramètre de forme de la NB est entier, la distribution est aussi appelée distribution de Pascal). Sa densité peut peut être réécrite sous la forme d'une EDM discrète (voir Tableau 3.2). Comme pour les deux lois précédentes, nous pouvons faire apparaître un terme correspondant au nombre de Stirling de troisième espèce, plus communément appelé nombre de Lah et défini par : $St_3(y,n) = \binom{y-1}{n-1}\frac{y!}{n!}$.

La distribution marginale de la variable y est une loi de Pólya–Aeppli [Pól30]. D'après la propriété 3.1, on a $y \sim \text{Poisson}(\lambda) \bigvee \text{sGeo}(1-p)$ qui est équivalent à $y \sim \text{Poisson}(\frac{\lambda}{p}) \bigvee \text{Geo}(1-p)$, la loi géométrique tronquée en zéro étant égale à la loi géométrique translatée.

Nombres de Stirling

Les trois distributions présentées peuvent donc s'écrire sous la forme d'une EDM discrète avec un paramètre de dispersion $\kappa = 1$, et une mesure de base faisant apparaître un nombre de Stirling :

$$h(x,\kappa) = \frac{x!}{\kappa!} St_j(x,\kappa), \tag{3.24}$$

où St_j représente les nombres de Stirling non signés de première ($j = 1$), seconde ($j = 2$) et troisième espèce ($j = 3$).

Les nombres de Stirling des trois espèces sont autant de façons de répartir y éléments en n groupes [Rio12]. La Figure 3.2 illustre les nombres de Stirling pour $y = 3$ éléments à répartir entre $n = 1$ groupe.

— Le nombre de Stirling de première espèce correspond au nombre de façons de répartir y éléments en n cycles disjoints. Il est obtenu grâce à la formule de récurrence $St_1(y+1, n) = y\ St_1(y, n) + St_1(y, n-1)$.

— Le nombre de Stirling de seconde espèce correspond au nombre de façons de répartir y éléments en n sous ensembles non vides. Il peut être calculé analytiquement : $St_2(y, n) = \frac{1}{n!}\sum_{j=0}^{n}(-1)^{n-j}\binom{n}{j}j^y$. Lorsque y est très grand, son calcul peut souffrir de problèmes numériques, mais de bonnes approximations existent [BW74].

— Le nombre de Stirling de troisième espèce (ou nombre de Lah) correspond au nombre de façons de répartir y éléments en n sous ensembles ordonnés non vides. Il se calcule ainsi : $St_3(y, n) = \binom{y-1}{n-1}\frac{y!}{n!}$. Cette définition semble particulièrement adaptée lorsque les groupes résultent de phénomènes temporels.

3.4.2. Distribution binomiale négative translatée

Dans cette section, nous introduisons une quatrième distribution élémentaire, la loi NB translatée, notée sNB. Lorsque $x_l \sim \text{sNB}(a, p)$, cela correspond à $x_l - 1 \sim \text{NB}(a, p)$. Le paramètre de forme a contrôle la queue de la distribution et $p \in (0, 1)$ est un paramètre de probabilité. La loi NB translatée est une EDM translatée et n'appartient pas exactement à la famille EDM. Cependant, on peut écrire la densité de $y = \sum_{l=1}^{n} x_l$ comme :

$$p(y; n, \theta, \kappa) = \exp(y\theta - n\boldsymbol{\kappa}^T\psi(\theta))h(y, n\kappa), \tag{3.25}$$

où $y \in \{n, \ldots, +\infty\}$, $\boldsymbol{\kappa} = (\kappa_1, \kappa_2)^T = (1, a)^T$ et $\psi(\theta) = (\theta, -\log(1-e^\theta))^T$. Il est important de noter que $\boldsymbol{\kappa}$ et $\psi(\theta)$ sont ici des vecteurs de dimension 2. Contrairement aux trois précédents exemples, le paramètre de dispersion n'est pas fixé à 1 et offre donc un degré de liberté supplémentaire. Ainsi, le paramètre κ_1 contrôle la translation et κ_2 la forme de la loi NB.

Les lois élémentaires translatées ont une interprétation particulièrement intéressante dans notre contexte. En effet, avec ce type de loi, on modélise le nombre d'événements «enfants» découlant de l'événement «parent» et non pas le nombre d'événement total. Dans notre exemple, la loi NB modélise le nombre d'écoutes successives à une première écoute, le nombre

total d'écoute étant alors distribué selon une loi sNB. C'est cette même réflexion qui a amené Thomas [Tho49] a utiliser une loi de Poisson translatée pour modéliser la distribution de plantes.

À notre connaissance, la loi marginale de la variable y n'a pas de nom connu dans la littérature. Elle peut être vue comme une généralisation de la distribution de Thomas [Tho49] qui généralise la loi élémentaire Poisson translatée. À noter que cette distribution marginale est bien différente de la loi Pólya–Aeppli généralisée [Ske52], qui généralise une loi NB (non translatée) ou une loi NB tronquée en zéro d'après la proposition 3.1.

3.5. Un compromis entre la factorisation Poisson appliquée aux données brutes et aux données binarisées

Dans cette section, nous montrons que la dcPF généralise la PF ; la PF appliquée aux données brutes et la PF appliquée aux données binarisées étant vues comme deux cas limites. Pour un paramètre de dispersion κ fixé, le paramètre naturel contrôle le niveau d'information contenu dans les observations $\mathbf{Y}$:

— Quand θ tend vers une certaine limite θ^{brut}, la dcPF devient équivalente à la PF appliquée aux données brutes.

— Quand θ tend vers une certaine limite θ^{bin}, la loi a posteriori de $\mathbf{W}$ et $\mathbf{H}$ sachant les données brutes $\mathbf{Y}$ de dcPF devient équivalente à celle de PF appliquée aux données binarisées. En d'autres termes, appliquer dcPF sur les données brutes devient équivalent à appliquer PF sur les données binarisées. À noter que, dans ce cas, la distribution de $y|n$ devient dégénérée, mais la distribution a posteriori reste bien définie [Rob07].

— Entre ces deux cas limites, nous obtenons un continuum contrôlé par le paramètre naturel θ. Il contrôle le degré de *distorsion implicite* des observations.

Ces résultats sont formalisés dans les propositions suivantes :

Proposition 3.3. *S'il existe θ^{brut} tel que $\lim_{\theta\to\theta^{brut}} \kappa^T\psi(\theta) = -\infty$, alors la loi a posteriori de dcPF tend vers la loi a posteriori de PF lorsque θ tend vers θ^{brut}.*

Proposition 3.4. *S'il existe θ^{bin} tel que $\lim_{\theta\to\theta^{bin}} \kappa^T\psi(\theta) = +\infty$, alors la loi a posteriori de dcPF tend vers la loi a posteriori de PF appliqué aux données binarisées lorsque θ tend vers θ^{bin}, i.e. : $\lim_{\theta\to\theta^{bin}} p(\mathbf{W},\mathbf{H}|\mathbf{Y}) = p(\mathbf{W},\mathbf{H}|\mathbf{N} = \mathbf{Y}^b)$.*

Les quatre distributions décrites en Section 3.4 respectent les hypothèses des deux propositions. Les cas limites du paramètre naturel sont présentés dans le Tableau 3.2.

Démonstration. Soit $\lambda \in \mathbb{R}_+$, $n \sim \text{Poisson}(\lambda)$ et $y|n \sim \text{ED}(\theta, n\kappa)$ dont le support est $S = \{n, \dots, +\infty\}$. On note $r = \lambda e^{-\kappa^T \psi(\theta)}$.

La distribution a posteriori de la variable n pour $y > 0$ est donnée par :

$$p(n|y) = \frac{r^n h(y, n\kappa)(n!)^{-1}}{\sum_{m=1}^{y} r^m h(y, m\kappa)(m!)^{-1}},\ n \in \{1, \dots, y\}. \tag{3.26}$$

Ainsi, pour κ et $y > 0$ fixés, on a les équivalences :

$$\sum_{m=1}^{y} r^m h(y, m\kappa)(m!)^{-1} \underset{r \to +\infty}{\sim} r^y h(y, y\kappa)(y!)^{-1}, \tag{3.27}$$

$$\underset{r \to 0}{\sim} r h(y, \kappa). \tag{3.28}$$

On en déduit les deux cas limites : $\lim_{r \to +\infty} p(n|y) = \delta_y(n)$ et $\lim_{r \to 0} p(n|y) = \delta_1(n)$. De ces résultats, on peut déduire que pour la dcPF, en supposant que :

— il existe θ^{brut} tel que $\lim_{\theta \to \theta^{\text{brut}}} \kappa^T \psi(\theta) = -\infty$ (et donc $r \to +\infty$),

— il existe θ^{bin} tel que $\lim_{\theta \to \theta^{\text{bin}}} \kappa^T \psi(\theta) = +\infty$ (et donc $r \to 0$),

on a les cas limites suivants :

$$p(\mathbf{N}|\mathbf{Y}) = \int_{\mathbf{W},\mathbf{H}} p(\mathbf{N}|\mathbf{Y}, \mathbf{W}, \mathbf{H}) p(\mathbf{W}, \mathbf{H}|\mathbf{Y}) \mathrm{d}\mathbf{W} \mathrm{d}\mathbf{H} \tag{3.29}$$

$$\xrightarrow[\theta \to \theta^{\text{brut}}]{} \int_{\mathbf{W},\mathbf{H}} \delta_{\mathbf{Y}}(\mathbf{N})\ p(\mathbf{W}, \mathbf{H}|\mathbf{Y}) \mathrm{d}\mathbf{W} \mathrm{d}\mathbf{H} = \delta_{\mathbf{Y}}(\mathbf{N}) \tag{3.30}$$

$$\xrightarrow[\theta \to \theta^{\text{bin}}]{} \int_{\mathbf{W},\mathbf{H}} \delta_{\mathbf{Y}^b}(\mathbf{N})\ p(\mathbf{W}, \mathbf{H}|\mathbf{Y}) \mathrm{d}\mathbf{W} \mathrm{d}\mathbf{H} = \delta_{\mathbf{Y}^b}(\mathbf{N}). \tag{3.31}$$

Et finalement, pour la distribution a posteriori :

$$p(\mathbf{W}, \mathbf{H}|\mathbf{Y}) = \int_{\mathbf{N}} p(\mathbf{W}, \mathbf{H}|\mathbf{N}) p(\mathbf{N}|\mathbf{Y}) d\mathbf{N} \tag{3.32}$$

$$\xrightarrow[\theta \to \theta^{\text{brut}}]{} p(\mathbf{W}, \mathbf{H}|\mathbf{N} = \mathbf{Y}) \tag{3.33}$$

$$\xrightarrow[\theta \to \theta^{\text{bin}}]{} p(\mathbf{W}, \mathbf{H}|\mathbf{N} = \mathbf{Y}^b), \tag{3.34}$$

où $p(\mathbf{W}, \mathbf{H}|\mathbf{N})$ est la loi a posteriori d'un modèle PF avec des données brutes ou binarisées respectivement. □

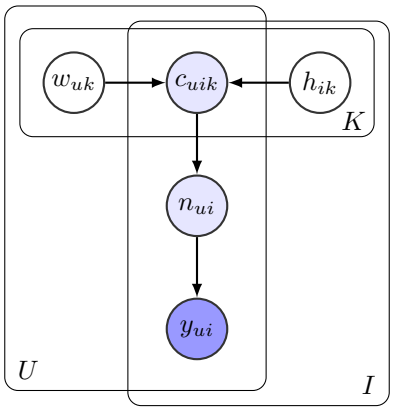


FIGURE 3.3. – Représentation graphique du modèle augmenté dcPF. En bleu foncé : les variables observées ; en bleu clair : les variables partiellement observées.

Apprentissage du paramètre naturel Il est particulièrement intéressant d'apprendre le paramètre naturel θ puisque celui-ci caractérise les données. Si θ est proche de θ^{brut}, les observations n'ont pas besoin d'être «distordues» (elles ne sont pas trop sur-dispersées) et PF sur les données brutes est efficace. Si θ est proche de θ^{bin}, les valeurs strictement positives de $\mathbf{Y}$ sont non informatives et une étape de pré-traitement rendant les données binaires est bienvenue. Entre ces deux cas extrêmes, la valeur de θ quantifie le «lissage» dont les données ont besoin avant d'appliquer une PF.

3.6. Estimation bayésienne

3.6.1. Inférence variationnelle

Dans cette section, nous nous intéressons à l'estimation de la distribution a posteriori $p(\mathbf{W}, \mathbf{H}|\mathbf{Y})$. Pour cela, nous considérons le modèle augmenté suivant :

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta_u^W), \; h_{ik} \sim \text{Gamma}(\alpha^H, \beta_i^H), \tag{3.35}$$

$$c_{uik}|\mathbf{W}, \mathbf{H} \sim \text{Poisson}(w_{uk}h_{ik}), \tag{3.36}$$

$$n_{ui} = \sum_k c_{uik}, \tag{3.37}$$

$$y_{ui} \sim \text{ED}(\theta, \kappa n_{ui}). \tag{3.38}$$

On note $\mathbf{Z} = \{\mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes et $\Phi = \Phi_1 \cup \Phi_2$ l'ensemble des paramètres avec $\Phi_1 = \{\theta, \kappa\}$ et $\Phi_2 = \{\alpha^W, \alpha^H, \boldsymbol{\beta}^W, \boldsymbol{\beta}^H\}$.

On utilise une fois encore l'inférence variationnelle et l'hypothèse du champ moyen afin d'approximer la loi a posteriori insoluble $p(\mathbf{Z}|\mathbf{Y})$ par une distribution de la forme :

$$q(\mathbf{Z}) = \prod_{ui} q(n_{ui}, \mathbf{c}_{ui}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \tag{3.39}$$

Il est important de noter que les variables n_{ui} et $\mathbf{c}_{ui}$ restent ici couplées et nous utilisons la décomposition suivante :

$$q(n_{ui}, \mathbf{c}_{ui}) = q(\mathbf{c}_{ui}|n_{ui})q(n_{ui}). \tag{3.40}$$

Le Tableau 3.3 récapitule l'expression analytique des différentes distributions variationnelles. L'Algorithme 4 présente les règles de mise à jour pour les paramètres variationnels. En particulier, on a $\mathbb{E}_q(c_{uik}) = \mathbb{E}_q(n_{ui})\tilde{\phi}_{uik}$. Comparée à la PF, la dcPF nécessite le calcul supplémentaire de $\mathbb{E}_q(n_{ui})$ pour chaque couple (u, i) tel que $y_{ui} > 0$. Grâce à la propriété présentée en Section 3.3.4, cette quantité est disponible analytiquement :

$$\mathbb{E}_q(n_{ui}) = \begin{cases} 0, \text{ si } y_{ui} = 0, \\ \sum_{n=1}^{y_{ui}} nq(n_{ui} = n), \text{ sinon.} \end{cases} \tag{3.41}$$

Nous rappelons que lorsque la distribution élémentaire est la loi logarithmique nous avons : $\mathbb{E}_q(n_{ui}) = r_{ui}(\Psi(y_{ui} + r_{ui}) - \Psi(r_{ui}))$ [ZC15]. Comme prévu, nous retrouvons les règles de mise à jour de la PF (voir Section 1.4) appliquée aux données brutes si $\mathbb{E}_q(n_{ui}) = y_{ui}$ et aux données binarisées si $\mathbb{E}_q(n_{ui}) = \mathbb{1}[y_{ui} > 0]$.

Distribution prédictive a posteriori. La distribution prédictive a posteriori de nouvelles données $\mathbf{Y}^*$ sachant les données observées $\mathbf{Y}$ peut être approximée en utilisant la distribution variationnelle q. En particulier, on peut obtenir l'approximation : $p(\mathbf{Y}^*, \mathbf{W}, \mathbf{H}|\mathbf{Y}) \approx p(\mathbf{Y}^*|\mathbf{W}, \mathbf{H})q(\mathbf{W})q(\mathbf{H})$. Cette distribution sera utilisée en Section 3.7.4 pour effectuer une vérification prédictive a posteriori.

En utilisant le fait que : $\mathbb{E}(\mathbf{Y}|\mathbf{N}) = \mathbf{N}\mathbb{E}(x)$, où $\mathbb{E}(x) = \kappa\psi'(\theta)$ est l'espérance de la distribution élémentaire, on peut approximer l'espérance de cette loi prédictive a posteriori par :

$$\mathbb{E}(\mathbf{Y}^*|\mathbf{Y}) = \mathbb{E}(x)\mathbb{E}(\mathbf{N}^*|\mathbf{Y}) \tag{3.42}$$

$$\approx \kappa\psi'(\theta)\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T). \tag{3.43}$$

Tableau 3.3. – Expression des distributions variationnelles pour le modèle dcPF.

Variable	Distribution q
$\mathbf{C}$	$q(\mathbf{c}_{ui}\|n_{ui}) = \text{Mult}\left(\mathbf{c}_{ui}; n_{ui},\ \tilde{\boldsymbol{\phi}}_{ui}\right)$
$\mathbf{N}$	$q(n_{ui} = n) \propto (r_{ui})^n \frac{h(y_{ui}, n\kappa)}{n!}, n \in \{1, \ldots, y_{ui}\}$
$\mathbf{W}$	$q(w_{uk}) = \text{Gamma}\left(w_{uk}; \tilde{\alpha}^W_{uk},\ \tilde{\beta}^W_{uk}\right)$
$\mathbf{H}$	$q(h_{ik}) = \text{Gamma}\left(h_{ik}; \tilde{\alpha}^H_{ik},\ \tilde{\beta}^H_{ik}\right)$

Ainsi, comme pour la PF ou la NBF, on peut utiliser le score $\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T)$ pour établir la liste personnalisée de recommandations pour chaque utilisateur.

3.6.2. Estimation des paramètres

Dans cette section, nous nous intéressons à l'optimisation de la ELBO par rapport aux paramètres Φ du modèle pour une distribution variationnelle q fixée, i.e.,

$$\max_{\Phi} \text{ELBO}(q, \Phi). \tag{3.44}$$

Cela équivaut aux deux sous-problèmes d'optimisation : $\max_{\Phi_1} \mathbb{E}_q(\log p(\mathbf{Y}|\mathbf{N}; \Phi_1))$ et $\max_{\Phi_2} \mathbb{E}_q(\log p(\mathbf{W}, \mathbf{H}; \Phi_2))$. Le premier problème traite des paramètres de la distribution élémentaire $\Phi_1 = \{\theta, \kappa\}$ qui détermine le niveau de sur-dispersion du modèle.

Paramètre naturel. Le paramètre naturel θ joue un rôle très important dans le modèle dcPF puisqu'il définit le continuum entre les deux cas limites PFbrut et PFbin (voir Section 3.5). L'optimisation de la ELBO par rapport à ce paramètre est équivalent à maximiser :

$$\mathbb{E}_q(\log p(\mathbf{Y}|\mathbf{N}; \Phi_1)) = \sum_{ui} \Big(y_{ui}\theta - \mathbb{E}_q(n_{ui})\kappa\psi(\theta) \Big) + cste, \tag{3.45}$$

où *cste* est une constante par rapport à θ. Cela revient à résoudre l'équation suivante :

$$\kappa\psi'(\theta) = \frac{\sum_{ui} y_{ui}}{\sum_{ui} \mathbb{E}_q(n_{ui})}. \tag{3.46}$$

Algorithme 4 : Algorithme CAVI pour la dcPF

Données : Matrice d'observation $\mathbf{Y}$
Résultat : Distribution variationnelle q

Initialisation aléatoire des paramètres variationnels et des paramètres Φ ;
répéter
  pour chaque *couple* (u, i) *tel que* $y_{ui} > 0$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$; $r_{ui} = \Lambda_{ui} e^{-\kappa\psi(\theta)}$;
    Calculer $q(n_{ui} = n)$ pour $n \in \{1, \dots, y_{ui}\}$;
    Calculer $\mathbb{E}_q(n_{ui})$ tel que décrit en Éq. (3.41) ;
  fin
  pour chaque *utilisateur* $u \in \{1, \dots, U\}$ **faire**
    $\tilde{\alpha}^W_{uk} = \alpha^W + \sum_i \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^W_{uk} = \beta^W_u + \sum_i \mathbb{E}_q(h_{ik})$;
  fin
  pour chaque *article* $i \in \{1, \dots, I\}$ **faire**
    $\tilde{\alpha}^H_{ik} = \alpha^H + \sum_u \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^H_{ik} = \beta^H_i + \sum_u \mathbb{E}_q(w_{uk})$;
  fin
  Paramètre naturel : résoudre Éq. (3.47) ;
  Paramètre de dispersion (pour sNB) : Éq. (3.52) ;
  Paramètres d'intensité : Éq. (3.53) ;
  Calculer ELBO(q, Φ)
jusqu'à *ELBO converge*;

Dans le cas de la loi NB translatée, $\psi(\theta) \in \mathbb{R}^2$ et ψ' correspond à son gradient. Pour les quatre exemples présentés en Section 3.4, on obtient les formes suivantes, avec $\theta = \log p$:

$$\kappa\psi'(\theta) = \begin{cases} \frac{1}{-\log(1-p)} \frac{p}{1-p} & \text{pour Log,} \\ \frac{p}{1-e^{-p}} & \text{pour ZTP,} \\ \frac{1}{1-p} & \text{pour Geo,} \\ 1 + \frac{ap}{1-p} & \text{pour sNB.} \end{cases} \tag{3.47}$$

La solution de l'Éq. (3.46) est donc connue analytiquement pour les lois élémentaires géométrique et NB translatée. Pour les lois logarithmique et ZTP, nous implémentons un algorithme de Newton-Raphson pour résoudre cette équation.

Paramètre de dispersion. Lorsque la NB translatée est choisie comme loi élémentaire, le paramètre de dispersion $\kappa_2 = a$, qui contrôle la forme de la loi NB, peut être appris. Son

optimisation n'est pas directe puisqu'un terme de la forme $\mathbb{E}_q(h(y_{ui}, n_{ui}\kappa))$ apparaît dans la ELBO et est très coûteux à optimiser.

Pour contourner ce problème, on utilise la formulation Poisson composée de la loi NB (voir Section 3.4). Ainsi, $y_{ui} - n_{ui} \sim \text{NB}(an_{ui}, p)$ est équivalent à :

$$m_{ui}|n_{ui} \sim \text{Poisson}\Big(an_{ui}(-\log(1-p))\Big) \tag{3.48}$$

$$y_{ui} - n_{ui} \sim \text{SumLog}(m_{ui}, p). \tag{3.49}$$

La loi conditionnelle de la variable m_{ui} est [Zho17] :

$$m_{ui}|y_{ui}, n_{ui} \sim \text{CRT}(y_{ui} - n_{ui}, an_{ui}), \tag{3.50}$$

avec $\mathbb{E}(m_{ui}|y_{ui}, n_{ui}) = an_{ui}a\big(\Psi(y_{ui} - n_{ui} + an_{ui}) - \Psi(an_{ui})\big)$ et si $y_{ui} = 0$ alors $m_{ui} = 0$.

Nous obtenons donc les règles de mise à jour itératives suivantes :

$$\mathbb{E}_q(m_{ui}) = \sum_{n=1}^{y_{ui}} anq(n_{ui} = n)\big(\Psi(y_{ui} - n + an) - \Psi(an)\big), \tag{3.51}$$

$$a = \frac{1}{-\log(1-e^{\theta})} \frac{\sum_{ui} \mathbb{E}_q(m_{ui})}{\sum_{ui} \mathbb{E}_q(n_{ui})}. \tag{3.52}$$

Elles peuvent être calculées au même moment que la mise à jour de la variable $\mathbf{N}$ présentée en Éq. (3.41).

Paramètres liés à l'activité et la popularité. L'optimisation de la ELBO par rapport aux paramètres Φ_2 est exactement la même que celle présentée en Section 1.4 pour la PF. Les paramètres de forme $\{\alpha^W, \alpha^H\}$ sont supposés connus et fixés. Seuls les paramètres liés à l'intensité et à la popularité $\{\boldsymbol{\beta}^W, \boldsymbol{\beta}^H\}$ sont appris. Comme pour la PF nous obtenons les mises à jour suivantes :

$$\beta_u^W = \frac{\sum_k \mathbb{E}_q(w_{uk})}{K\alpha^W};\ \beta_i^H = \frac{\sum_k \mathbb{E}_q(h_{ik})}{K\alpha^H}. \tag{3.53}$$

3.7. Résultats expérimentaux

3.7.1. Protocole expérimental

Tableau 3.4. – Structure des jeux de données TP, NIPS et Last.fm après pré-traitement.

	TP	**NIPS**	**Last.fm**
Nombre de lignes U	16 301	5 811	781
Nombre de columns I	12 118	11 463	11 172
Nombre de valeurs non nulles	1 176 086	4 033 830	402 058
% de valeurs non nulles	0.60%	6.06%	4.61%

Jeux de données. On considère les trois jeux de données suivants, dont la structure est résumée dans le Tableau 3.4.

— Le jeu de données Taste Profile (TP) [Ber+11] contient le nombre d'écoutes de chansons par des utilisateurs. Nous pré-traitons les données comme en Section 1.4.6. L'histogramme des comptes d'écoutes est affiché en Figure 3.6.

— Le jeu de données Last.fm [Cel10] contient lui aussi l'historique d'écoutes d'utilisateurs sur des chansons avec des informations temporelles additionnelles. Nous sélectionnons les données correspondant à l'année 2008 et appliquons le même pré-traitement que pour le jeu de données TP.

— Le jeu de données NIPS [Per+16] contient la représentation sous forme de sac de mots (*bag of words* en anglais) des articles scientifiques publiés entre 1987 et 2015 à la conférence NIPS (maintenant appelée NeurIPS). Nous faisons une analogie entre «des utilisateurs qui écoutent des chansons» et «des documents écrits avec des mots». Le but est donc ici de recommander des mots non employés aux auteurs des papiers.

Méthode d'évaluation. Nous utilisons la même méthode d'évaluation que celle présentée en Section 2.5 pour la NBF. Chaque jeu de données est donc divisé en un ensemble d'entraînement $\mathbf{Y}^{\text{train}}$ contenant 80% des valeurs non nulles du jeu de données original et un ensemble de test $\mathbf{Y}^{\text{test}}$ contenant les 20% restants (ces valeurs sont mises à zéro dans l'ensemble d'entraînement). Chaque algorithme est entraîné sur l'ensemble d'entraînement et fournit une liste de recommandations de 100 articles à chaque utilisateur fondée sur le score de prédiction : $[\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T)]_{ui}$ comme discuté en Section 3.6.1. La qualité de ces listes est ensuite évaluée sur l'ensemble de test à l'aide de la métrique NDCG présentée en Section 1.1.3.

Méthodes comparées. On compare la dcPF avec ses deux cas limites : la PF appliquée aux données brutes (PFbrut) ou aux données binarisées (PFbin). Au vu des expériences

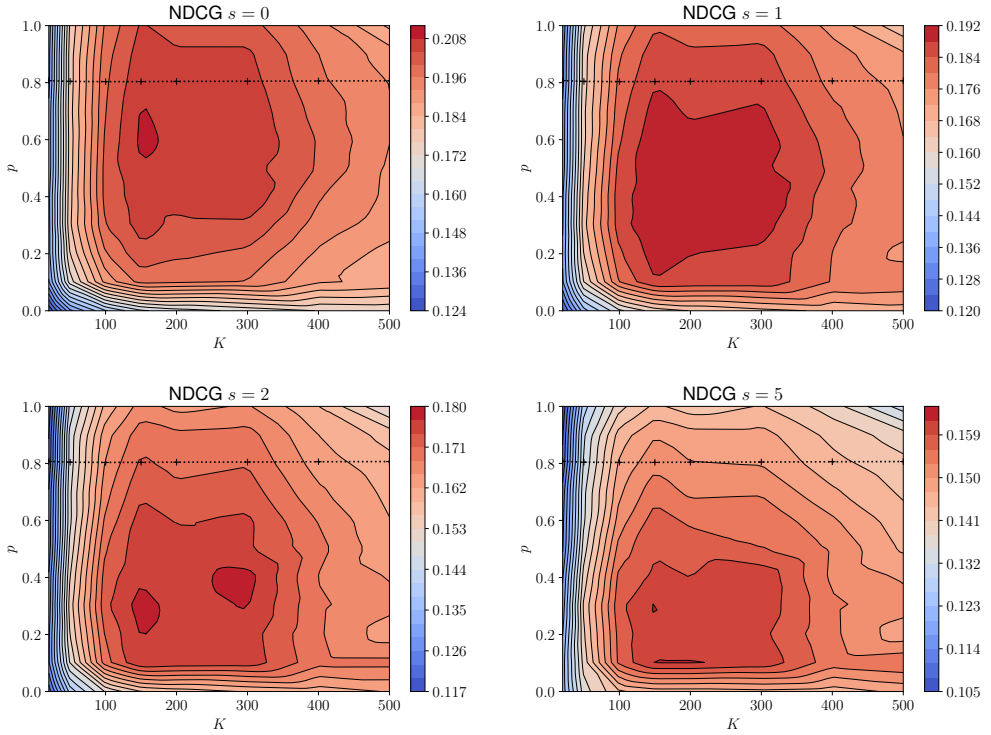


FIGURE 3.4. – Influence des paramètres K et $\theta = \log p$ sur le score de recommandation NDCG pour différents seuils $s \in \{0, 1, 2, 5\}$. Cartes de niveau pour la dcPF avec distribution élémentaire Log.

réalisées en Section 1.4.6, nous sélectionnons les hyper-paramètres $\alpha^W = \alpha^H = 0.3$. Le nombre de facteurs latents K sera testé sur une grille de recherche dans la Section 3.7.2. Pour les jeux de données NIPS et Last.fm, qui sont de taille plus petite que le jeu de données TP, nous sélectionnons $\alpha^W = \alpha^H = 0.3$ et $K = 50$. Le critère d'arrêt des différents algorithmes est fixé à $\tau = 10^{-5}$. Pour chaque expérience, les algorithmes sont exécutés cinq fois avec des initialisations aléatoires.

3.7.2. Résultats de prédiction

La Figure 3.4 illustre les résultats de recommandation moyens obtenus sur 5 exécutions de l'algorithme de dcPF avec une loi élémentaire Log, pour différentes valeurs de K et $p = \log \theta$ (ici, le paramètre naturel n'est pas mis à jour dans l'Algorithme 4). Le nombre de facteurs latents est testé sur la grille $K \in \{20, 50, 100, 150, 200, 300, 400, 500\}$, et le paramètre naturel $p = \log \theta$ varie de 0 (PFbrut) à 1 (PFbin) avec un pas de 0.1. Chacune des quatre figures

représente le NDCG pour les seuils $s \in \{0, 1, 2, 5\}$ (voir Section 1.1.3). Les croix noires correspondent à l'Algorithme 4 où le paramètre naturel est appris durant l'inférence. On constate que les meilleurs résultats se concentrent dans une région définie par $K \in [150, 300]$ et par un paramètre p qui décroit lorsque le seuil s augmente. Pour chaque figure, le NDCG est maximal pour une valeur p différente de 0 (PFbrut) et 1 (PFbin). Cela montre l'utilité du continuum proposé par la dcPF entre PFbin et PFbrut pour améliorer le NDCG. La courbe noire nous indique que le paramètre naturel appris ne dépend pas du nombre de facteurs latents. D'après ces observations, nous choisissons donc de fixer le nombre de facteurs latents à $K = 150$ pour les modèles dcPF.

Nous nous intéressons maintenant aux résultats obtenus par les différentes versions de la dcPF (Log, ZTP, sGeo, sNB) qui sont rapportés dans le Tableau 3.5. Une observation générale est que la dcPF donne de meilleurs résultats que les deux modèles de références PFbrut et PFbin pour chacune des métriques et pour chaque distribution élémentaire (à l'exception de la loi ZTP pour NDCG0 et NDCG1). PFbin donne de meilleurs résultats que PFbrut jusqu'au seuil $s = 2$. Cela confirme l'utilité de l'étape de binarisation lors de l'application de PF, mais seulement jusqu'à un certain seuil s. En effet, lorsque le seuil s augmente, PFbin perd de son pouvoir prédictif et l'écart avec dcPF augmente. Cela paraît sensé puisque PFbin n'exploite pas la valeur des données non nulles du jeu de données original.

Pour les deux autres jeux de données NIPS et Last.fm, le Tableau 3.6 montre que dcPF bat les méthodes de référence pour toutes les distributions élémentaires. On peut noter que pour le jeu de données NIPS, qui est un cas légèrement différent de la recommandation musicale, PFbrut est performant et fait mieux que PFbin dès que le seuil s est plus grand que 1. D'après les Tableaux 3.5 et 3.6, nous pouvons conclure que la loi NB translatée offre un bon compromis parmi les distributions élémentaires.

Tableau 3.5. – Performance des modèles dcPF et PF sur le jeu de données TP. Le nombre de facteurs latents est fixé à $K = 150$ pour PFbin et tous les modèles de dcPF, et est fixé à $K = 1000$ pour PFbrut. En gras : les deux meilleurs scores NDCG. Entre parenthèses : l'écart-type sur les 5 exécutions effectuées.

Modèle	p	κ	**NDCG0**	**NDCG1**	**NDCG2**	**NDCG5**
Log	0.8	1	**0.208** ($\mathbf{3 \cdot 10^{-3}}$)	0.188 ($3 \cdot 10^{-3}$)	**0.172** ($\mathbf{2 \cdot 10^{-3}}$)	0.152 ($2 \cdot 10^{-3}$)
ZTP	1.9	1	0.198 ($3 \cdot 10^{-3}$)	0.183 ($2 \cdot 10^{-3}$)	**0.172** ($\mathbf{2 \cdot 10^{-3}}$)	**0.160** ($\mathbf{2 \cdot 10^{-3}}$)
sGeo	0.6	1	0.207 ($3 \cdot 10^{-3}$)	**0.189** ($\mathbf{4 \cdot 10^{-3}}$)	**0.173** ($\mathbf{3 \cdot 10^{-3}}$)	**0.156** ($\mathbf{4 \cdot 10^{-3}}$)
sNB	0.9	$(1, 0.2)^T$	**0.208** ($\mathbf{2 \cdot 10^{-3}}$)	**0.189** ($\mathbf{2 \cdot 10^{-3}}$)	**0.172** ($\mathbf{2 \cdot 10^{-3}}$)	0.151 ($2 \cdot 10^{-3}$)
PFbin	.	.	0.205 ($2 \cdot 10^{-3}$)	0.184 ($1 \cdot 10^{-3}$)	0.166 ($1 \cdot 10^{-3}$)	0.145 ($6 \cdot 10^{-4}$)
PFbrut	.	.	0.173 ($5 \cdot 10^{-3}$)	0.172 ($5 \cdot 10^{-3}$)	0.166 ($5 \cdot 10^{-3}$)	0.153 ($5 \cdot 10^{-3}$)

Tableau 3.6. – Performance des modèles dcPF et PF avec les jeux de données NIPS et Last.fm, pour $K = 50$.

	NIPS		**Last.fm**	
Modèle	**NDCG0**	**NDCG1**	**NDCG0**	**NDCG1**
Log	**0.394**	**0.430**	**0.142**	**0.129**
ZTP	0.381	0.422	0.122	0.113
sGeo	0.390	0.429	0.139	0.128
sNB	**0.396**	**0.431**	**0.143**	**0.130**
PFbrut	0.358	0.405	0.091	0.088
PFbin	0.378	0.392	0.122	0.108

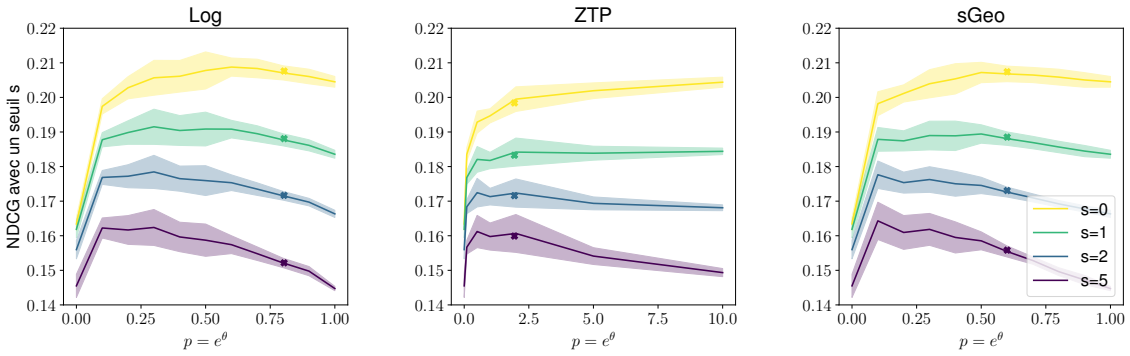


FIGURE 3.5. – Influence du paramètre naturel $\theta = \log p$ pour $K = 150$ sur le score de recommandation NDCG pour différents seuils $s \in \{0, 1, 2, 5\}$.

3.7.3. Influence du paramètre naturel

Comme expliqué en Section 3.5, l'estimation du paramètre naturel nous indique le degré d'information exploité par le modèle dcPF. Le tableau 3.5 montre en effet que la dcPF offre un compromis intéressant entre PFbrut et PFbin, puisque le paramètre estimé θ se situe entre les deux cas limites θ^{brut} et θ^{bin}. Pour évaluer la qualité de la procédure d'estimation de θ pour les lois Log, ZTP et sGeo décrites en Section 3.4, nous comparons dans la Figure 3.5 les résultats de recommandation de deux versions de l'Algorithme 4 : l'une où le paramètre naturel est appris durant l'inférence, et l'autre où le paramètre naturel est supposé fixé durant l'inférence et est recherché sur une grille prédéfinie. Pour les lois Log et sGeo, on recherche le paramètre p entre les valeurs 0 (PFbrut) et 1 (PFbin) avec un pas de 0.1. Pour la loi ZTP, on recherche p parmi l'ensemble de valeurs $\{0, 0.1, 0.5, 1, 2, 10, 100\}$. Il apparaît que l'algorithme VBEM sur-estime légèrement la valeur optimale (au sens du critère NDCG) du paramètre naturel pour chacune des trois lois. Cela peut s'expliquer notamment par la présence de valeurs aberrantes dans les données. Néanmoins, la procédure d'estimation proposée reste une robuste par rapport au score NDCG.

La Figure 3.5 illustre les performances de la dcPF avec les trois lois élémentaires Log, ZTP et sGeo. Le nombre de facteurs latents fixé à $K = 150$, et le paramètre p est recherché entre les valeurs 0 (PFbrut) et 1 (PFbin) avec un pas de 0.1 pour les lois Log et sGeo, et parmi l'ensemble $\{0, 0.1, 0.5, 1, 2, 10, 100\}$ pour la loi ZTP. Plus précisément, on choisit le paramètre $\theta = \log p$ qui maximise le critère NDCG5 pour un ensemble de valeurs prédéfinies. Pour les lois Log et sGeo, p est recherché entre les valeurs 0 et 1 avec un pas de 0.1. Pour la loi ZTP, p est recherché parmi l'ensemble de valeurs $\{0, 0.1, 0.5, 1, 2, 10, 100, +\infty\}$.

Tableau 3.7. – Corrélation entre la sur-dispersion des données et le paramètre naturel inféré. Moyenne, variance et ratio variance/moyenne des valeurs non nulles de chaque jeu de données. Paramètres appris pour chaque modèle et chaque jeu de données.

Jeu de données	**NIPS**	**TP**	**Last.fm**
Moyenne	2.7	2.7	3.9
Variance	20.9	25.9	65.7
Ratio var./moy.	7.6	9.8	17.0
Log - p	0.74	0.80	0.90
ZTP - p	1.40	1.95	2.35
sGeo - p	0.51	0.60	0.69

Le Tableau 3.7 illustre la corrélation entre le paramètre naturel inféré et la sur-dispersion des données. Pour cela, on compare le paramètre naturel appris pour chaque modèle et pour les trois jeux de données présentés. Chaque jeu de données a un niveau de sur-dispersion différent. On constate alors que plus la sur-dispersion est élevée plus le paramètre naturel se rapproche de θ^{brut}. Le modèle dcPF applique une distorsion des données plus importante lorsque les données sont fortement dispersées.

3.7.4. Vérification prédictive a posteriori

Dans cette section, nous nous intéressons à la vérification predictive a posteriori (PPC, de l'anglais *posterior predictive check*) de la distribution des comptes d'écoutes dans le jeu de données TP (voir Figure 3.6). Un PPC consiste à simuler un nouvel ensemble de données $\mathbf{Y}^{\text{PPC}}$ à partir d'un modèle génératif et de ses paramètres inférés, puis de comparer, selon un critère d'intérêt, les données simulées avec les données réelles. Ici, nous générons de nouvelles données d'après le modèle dcPF décrit en Section 3.3.1 à partir des variables latentes $\mathbf{W}$ et $\mathbf{H}$ et des paramètres inférés κ et θ (voir Section 3.7.2). Ensuite, nous comparons l'histogramme des valeurs de $\mathbf{Y}^{\text{train}}$ et $\mathbf{Y}^{\text{PPC}}$.

Le PPC des deux cas limites (PFbrut et PFbin) est très instructif. PFbrut essaie de s'ajuster à la queue lourde des données mais, ce faisant, détruit la représentation des valeurs nulles (1.02% de valeurs non nulles par rapport à 0.48% dans le jeu de données réel). Cela peut en partie expliquer les performances décevantes de PFbrut pour le score NDCG0. Au contraire, PFbin s'ajuste mieux à la parcimonie des données mais ne permet pas de décrire les grandes valeurs (phénomène attendu puisque PFbin n'avait pas accès à ces informations

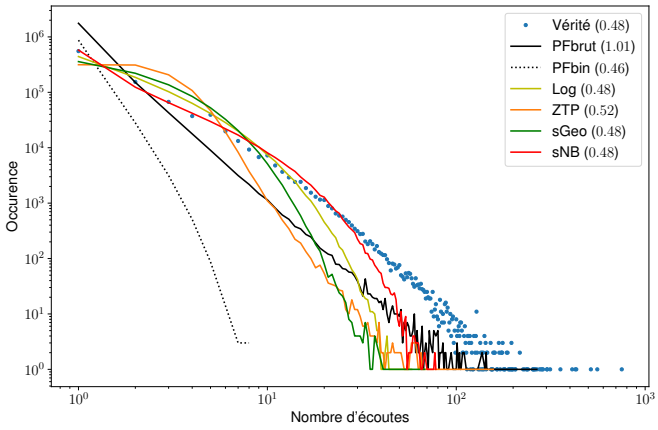


FIGURE 3.6. – PPC de la distribution des valeurs non nulles dans le jeu de données TP. Les points bleus représentent l'histogramme empirique des valeurs non nulles de l'ensemble d'entraînement. Les courbes colorées représentent les histogrammes simulés obtenus à partir des différents modèles déduits de dcPF ou de PF. Les pourcentages des valeurs non nulles sont écrits entre parenthèses.

lors de la phase d'apprentissage). Dans les deux cas, PF a du mal à équilibrer correctement l'influence des grandes valeurs par rapport aux petites. En comparaison, dcPF propose une pondération plus douce entre les grandes et petites valeurs. dcPF respecte à la fois la parcimonie et la queue lourde des données pour les quatre distributions élémentaires. ZTP semble surestimer l'influence des comptes moyens (de 1 à 5), alors que la NB translatée s'ajuste le mieux à l'histogramme. Nous observons que quel que soit le modèle, il reste difficile d'expliquer les très grandes valeurs (> 100), mais nous pouvons considérer qu'après un certain seuil, les comptes ne contiennent plus d'informations utiles.

3.8. Discussion

Dans ce chapitre, nous avons étudié la dcPF qui est une instance de la cPF proposée par [BE16]. La dcPF est spécialement conçue pour traiter les données de comptage sur-dispersées collectées en CF. En particulier, la dcPF préserve la propriété de passage à l'échelle de la PF, en assurant que les zéros de la variable latente $\mathbf{N}$ soient les mêmes que ceux des données observées $\mathbf{Y}$. Contrairement à la NBF, les rôles des paramètres régissant la dcPF sont parfaitement définis. En effet, les paramètres additionnels θ et κ de l'EDM utilisée se focalisent uniquement sur la sur-dispersion des données et n'influent pas sur leur

niveau de parcimonie.

Nous avons montré que la dcPF offre un continuum entre deux versions de la PF : l'une, appliquée sur les données brutes (PFbrut) et l'autre, appliquée aux données binarisées (PFbin). La variable latente additionnelle $\mathbf{N}$ de la dcPF peut être interprétée comme une distorsion des données brutes, permettant de les rendre plus «factorisable» au sens de la PF. Cette distorsion est plus souple qu'une étape de binarisation et est adaptative. De plus, la dcPF dispose de deux cas limites au comportement stable (PFbrut et PFbin), ce qui lui évite de rencontrer des cas pathologiques, comme cela peut être le cas pour la NBF (lorsque le paramètre α est petit). Le choix du paramètre θ n'est donc pas critique et permet de donner plus ou moins d'importance aux données brutes. Nous avons proposé d'estimer ce paramètre au sens du maximum de vraisemblance, mais d'autres méthodes pourraient être envisagées, en utilisant un ensemble de validation par exemple [Lia+18]. Nous détaillons dans ce qui suit quelques perspectives de ce travail.

Individualisation des paramètres. Une première amélioration possible de la dcPF consisterait à individualiser le rôle des paramètres θ et κ. En l'état actuel, ces paramètres régissent la sur-dispersion des données pour la totalité de la matrice $\mathbf{Y}$. L'estimation de ces paramètres peut donc être biaisée par la présence de données aberrantes. Cela semble être le cas dans les problèmes de CF où de très grands comptes d'écoutes peuvent être observés. L'utilisation de paramètres de la forme θ_u et κ_u pour $u \in \{1, \dots, U\}$ pourrait permettre de modéliser plusieurs types de comportements au sein des utilisateurs. Certains utilisateurs écoutent des contenus variés à des doses raisonnables. Leurs données sont alors peu dispersées et peuvent être traitées en l'état par la PF. Au contraire, d'autres utilisateurs écoutent un petit ensemble de chansons un grand nombre de fois (par exemple un bar qui utilise la même playlist chaque jour). Leurs données sont alors fortement sur-dispersées et nécessitent d'être binarisées avant d'appliquer la PF.

Paramètre de dispersion. Dans ce chapitre, nous avons étudié l'influence du paramètre naturel θ sur la dcPF lorsque le paramètre de dispersion κ est fixé. C'est notamment le cas par exemple pour les distributions de Stirling qui imposent $\kappa = 1$. Une perspective intéressante est d'étudier le rôle du paramètre de dispersion κ sur la dcPF. Cela permettrait de mieux comprendre l'influence jointe des deux paramètres θ et κ et dobtenir un réglage plus fin du modèle.

Nous avons proposé une méthode d'estimation du paramètre de dispersion κ seulement dans le cas de la distribution élémentaire sNB où nous utilisons une astuce d'augmentation

de modèle. Comme nous l'avons signalé, l'estimation de ce paramètre au sens du maximum de vraisemblance dans un cadre général est coûteuse. Une perspective possible est donc de chercher une nouvelle méthode d'apprentissage de ce paramètre qui soit raisonnable en temps de calcul.

Une dernière perspective de travail à propos du paramètre de dispersion est d'utiliser des lois de la forme : $x_{l,ui} \sim \mathrm{ED}(\theta, \kappa_l),\ \forall l \in \{1, \dots, n_{ui}\}$ dans le modèle génératif. Cela signifierait que la distribution du nombre d'écoutes pour chaque session dépendrait de l'indice de cette session. Par exemple, on pourrait imaginer que plus le nombre de sessions d'écoutes d'un utilisateur pour une chanson est élevé, plus il va l'écouter la chanson lors ces ces sessions. En utilisant la propriété de superposition des EDM, le modèle génératif deviendrait alors :

$$y_{ui}|n_{ui} \sim \mathrm{ED}\left(\theta, \sum_{l=1}^{n_{ui}} \kappa_l\right), \tag{3.54}$$

où le terme non-linéaire $\phi(n) = \sum_{l=1}^{n} \kappa_l$ remplacerait le terme $n\kappa$ de la dcPF. L'introduction d'une non-linéarité a été proposée dans [BE17] où la fonction ϕ est prédéfinie et n'est pas nécessairement une fonction croissante (ce qui rend l'interprétation délicate). Les paramètres $\kappa_l > 0$ pourraient donc être estimés afin de mieux modéliser la distribution des valeurs surdispersées (et notamment les très grandes valeurs).

Modèles de mélange fondés sur la PF. La dcPF est un modèle hiérarchique introduisant une variable latente $\mathbf{N}$ (voir Section 3.3.3). Cette variable est liée aux données $\mathbf{Y}$ par la distribution $p(y_{ui}|n_{ui})$ pour chaque couple (u, i). Dans le cadre de la dcPF, cette distribution correspond à la convolution de n_{ui} distributions élémentaires. Cependant, la dcPF peut s'inscrire dans un cadre encore plus général où la distribution $p(y_{ui}|n_{ui})$ correspondrait à un modèle de mélange. Ainsi, le modèle pourrait être formalisé sous la forme :

$$n_{ui} \sim \mathrm{Poisson}([\mathbf{W}\mathbf{H}^T]_{ui}) \tag{3.55}$$

$$y_{ui} \sim \delta_0,\ \text{si } n_{ui} = 0, \tag{3.56}$$

$$y_{ui} \sim f(\mu_{n_{ui}}),\ \text{sinon.} \tag{3.57}$$

f est une distribution dont le support exclut 0 (cela permet de préserver la propriété de passage à l'échelle de la PF). La variable $n_{ui} \in \mathbb{N}$ ne correspond plus à un nombre de sessions d'écoutes, mais à l'indice d'une classe définissant le comportement de l'utilisateur u vis-à-vis de la chanson i. Les paramètres μ_n contrôlent la distribution f du nombre d'écoutes lié à la

classe n. Nous avons donc un nombre infini de paramètres à estimer puisque $n_{ui} \in \mathbb{N}$. Pour contourner ce problème on peut approximer le modèle en supposant que $\mu_n = \mu_T$ dès que $n \geq T$.

Ce modèle est plus général que la dcPF puisqu'il peut notamment modéliser des données $\mathbf{Y}$ qui sont bornées [4]. Cependant, on peut s'interroger sur la pertinence d'imposer un a priori poissonnien à l'indice des classes n_{ui} dans ce type de modèle. Ces réflexions correspondent à un travail en cours.

4. Le support d'une distribution Poisson composée ne peut pas être borné.

Chapitre 4.

NMF pour données ordinales

Le contenu de ce chapitre sera soumis prochainement pour publication.

Contents

4.1. Introduction

Dans le Chapitre 1, nous avons montré qu'une étape de binarisation des données entraînait une perte d'information. Pour pallier ce problème, nous avons proposé dans les Chapitres 2 et 3 de modéliser directement les données de comptage sur-dispersées par le biais de modèles

généralisant la PF, donnant lieu à la NBF et à la dcPF. Cependant, factoriser les données brutes est un problème difficile et sensible aux très grandes valeurs présentes dans les données. Dans ce chapitre, nous considérons une autre approche utilisant une quantification des données plutôt qu'une binarisation (voir Tableau 4.2 pour un exemple de quantification). Ce pré-traitement est moins radical que la binarisation et permet de conserver plus d'information comprise dans les données.

Les données quantifiées font parties de la classe des données *ordinales* [Ste+46]. Les données ordinales sont des données nominales possédant une relation d'ordre naturelle (par exemple : froid $\prec$ tiède $\prec$ chaud). Il est important de noter que pour ce type de données, il n'existe pas de notion de distance entre les différentes classes rencontrées. Cela implique notamment que la statistique de la moyenne n'est pas adaptée à ces données, contrairement à la médiane [Ste+46]. Sans perte de généralité, nous travaillons dans la suite du chapitre avec des données pré-traitées appartenant à $\{0, \dots, V\}$ [1], c'est-à-dire que nous utilisons une quantification sur $V + 1$ niveaux.

Plusieurs méthodes de régression ont été développées pour les données ordinales, appelées méthodes de régression ordinale. Nous invitons le lecteur intéressé à se référer à [Gut+15] pour une revue de la littérature, les auteurs y proposent notamment une taxonomie claire des différents modèles existants. Il existe deux façons naïves de traiter les données ordinales. La première est de voir ces données comme de simples données nominales afin de leur appliquer des méthodes de classification. La relation d'ordre qui lie les différentes classes est alors ignorée. La seconde est de considérer ces données comme des valeurs réelles afin de leur appliquer des modèles de régression. En faisant cela, on crée alors artificiellement une distance entre les différentes classes. Ces deux méthodes naïves ne considèrent pas entièrement les spécificités des données ordinales, puisqu'elles enlèvent ou ajoutent de l'information aux données.

Dans ce chapitre, nous nous intéressons aux modèles à seuillage latent (*threshold models* en anglais) [WD67 ; McC80]. Cette approche populaire suppose que les données ordinales résultent de la quantification de variables latentes continues par rapport à une suite croissante de seuils. Le but de ces modèles est alors d'entraîner un modèle prédictif sur la variable latente, et d'apprendre les seuils de la quantification. Les modèles à seuillage latents peuvent ainsi être vus comme une extension des modèles de régression naïfs, où les distances entre les différentes classes sont apprises par le biais des seuils de quantification. Plus particuliè-

1. Nous choisissons de commencer la numérotation des catégories par 0, car cette catégorie correspondra aux données de comptage nulles. Dans la littérature, il est plus courant que cette numérotation commence à 1.

rement, nous nous focalisons sur la famille des *cumulative link models* (en anglais) [AK11] qui proposent d'estimer la fonction de répartition des données ordinales.

Dans ce chapitre, nous développons un modèle de NMF probabiliste pour les données ordinales (pour lequel nous utilisons l'acronyme de OrdNMF pour *NMF pour données ordinales*). La OrdNMF est un modèle à seuillage latent où la variable latente a une structure NMF. Autrement dit, cela revient à définir l'approximation $\mathbf{Y} \approx \mathbf{W}\mathbf{H}^T$ pour $\mathbf{Y}$ une matrice ordinale, $\mathbf{W}$ et $\mathbf{H}$ des matrices non-négatives. La OrdNMF peut être vue comme une nouvelle extension de la PFbin où le pré-traitement appliqué aux données est plus souple. Contrairement à la NBF ou à la dcPF qui travaillent sur les données brutes, la OrdNMF permet de prendre en compte toutes les échelles de valeurs (notamment les très grandes valeurs présentes dans les données). De plus, la OrdNMF peut traiter le cas de données implicites continues, puisqu'elle s'intéresse à une quantification de celles-ci.

Les données explicites rencontrées dans les systèmes de recommandation sont aussi des données ordinales. Elles correspondent le plus souvent à des notes d'utilisateurs appartenant à une échelle du type : mauvais $\prec$ moyen $\prec$ bon $\prec$ très bon. Notre modèle, conçu pour des données implicites pré-traitées, permet aussi de s'adapter aux données explicites. Nous développons deux versions de l'algorithme OrdNMF pour ce type de données, l'une fondée sur l'hypothèse MAR et l'autre sur l'hypothèse MNAR (voir Section 1.2.1) [LR14 ; Sea+13].

Les contributions de ce chapitre sont les suivantes.

— Nous proposons un modèle de NMF pour données ordinales (OrdNMF) fondé sur la présence d'un bruit multiplicatif. En particulier, nous étudions une instance de ce modèle où le bruit est supposé être tiré d'une loi IG. Nous montrons notamment que ce modèle est une extension du modèle Bernoulli-Poisson (BePo) proposé dans [AGZ15].

— Nous utilisons une astuce d'augmentation de modèle afin de mettre en place un algorithme variationnel très simple, à la fois pour les règles de mise à jour des facteurs latents $\mathbf{W}$ et $\mathbf{H}$, et pour celles des seuils $\mathbf{b}$.

— Nous comparons les performances de la OrdNMF avec celles de la dcPF sur des tâches de recommandation sur le jeu de données Taste Profile. De plus, nous nous intéressons au PPC de la distribution des différentes classes et montrons la capacité de la OrdNMF à prendre en compte les très grandes valeurs.

— Nous adaptons la OrdNMF au traitement des données explicites. Nous développons notamment deux algorithmes fondés sur les hypothèses MAR et MNAR. Là aussi, nous effectuons un PPC pour montrer la capacité de la OrdNMF à s'adapter à ce type de données.

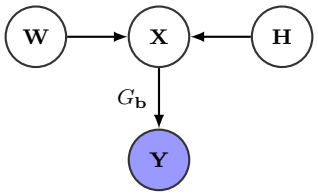


FIGURE 4.1. – Représentation graphique de la factorisation de matrices pour données ordinales. Une variable latente $\mathbf{X}$ est introduite pour faire le lien entre la MF $\mathbf{WH}^T$ et les données ordinales $\mathbf{Y}$.

Le reste du chapitre est organisé comme suit. Dans la Section 4.2, nous présentons quelques pré-requis sur les modèles à seuillage latents pour données ordinales. De plus, nous présentons le modèle de factorisation BePo (BePoF) qui étend la PF au traitement de données binaires. Dans la Section 4.3, nous présentons le modèle OrdNMF et en étudions une instance particulière. Dans la Section 4.4, nous développons un algorithme de VI fondé sur une augmentation de modèle similaire à la BePoF. Dans la Section 4.5, nous testons cet algorithme sur le jeu de données Taste Profile comme dans le Chapitre 3. Dans la Section 4.6, nous adaptons le modèle proposé au traitement de données explicites et testons les algorithmes développés sur le jeu de données MovieLens. Enfin, dans la Section 4.7, nous discutons des limites et perspectives de la OrdNMF.

4.2. Pré-requis

4.2.1. Factorisation de matrices pour données ordinales

Dans cette section, nous présentons la famille des *cumulative link models* qui ont d'abord été proposés pour des modèles de régression [AK11]. Nous nous intéressons ici au problème de MF représenté par l'approximation $\mathbf{Y} \approx \mathbf{WH}^T$, où $\mathbf{Y} \in \{0, \dots, T\}^{U \times I}$ est une matrice ordinale, et où $\mathbf{W} \in \mathbb{R}^{U \times K}$ et $\mathbf{H} \in \mathbb{R}^{I \times K}$ sont des variables latentes. Chaque coefficient de la matrice $\mathbf{Y}$ correspond à l'étiquette d'une classe (ici, ces classes sont numérotées de 0 à V).

Comme nous l'avons évoqué en introduction, l'idée des modèles à seuillage latent est d'introduire une variable latente continue $x_{ui} \in \mathbb{R}$ permettant de faire le lien avec les données ordinales y_{ui}. Pour cela, on introduit une suite croissante de seuils $b_{-1} = -\infty < b_0 < \cdots < b_{V-1} < b_V = +\infty$, notée $\mathbf{b}$, permettant de quantifier la variable latente x_{ui}. On définit la

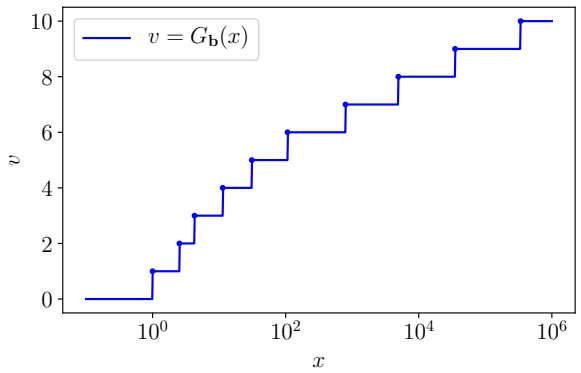


FIGURE 4.2. – Fonction de quantification $G_{\mathbf{b}}$, où les seuils $\mathbf{b}$ ont été inférés avec le jeu de données Taste Profile quantifié.

fonction de quantification, illustrée en Figure 4.2, comme :

$$\begin{array}{rccl} G_{\mathbf{b}}: & \mathbb{R} & \to & \{0,\dots,V\} \\ & x & \mapsto & v \text{ tel que } x \in [b_{v-1}, b_v). \end{array} \tag{4.1}$$

Par conséquent, les données ordinales résultent de la quantification de la variable x_{ui} par la fonction en escalier $G_{\mathbf{b}}$, i.e., $y_{ui} = G_{\mathbf{b}}(x_{ui})$. La variable latente x_{ui} correspond à la variable $\lambda_{ui} = [\mathbf{W}\mathbf{H}^T]_{ui} \in \mathbb{R}$ perturbée par un bruit additif ε_{ui}, dont la fonction de répartition est notée $F_\varepsilon : \mathbb{R} \to [0,1]$. On obtient donc le modèle génératif (illustré en Figure 4.1) suivant :

$$x_{ui} = \lambda_{ui} + \varepsilon_{ui}, \tag{4.2}$$

$$y_{ui} = G_{\mathbf{b}}(x_{ui}). \tag{4.3}$$

Le but de ces modèles de MF pour données ordinales est d'inférer conjointement les variables $\mathbf{W}$ et $\mathbf{H}$, ainsi que la suite de seuils $\mathbf{b}$ permettant de faire le lien entre $\mathbb{R}$ et les données ordinales.

Fonction de répartition. La fonction de répartition associée à la variable aléatoire y_{ui} peut être calculée de la sorte :

$$\begin{aligned}
\mathbb{P}[y_{ui} \leq v|\lambda_{ui}] &= \mathbb{P}[G_{\mathbf{b}}(x_{ui}) \leq v|\lambda_{ui}] && (4.4)\\
&= \mathbb{P}[\lambda_{ui} + \varepsilon_{ui} < b_v] && (4.5)\\
&= \mathbb{P}[\varepsilon_{ui} < b_v - \lambda_{ui}] && (4.6)\\
&= F_\varepsilon\left(b_v - \lambda_{ui}\right). && (4.7)
\end{aligned}$$

On a bien que la fonction $v \mapsto \mathbb{P}[y_{ui} \leq v|\lambda_{ui}]$ est croissante puisque la suite des seuils est elle-même croissante. De plus, la fonction de masse associée aux données ordinales peut s'écrire sous la forme :

$$\begin{aligned}
\mathbb{P}[y_{ui} = v|\lambda_{ui}] &= \mathbb{P}[y_{ui} \leq v|\lambda_{ui}] - \mathbb{P}[y_{ui} \leq v-1|\lambda_{ui}] && (4.8)\\
&= F_\varepsilon\left(b_v - \lambda_{ui}\right) - F_\varepsilon\left(b_{v-1} - \lambda_{ui}\right). && (4.9)
\end{aligned}$$

Quelques exemples. Si la fonction de répartition est strictement croissante, on peut réécrire l'Éq. (4.7) sous la forme :

$$F_\varepsilon^{-1}(\mathbb{P}[y_{ui} \leq v|\lambda_{ui}]) = b_v - \lambda_{ui}. \tag{4.10}$$

D'où le nom de *cumulative link models*, puisque le modèle de factorisation est lié à la fonction de répartition des données ordinales par une fonction de lien $F_\varepsilon^{-1} : [0,1] \to \mathbb{R}$. Plusieurs choix de bruits (ou de fonctions de lien F_ε^{-1}) ont été proposés dans la littérature. Nous présentons quelques uns de ces choix dans ce qui suit.

— Fonction logit. L'utilisation de la fonction logit a d'abord été proposée dans [WD67], puis le modèle a été popularisé et appelé *proportional odds model* par [McC80]. Le modèle peut alors être réécrit sous la forme :

$$\operatorname{logit} \mathbb{P}[y_{ui} \leq v|\lambda_{ui}] = \log \frac{\mathbb{P}[y_{ui} \leq v|\lambda_{ui}]}{\mathbb{P}[y_{ui} > v|\lambda_{ui}]} = b_v - \lambda_{ui} \tag{4.11}$$

— Fonction probit. Un choix commun pour le bruit additif est de supposer que $\varepsilon_{ui} \sim \mathcal{N}(0, \sigma^2)$ [CG05 ; PTW12 ; HHG14] et la fonction de lien F_ε^{-1} est la fonction probit. L'inférence du modèle peut être menée avec un algorithme EM introduisant la variable latente x_{ui}.

— D'autres choix comme les fonctions log-log ou cauchit ont aussi été étudiées dans la

littérature [AK11]. La revue de littérature [AK97] récapitule notamment quelques uns de ces choix.

4.2.2. Bernoulli-Poisson factorisation (BePoF)

Dans cette section, et sans lien avec la section précédente, nous nous intéressons à une variante de la PF pour les données binaires. Cette variante fait apparaître une augmentation de modèle que nous réutilisons dans la suite du chapitre.

La distribution Poisson peut être facilement augmentée pour s'adapter à des données binaires $y_{ui} \in \{0, 1\}$. Pour cela, il suffit d'introduire un seuillage pour binariser les données [AGZ15]. Le modèle génératif hiérarchique est alors donné par :

$$n_{ui} \sim \text{Poisson}([\mathbf{W}\mathbf{H}^T]_{ui}), \tag{4.12}$$

$$y_{ui} = \mathbb{1}[n_{ui} > 0], \tag{4.13}$$

où $n_{ui} \in \mathbb{N}$ est une variable latente. On peut marginaliser cette variable en notant que $\mathbb{P}[y_{ui} = 0] = e^{-[\mathbf{W}\mathbf{H}^T]_{ui}}$. On obtient donc :

$$y_{ui} \sim \text{Bern}(1 - e^{-[\mathbf{W}\mathbf{H}^T]_{ui}}). \tag{4.14}$$

Ce modèle fait le pont entre la loi Bernoulli et la loi Poisson. Nous utiliserons l'acronyme BePoF pour Bernoulli-Poisson factorisation. La distribution conditionnelle de la variable latente n_{ui} est donnée par :

$$n_{ui}|y_{ui} \sim \begin{cases} \delta_0 & \text{si } y_{ui} = 0, \\ \text{ZTP}([\mathbf{W}\mathbf{H}^T]_{ui}) & \text{si } y_{ui} = 1. \end{cases} \tag{4.15}$$

Ainsi, on peut mettre en place des algorithmes de Gibbs ou de VI pour l'estimation de la loi à posteriori du modèle augmenté $p(\mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}|\mathbf{Y})$, où $\mathbf{N}$ est la matrice contenant les variables latentes n_{ui} et $\mathbf{C}$ est la variable latente telle qu'introduite en Section 1.4.2.

Remarque. Le modèle génératif marginalisé présenté en Éq. (4.14) est de la forme $y_{ui} \sim \text{Bern}(f([\mathbf{W}\mathbf{H}^T]_{ui}))$ où la fonction $f : \mathbb{R}$ (ou $\mathbb{R}_+$) $\to [0, 1]$ fait le lien entre le domaine de $[\mathbf{W}\mathbf{H}^T]_{ui}$ et le paramètre de probabilité de la loi Bernoulli. Lorsque $[\mathbf{W}\mathbf{H}^T]_{ui} \in \mathbb{R}$, la fonction f peut être la réciproque de la fonction probit [CM07] ou de la fonction logit par exemple. Ces deux cas sont alors des cas particuliers du modèle présenté en Section 4.2.1 où

$V = 1$. Des modèles de MF paramétrés par la moyenne ont aussi été développés [LFF18], i.e., $f = \mathrm{Id}$. Ils nécessitent cependant des contraintes supplémentaires sur $\mathbf{W}$ et $\mathbf{H}$ afin de s'assurer que $[\mathbf{W}\mathbf{H}^T]_{ui} \in [0, 1]$.

Extension aux données bornées. La PF peut être aussi étendue aux données bornées $y_{ui} \in \{0, \dots, V\}$ en utilisant la même fonction de lien que précédemment :

$$y_{ui} \sim \mathrm{Bin}(V, 1 - e^{-[\mathbf{W}\mathbf{H}^T]_{ui}}), \tag{4.16}$$

où Bin correspond à la loi binomiale. Ce modèle peut aussi être augmenté en utilisant le fait qu'une loi binomiale de paramètre V correspond à la somme de V lois Bernoulli.

4.3. NMF bayésienne pour données ordinales

Dans cette section, nous introduisons la OrdNMF, qui est un modèle de NMF pour données ordinales. Contrairement à la Section 4.2.1, nous imposons ici que les matrices $\mathbf{W}$ et $\mathbf{H}$ soient non-négatives, ainsi nous avons $[\mathbf{W}\mathbf{H}^T]_{ui} \in \mathbb{R}_+$ et non plus $[\mathbf{W}\mathbf{H}^T]_{ui} \in \mathbb{R}$. Comme dans les Chapitres 2 et 3, nous imposons des a priori gamma sur chacun des coefficients des matrices $\mathbf{W}$ et $\mathbf{H}$:

$$w_{uk} \sim \mathrm{Gamma}(\alpha^W, \beta_u^W), \;\; h_{ik} \sim \mathrm{Gamma}(\alpha^H, \beta_i^H). \tag{4.17}$$

On note $\lambda_{uik} = w_{uk} h_{ik}$ et $\lambda_{ui} = \sum_k \lambda_{uik} = [\mathbf{W}\mathbf{H}^T]_{ui}$.

4.3.1. Quantification de la droite des réels positifs

À l'instar des méthodes de MF pour données ordinales, notre modèle propose de quantifier la droite des réels positifs $\mathbb{R}_+$. Pour cela, on pose la suite croissante des seuils $\mathbf{b}$ donnée par $b_{-1} = 0 < b_0 < \cdots < b_{V-1} < b_V = +\infty$ (les seuils sont ici non-négatifs). De plus, on définit la fonction de quantification $G_{\mathbf{b}} : \mathbb{R}_+ \to \{0, \dots, V\}$ comme en Éq. (4.1) (le domaine de définition est ici donné par $\mathbb{R}_+$).

Contrairement à la Section 4.2.1, nous allons ici supposer la présence d'un bruit multiplicatif non-négatif. Ce type de bruit semble plus adapté aux données sur-dispersées qu'un bruit additif. Soit ε_{ui} une variable aléatoire réelle positive, de fonction de répartition F_ε,

nous proposons le modèle génératif suivant :

$$x_{ui} = \lambda_{ui} \cdot \varepsilon_{ui}, \tag{4.18}$$

$$y_{ui} = G_{\mathbf{b}}(x_{ui}). \tag{4.19}$$

Comme précédemment notre but est d'inférer conjointement les variables $\mathbf{W}$ et $\mathbf{H}$ ainsi que la suite des seuils $\mathbf{b}$. Dans notre modèle, la fonction de répartition associée à la variable aléatoire ordinale y_{ui} est :

$$\begin{aligned}
\mathbb{P}[y_{ui} \leq v|\lambda_{ui}] &= \mathbb{P}[G_{\mathbf{b}}(x_{ui}) \leq v|\lambda_{ui}] && (4.20)\\
&= \mathbb{P}[\lambda_{ui} \cdot \varepsilon_{ui} < b_v] && (4.21)\\
&= \mathbb{P}\left[\varepsilon_{ui} < \frac{b_v}{\lambda_{ui}}\right] && (4.22)\\
&= F_\varepsilon\left(\frac{b_v}{\lambda_{ui}}\right). && (4.23)
\end{aligned}$$

Par conséquent, on peut en déduire que la fonction de masse est donnée par :

$$\begin{aligned}
\mathbb{P}[y_{ui} = v|\lambda_{ui}] &= \mathbb{P}[y_{ui} \leq v|\lambda_{ui}] - \mathbb{P}[y_{ui} \leq v-1|\lambda_{ui}] && (4.24)\\
&= F_\varepsilon\left(\frac{b_v}{\lambda_{ui}}\right) - F_\varepsilon\left(\frac{b_{v-1}}{\lambda_{ui}}\right). && (4.25)
\end{aligned}$$

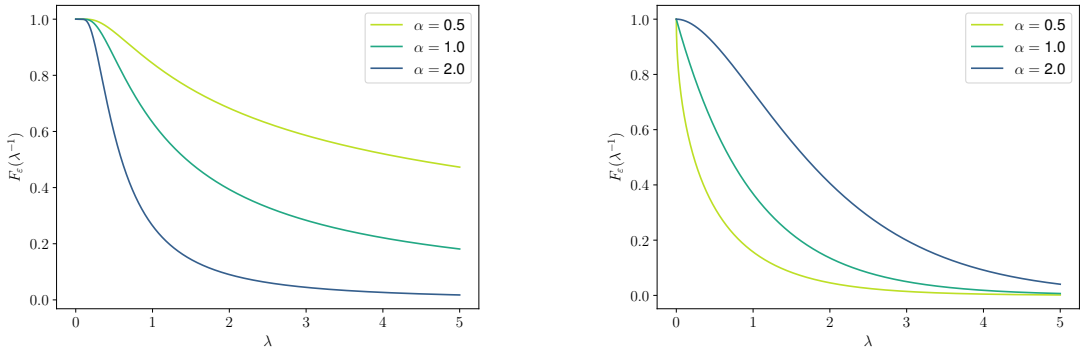
Plusieurs fonctions F_ε peuvent être utilisées dans le modèle présenté ci-dessus, correspondant à différents modèles de bruit. La Figure 4.3 illustre la fonction $\lambda \mapsto F_\varepsilon(\lambda^{-1})$ pour les exemples décrits ci-dessous.

— Bruit gamma : $\varepsilon_{ui} \sim \text{Gamma}(\alpha, 1)$. [2] La fonction de répartition est donnée par $F_\varepsilon(x) = \dfrac{\gamma(\alpha, x)}{\Gamma(\alpha)}$ où $\gamma(\alpha, x) = \int_0^x t^{\alpha-1}e^{-t}\mathrm{d}t$ est la fonction gamma incomplète inférieure. Si $\alpha = 1$, on retrouve un bruit exponentiel $\varepsilon_{ui} \sim \text{Exp}(1)$ dont la fonction de répartition est $F_\varepsilon(x) = 1 - e^{-x}$.

— Bruit inverse-gamma (IG) : $\varepsilon_{ui} \sim \text{IG}(\alpha, 1)$. [2] La fonction de répartition est donnée par $F_\varepsilon(x) = \dfrac{\Gamma(\alpha, x^{-1})}{\Gamma(\alpha)}$ où $\Gamma(\alpha, x) = \int_x^\infty t^{\alpha-1}e^{-t}\mathrm{d}t$ est la fonction gamma incomplète supérieure. Si $\alpha = 1$, on obtient une fonction de répartition de la forme $F_\varepsilon(x) = e^{-1/x}$.

— Tout autre fonction croissante $F_\varepsilon : \mathbb{R}_+ \to [0, 1]$ définit une variable aléatoire positive et pourrait être utilisée dans ce modèle à bruit multiplicatif.

2. Le paramètre d'échelle β est ici fixé à 1 du fait de l'invariance par changement d'échelle existante par rapport à la variable λ_{ui}.



(a) Bruit gamma.

(b) Bruit inverse-gamma.

FIGURE 4.3. – Fonctions $\lambda \mapsto F_\varepsilon(\lambda^{-1})$ associées à des bruits gamma (a) ou inverse-gamma (b).

4.3.2. OrdNMF avec bruit multiplicatif IG

Dans la suite de ce chapitre, nous nous focalisons sur le cas où le bruit multiplicatif est IG avec un paramètre de forme $\alpha = 1$, i.e., $\varepsilon_{ui} \sim \mathrm{IG}(1,1)$ [3]. Nous utiliserons l'acronyme IG-OrdNMF pour cette instance du modèle OrdNMF.

On pose $\theta_v = b_v^{-1}$. La suite $\boldsymbol{\theta}$ correspond à l'inverse des seuils et est donc décroissante. On a $\theta_{-1} = +\infty > \theta_0 > \cdots > \theta_{V-1} > \theta_V = 0$. De plus, on note $\boldsymbol{\Delta}$ la suite positive des décréments définie par $\Delta_v = \theta_{v-1} - \theta_v$ pour $v \in \{1, \dots, V\}$. On a donc la relation $\theta_v = \sum_{l=v+1}^{V} \Delta_l$ avec en particulier $\theta_{V-1} = \Delta_V$.

Interprétation. La fonction de répartition associée à une donnée ordinale y_{ui} dans le modèle IG-OrdNMF est donc donnée par :

$$\mathbb{P}[y_{ui} \leq v | \lambda_{ui}] = e^{-\lambda_{ui}\theta_v}, \tag{4.26}$$

$$\text{soit } \mathbb{P}[y_{ui} > v | \lambda_{ui}] = 1 - e^{-\lambda_{ui}\theta_v}, \tag{4.27}$$

pour $v \in \{0, \dots, V\}$. Par conséquent, la BePoF présentée en Section 4.2.2 est un cas particulier de la IG-OrdNMF pour $V = 1$ et $\theta_0 = 1$.

Cette formulation permet une nouvelle interprétation du modèle IG-OrdNMF. En effet, l'événement $\{y_{ui} > v\}$ est une variable aléatoire binaire suivant une loi Bernoulli : $\{y_{ui} > v\} \sim \mathrm{Bern}(1 - e^{-\lambda_{ui}\theta_v})$. On peut donc voir la IG-OrdNMF comme la fusion de V modèles de

3. L'espérance de la variable IG n'est pas définie lorsque $\alpha \leq 1$, cependant le modèle reste bien posé.

BePoF pour différents seuils $v \in \{0, \dots, V-1\}$ de binarisation. Les coefficients θ_v permettent de régler les poids liés à chaque factorisation.

Fonction de masse. La fonction de masse :

$$\mathbb{P}[y_{ui} = v|\lambda_{ui}] = \begin{cases} e^{-\lambda_{ui}\theta_0}, & \text{si } v = 0, \\ e^{-\lambda_{ui}\theta_v} - e^{-\lambda_{ui}\theta_{v-1}}, & \text{si } v \in \{1, \dots, V-1\}, \\ 1 - e^{-\lambda_{ui}\theta_{V-1}}, & \text{si } v = V. \end{cases} \tag{4.28}$$

On peut réécrire la log-vraisemblance sous la forme :

$$\log \mathbb{P}[y_{ui} = v|\lambda_{ui}] = \begin{cases} -\lambda_{ui}\theta_0, & \text{si } v = 0, \\ -\lambda_{ui}\theta_v + \log(1 - e^{-\lambda_{ui}\Delta_v}), & \text{si } v \in \{1, \dots, V\}. \end{cases} \tag{4.29}$$

Cette réécriture fait apparaître un terme linéaire en λ_{ui} et un terme non-linéaire de la forme $x \mapsto \log(1 - e^{-x})$ faisant apparaître la fonction de lien BePo.

De plus, l'espérance des observations est bien définie et est donnée par :

$$\mathbb{E}(y_{ui}|\lambda_{ui}) = V - \sum_{v=0}^{V-1} e^{-\lambda_{ui}\theta_v}. \tag{4.30}$$

Nous rappelons que, dans le cadre du traitement de données ordinales, l'espérance n'est pas une bonne statistique puisqu'elle suppose implicitement une notion de distance entre les différentes classes. Cependant, cette valeur nous sera utile par la suite pour construire nos listes de recommandation. On remarque notamment que la fonction $\lambda_{ui} \mapsto \mathbb{E}(y_{ui}|\lambda_{ui})$ est bien une fonction croissante. Ainsi, on retrouve le même comportement que pour la PF, la NBF ou la dcPF : plus le produit scalaire λ_{ui} entre un utilisateur et un article est élevé, plus le niveau d'interaction sera élevé (en espérance).

4.4. Inférence bayésienne

4.4.1. Modèle augmenté

Comme nous l'avons vu précédemment, la log-vraisemblance des données ordinales pour $v \in \{1, \dots, V\}$ fait apparaître un terme en $\log(1 - e^{-x})$ qui n'est pas conjugué avec la distribution gamma et rend l'inférence compliquée. Pour résoudre ce problème, nous utilisons

l'astuce présentée en Section 4.2.2 en augmentant notre modèle avec la variable latente :

$$n_{ui}|y_{ui},\lambda_{ui} \sim \begin{cases} \delta_0, & \text{si } y_{ui}=0, \\ \text{ZTP}(\lambda_{ui}\Delta_{y_{ui}}), & \text{si } y_{ui}>0. \end{cases} \tag{4.31}$$

De plus, comme dans les chapitres précédents, nous augmentons notre modèle avec la variable latente $\mathbf{c}_{ui}|n_{ui},\lambda_{ui} \sim \text{Mult}(n_{ui},\boldsymbol{\phi}_{ui})$, où $\boldsymbol{\phi}_{ui}$ est un vecteur de probabilité ayant pour coefficients $\frac{\lambda_{uik}}{\lambda_{ui}}$.

De ce fait, pour les données $y_{ui} \in \{1,\dots,V\}$, nous obtenons la log-vraisemblance jointe suivante (les dérivations sont détaillées en Annexe A.4) :

$$\log p(y_{ui},n_{ui},\mathbf{c}_{ui}|\lambda_{ui}) = \log p(y_{ui}|\lambda_{ui}) + \log p(n_{ui}|y_{ui},\lambda_{ui}) + \log p(\mathbf{c}_{ui}|n_{ui},\lambda_{ui}) \tag{4.32}$$

$$= -\lambda_{ui}\theta_{y_{ui}-1} + n_{ui}\log\Delta_{y_{ui}} + \sum_k \left(c_{uik}\log\lambda_{uik} - \log c_{uik}!\right), \tag{4.33}$$

s.c. $n_{ui} \in \mathbb{N}^*$ et $n_{ui} = \sum_k c_{uik}$.

Log-vraisemblance jointe du modèle IG-OrdNMF. On note $\mathbf{Z} = \{\mathbf{N},\mathbf{C},\mathbf{W},\mathbf{H}\}$ l'ensemble des variables latentes du modèle augmenté. De plus, on définit H_v tel que :

$$H_v = \begin{cases} \theta_0, & \text{si } v=0, \\ \theta_{v-1}, & \text{si } v>0. \end{cases} \tag{4.34}$$

La log-vraisemblance jointe de la IG-OrdNMF est donc donnée par :

$$\log p(\mathbf{Y},\mathbf{N},\mathbf{C}|\mathbf{W},\mathbf{H}) = \sum_{ui}\left[n_{ui}\log\Delta_{y_{ui}} + \sum_k \left(c_{uik}\log\lambda_{uik} - \log c_{uik}!\right) - \lambda_{ui}H_{y_{ui}}\right]. \tag{4.35}$$

Il est important de rappeler que $n_{ui}=0$ et $\mathbf{c}_{ui}=\mathbf{0}_K$ lorsque $y_{ui}=0$. Par conséquent, les variables $\mathbf{N}$ et $\mathbf{C}$ sont partiellement observées, comme c'était le cas pour la dcPF en Chapitre 3.

4.4.2. Inférence variationnelle

La distribution a posteriori $p(\mathbf{Z}|\mathbf{Y})$ est insoluble. Nous utilisons donc la VI pour approximer cette distribution par une distribution variationnelle q plus simple. Nous supposons que

Tableau 4.1. – Expression des distributions variationnelles pour le modèle OrdNMF.

Variable	Distribution
$\mathbf{C}$	$q(\mathbf{c}_{ui}\|n_{ui}) = \text{Mult}\left(\mathbf{c}_{ui};\ n_{ui},\ \tilde{\boldsymbol{\phi}}_{ui}\right)$
$\mathbf{N}$	$q(n_{ui}) = \begin{cases} \text{ZTP}(n_{ui};\ \Lambda_{ui}\Delta_{y_{ui}}), & \text{si } y_{ui} > 0 \\ \delta_0, & \text{si } y_{ui} = 0 \end{cases}$
$\mathbf{W}$	$q(w_{uk}) = \text{Gamma}\left(w_{uk};\ \tilde{\alpha}^W_{uk},\ \tilde{\beta}^W_{uk}\right)$
$\mathbf{H}$	$q(h_{ik}) = \text{Gamma}\left(h_{ik};\ \tilde{\alpha}^H_{ik},\ \tilde{\beta}^H_{ik}\right)$

q appartient à la famille du champ moyen et s'écrit donc sous la forme factorisée suivante :

$$q(\mathbf{Z}) = \prod_{ui} q(n_{ui}, \mathbf{c}_{ui}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \tag{4.36}$$

Encore une fois, nous gardons les variables $\mathbf{N}$ et $\mathbf{C}$ couplées (comme dans le Chapitre 3).

Nous utilisons un algorithme CAVI qui nous permet d'obtenir les expressions analytiques des distributions variationnelles décrites en Tableau 4.1. Les règles de mise à jour des paramètres variationnels sont résumées en Algorithme 5. Les dérivations des calculs menant à ces résultats sont détaillées en Annexe A.4.

Approximation et lien avec la PFbin. L'Algorithme 5 peut être simplifié en supposant que $q(n_{ui}) = \delta_1$ si $y_{ui} > 0$. Cela revient à remplacer le terme non-linéaire $\log(1 - e^{-x})$ par $\log x$ dans l'Éq. (4.29) (voir Figure 4.4). Cependant, cette approximation ne produira des résultats similaires que si x est très petit, puisque $\log(1 - e^{-x}) = \log x + o(x)$. En pratique, on ne peut vérifier cela qu'a posteriori en constatant que $\mathbb{E}_q(n_{ui}) \approx 1$.

Comme nous l'avons évoqué précédemment, la BePoF est un cas particulier de la IG-OrdNMF pour $V = 1$ et $\theta_0 = 1$. De ce fait, on peut remarquer que l'algorithme PFbin est une approximation de l'algorithme BePoF avec $q(n_{ui}) = \delta_1$ si $y_{ui} = 1$.

4.4.3. Estimation des seuils

Un élément clé des modèles à seuillage latent est l'apprentissage des seuils (correspondants ici aux paramètres $\boldsymbol{\theta}$). Pour cela, on utilise un algorithme VBEM comme décrit en Section 1.3.4. On cherche alors à maximiser le terme $\mathbb{E}_q(\log p(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta}))$ par rapport aux

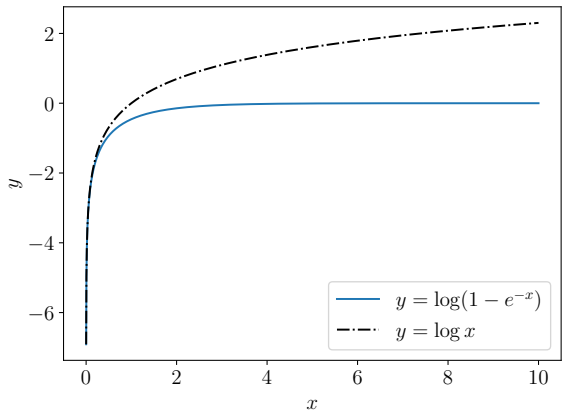


FIGURE 4.4. – Représentation des fonctions $x \mapsto \log(1 - e^{-x})$ et $x \mapsto \log x$.

variables $\boldsymbol{\theta}$, qui est donné par :

$$\begin{aligned} &\mathbb{E}_q(\log p(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta})) = \sum_{ui} \Big[\mathbb{E}_q(n_{ui}) \log \Delta_{y_{ui}} - \mathbb{E}_q(\lambda_{ui}) H_{y_{ui}}\Big] + cste, \\ &\text{s.c. } \theta_0 > \theta_1 > \cdots > \theta_{V-1} > \theta_V = 0. \end{aligned} \tag{4.37}$$

On rappelle que H_v (défini en Éq. (4.34)) et $\Delta_v = \theta_{v-1} - \theta_v > 0$ sont des termes dépendants de la suite $\boldsymbol{\theta}$.

Optimisation des décréments. On choisit de travailler avec la suite des décréments $\boldsymbol{\Delta}$ plutôt qu'avec la suite des seuils inversés $\boldsymbol{\theta}$. En effet, en faisant cela, la contrainte de décroissance devient une contraire de positivité des décréments et nous obtenons seulement des termes en x et $\log x$ dans la fonction à maximiser. Le problème peut alors être résolu analytiquement.

On peut réécrire le terme H_v en fonction de la suite $\boldsymbol{\Delta}$ en notant que :

$$H_v = \sum_{l=1}^{V} \mathbb{1}[v \leq l] \Delta_l, \, \forall v \in \{0, \ldots, V\}. \tag{4.38}$$

Par conséquent, le problème d'optimisation présenté en Éq. (4.37) revient à maximiser la

Algorithme 5 : Algorithme CAVI pour OrdNMF pour des données implicites.

Données : Matrice d'observation $\mathbf{Y}$
Résultat : Distribution variationnelle q et seuils $\boldsymbol{\theta}$

Initialisation des paramètres variationnels et des seuils $\boldsymbol{\theta}$;
répéter
  pour chaque *couple tel que* $y_{ui} > 0$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$;
    $\mathbb{E}_q(n_{ui}) = \dfrac{\Lambda_{ui}\Delta_{y_{ui}}}{1 - e^{-\Lambda_{ui}\Delta_{y_{ui}}}}$; $\mathbb{E}_q(c_{uik}) = \mathbb{E}_q(n_{ui})\dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
  fin
  pour chaque *utilisateur* $u \in \{1, \dots, U\}$ **faire**
    $\tilde{\alpha}^W_{uk} = \alpha^W + \sum_i \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^W_{uk} = \beta^W + \sum_i H_{y_{ui}} \mathbb{E}_q(h_{ik})$;
  fin
  pour chaque *article* $i \in \{1, \dots, I\}$ **faire**
    $\tilde{\alpha}^H_{ik} = \alpha^H + \sum_u \mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^H_{ik} = \beta^H + \sum_u H_{y_{ui}} \mathbb{E}_q(w_{uk})$;
  fin
  Mise à jour des seuils selon Éq. (4.40) et Éq. (4.41) ;
  Mise à jour des paramètres d'intensité ;
  Calculer ELBO$(q, \boldsymbol{\theta})$;
jusqu'à *ELBO converge*;

fonction suivante :

$$\mathbb{E}_q(\log p(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\Delta})) = \sum_{ui} \sum_{l=1}^{V} \left(\mathbb{1}[y_{ui} = l] \mathbb{E}_q(n_{ui}) \log \Delta_l - \mathbb{1}[y_{ui} \leq l] \mathbb{E}_q(\lambda_{ui}) \Delta_l \right) + cste,$$
$$\text{s.c. } \boldsymbol{\Delta} \geq 0. \tag{4.39}$$

On obtient donc les règles de mise à jour suivantes :

$$\Delta_l = \frac{\sum_{ui} \mathbb{1}[y_{ui} = l] \mathbb{E}_q(n_{ui})}{\sum_{ui} \mathbb{1}[y_{ui} \leq l] \mathbb{E}_q(\lambda_{ui})}, \text{ pour tout } l \in \{1, \dots, V\}, \tag{4.40}$$

$$\theta_v = \sum_{l=v+1}^{V} \Delta_l, \qquad \text{pour tout } v \in \{0, \dots, V-1\}. \tag{4.41}$$

Espérance prédictive a posteriori. L'espérance prédictive a posteriori nous permet d'établir des listes de recommandation pour chaque utilisateur. En utilisant la VI, on peut

Tableau 4.2. – Pré-traitement appliqué aux données.

Classe	Nombre d'écoutes
0	0
1	1
2	2
3	$3-5$
4	$6-10$
5	$11-20$
6	$21-50$
7	$51-100$
8	$101-200$
9	$201-500$
10	>500

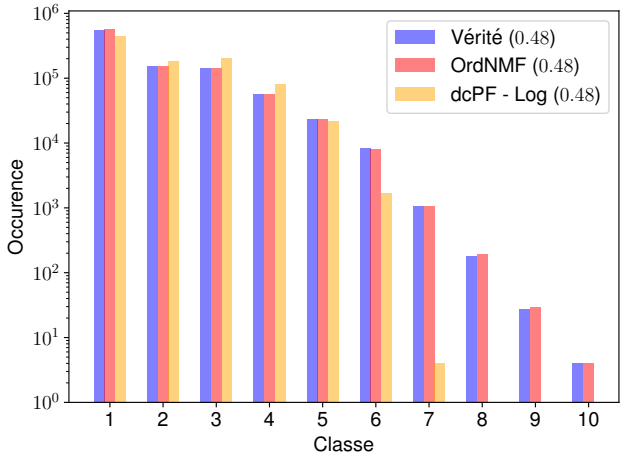


FIGURE 4.5. – PPC de l'histogramme des classes pour le modèle OrdNMF.

l'approximer par :

$$\mathbb{E}(\mathbf{Y}^*|\mathbf{Y}) \approx \int_{\mathbf{W},\mathbf{H}} \mathbb{E}(\mathbf{Y}^*|\mathbf{W},\mathbf{H}) q(\mathbf{W}) q(\mathbf{H}) \mathrm{d}\mathbf{W} \mathrm{d}\mathbf{H}. \tag{4.42}$$

Contrairement à la PF, à la NBF et à la dcPF, cette approximation ne peut pas être calculée analytiquement. Cependant, nous ne sommes intéressés que par l'ordre de cette espérance par rapport aux articles et non pas par sa valeur en elle-même. La fonction $\lambda_{ui} \mapsto \mathbb{E}(y^*_{ui}|\lambda_{ui})$ étant croissante, on peut donc établir que l'ordre des recommandations donné par le score de l'Éq. (4.42) est le même que celui obtenu par le score $s_{ui} = [\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H})^T]_{ui}$.

4.5. Résultats expérimentaux

4.5.1. Protocole expérimental

Jeu de données et pré-traitement. Nous considérons le même sous-ensemble du jeu de données Taste Profile que dans les Sections 1.4.6 et 3.7. Comme expliqué dans l'introduction du chapitre, nous utilisons un pré-traitement prédéfini des données afin d'obtenir des données quantifiées. Ce pré-traitement est décrit dans le Tableau 4.2. Nous avons choisi une quantification dans laquelle la taille des intervalles d'écoutes croit exponentiellement. La Figure 4.5 présente en bleu la distribution des différentes classes ainsi construites.

Nous invitons le lecteur à bien faire attention à ne pas confondre la quantification des données (qui correspond à un pré-traitement dont les seuils sont prédéfinis et qui permet d'obtenir des données ordinales), avec la quantification de la variable latente présente dans le modèle OrdNMF (dont les seuils $\mathbf{b}$ sont estimés durant l'inférence du modèle).

Méthode d'évaluation. Nous utilisons la même méthode d'évaluation que celle présentée en Section 2.5 pour la NBF. Nous divisons donc le jeu de données en un ensemble d'entraînement $\mathbf{Y}^{\text{train}}$ contenant 80% des valeurs non nulles du jeu de données original et un ensemble de test $\mathbf{Y}^{\text{test}}$ contenant les 20% restants (ces valeurs sont mises à zéro dans l'ensemble d'entraînement). Chaque algorithme est entraîné sur l'ensemble d'entraînement et fournit une liste de recommandations de 100 articles à chaque utilisateur fondée sur le score de prédiction : $[\mathbb{E}_q(\mathbf{W})\mathbb{E}_q(\mathbf{H}^T)]_{ui}$ comme discuté en Section 4.4.2. La qualité de ces listes est ensuite évaluée sur l'ensemble de test à l'aide de la métrique NDCG présentée en Section 1.1.3.

Méthodes comparées. On compare la OrdNMF avec les différentes méthodes étudiées dans cette thèse, à savoir : BePoF (cas particulier de la OrdNMF avec $V = 1$), PFbin (approximation de la BePoF) et dcPF avec loi élémentaire Log qui est appliquée sur les données brutes. Nous sélectionnons les hyper-paramètres $\alpha^W = \alpha^H = 0.3$ pour tous les modèles. Nous sélectionnons $K = 150$ pour la PFbin et la dcPF, et $K = 200$ pour la OrdNMF et la BePoF, qui sont les nombres de facteurs latents qui donnent les meilleurs résultats en terme de NDCG0 parmi $K \in \{150, 200, 300\}$. Le critère d'arrêt des différents algorithmes est fixé à $\tau = 10^{-5}$. Pour chaque expérience, les algorithmes sont exécutés cinq fois avec des initialisations aléatoires.

4.5.2. Résultats de prédiction

Le Tableau 4.3 reporte les résultats des quatre modèles pour la métrique d'évaluation NDCG pour les seuils $s \in \{0, 1, 5, 10\}$. Les deux dernières lignes du tableau sont les mêmes que dans le Tableau 3.5.

Premièrement, ce tableau nous permet de comparer la BePoF et son approximation PFbin (voir la remarque de la Section 4.4.2). La BePoF surpasse légèrement PFbin en terme de recommandation. De plus, le nombre de facteurs latents est plus grand pour la BePoF. Cela peut s'expliquer par le fait que la BePoF pénalise moins les grandes valeurs de $\mathbf{WH}^T$ que PFbin lorsque $y_{ui} > 0$. En effet, nous avons que $\lim_{x \to +\infty} \log(1 - e^{-x}) = 0$ pour la BePoF

Tableau 4.3. – Performance des modèles OrdNMF, BePoF, PFbin et dcPF (avec loi élémentaire logarithmique) sur le jeu de données TP. En gras : les deux meilleurs scores NDCG. Entre parenthèses : l'écart-type des 5 exécutions effectuées.

Modèle	K	NDCG0	NDCG1	NDCG5	NDCG10
OrdNMF	200	**0.207** ($\mathbf{4 \cdot 10^{-3}}$)	**0.187** ($\mathbf{4 \cdot 10^{-3}}$)	**0.149** ($\mathbf{4 \cdot 10^{-3}}$)	**0.134** ($\mathbf{5 \cdot 10^{-3}}$)
BePoF	200	0.206 ($4 \cdot 10^{-3}$)	0.185 ($4 \cdot 10^{-3}$)	0.145 ($3 \cdot 10^{-3}$)	0.130 ($3 \cdot 10^{-3}$)
PFbin	150	0.205 ($2 \cdot 10^{-3}$)	0.184 ($1 \cdot 10^{-3}$)	0.145 ($6 \cdot 10^{-4}$)	0.130 ($9 \cdot 10^{-4}$)
dcPF - Log	150	**0.208** ($\mathbf{3 \cdot 10^{-3}}$)	**0.188** ($\mathbf{3 \cdot 10^{-3}}$)	**0.152** ($\mathbf{2 \cdot 10^{-3}}$)	**0.139** ($\mathbf{2 \cdot 10^{-4}}$)

alors que $\lim_{x \to +\infty} \log x = +\infty$ pour PFbin. Enfin, nous pouvons vérifier la qualité de cette approximation en observant la variable latente $\mathbf{N}$ pour la BePoF. Nous avons que la valeur moyenne des $\mathbb{E}_q(n_{ui})$ pour les valeurs $y_{ui} > 0$ est d'environ 1.069. Nous pouvons donc établir que la PFbin est une bonne approximation de la BePoF, même si elle dégrade légèrement les performances de recommandation.

En comparant les deux premières lignes du tableau, on peut constater que la OrdNMF améliore bien les performances de recommandation par rapport à la BePoF (qui en est un cas particulier travaillant sur les données binarisées). Cette amélioration est d'autant plus nette que le seuil augmente. Cela confirme les conclusions de la Section 1.4.6 relatant que les valeurs des données de comptage contiennent de l'information utile à la recommandation. Cependant, le modèle OrdNMF a des performances légèrement plus faibles que la dcPF avec loi élémentaire Log comme présentée en Chapitre 3.

4.5.3. Vérification prédictive a posteriori (PPC)

Similairement à la Section 3.7.4, nous présentons un PPC de la distribution des différentes classes $\{0, 1, \ldots, 10\}$ de nos données, pour les modèles OrdNMF et dcPF. Pour cela, nous générons artificiellement de nouvelles données à partir de la distribution prédictive a posteriori $p(\mathbf{Y}^*, \mathbf{W}, \mathbf{H}|\mathbf{Y}) \approx p(\mathbf{Y}^*|\mathbf{W}, \mathbf{H})q(\mathbf{W})q(\mathbf{H})$. La dcPF générant directement des données de comptage, nous appliquons le pré-traitement décrit dans le Tableau 4.2 afin de pouvoir comparer les résultats avec la OrdNMF qui génère des classes.

La Figure 4.5 présente les résultats de ces PPC, les barres bleues correspondent à l'histogramme empirique des données ($\mathbf{Y}^{\text{train}}$), les barres rouges et oranges correspondent elles aux histogrammes des données simulées avec la OrdNMF et la dcPF respectivement. Comme nous l'avions constaté en Section 3.7.4, la dcPF ne parvient pas à modéliser les très grandes valeurs présentes dans les données (à partir de la classe 7, ce qui correspond aux valeurs

supérieures à 50 écoutes). Au contraire, la OrdNMF parvient à s'adapter parfaitement au pré-traitement appliqué aux données, comme en témoigne la proximité entre les barres bleues et rouges.

4.6. Données explicites

Dans cette section, nous nous intéressons au problème de CF pour des données explicites. Le plus souvent, ces données correspondent à des notes d'utilisateurs sur des articles. Ici, nous prenons l'exemple d'un sous-ensemble du jeu de données MovieLens qui contient les notes de $U = 20$k utilisateurs sur $I = 12$k films. Les notes vont de 0.5 pour la note minimale, à 5 pour la note maximale, avec un pas de 0.5. La distribution des notes dans le jeu de données est illustrée en Figure 4.6 (barres bleues). On constate notamment que l'échelle de notation est «discontinue», les utilisateurs préférant utiliser les notes entières plutôt que les demi-points.

Les notes données par les utilisateurs sont des données ordinales puisqu'elles possèdent une relation d'ordre permettant de classer leurs préférences, la note n'étant considérée que comme l'étiquette associée à la classe. De ce fait, on considérera ici que les notes appartiennent à $\{0, 1, \dots, V\}$ avec $V = 9$. La classe 0 correspond à la note 0.5 et la classe 9 correspond à la note 5.

L'une des particularités des données explicites est que la matrice $\mathbf{Y}$ contenant les notes n'est que partiellement observée (les utilisateurs n'ont noté qu'une partie des films disponibles). Le but est donc de prédire les notes manquantes de cette matrice afin de pouvoir indiquer aux utilisateurs quels films ils devraient regarder. On reprend les notations présentées en Section 1.2.1 : $\mathcal{O}$ correspond à l'ensemble des couples utilisateurs-articles (u, i) tels qu'une note est observée ; $\mathbf{M}$ est la matrice binaire indicatrice du motif d'absence des données, i.e., $m_{ui} = \mathbb{1}[(u, i) \in \mathcal{O}]$. On supposera dans la suite que le motif d'absence $\mathbf{M}$ est lui aussi aléatoire.

4.6.1. Hypothèse MAR

Dans cette première section, on suppose que les données sont manquantes au hasard (MAR). Dans ce cas, le motif d'absence $\mathbf{M}$ peut être ignoré lors de l'inférence. La matrice $\mathbf{M}$ est alors considérée comme un masque qui cache les valeurs manquantes de la matrice $\mathbf{Y}$. L'algorithme de OrdNMF sous cette hypothèse peut facilement être adapté en introduisant le masque $\mathbf{M}$. L'Algorithme 6 résume les différentes règles de mise à jour. Nous entraînons

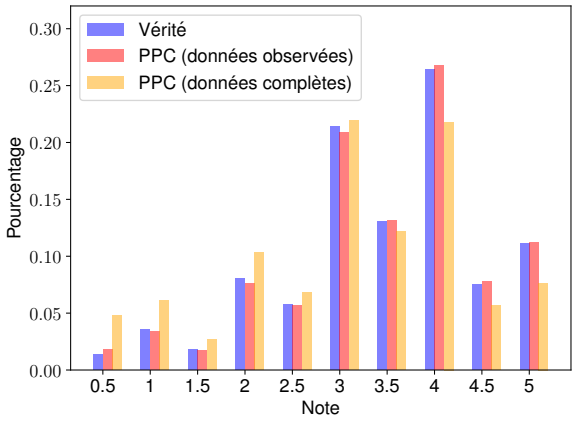


FIGURE 4.6. – PPC de la OrdNMF sous hypothèse MAR.

le modèle OrdNMF sur le jeu de données MovieLens. Le nombre de facteurs latents est fixé à $K = 150$ et le critère d'arrêt est fixé à $\tau = 10^{-5}$.

Nous réalisons une vérification prédictive a posteriori (PPC) de la distribution des notes dans le jeu de données. Pour cela, nous générons une nouvelle matrice de note $\mathbf{Y}^*$ à partir des matrices $\mathbf{W}$, $\mathbf{H}$ et des seuils $\mathbf{b}$ estimés. La Figure 4.6 illustre le PPC réalisé à partir de ces simulations. Nous comparons alors la distribution empiriques des notes, avec celle des notes simulées. Les barres bleues correspondent à la distribution empirique des notes de la matrice $\mathbf{Y}$. Les barres orange correspondent à la distribution des notes simulées pour la totalité de la matrice $\mathbf{Y}^*$. Les barres rouges correspondent à la distribution des notes simulées $\mathbf{Y}^*$ pour lesquelles $m_{ui} = 1$. Comme pour la OrdNMF sur données implicites, on remarque que l'histogramme simulé (rouge) et empirique (bleu) sont très proche l'un de l'autre. Cela montre encore une fois que la OrdNMF est capable de s'adapter à tout type de distribution des données. Les barres oranges nous indiquent que les notes cachées par le masque $\mathbf{M}$ sont légèrement moins élevées que celles observées.

4.6.2. Hypothèse MNAR

Données manquantes en CF. Les auteurs de l'article [MZ09] ont montré que les hypothèses MAR n'étaient pas vérifiées en CF. Pour cela, ils ont étudié deux jeux de données fournis par Yahoo, contenant les notes d'utilisateurs sur 1 000 chansons. Le premier jeu de données correspond aux notes attribuées par les utilisateurs qui ont librement interagi avec le catalogue. Au contraire, le second jeu de données correspond à un sondage où il

Algorithme 6 : Algorithme CAVI pour OrdNMF pour des données explicites sous hypothèse MAR

Données : Matrices d'observation $\mathbf{Y}$ et de masque $\mathbf{M}$
Résultat : Distribution variationnelle q et seuils $\boldsymbol{\theta}$

Initialisation des paramètres variationnels et des seuils $\boldsymbol{\theta}$;
répéter
  pour chaque *couple* $(u, i) \in \mathcal{O}$ *tel que* $y_{ui} > 0$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$;
    $\mathbb{E}_q(n_{ui}) = \dfrac{\Lambda_{ui}\Delta_{y_{ui}}}{1 - e^{-\Lambda_{ui}\Delta_{y_{ui}}}}$; $\mathbb{E}_q(c_{uik}) = \mathbb{E}_q(n_{ui})\dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
  fin
  pour chaque *utilisateur* $u \in \{1, \dots, U\}$ **faire**
    $\tilde{\alpha}^W_{uk} = \alpha^W + \sum_i m_{ui}\mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^W_{uk} = \beta^W + \sum_i m_{ui} H_{y_{ui}} \mathbb{E}_q(h_{ik})$;
  fin
  pour chaque *article* $i \in \{1, \dots, I\}$ **faire**
    $\tilde{\alpha}^H_{ik} = \alpha^H + \sum_u m_{ui}\mathbb{E}_q(c_{uik})$;
    $\tilde{\beta}^H_{ik} = \beta^H + \sum_u m_{ui} H_{y_{ui}} \mathbb{E}_q(w_{uk})$;
  fin
  Mise à jour des seuils selon Éq. (4.40) et Éq. (4.41) ;
  Mise à jour des paramètres d'intensité ;
  Calculer ELBO$(q, \boldsymbol{\theta})$;
jusqu'à *ELBO converge*;

est demandé à chaque participant de noter 10 chansons sélectionnées aléatoirement et de manière uniforme parmi le catalogue. Les résultats de cette expérience montrent que les notes du second jeu de données sont bien plus basses que celles du premier. À partir de cette observation, plusieurs modèles de MF ont été proposés pour lever l'hypothèse MAR au profit d'une hypothèse MNAR [Mar+07 ; MZ09 ; PTW12 ; HHG14].

OrdNMF avec hypothèse MNAR. Dans cette section, nous proposons une version de la OrdNMF permettant de travailler efficacement sur les hypothèses MNAR. Pour cela, nous supposons que la matrice $\mathbf{M}$ est aussi une matrice ordinale possédant la relation d'ordre suivante :

$$m_{ui} = 0 \text{ (donnée manquante)} \prec m_{ui} = 1 \text{ (donnée observée)}. \tag{4.43}$$

Cela signifie que le fait d'observer un retour d'utilisateur est un indicateur de son intérêt pour l'article. Cette relation d'ordre illustre le fait que les utilisateurs ne choisissent pas l'article avec lequel ils vont interagir au hasard. Au contraire, ils le choisissent en présumant de leur intérêt pour celui-ci. Par exemple, un utilisateur qui n'aime pas les films d'horreur a peu de chance d'aller en regarder un au cinéma, puisqu'il suppose qu'il ne va pas aimer le film.

Modèle génératif. Nous travaillons désormais avec deux modalités de données ordinales : la matrice de notes $\mathbf{Y}$ et le motif d'absence $\mathbf{M}$. Comme précédemment, nous lions ces deux aspects des données avec la droite des réels positifs $\mathbb{R}^+$ à l'aide des seuils $\mathbf{b}$ pour les notes des utilisateurs, et du seuil $b^{\text{mis}} = (\theta^{\text{mis}})^{-1}$ pour le motif d'absence. Nous introduisons un bruit multiplicatif pour chacune de ces deux modalités. Par conséquent, le modèle génératif considéré est le suivant :

$$w_{uk} \sim \text{Gamma}(\alpha^W, \beta_u^W),\ h_{ik} \sim \text{Gamma}(\alpha^H, \beta_i^H), \tag{4.44}$$

$$\lambda_{ui} = [\mathbf{W}\mathbf{H}^T]_{ui}, \tag{4.45}$$

$$\varepsilon_{ui}^{\text{mis}} \sim \text{IG}(1,1),\ m_{ui} = G_{b^{\text{mis}}}(\lambda_{ui}\varepsilon_{ui}^{\text{mis}}), \tag{4.46}$$

$$\varepsilon_{ui} \sim \text{IG}(1,1),\ y_{ui} = G_{\mathbf{b}}(\lambda_{ui}\varepsilon_{ui}). \tag{4.47}$$

Nous pouvons aussi voir ce problème comme un problème de co-factorisation de matrices donné par $\mathbf{M} \approx \mathbf{W}\mathbf{H}^T$ et $\mathbf{Y} \approx \mathbf{W}\mathbf{H}^T$. Nous discuterons plus largement des problèmes de co-factorisation dans le Chapitre 5.

Algorithme CAVI. Nous utilisons les mêmes astuces d'augmentation que celles présentées en Section 4.4.1 pour chacune des deux modalités. Nous avons donc les augmentations suivantes :

$$n_{ui} \sim \text{ZTP}(\lambda_{ui}\Delta_{y_{ui}}),\ \mathbf{c}_{ui} \sim \text{Mult}(n_{ui}, \boldsymbol{\phi}_{ui}),\ \text{pour } y_{ui} \in \{1, \dots, V\}, \tag{4.48}$$

$$n_{ui}^{\text{mis}} \sim \text{ZTP}(\lambda_{ui}\theta^{\text{mis}}),\ \mathbf{c}_{ui}^{\text{mis}} \sim \text{Mult}(n_{ui}^{\text{mis}}, \boldsymbol{\phi}_{ui}),\ \text{pour } m_{ui} = 1, \tag{4.49}$$

où $\boldsymbol{\phi}_{ui}$ est un vecteur de probabilité ayant pour coefficients $\frac{\lambda_{uik}}{\lambda_{ui}}$.

Nous notons $\mathbf{Z} = \{\mathbf{N}, \mathbf{C}, \mathbf{N}^{\text{mis}}, \mathbf{C}^{\text{mis}}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes. La distribution a posteriori $p(\mathbf{Z}|\mathbf{Y}, \mathbf{M})$ est insoluble. Nous l'approximons donc par la distribution

Algorithme 7 : Algorithme CAVI pour OrdNMF pour des données explicites sous hypothèse MNAR

Données : Matrices d'observation $\mathbf{Y}$ et de masque $\mathbf{M}$
Résultat : Distribution variationnelle q et seuils $\boldsymbol{\theta}$, θ^{mis}

Initialisation aléatoire des paramètres variationnels et des seuils $\boldsymbol{\theta}$, θ^{mis} ;
répéter
  pour chaque *couple* $(u, i) \in \mathcal{O}$ **faire**
    $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$; $\Lambda_{ui} = \sum_k \Lambda_{uik}$;
    pour $y_{ui} > 0$: $\mathbb{E}_q(n_{ui}) = \dfrac{\Lambda_{ui}\Delta_{y_{ui}}}{1 - e^{-\Lambda_{ui}\Delta_{y_{ui}}}}$; $\mathbb{E}_q(c_{uik}) = \mathbb{E}_q(n_{ui})\dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
    $\mathbb{E}_q(n_{ui}^{\text{mis}}) = \dfrac{\Lambda_{ui}\theta^{\text{mis}}}{1 - e^{-\Lambda_{ui}\theta^{\text{mis}}}}$; $\mathbb{E}_q(c_{uik}^{\text{mis}}) = \mathbb{E}_q(n_{ui}^{\text{mis}})\dfrac{\Lambda_{uik}}{\Lambda_{ui}}$;
  fin
  pour chaque *utilisateur* $u \in \{1, \dots, U\}$ **faire**
    $\tilde{\alpha}_{uk}^W = \alpha^W + \sum_i m_{ui}\left(\mathbb{E}_q(c_{uik}) + \mathbb{E}_q(c_{uik}^{\text{mis}})\right)$;
    $\tilde{\beta}_{uk}^W = \beta_u^W + \sum_i m_{ui} H_{y_{ui}} \mathbb{E}_q(h_{ik}) + \theta^{\text{mis}} \sum_i \mathbb{E}_q(h_{ik})$;
  fin
  pour chaque *article* $i \in \{1, \dots, I\}$ **faire**
    $\tilde{\alpha}_{ik}^H = \alpha^H + \sum_u m_{ui}\left(\mathbb{E}_q(c_{uik}) + \mathbb{E}_q(c_{uik}^{\text{mis}})\right)$;
    $\tilde{\beta}_{ik}^H = \beta_i^H + \sum_u m_{ui} H_{y_{ui}} \mathbb{E}_q(w_{uk}) + \theta^{\text{mis}} \sum_u \mathbb{E}_q(w_{uk})$;
  fin
  Mise à jour des seuils selon Éq. (4.40) et Éq. (4.41) ;
  Mise à jour du seuil : $\theta^{\text{mis}} = \dfrac{\sum_{ui} \mathbb{E}_q(n_{ui}^{\text{mis}})}{\sum_{ui} \mathbb{E}_q(\lambda_{ui}^{\text{mis}})}$;
  Mise à jour des paramètres d'intensité ;
  Calculer ELBO$(q, \boldsymbol{\theta}, \Phi)$;
jusqu'à *ELBO converge*;

variationnelle q qui est choisie parmi la famille du champ moyen :

$$q(\mathbf{Z}) = \prod_{(u,i)\in\mathcal{O}} q(n_{ui}, \mathbf{c}_{ui}) \prod_{ui} q(n_{ui}^{\text{mis}}, \mathbf{c}_{ui}^{\text{mis}}) \prod_{uk} q(w_{uk}) \prod_{ik} q(h_{ik}). \tag{4.50}$$

L'algorithme CAVI permettant d'obtenir q ainsi qu'une estimation des seuils $\mathbf{b}$ est décrit en Algorithme 7.

Vérification prédictive a posteriori (PPC). Comme précédemment nous réalisons un PPC de la distribution des notes pour le modèle OrdNMF sous hypothèse MAR. Cette fois-ci, nous générons à la fois une matrice de notes $\mathbf{Y}^*$ ainsi qu'une matrice indicatrice des

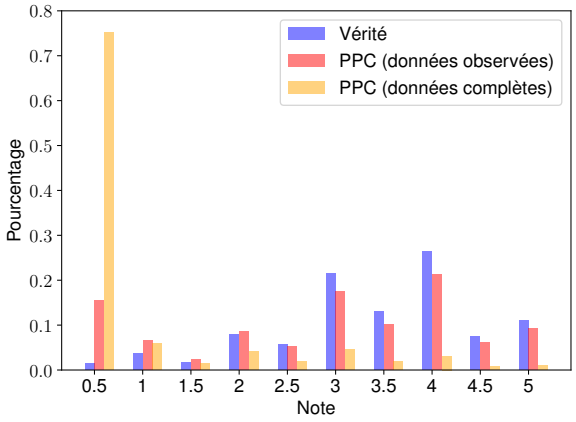


FIGURE 4.7. – PPC de la OrdNMF sous hypothèse MNAR.

absences $\mathbf{M}^*$, à partir des matrices $\mathbf{W}$, $\mathbf{H}$ et des seuils $\mathbf{b}$ et $b^{\min}$ inférés. La Figure 4.7 illustre ce PPC réalisé à partir de ces simulations. On note que les barres rouges correspondent ici à la distribution des notes simulées $\mathbf{Y}^*$ pour lesquelles $m^*_{ui} = 1$.

Comme pour le premier modèle les barres bleues et rouges sont assez proches l'une de l'autre. Néanmoins, on observe que les petites notes sont sur-estimées par rapport à l'histogramme empirique. Cela peut s'expliquer de deux manières : le modèle MNAR est plus difficile à inférer que le modèle MAR (introduction de nouvelles variables latentes), le PPC est plus variable que précédemment puisque la matrice de motif d'absence $\mathbf{M}^{\text{mis}}$ est aussi générée à partir des paramètres estimés.

Contrairement au premier modèle sous hypothèse MAR, nous constatons ici une très grande différence entre la distribution des notes observées (en rouge) et la distribution des notes complètes (en orange). Les notes manquantes sont bien plus basses que celles observées, puisque l'hypothèse MNAR suppose qu'une note manquante traduit aussi un désintérêt pour le film.

Ces PPC réalisés sur les données explicites semblent prometteurs. Les futurs travaux consisteront à mener de plus amples expériences avec notamment des évaluations de recommandation (via les métriques MAE, MSE ou NDCG par exemple).

4.7. Discussion

Dans ce chapitre, nous avons proposé une nouvelle approche pour prendre en compte les données implicites sur-dispersées rencontrées en CF. Contrairement aux Chapitres 2 et 3, nous n'avons pas cherché à modéliser directement ces données, mais nous avons choisi de travailler sur une version quantifiée des données (ce qui est un pré-traitement plus souple que la binarisation introduite en Section 1.4). Nous avons développé un modèle de NMF capable de d'adapter à ces données ordinales. En particulier, nous avons étudié le modèle IG-OrdNMF qui est une extension du modèle BePoF. Nous avons conduit des expériences sur le même jeu de données que les chapitres précédents. À partir de ces expériences, nous pouvons conclure que la OrdNMF s'adapte bien aux données quantifiées. En particulier, elle prend aussi en compte les grandes classes qui correspondent aux grandes valeurs présentes dans les données. Plusieurs perspectives de travail peuvent être évoquées.

Modèles de bruit. Tout d'abord, comme nous l'avons décrit en Section 4.3.1, le modèle OrdNMF peut se décliner pour différents choix de bruit multiplicatif. Par exemple, la Figure 4.3 illustre les choix d'un bruit gamma ou IG avec différents paramètres de forme. On peut notamment remarquer que les fonctions associées aux lois gamma ou IG avec un paramètre de forme $\alpha > 1$ présentent toutes un point d'inflexion. Ainsi, ces fonctions exhibent deux paliers permettant de mieux discriminer les valeurs de λ. Par conséquent, ce genre de bruit permettrait sûrement de mieux discriminer les classes des données ordinales.

Parmi ces choix, le bruit exponentiel (bruit gamma avec $\alpha = 1$) retient toute notre attention. En effet, la fonction de répartition est connue en forme analytique pour ce cas. De plus, nous pouvons à nouveau utiliser l'astuce d'augmentation présentée en Section 4.2.2 afin de simplifier le modèle. Nous obtenons alors une vraisemblance jointe faisant apparaître une divergence d'IS pondérée. Comme pour la IG-OrdNMF, des algorithmes de VI pourraient être mis en place, en utilisant la propriété de conjugaison de la loi GIG.

Quantification des données. Une des limites du modèle proposé dans ce chapitre est qu'il dépend du pré-traitement choisi pour quantifier les données. Une perspective serait d'apprendre à la fois cette quantification et d'inférer la OrdNMF. Une autre idée pourrait être d'appliquer la OrdNMF directement sur les données de comptage (sans pré-traitement) en ajoutant des lois a priori sur les seuils $\mathbf{b}$.

Chapitre 5.

Co-factorisation de matrices pour données multimodales

Ce chapitre est adapté de l'article [GOF18a] publié à la conférence International Society for Music Information Retrieval *(ISMIR) en 2018.*

Contents

5.1. Introduction : le problème du démarrage à froid

Dans les Chapitres 2, 3 et 4, nous avons proposé différents modèles afin de s'adapter à la sur-dispersion des données implicites rencontrées en CF. Comme toute méthode de CF, ces modèles souffrent du problème de démarrage à froid [Sch+02 ; Lam+08]. Cela signifie qu'ils sont dans l'incapacité de proposer de nouvelles recommandations pour de nouveaux utilisateurs ou de nouveaux articles. En effet, les méthodes de CF nécessitent de posséder des historiques d'interactions suffisants afin de pouvoir inférer les vecteurs de préférences ou d'attributs. Une des façons de résoudre ce problème est de mettre en place des méthodes hybrides de recommandation, mêlant le CF avec le filtrage fondé sur le contenu, en introduisant de nouvelles modalités.

Comme pour les chapitres précédents, nous utilisons le jeu de données Taste Profile contenant les historiques d'écoutes d'utilisateurs. Nous ajoutons à cela de nouvelles informations sur les chansons, sous la forme de tags. Ces tags ont été attribués à chaque chanson par des utilisateurs et collectés par le site internet Last.fm. À partir de ces données multimodales, nous nous fixons deux objectifs : pouvoir recommander des chansons jamais écoutées sur la base de leurs tags associés, et pouvoir attribuer des tags à des chansons non labellisées sur la base de leur audience.

Pour cela, nous nous mettons en place une méthode de co-factorisation de matrices (McF, *matrix co-factorization* en anglais) [FS11 ; WB11] permettant de factoriser conjointement plusieurs matrices (correspondant à plusieurs modalités) tout en ajoutant des contraintes sur les matrices de préférences ou d'attributs. Dans ce chapitre nous introduisons un modèle McF fondé sur la divergence KL (liée à la loi Poisson) qui permet de prendre en compte les phénomènes d'échelle pour chacune des modalités. Contrairement aux chapitres précédents, nous adaptons un point de vue fréquentiste et présentons notre modèle sous la forme d'un problème d'optimisation. Le cœur de ce chapitre étant d'étudier les méthodes de McF, nous choisissons de travailler avec des données binarisées afin de simplifier le problème.

Le reste du chapitre est organisé comme suit. Dans la Section 5.2, nous présentons une revue de la littérature sur les techniques de co-factorisation. Dans la Section 5.3, nous présentons notre modèle de co-factorisation sous la forme d'un problème d'optimisation. Dans la Section 5.5, nous fournissons un algorithme MM permettant d'estimer les différents paramètres du modèle. Dans la Section 5.6, nous testons ce modèle sur des tâches de recommandation musicale et d'attribution automatique de tags [Eck+08]. Enfin, dans la Section 5.7, nous concluons ce chapitre et discutons des perspectives liées à ce travail.

5.2. Co-factorisation de matrices

Un moyen de résoudre le problème de démarrage à froid est d'introduire de nouvelles modalités [FS11 ; GCB14 ; LZE15]. Ainsi, les méthodes de co-factorisation ont été développées pour factoriser conjointement plusieurs matrices d'observation (correspondant à plusieurs modalités). Nous nous restreignons ici au cas où deux modalités sont disponibles. Nous cherchons les approximations de rang faible $\mathbf{Y}_A \approx \mathbf{W}_A\mathbf{H}_A^T$ et $\mathbf{Y}_B \approx \mathbf{W}_B\mathbf{H}_B^T$, et lions les matrices d'attributs $\mathbf{H}_A \approx \mathbf{H}_B$ afin d'en exploiter l'information mutuelle.

5.2.1. Co-factorisation stricte

La co-factorisation stricte [FS11 ; Sei+14] suppose que le lien entre les attributs se traduit par une contrainte d'égalité : $\mathbf{H}_A = \mathbf{H}_B = \mathbf{H}$. Cela revient à concaténer les observations $\mathbf{Y}_A$ et $\mathbf{Y}_B$, ainsi que les dictionnaires $\mathbf{W}_A$ et $\mathbf{W}_B$. le problème d'optimisation revient alors à minimiser la fonction de coût :

$$C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{H}) = \mathrm{KL}(\mathbf{Y}_A|\mathbf{W}_A\mathbf{H}^T) + \gamma\,\mathrm{KL}(\mathbf{Y}_B|\mathbf{W}_B\mathbf{H}^T) \tag{5.1}$$

$$= \mathrm{KL}\left(\begin{pmatrix}\mathbf{Y}_A\\ \gamma\mathbf{Y}_B\end{pmatrix} \middle| \begin{pmatrix}\mathbf{W}_A\\ \gamma\mathbf{W}_B\end{pmatrix}\mathbf{H}^T\right), \tag{5.2}$$

où $\gamma \in \mathbb{R}^+$ est un hyper-paramètre de pondération.

5.2.2. Co-factorisation souple

La co-factorisation souple [Sei+14] relâche la contrainte d'égalité en la remplaçant par l'ajout d'une pénalité, contrôlée par un hyper-paramètre $\delta \in \mathbb{R}^+$ qui règle la pondération :

$$\begin{aligned} C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{H}) = &\,\mathrm{KL}(\mathbf{Y}_A|\mathbf{W}_A(\mathbf{H}_A)^T) + \gamma\,\mathrm{KL}(\mathbf{Y}_B|\mathbf{W}_B(\mathbf{H}_B)^T) \\ &+ \delta D(\mathbf{H}_A, \mathbf{H}_B). \end{aligned} \tag{5.3}$$

Un choix possible pour la pénalité est la norme ℓ_1, i.e., $D(\mathbf{H}_A, \mathbf{H}_B) = \|\mathbf{H}_A - \mathbf{H}_B\|_1$. Elle est particulièrement adaptée lorsque les modalités partagent les mêmes activations, excepté en quelques endroits parcimonieux où elles peuvent différer significativement.

5.2.3. Co-factorisation bayésienne

Des formulations bayésiennes des problèmes de co-factorisation ont aussi été développées. Ces modèles bayésiens font appels à une variable latente dite de compensation (*offset* en anglais) [WB11 ; GCB14]. Le lien entre les attributs est alors donné par :

$$h_{ik}^B = h_{ik}^A + \varepsilon_{ik}, \tag{5.4}$$

où $\varepsilon_{ik} \in \mathbb{R}_+$ est la variable latente de compensation.

En particulier dans [GCB14], un modèle de co-factorisation est développé à partir de la PF, avec $\varepsilon_{ik} \sim \text{Gamma}(\alpha, \beta)$. Ce choix est motivé par la conjugaison de la loi gamma par rapport à la loi Poisson. Il est important de noter que ce modèle n'est pas symétrique par rapport aux matrices $\mathbf{H}_A$ et $\mathbf{H}_B$ puisque $h_{ik}^B \geq h_{ik}^A$ par construction. De ce fait, il ne peut résoudre le problème de démarrage à froid que pour la modalité A, et pas pour la modalité B.

5.3. Modèle de co-factorisation de matrices

Comme nous l'avons évoqué en introduction, nous nous intéressons au cas où la modalité A correspond aux comptes d'écoute de U utilisateurs sur I chansons, et où la modalité B correspond aux tags assignés à chacune de ces I chansons (parmi un ensemble de V tags). $\mathbf{W}_A$ est la matrice des préférences des utilisateurs, $\mathbf{W}_B$ est la matrice des dictionnaires de tags et $\mathbf{H}_A$ et $\mathbf{H}_B$ sont les attributs des chansons pour chacune des modalités.

5.3.1. Lien entre les attributs

Nous proposons de lier les attributs de la manière suivante :

$$\mathbf{H}_A = \mathbf{D}\mathbf{N}_A, \ \mathbf{H}_B = \mathbf{D}\mathbf{N}_B, \tag{5.5}$$

où $\mathbf{D} \geq 0$ est une matrice commune de taille $I \times K$ telle que chacune de ses lignes somme à 1, et $\mathbf{N}_A \geq 0$ et $\mathbf{N}_B \geq 0$ sont deux matrices diagonales de tailles $I \times I$. On note par n_i^A et n_i^B les éléments de la diagonale des matrices $\mathbf{N}_A$ et $\mathbf{N}_B$. Les vecteurs lignes des matrices $\mathbf{H}_A$, $\mathbf{H}_B$ et $\mathbf{D}$ sont notés $\mathbf{h}_i^A$, $\mathbf{h}_i^B$ et $\mathbf{d}_i$, et leurs coefficients sont notés h_{ik}^A, h_{ik}^B et d_{ik}.

Ainsi lorsque $n_i^A > 0$ et $n_i^B > 0$, le lien présenté ci-dessus correspond à une contrainte d'égalité sur les vecteurs d'attributs normalisés, i.e, pour chaque article $i \in \{1, \ldots, I\}$, on a

l'égalité :

$$\frac{\mathbf{h}_i^A}{n_i^A} = \frac{\mathbf{h}_i^B}{n_i^B} = \mathbf{d}_i. \tag{5.6}$$

Ainsi, dans notre modèle, chaque chanson $i \in \{1, \dots, I\}$ est représentée par trois variables :

— Un vecteur normalisé des attributs $\mathbf{d}_i$ sommant à 1. Cette information est partagée pour les deux modalités. C'est elle qui va pouvoir discriminer les différentes chansons du catalogue. On s'attend par exemple à ce que les K facteurs latents soient reliés à des informations sur le genre de la chanson (rock, rap, classique, métal, etc.). Le fait que ce vecteur soit contraint à sommer à 1 permet de retirer l'effet d'échelle.

— Un paramètre d'échelle n_i^A lié à la modalité A (écoutes d'utilisateurs). Ce paramètre permet de capter l'information de popularité liée à la chanson (voir Section 1.4). Plus l'audience liée à cette chanson est grande, plus ce paramètre augmente. En particulier, une nouvelle chanson qui ne possède pas encore d'audience (aucun utilisateur ne l'a encore écoutée) a un paramètre d'échelle nul, i.e., $n_i^A = 0$.

— Un paramètre d'échelle n_i^B lié à la modalité B (tags). Similairement au paramètre n_i^A, ce paramètre permet de modéliser la quantité de tags associée à une chanson. Plus une chanson est labellisée, plus ce paramètre augmente. À l'inverse, une chanson ne possédant aucun tag a un paramètre d'échelle nul, i.e., $n_i^B = 0$.

Deux chansons $i \neq j$ peuvent partager les mêmes motifs d'attributs $\mathbf{d}_i = \mathbf{d}_j$ mais posséder des paramètres d'échelle différents. Par exemple, la première chanson peut être très populaire et rencontrer une large audience, i.e., $n_i^A \gg 0$, mais n'avoir été que peu décrite par des tags, i.e., $n_i^B \approx 0$. De la même façon, la seconde chanson peut être peu populaire (parce qu'elle est nouvelle ou mal reçue), i.e., $n_j^A \approx 0$, mais avoir beaucoup de tags associés, i.e., $n_j^B \gg 0$.

Il est important de noter que ce modèle est parfaitement symétrique par rapport aux modalités A et B. Il permet donc de résoudre à la fois le problème de démarrage à froid pour la recommandation de chansons à des utilisateurs et la labellisation de tags.

5.3.2. Fonction de coût

Le problème d'optimisation de ce modèle de factorisation, illustré en Figure 5.1, consiste à minimiser la fonction de coût :

$$C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{D}, \mathbf{N}_A, \mathbf{N}_B) = \mathrm{KL}(\mathbf{Y}_A | \mathbf{W}_A \mathbf{D}^T \mathbf{N}_A) + \gamma\, \mathrm{KL}(\mathbf{Y}_B | \mathbf{W}_B \mathbf{D}^T \mathbf{N}_B), \tag{5.7}$$

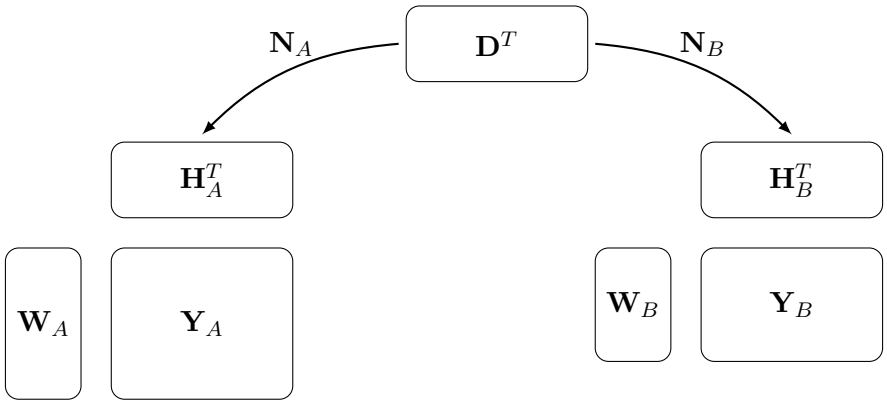


FIGURE 5.1. – Illustration du modèle de co-factorisation.

s.c. $\mathbf{W}_A \geq 0$, $\mathbf{W}_B \geq 0$, et où $\mathbf{D}$, $\mathbf{N}_A$ et $\mathbf{N}_B$ sont telles que décrites en Section 5.3.1. On note $\mathbf{Z} = \{\mathbf{W}_A, \mathbf{W}_B, \mathbf{N}_A, \mathbf{N}_B, \mathbf{D}\}$ l'ensemble des paramètres à estimer.

Invariances d'échelle. Nous avons deux invariances par changement d'échelle liées à la fonction de coût décrite ci-dessus.

— Soit $\Theta = \text{diag}((\theta_1, \ldots, \theta_I))$ une matrice diagonale de taille $I \times I$ ayant des coefficients strictement positifs. Il est aisé de montrer l'invariance par changement d'échelle suivante :

$$C(\mathbf{W}_A, \mathbf{W}_B, \Theta^{-1}\mathbf{D}, \mathbf{N}_A\Theta, \mathbf{N}_B\Theta) = C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{D}, \mathbf{N}_A, \mathbf{N}_B). \tag{5.8}$$

Cette invariance nous permet d'imposer la contrainte de somme à 1 sur $\mathbf{D}$ par une simple étape de renormalisation (voir Section 5.5).

— Nous avons une seconde invariance dans ce modèle. Soit $\Lambda = \text{diag}((\lambda_1, \ldots, \lambda_K))$ une matrice diagonale de taille $K \times K$ ayant des coefficients strictement positifs. Il est aisé de montrer l'invariance par changement d'échelle suivante :

$$C(\mathbf{W}_A\Lambda^{-1}, \mathbf{W}_B\Lambda^{-1}, \mathbf{D}\Lambda, \mathbf{N}_A, \mathbf{N}_B) = C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{D}, \mathbf{N}_A, \mathbf{N}_B). \tag{5.9}$$

En pratique, cette invariance ne pose pas de problème de dégénérescence et nous ne procédons donc pas à une étape de renormalisation.

Ces deux invariances par changement d'échelle peuvent être résumées dans l'équation suivante :

$$C(\mathbf{W}_A\Lambda^{-1}, \mathbf{W}_B\Lambda^{-1}, \Theta^{-1}\mathbf{D}\Lambda, \mathbf{N}_A\Theta, \mathbf{N}_B\Theta) = C(\mathbf{W}_A, \mathbf{W}_B, \mathbf{D}, \mathbf{N}_A, \mathbf{N}_B). \tag{5.10}$$

On note $\bar{\mathbf{D}} = \Theta^{-1}\mathbf{D}\Lambda$, $\bar{\mathbf{W}}_A = \mathbf{W}_A\Lambda^{-1}$,$\bar{\mathbf{W}}_A = \mathbf{W}_B\Lambda^{-1}$,$\bar{\mathbf{N}}_A = \mathbf{N}_A\Theta$ et $\bar{\mathbf{N}}_B = \mathbf{N}_B\Theta$. On remarque que la matrice $\bar{\mathbf{D}}$ respecte la contrainte de somme à 1 si et seulement si $\theta_i = \sum_k d_{ik}\lambda_k$. Nous discuterons à nouveau de cette invariance dans la section suivante.

5.4. Tâches de recommandation

Jusqu'à maintenant, nous nous sommes intéressés à la recommandation de chansons déjà écoutées par des utilisateurs (nous utiliserons le terme anglais de *in-matrix recommendation*). Ici, nous voulons aussi pouvoir faire des recommandation sur des chansons pour lesquelles une des modalités est manquante, nous les appelons «recommandations à froid» (*cold-start* ou *out-matrix recommendation* en anglais).

5.4.1. *In-matrix recommendation*

Pour cette tâche de recommandation, nous utilisons le même score qu'habituellement pour trier les chansons et qui est donné par :

$$s^A_{ui} = \sum_k w^A_{uk} h^A_{ik}. \tag{5.11}$$

Ce score a les mêmes invariances que la fonction de coût décrite en Éq. (5.9). En effet, en utilisant les notations de l'Éq (5.10), on a bien $\bar{s}^A_{ui} = \sum_k \bar{w}^A_{uk}\bar{h}^A_{ik} = s^A_{ui}$.

5.4.2. Recommandation à froid

Dans cette section on prendra l'exemple d'un problème de démarrage à froid pour la modalité A, i.e., pour des chansons qui n'ont pas encore été écoutées mais qui ont des tags qui leur sont associés. Les remarques suivantes sont aussi valables pour la modalité B (le problème est symétrique).

Pour les recommandations à froid, on définit le score suivant :

$$f_{ui}^A = \sum_k w_{uk}^A d_{ik}. \tag{5.12}$$

Contrairement à la section précédente, on utilise la matrice $\mathbf{D}$ et non pas la matrice $\mathbf{H}_A = \mathbf{N}_A\mathbf{D}$, puisque le paramètre d'intensité associé à ses chansons est nul, i.e., $n_i^A = 0$ et donc $s_{ui}^A = 0$.

Il est important de noter que le score f_{ui}^A et la fonction de coût C n'ont pas les mêmes invariances d'échelle. En effet, en utilisant les notations de l'Éq (5.10), on a :

$$\bar{f}_{ui}^A = \sum_k \bar{w}_{uk}^A \bar{d}_{ik} = \sum_k w_{uk}^A \theta_i^{-1} d_{ik} = \theta_i^{-1} f_{ui}^A, \tag{5.13}$$

avec $\theta_i = \theta_i = \sum_k d_{ik}\lambda_k$ où $(\lambda_1, \ldots, \lambda_K)$ sont des scalaires arbitraires. Cela signifie que nous ne pouvons pas établir de liste personnalisée de recommandations pour les utilisateurs à partir de ce score, puisque l'ordre de la liste dépendrait des paramètres $(\lambda_1, \ldots, \lambda_K)$.

De ce fait, pour proprement évaluer les recommandations à froid, nous devons le faire pour une chanson fixée et proposer une liste ordonnée d'utilisateurs ou de tags. Pour la modalité A, cela revient à répondre à la question : «Quels utilisateurs doit-on avertir de la sortie d'une nouvelle chanson ?», alors que pour la modalité B, cela revient à répondre à : «Quels tags peut-on associer à une chanson qui n'en a pas encore ?»

5.5. Estimation par majoration-minimisation

La fonction de coût C n'a pas de minimum disponible en forme analytique et n'est pas convexe. Comme en Section 2.3.2, on utilise un algorithme MM [FI11] pour atteindre un minimum local.

Notre fonction de coût correspond à une somme de deux divergences KL. Les terme de la forme $\log(\sum_i x_i)$ peuvent être majorés en utilisant une inégalité de type Jensen [FI11]. On obtient une majorante de la fonction de coût définie par :

$$\begin{aligned} G(\mathbf{Z}|\tilde{\mathbf{D}}, \tilde{\mathbf{W}}_A, \tilde{\mathbf{W}}_B) = &\sum_{uik} \left[-c_{uik}^A \log(w_{uk}^A n_i^A d_{ik}) + w_{uk}^A n_i^A d_{ik} \right] \\ &+ \gamma \sum_{vik} \left[-c_{vik}^B \log(w_{vk}^B n_i^B d_{ik}) + w_{vk}^B n_i^B d_{ik} \right] + cste, \end{aligned}$$

Algorithme 8 : Algorithme MM pour la co-NMF

Données : Matrices d'observation $\mathbf{Y}_A$ et $\mathbf{Y}_B$.
Résultat : Matrices $\mathbf{W}_A$, $\mathbf{W}_B$, $\mathbf{H}_A$, $\mathbf{H}_B$.

Initialisation des matrices $\mathbf{W}_A$, $\mathbf{W}_B$, $\mathbf{D}$, $\mathbf{N}_A$, $\mathbf{N}_B$;
répéter
  pour *chaque couple* (u,i) *tel que* $y^A_{ui} > 0$ *et chaque couple* (v,i) *tel que* $y^B_{vi} > 0$ **faire**
    Calculer c^A_{uik} tel que décrit en Éq. (5.14) ;
    Calculer c^B_{vik} tel que décrit en Éq. (5.15) ;
  fin
  pour *chaque utilisateur* $u \in \{1,\dots,U\}$ *et chaque tag* $v \in \{1,\dots,V\}$ **faire**
    $w^A_{uk} = \dfrac{\sum_i c^A_{uik}}{\sum_i n^A_i d_{ik}}$; $w^B_{vk} = \dfrac{\sum_i c^B_{vik}}{\sum_i n^B_i d_{ik}}$;
  fin
  pour *chaque chanson* $i \in \{1,\dots,I\}$ **faire**
    $n^A_i = \dfrac{\sum_u y^A_{ui}}{\sum_{uk} w^A_{uk} d_{ik}}$; $n^B_i = \dfrac{\sum_v y^B_{vi}}{\sum_{vk} w^B_{vk} d_{ik}}$;
    $d_{ik} = \dfrac{\sum_u c^A_{uik} + \gamma \sum_v c^B_{vik}}{n^A_i \sum_u w^A_{uk} + \gamma n^B_i \sum_v w^B_{vk}}$;
    Renormalisation :
    $\theta_i = \sum_k d_{ik}/K$;
    $\mathbf{d}_i \leftarrow \mathbf{d}_i \theta_i^{-1}$, $n^A_i \leftarrow n^A_i \theta_i$, $n^B_i \leftarrow n^B_i \theta_i$.
  fin
jusqu'à *C converge*;

où *cste* est une constante par rapport à nos paramètres d'intérêt $\mathbf{Z}$, et où c^A_{uik} et c^B_{uik} sont définis par :

$$\phi^A_{uik} = \frac{\tilde{w}^A_{uk}\tilde{d}_{ik}}{\sum_k \tilde{w}^A_{uk}\tilde{d}_{ik}}, \quad c^A_{uik} = y^A_{ui}\phi^A_{uik}, \tag{5.14}$$

$$\phi^B_{vik} = \frac{\tilde{w}^B_{vk}\tilde{d}_{ik}}{\sum_k \tilde{w}^B_{vk}\tilde{d}_{ik}}, \quad c^B_{vik} = y^B_{vi}\phi^B_{vik}. \tag{5.15}$$

La fonction auxiliaire G peut être minimisée en utilisant un algorithme de descente par bloc. À chaque itération, on minimise la fonction G par rapport à une des variables latentes, en laissant toutes les autres variables fixées. En procédant de la sorte on obtient les règles de mise à jour décrites dans l'Algorithme 8.

Comme discuté en Section 5.3.2, nous appliquons une étape de renormalisation à la fin de chaque itération pour respecter la contrainte liée à la variable $\mathbf{D}$.

Comme pour les algorithmes de PF, notre algorithme ne traite que les valeurs non nulles de chaque modalité durant les mises à jour des variables c^A_{uik} et c^B_{uik}. Par conséquent, notre algorithme passe à l'échelle et est particulièrement adapté aux larges jeux de données parcimonieux, comme c'est le cas en recommandation (voir Tableau 5.1). L'algorithme est stoppé lorsque que l'incrément relatif de la fonction de coût entre deux itérations consécutives passe sous le seuil $\tau = 10^{-5}$.

5.6. Résultats expérimentaux

5.6.1. Protocole expérimental

Jeux de données. Nous utilisons deux jeux de données extraits du *Million Song Dataset* (MSD) [Ber+11] et les fusionnons selon les chansons :

— Comme dans les chapitres précédents, nous utilisons le jeu de données Taste Profile [McF+12] que nous pré-traitons les donnons pour ne sélectionner que des utilisateurs et des chansons qui détiennent plus de 20 interactions.

— Le jeu de données Last.fm contient des tags associés à environ 500k chansons. Ces tags ont été extraits via l'API Last.fm [Cel10]. Ces tags proviennent d'annotations d'utilisateurs et sont donc assez bruités. Nous pré-traitons ces données en ne sélectionnant que les 1 000 tags les plus utilisés. Pour chaque paire chanson-tag, une valeur entre 0 et 100 indique la confiance liée au tag, nous gardons seulement les tags qui ont une confiance plus grande que 10. Les 10 tags les plus rencontrés après le pré-traitement sont affichés dans le Tableau 5.2.

Dans ce chapitre, nous utilisons la divergence de KL qui est associée à la distribution Poisson. Comme nous l'avons vu dans la Section 1.4, cette loi n'est pas adaptée aux données sur-dispersées que nous rencontrons ici. Ainsi, nous binarisons les deux jeux de données de telle sorte que $\mathbf{Y}_A \in \{0,1\}^{U\times I}$ et $\mathbf{Y}_B \in \{0,1\}^{V\times I}$. La structure finale des deux jeux de données après pré-traitement est décrite dans le Tableau 5.1.

Méthodes comparatives. Pour chaque expérience, on compare les performances de notre modèle, noté S-coNMF (pour *scale-free non-negative matrix co-factorization* en anglais) avec deux autres méthodes :

— KL-NMF, présenté en Section 1.4. Cette méthode souffre du problème de démarrage à froid et ne peut donc être utilisée que pour des tâches de *in-matrix recommendation*.

Tableau 5.1. – Structure des jeux de données correspondant aux deux modalités.

	TP	**Last.fm**
Nombre de chansons I	15 667	15 667
Nombre de lignes (U ou V)	16 203	620
Nombre de valeurs non nulles	792 761	128 652
% de valeurs non nulles	0.31%	1.32%

Tableau 5.2. – Occurrences (Occ.) des 10 tags les plus utilisés dans le jeu de données Last.fm après pré-traitement.

Tags	**Occ.**	**Tags**	**Occ.**
rock	6 703	electronic	2 413
alternative	4 949	female vocalists	2 407
indie	4 151	indie rock	2 171
pop	3 853	Love	1 875
alternative rock	2 854	singer-songwriter	1 786

— La co-factorisation stricte (H-coNMF), présentée en Section 5.2.1 et qui utilise l'algorithme de KL-NMF sur les données concaténées. Pour les recommandations à froid, nous utilisons une matrice de masque qui indique quelles sont les données manquantes. La fonction de coût est donc :

$$C(\mathbf{W}, \mathbf{H}) = \mathrm{KL}(\mathbf{M} \odot \mathbf{Y} | \mathbf{M} \odot \mathbf{W}\mathbf{H}^T), \tag{5.16}$$

où $\odot$ est la multiplication terme à terme de matrices, et $\mathbf{M}$ est la matrice de masque.

Pour chaque méthode, on choisit $K = 100$ facteurs latents. L'hyper-paramètre de pondération est fixé à $\gamma = \frac{U}{V}$, de sorte qu'il compense la différence de taille entre les deux jeux de données ($V \ll U$).

Évaluation des listes de recommandations. Comme nous considérons que les matrices d'observation $\mathbf{Y}_A$ et $\mathbf{Y}_B$ sont binaires, les évaluations des listes de recommandations se font donc à l'aide de la métrique NDCG0 (voir Section 1.1.3). Cependant, nous rapportons les scores NDCG pour différentes tailles de listes $L \in \{1, 10, 20, 100, I\}$ suivant les expériences. Nous notons NDCG@L le score NDCG0 dépendant de la taille L des listes de

Tableau 5.3. – Performance des trois modèles : S-coNMF, H-coNMF, KL-NMF, pour trois tâches différentes : recommandation à froid (OUT-A), annotation automatique de tags (OUT-B), *in-matrix recommendation* (IN-A). Chaque algorithme est exécuté 5 fois, la moyenne et la variance des métriques NDCG sont affichées.

Expérience	OUT-A		OUT-B		IN-A	
Score	NDCG@20	NDCG@200	NDCG@1	NDCG@10	NDCG@100	NDCG@I
S-coNMF	0.082 $(1 \cdot 10^{-5})$	0.122 $(1 \cdot 10^{-5})$	**0.416** $(6 \cdot 10^{-4})$	**0.266** $(2 \cdot 10^{-4})$	0.129 $(4 \cdot 10^{-6})$	0.286 $(3 \cdot 10^{-6})$
H-coNMF	**0.087** $(1 \cdot 10^{-5})$	**0.131** $(2 \cdot 10^{-5})$	0.391 $(1 \cdot 10^{-4})$	0.264 $(1 \cdot 10^{-4})$	0.122 $(6 \cdot 10^{-6})$	0.283 $(3 \cdot 10^{-6})$
KL-NMF	.	.	.	.	**0.163** $(5 \cdot 10^{-7})$	**0.313** $(2 \cdot 10^{-7})$

recommandations.

5.6.2. Recommandation avec démarrage à froid

Dans cette section, on évalue les tâches de recommandation à froid pour les modalités A et B. Pour cela, on remplace artificiellement des colonnes de $\mathbf{Y}_A$ et de $\mathbf{Y}_B$ par des colonnes de zéros. Ainsi nous créons les ensembles d'entraînement $\mathbf{Y}_A^{\text{train}}$ et $\mathbf{Y}_B^{\text{train}}$ de telle sorte à ce que 10% des chansons ont uniquement des informations de comptes d'écoute, 10% des chansons ont uniquement des informations de tags, et 80% ont les deux informations. Les colonnes retirées forment les ensembles de test $\mathbf{Y}_A^{\text{test}}$ et $\mathbf{Y}_B^{\text{test}}$.

Pour chaque chanson qui n'a jamais été écoutée, nous proposons une liste ordonnée d'utilisateurs qui pourraient l'apprécier (voir Section 5.4). Nous entraînons donc l'algorithme sur les bases $\mathbf{Y}_A^{\text{train}}$ et $\mathbf{Y}_B^{\text{train}}$. Pour chaque chanson, nous créons une liste ordonnée d'utilisateurs à partir du score présenté en Section 5.4. Nous évaluons la qualité de cette liste avec la métrique NDCG@L avec $L \in \{20, 200\}$. De la même façon, pour chaque chanson qui n'a aucune annotation, nous proposons une liste ordonnée de tags qui pourraient la décrire, et l'évaluons avec la métrique NDCG@L avec $L \in \{1, 10\}$.

Les colonnes OUT-A et OUT-B du Tableau 5.3 présentent les résultats des modèles S-coNMF et H-coNMF sur ces deux problèmes de recommandation à froid. Pour recommander de potentiels auditeurs (OUT-A), H-coNMF semble être sensiblement meilleure que notre méthode. Cependant, S-coNMF surpasse H-coNMF sur la tâche d'étiquetage. S-coNMF

présente un taux de réussite de 42% sur la prédiction du premier tag. Ceci est un bon résultat au regard du bruit présent dans ce jeu de données. En effet, les chansons n'ont pas été annotées par des experts mais par des utilisateurs, ce qui explique certaines incohérences dans les tags utilisés. Plus de détails sur les annotations sont donnés en Section 5.6.4.

5.6.3. Recommandation sans démarrage à froid

On peut aussi évaluer notre algorithme sur des tâches plus classiques de recommandation (*in-matrix recommendation*) comme lors des précédents chapitres. Le but est de recommander des listes personnalisées de chansons aux utilisateurs. Puisque l'on ne traite plus ici du problème de démarrage à froid, l'algorithme KL-NMF peut être aussi évalué sur cette tâche.

Nous appliquons le même protocole expérimental que dans le Chapitre 2 et retirons artificiellement 20% des valeurs non nulles de la modalité A. Ainsi, nous construisons deux ensembles $\mathbf{Y}_A^{\text{train}}$ et $\mathbf{Y}_A^{\text{test}}$. Pour chaque utilisateur, une liste de chansons est proposée à partir du score défini en Section 5.4.1, parmi les chansons qu'il n'a pas encore écoutées. Nous évaluons la qualité de cette liste avec la métrique NDCG@L avec $L \in \{100, I\}$.

Les résultats sont présentés dans la troisième colonne (IN-A) du Tableau 5.3. S-coNMF est sensiblement meilleur que H-coNMF, mais nous observons que KL-NMF obtient les meilleures performances sur cette tâche. Ce n'est pas totalement surprenant, l'ajout d'une nouvelle modalité (ici des tags) peut être vu comme un terme de régularisation sur les attributs $\mathbf{H}_A$. Nous perdons donc en précision dans la tâche de recommandation pour les utilisateurs mais résolvons le problème de démarrage à froid. Ce compromis peut être réglé avec le paramètre de pondération γ.

5.6.4. Analyse exploratoire : prédiction de tags

Dans le Tableau 5.4, nous présentons les attributs obtenus pour trois facteurs latents.

— Dans la première colonne du tableau, nous affichons les tags correspondant aux 5 plus grandes valeurs des trois colonnes de $\mathbf{W}_B$ associées à ces facteurs.

— Dans la deuxième colonne du tableau, nous affichons les chansons correspondant aux 5 plus grandes valeurs des trois colonnes de $\mathbf{H}_A = \mathbf{N}_A\mathbf{D}$ associées à ces facteurs.

— Dans la troisième colonne du tableau, nous affichons les chansons correspondant aux 5 plus grandes valeurs des trois colonnes $\mathbf{D}$ associées à ces facteurs.

Les tags associés à chaque facteur sont consistants, par exemple, des genres comme «new wave» et «post-punk» appartiennent au même facteur latent. Le modèle est aussi robuste

aux différentes orthographes utilisées par les utilisateurs («post-punk» et «Post punk» par exemple). Nous constatons que les chansons associées à chaque facteur sont bien reliées avec les tags associés. Eminem, 50 Cent et The Notorious B.I.G. sont des artistes de rap. The Cure, The Smiths et Joy Division sont des groupes de new wave. TV On The Radio, The Mars Volta et Animal Collective sont des groupes de rock expérimental. Finalement, la popularité des chansons $\mathbf{N}_A$ a une forte influence sur la diversité de chaque facteur. En effet, lorsque cette notion est retirée (dernière colonne du tableau), des chansons et des groupes moins populaires apparaissent en tête de chaque facteur.

5.7. Discussion

Dans ce chapitre, nous avons proposé un modèle de McF où chaque modalité dispose de ses propres paramètres d'échelle. L'inférence de ce modèle symétrique nous permet de résoudre le problème de démarrage à froid à la fois pour la modalité A et la modalité B. Ainsi, nous pouvons recommander des utilisateurs à de nouvelles chansons à partir leurs tags associés, et nous pouvons recommander des tags pertinents pour les chansons à partir de leur audience. Nous détaillons des perspectives liées à ce travail dans ce qui suit.

Co-factorisation bayésienne. Une première perspective serait donc d'adopter un point de vue bayésien en imposant des lois a priori aux matrices $\mathbf{D}$, $\mathbf{N}_A$ et $\mathbf{N}_B$. Par exemple, on pourrait imaginer l'ensemble de lois a priori suivant :

$$\mathbf{d}_i \sim \mathrm{Dir}(\alpha), \tag{5.17}$$

$$h_i^A = n_i^A \mathbf{d}_i \ ; \ n_i^A \sim \mathrm{Gamma}(K\alpha, \beta), \tag{5.18}$$

$$h_i^B = n_i^B \mathbf{d}_i \ ; \ n_i^B \sim \mathrm{Gamma}(K\alpha, \beta). \tag{5.19}$$

De cette façon, les distributions marginales des matrices $\mathbf{W}$ et $\mathbf{H}$ seraient encore une fois des lois gamma, i.e., $h_i^A \sim \mathrm{Gamma}(\alpha, \beta)$ et $h_i^B \sim \mathrm{Gamma}(\alpha, \beta)$.

Extension aux données non binaires. Une deuxième perspective serait de combiner le modèle S-coNMF avec les modèles étudiés en Chapitres 2, 3 ou 4, afin d'éviter l'étape de binarisation que nous avons appliquée aux modalités A et B. Avec la dcPF, on pourrait apprendre le paramètre naturel de la distribution élémentaire pour chacune des deux modalités. Ainsi, le modèle s'adapterait à la qualité des informations détenues dans chacune des modalités.

Rang de la factorisation pour chaque modalité. Dans ce chapitre, nous avons supposé que le rang des factorisations de matrices était le même pour les modalités A et B. Cependant, les matrices observées peuvent présenter des dynamiques différentes selon les modalités. Par conséquent, une dernière perspective pourrait être de développer un modèle capable de s'adapter à des rangs de factorisation différents, par le biais de méthodes de tri-factorisation par exemple [YC09].

Tableau 5.4. – Trois exemples de facteurs latents, avec, pour chacun d'entres eux, les 5 meilleurs tags associés, et les 5 meilleures chansons associées, avec ou sans la notion de popularité.

Meilleurs tags	**Meilleures chansons selon $\mathbf{H}_A$**	**Meilleures chansons selon D**
«Hip-Hop»	Eminem - «Mockingbird»	DMX - «Where The Hood At»
«hip hop»	Eminem - «Without Me»	Lil Jon - «Crunk Juice»
«classic»	Kid Cudi - «Day 'N' Nite»	50 Cent - «Straight To The Bank»
«rap»	Kid Cudi - «Up Up & Away»	Eminem - «The Kiss»
«Gangsta Rap»	Kid Cudi - «Cudi Zone»	The Notorious B.I.G. - «Respect»
«new wave»	The Cure - «Boys Don't Cry»	New Order - «The Perfect Kiss»
«post-punk»	The Smiths - «There Is A Light [...]»	Talking Heads - «Burning Down The House»
«Guilty Pleasures»	The Smiths - «This Charming Man»	Joy Division - «Disorder»
«intense»	The Smiths - «What Difference Does It Make ?»	Tears For Fears - «Goodnight Song»
«Post punk»	Wolfsheim - «Once In A Lifetime»	The Smiths - «Miserable Lie»
«experimental»	Animal Collective - «Fireworks»	The Mars Volta - «Tira Me a Las Aranas»
«Experimental Rock»	Sigur Ros - «Staralfur»	Cocorosie - «Gallows»
«Avant-Garde»	Sonic Youth - «Youth Against Fascism»	The Mars Volta - «Concertina»
«noise»	Grizzly Bear - «Little Brother»	The Mars Volta - «Roulette Dares»
«weird»	TV On The Radio - «Crying»	TV On The Radio - «Golden Age»

Conclusion

Dans cette thèse, nous nous sommes intéressés à des méthodes bayésiennes de NMF pour les données de comptage sur-dispersées rencontrées en CF. La PF est une variante de la NMF adaptée pour les données de comptage. Comme nous l'avons montré dans le Chapitre 1, elle ne permet pas de modéliser la sur-dispersion des données. Une étape de binarisation des données est un moyen efficace de contourner ce problème. Cependant, ce pré-traitement entraîne une perte d'information puisque la valeur des comptes est alors ignorée.

Dans les Chapitres 2, 3 et 4 de ce manuscrit, nous avons donc proposé des méthodes de NMF probabilistes permettant de mieux modéliser les données brutes et de minimiser l'effet des pré-traitements. La Figure 6.1 présente une taxonomie des modèles proposés dans ces chapitres et des liens qui les unissent.

Dans le Chapitre 2, nous avons développé la factorisation binomiale négative (NBF) qui est une extension de la PF. Nous avons montré que la NBF était capable de prendre en compte la sur-dispersion des données par le biais d'une variable dite d'exposition. Cependant, l'inférence de cette variable s'est avérée coûteuse pour les problèmes de CF. De plus, nous avons identifié un régime où la variable d'exposition pouvait nuire aux recommandations.

Afin de pallier ces limitations, nous avons proposé dans le Chapitre 3 la dcPF qui est une instance de la cPF [BE16] spécialement conçue pour les données de comptage sur-dispersées. Nous avons notamment montré que la dcPF, qui est contrôlée par le paramètre naturel de sa loi élémentaire, offre un continuum entre les deux versions de la PF : PFbrut et PFbin. De plus, l'algorithme développé pour la dcPF passe à l'échelle sur des données de grande dimension.

Dans le Chapitre 4, nous avons adopté une approche différente des deux chapitres précédents. Au lieu de travailler directement sur les données de comptage sur-dispersées, nous avons eu recours à une quantification des données, plus souple que la binarisation. Cette quantification nous a permis de conserver la relation d'ordre associée aux données en ne se focalisant que sur un nombre réduit de classes les représentant. Nous avons proposé un mo-

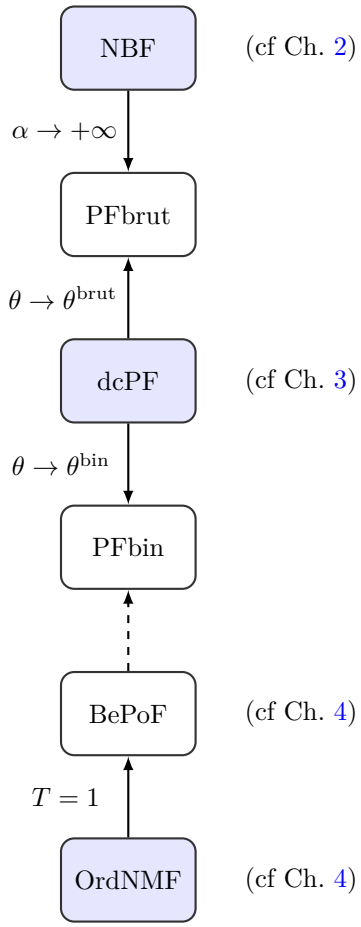


FIGURE 6.1. – Taxonomie des modèles étudiés dans ce manuscrit. Les flèches pleines correspondent à des cas limites, alors que la flèche en pointillé correspond à une approximation.

dèle de NMF spécialement conçu pour ces données quantifiées, appelé OrdNMF. Ce modèle peut être vu comme une nouvelle extension de PFbin permettant d'utiliser un pré-traitement plus fin que l'une binarisation des données. Nous avons utilisé une astuce d'augmentation de modèle afin d'obtenir un algorithme simple et passant à l'échelle. Les résultats de nos expériences confirment l'utilité d'exploiter les valeurs brutes de comptage.

Enfin, dans le Chapitre 5, nous nous sommes focalisés sur une des limites bien connue des techniques de CF : le problème de démarrage à froid. Ce problème peut être résolu grâce à l'introduction de nouvelles informations sur les articles à recommander. Dans ce chapitre, nous nous sommes intéressés à la recommandation musicale, et nous avons choisi d'intégrer des informations d'étiquetage (*tags*) sur les chansons. Nous avons proposé un modèle de co-factorisation de matrices permettant de prendre en compte les phénomènes

d'échelle propre à chaque modalité. Ce modèle entièrement symétrique nous a permis de résoudre le problème de démarrage à froid pour chacune des modalités.

Discussions et perspectives

Autres modélisations. Dans les Chapitres 2 et 3, nous nous sommes efforcés de proposer des méthodes bayésiennes capables de modéliser directement les données de comptage brutes (sans pré-traitement). Néanmoins, nous avons évalué nos modèles sur des tâches de recommandation de listes d'articles en utilisant des métriques de classement. Nos modèles ne sont pas directement optimisés pour ce type de métriques puisqu'ils cherchent à modéliser la valeur de chaque coefficient de la matrice d'observation. Cela semble être un effort trop important pour le type d'évaluation que nous avons choisi. Dans le Chapitre 4, nous avons relâché nos contraintes en acceptant d'utiliser une quantification des données. Nous nous sommes alors focalisés sur la modélisation de la relation d'ordre présente dans les données plutôt que sur la modélisation de leurs valeurs. Ce type de modèle est donc plus flexible et paraît mieux adapté aux métriques de classement. D'autres articles ont proposé proposé d'apprendre à classer (*learning-to-rank*) les articles plutôt que de prédire les retours des utilisateurs associés [Ren+09]. Les travaux du Chapitre 4 prennent donc cette direction et nous pourrions proposer de tels modèles sur la base du modèle OrdNMF présenté. Cela semble une perspective très excitante puisque les algorithmes que nous avons développés sont très simples à mettre en place.

Un autre aspect du travail proposé dans ce manuscrit dont nous pouvons discuter est le choix d'une modélisation probabiliste des données. Les modèles probabilistes nous ont permis de donner un cadre clair à nos travaux. Cependant les modèles d'optimisation semblent plus souples et permettent d'ajouter plus facilement de nouvelles contraintes et régularisations. Par exemple, dans [HKV08], les auteurs proposent d'utiliser les valeurs des retours d'utilisateurs sous la forme de coefficients de pondération, ce que nous ne nous sommes pas autorisés à faire dans cette thèse.

Enfin, nous pouvons évoquer les perspectives qu'offrent les méthodes d'apprentissage profond. Alors que les modèles de MF correspondaient à l'approche dominante en CF depuis le *Netflix prize*, des travaux récents ont commencé à intégrer des techniques d'apprentissage profond dans les systèmes de recommandation [KH17]. L'apprentissage profond possède une communauté très active et les premières applications à des tâches de recommandation semblent prometteuses.

Aspect temporel. Une autre perspective serait de proposer des modèles tenant compte de phénomènes temporels, tels que l'évolution des goûts des utilisateurs au cours du temps par exemple. Dans cette thèse, nous avons uniquement utilisé des données collectées correspondant aux retours d'utilisateurs pendant un laps de temps donné.

Il existe deux manières de prendre en compte les informations temporelles. La première façon est de discrétiser l'axe du temps et de collecter les données à intervalle régulier. La matrice d'observation devient alors un tenseur d'observation, et des lois a priori fondées sur des chaînes de Markov peuvent alors être placées sur les préférences des utilisateurs et les attributs des articles [Cha+15 ; JBE17]. Ainsi, il pourrait être intéressant d'adapter les modèles présentés dans cette thèse à ce type de problème. De plus, plusieurs choix de chaînes de Markov sont possibles et leur étude pourrait améliorer notre compréhension de ces modèles. Une autre manière d'aborder ce problème est d'utiliser des processus Poisson hétérogènes où le temps est continu. Certains modèles fondés sur les cascades de processus Poisson [SJ10] permettent de modéliser l'arrivée par rafale des données. Dans le Chapitre 3, nous avons notamment fait le lien entre la dcPF et ce type de modèle [Du+15 ; Hos+18 ; Kho+18 ; Zho17]. Nous pourrions également étudier ce lien plus en profondeur, et proposer une façon de modéliser les évolutions du comportement des utilisateurs au fil du temps.

Évaluation des systèmes de recommandation. Une des problématiques majeures des systèmes de recommandation concerne la façon d'évaluer les recommandations [Her+04]. Dans cette thèse, nous avons choisi d'évaluer nos modèles sur des tâches de recommandation de listes personnalisées d'articles. Ce choix correspond à des situations réelles, Deezer ou Spotify proposant des playlists hebdomadaires à leurs abonnés par exemple. Le calcul de ces métriques repose sur la notion de pertinence des articles recommandés. Nous avons travaillé avec des jeux de données publiques disponibles en ligne, et procédé à des évaluations dites hors-lignes de nos modèles. Ainsi, le calcul de la pertinence d'un article correspond à sa présence ou non dans un ensemble de test préalablement mis de côté. Ces méthodes d'évaluation hors-lignes récompensent donc davantage la prédiction des futures interactions d'un utilisateur, plutôt que la prédiction de ce que l'utilisateur aurait pu aimer. Cette nuance est importante puisqu'elle définit l'objectif du système de recommandation : anticiper les actions des utilisateurs ou leur proposer de nouveaux articles qu'ils pourraient aimer. Nous avons tenté d'inclure cette nuance dans nos évaluations avec l'introduction d'un seuillage pour définir la pertinence des articles, mais cela semble avoir un impact limité sur les scores. Une des solutions pourrait être d'évaluer les modèles traitant de données implicites pour lesquels il existe une vérité-terrain explicite. De plus, il serait intéressant d'intégrer des

notions de diversité ou de sérendipité dans les modèles de recommandations, sous la forme de régularisations par exemple.

Annexe A.

Dérivations des algorithmes CAVI

Dans cette annexe, nous donnons les éléments-clés des dérivations liées aux algorithmes CAVI développés pour les modèles PF, NBF, dcPF et IG-OrdNMF (sur données implicites quantifiées). Le lecteur est invité à se référer au Tableau A pour la forme des différentes distributions utilisées dans la suite. On rappelle que l'utilisation d'un algorithme CAVI alliée avec l'hypothèse de champ moyen nous donne la solution analytique suivante (voir Section 1.3.3) :

$$q(z_i) \propto \exp\left(\mathbb{E}_{q_{-i}}(\log p(\mathbf{y}, \mathbf{z}))\right). \tag{A.1}$$

A.1. Modèle PF

Dans le modèle PF augmenté, on a noté $\mathbf{Z} = \{\mathbf{C}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes.

Distribution jointe. La densité de la distribution jointe du modèle augmenté de PF peut se décomposer en deux termes :

$$\log p(\mathbf{Y}, \mathbf{C}, \mathbf{W}, \mathbf{H}) = \underbrace{\log p(\mathbf{Y}, \mathbf{C}|\mathbf{W}, \mathbf{H})}_{\text{attache aux données}} + \underbrace{\log p(\mathbf{W}) + \log p(\mathbf{H})}_{\text{a priori}}. \tag{A.2}$$

Le terme d'attache aux données peut s'écrire comme :

$$\log p(\mathbf{Y}, \mathbf{C}|\mathbf{W}, \mathbf{H}) = \sum_{uik} \left[c_{uik} \log(w_{uk} h_{ik}) - w_{uk} h_{ik} - \log c_{uik}!\right], \tag{A.3}$$

avec $y_{ui} = \sum_k c_{uik}$. Le terme de régularisation (correspondant aux lois a priori gamma) s'écrit sous la forme :

$$\log p(\mathbf{W}) = \sum_{uk} \left[(\alpha^W - 1) \log w_{uk} - \beta_u^W w_{uk}\right] + cste. \tag{A.4}$$

Variable C. Ainsi, pour la variable latente $\mathbf{C}$, on a :

$$\log q(\mathbf{C}) = \mathbb{E}_{q_{-\mathbf{C}}}(\log p(\mathbf{Y}, \mathbf{C}, \mathbf{W}, \mathbf{H})) + cste, \tag{A.5}$$

où *cste* est une constante par rapport à $\mathbf{C}$, et $q_{-\mathbf{C}}(\mathbf{W}, \mathbf{H}) = q(\mathbf{W})q(\mathbf{H})$. On obtient donc :

$$\begin{aligned}
\mathbb{E}_{q_{-\mathbf{C}}}(\log p(\mathbf{Y}, \mathbf{C}, \mathbf{W}, \mathbf{H})) &= \sum_{uik} \left[c_{uik}\mathbb{E}_q(\log(w_{uk}h_{ik})) - \log c_{uik}!\right] + cste && \text{(A.6)}\\
&= \sum_{uik} \left[c_{uik} \log \underbrace{\exp \mathbb{E}_q(\log(w_{uk}h_{ik}))}_{=\Lambda_{uik}} - \log c_{uik}!\right] + cste && \text{(A.7)}\\
&= \sum_{ui} \log \mathrm{Mult}(\mathbf{c}_{ui}; y_{ui}, \tilde{\boldsymbol{\phi}}_{ui}) + cste. && \text{(A.8)}
\end{aligned}$$

Et donc $q(\mathbf{c}_{ui}) = \mathrm{Mult}(\mathbf{c}_{ui}; y_{ui}, \tilde{\boldsymbol{\phi}}_{ui})$ où $\tilde{\boldsymbol{\phi}}_{ui}$ est un vecteur de taille K ayant pour coefficients $\tilde{\phi}_{uik} = \frac{\Lambda_{uik}}{\Lambda_{ui}}$, avec $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$.

Variables W et H. De la même façon, pour la variable $\mathbf{W}$, on a :

$$\log q(\mathbf{W}) = \mathbb{E}_{q_{-\mathbf{W}}}(\log p(\mathbf{Y}, \mathbf{C}, \mathbf{W}, \mathbf{H})) + cste, \tag{A.9}$$

où *cste* est une constante par rapport à $\mathbf{W}$, et $q_{-\mathbf{W}}(\mathbf{C}, \mathbf{H}) = q(\mathbf{C})q(\mathbf{H})$. On obtient donc :

$$\begin{aligned}
&\mathbb{E}_{q_{-\mathbf{W}}}(\log p(\mathbf{Y}, \mathbf{C}, \mathbf{W}, \mathbf{H})) && \text{(A.10)}\\
&= \sum_{uk} \left[\left(\alpha + \sum_i \mathbb{E}_q(c_{uik}) - 1\right) \log w_{uk} - \left(\beta + \sum_i \mathbb{E}_q(h_{ik})\right) w_{uk}\right] + cste && \text{(A.11)}\\
&= \sum_{uk} \log \mathrm{Gamma}\left(w_{uk}; \alpha^W + \sum_i \mathbb{E}_q(c_{uik}), \beta_u^W + \sum_i \mathbb{E}_q(h_{ik})\right) + cste. && \text{(A.12)}
\end{aligned}$$

Et par conséquent (le problème est symétrique pour $\mathbf{W}$ et $\mathbf{H}$) :

$$q(w_{uk}) = \mathrm{Gamma}\left(w_{uk}; \alpha^W + \sum_i \mathbb{E}_q(c_{uik}), \beta_u^W + \sum_i \mathbb{E}_q(h_{ik})\right), \tag{A.13}$$

$$q(h_{ik}) = \mathrm{Gamma}\left(h_{ik}; \alpha^H + \sum_u \mathbb{E}_q(c_{uik}), \beta_i^H + \sum_u \mathbb{E}_q(w_{uk})\right). \tag{A.14}$$

A.2. Modèle NBF

Dans le modèle NBF, l'ensemble des variables latentes est donné par $\mathbf{Z} = \{\mathbf{C}, \mathbf{A}, \mathbf{W}, \mathbf{H}\}$. Comme pour la PF, ce modèle augmenté est entièrement conjugué. On détaillera ici le calcul de la loi $q(\mathbf{A})$.

Distribution jointe. La densité de la distribution jointe du modèle augmenté est donnée par :

$$\log p(\mathbf{Y}, \mathbf{C}, \mathbf{A}, \mathbf{W}, \mathbf{H}) = \log p(\mathbf{Y}, \mathbf{C}|\mathbf{A}, \mathbf{W}, \mathbf{H}) + \log p(\mathbf{A}) + \log p(\mathbf{W}) + \log p(\mathbf{H}) \quad \text{(A.15)}$$

Le terme d'attache aux données s'écrit comme :

$$\begin{aligned}
&\log p(\mathbf{Y}, \mathbf{C}|\mathbf{A}, \mathbf{W}, \mathbf{H}) && \text{(A.16)}\\
&= \sum_{uik} \left[c_{uik} \log(a_{ui} w_{uk} h_{ik}) - a_{ui} w_{uk} h_{ik} - \log c_{uik}!\right] && \text{(A.17)}\\
&= \sum_{ui} \left[y_{ui} \log a_{ui} + \sum_k \left(c_{uik} \log(w_{uk} h_{ik}) - a_{ui} w_{uk} h_{ik} - \log c_{uik}! \right) \right], && \text{(A.18)}
\end{aligned}$$

avec $y_{ui} = \sum_k c_{uik}$. On peut remarquer dès à présent que la loi $q(\mathbf{C})$ va être la même que précédemment (la variable $\mathbf{c}_{ui}$ est indépendante de $\mathbf{A}$ sachant $\mathbf{Y}$, $\mathbf{W}$ et $\mathbf{H}$). La loi a priori de la variable $\mathbf{A}$ est donnée par :

$$\log p(\mathbf{A}) = \sum_{ui} \left[(\alpha - 1) \log a_{ui} - \alpha a_{ui}\right] + cste, \quad \text{(A.19)}$$

où *cste* est une constante par rapport à a_{ui}.

Variable A. On note $q_{-\mathbf{A}}(\mathbf{C}, \mathbf{W}, \mathbf{H}) = q(\mathbf{C})q(\mathbf{W})q(\mathbf{H})$. Pour la variable $\mathbf{A}$, on obtient donc :

$$\begin{aligned}
\log q(\mathbf{A}) &= \mathbb{E}_{q_{-\mathbf{A}}}(\log p(\mathbf{Y}, \mathbf{C}, \mathbf{A}|\mathbf{W}, \mathbf{H})) + cste && \text{(A.20)}\\
&= \sum_{ui} \left[y_{ui} \log a_{ui} - a_{ui} \sum_k \mathbb{E}_q(w_{uk}) \mathbb{E}_q(h_{ik}) \right] + \sum_{ui} \left[(\alpha - 1) \log a_{ui} - \alpha a_{ui}\right] && \text{(A.21)}\\
&= \sum_{ui} \left[(\alpha + y_{ui} - 1) \log a_{ui} - a_{ui} \left(\alpha + \sum_k \mathbb{E}_q(w_{uk}) \mathbb{E}_q(h_{ik}) \right) \right] && \text{(A.22)}\\
&= \sum_{ui} \log \text{Gamma} \left(a_{ui}; \alpha + y_{ui}, \alpha + \sum_k \mathbb{E}_q(w_{uk}) \mathbb{E}_q(h_{ik}) \right). && \text{(A.23)}
\end{aligned}$$

A.3. Modèle dcPF

Dans le modèle dcPF, l'ensemble des variables latentes est donné par $\mathbf{Z} = \{\mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}\}$. On détaillera ici le calcul de $q(\mathbf{N}, \mathbf{C})$.

Distribution jointe. La densité de la distribution jointe du modèle augmenté peut s'écrire comme :

$$\log p(\mathbf{Y}, \mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}) = \underbrace{\log p(\mathbf{Y}|\mathbf{N})}_{\text{mapping}} + \underbrace{\log p(\mathbf{N}, \mathbf{C}|\mathbf{W}, \mathbf{H})}_{\text{PF}} + \underbrace{\log p(\mathbf{W}) + \log p(\mathbf{H})}_{\text{a priori}}, \tag{A.24}$$

où le terme de «mapping» est donné par :

$$\log p(\mathbf{Y}|\mathbf{N}) = \sum_{ui} \left[y_{ui}\theta - n_{ui}\kappa\psi(\theta) + \log h(y_{ui}, n_{ui}\kappa) \right]. \tag{A.25}$$

On peut remarquer dès à présent, que les distributions $q(\mathbf{W})$ et $q(\mathbf{W})$ seront très similaire au modèle PF. Il suffit de substituer y_{ui} par $\mathbb{E}_q(n_{ui})$ dans les Éq. (A.13) et (A.13).

Variables N et C. On rappelle que nous avons choisi de garder les variables latentes $\mathbf{N}$ et $\mathbf{C}$ couplées. On note $q_{-\{\mathbf{N},\mathbf{C}\}}(\mathbf{W}, \mathbf{H}) = q(\mathbf{W})q(\mathbf{H})$. On obtient alors :

$$\log q(\mathbf{N}, \mathbf{C}) \tag{A.26}$$

$$= \mathbb{E}_{q_{-\{\mathbf{N},\mathbf{C}\}}}(\log p(\mathbf{Y}, \mathbf{N}, \mathbf{C}|\mathbf{W}, \mathbf{H})) + cste \tag{A.27}$$

$$= \sum_{ui} \left[-n_{ui}\kappa\psi(\theta) + \log h(y_{ui}, n_{ui}\kappa) + \sum_k c_{uik}\mathbb{E}_q(\log(w_{uk}h_{ik})) - \log c_{uik}! \right] + cste \tag{A.28}$$

$$= \sum_{ui} \left[-n_{ui}\kappa\psi(\theta) + \log h(y_{ui}, n_{ui}\kappa) + \sum_k c_{uik}\log\Lambda_{uik} - \log c_{uik}! \right] + cste \tag{A.29}$$

$$= \sum_{ui} \left[\log \frac{\left(\Lambda_{ui}e^{-\kappa\psi(\theta)}\right)^{n_{ui}} h(y_{ui}, n_{ui}\kappa)}{n_{ui}!} + \log \mathrm{Mult}(\mathbf{c}_{ui}; n_{ui}, \tilde{\boldsymbol{\phi}}_{ui}) \right] + cste, \tag{A.30}$$

où $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$ et $\tilde{\boldsymbol{\phi}}_{ui}$ est un vecteur de taille K ayant pour coefficients $\tilde{\phi}_{uik} = \frac{\Lambda_{uik}}{\Lambda_{ui}}$. Par conséquent on a $q(n_{ui}, \mathbf{c}_{ui}) = q(n_{ui})q(\mathbf{c}_{ui}|n_{ui})$ avec :

$$q(n_{ui} = n) \propto \frac{1}{n!}\left(\Lambda_{ui}e^{-\kappa\psi(\theta)}\right)^n h(y_{ui}, n\kappa), \tag{A.31}$$

$$q(\mathbf{c}_{ui}|n_{ui}) = \mathrm{Mult}(\mathbf{c}_{ui}; n_{ui}, \tilde{\boldsymbol{\phi}}_{ui}). \tag{A.32}$$

A.4. Modèle IG-OrdNMF

Le modèle IG-OrdNMF est augmenté avec les variables $\mathbf{N}$ et $\mathbf{C}$ telles que :

$$n_{ui}|y_{ui} \sim \begin{cases} \delta_0, & \text{si } y_{ui} = 0, \\ \text{ZTP}(\lambda_{ui}\Delta_{y_{ui}}), & \text{si } y_{ui} > 0, \end{cases} \tag{A.33}$$

$$\mathbf{c}_{ui}|n_{ui}, \mathbf{W}, \mathbf{H} \sim \text{Mult}(n_{ui}, \boldsymbol{\phi}_{ui}), \tag{A.34}$$

où $\boldsymbol{\phi}_{ui}$ est un vecteur de probabilité ayant pour coefficients $\frac{\lambda_{uik}}{\lambda_{ui}}$, avec $\lambda_{uik} = w_{uk}h_{ik}$ et $\lambda_{ui} = \sum_k \lambda_{uik}$. On note $\mathbf{Z} = \{\mathbf{N}, \mathbf{C}, \mathbf{W}, \mathbf{H}\}$ l'ensemble des variables latentes du modèle augmenté.

Distribution jointe. Lorsque $y_{ui} = 0$, on a $n_{ui} = 0$ et $\mathbf{c}_{ui} = \mathbf{0}_K$. La log-vraisemblance des valeurs nulles est donnée par $\log p(y_{ui} = 0) = -\theta_0\lambda_{ui}$.

Lorsque $y_{ui} > 0$, on a les densités suivantes :

$$\log p(y_{ui}|\mathbf{W}, \mathbf{H}) = -\theta_{y_{ui}}\lambda_{ui} + \log(1 - e^{\Delta_{y_{ui}}\lambda_{ui}}) \tag{A.35}$$

$$\log p(n_{ui}|y_{ui}, \mathbf{W}, \mathbf{H}) = -\log(1 - e^{\Delta_{y_{ui}}\lambda_{ui}}) + n_{ui}\log\Delta_{y_{ui}}\lambda_{ui} - \Delta_{y_{ui}}\lambda_{ui} - \log n_{ui}! \tag{A.36}$$

$$\log p(\mathbf{c}_{ui}|n_{ui}, \mathbf{W}, \mathbf{H}) = -n_{ui}\log\lambda_{uik} + \log n_{ui}! + \sum_k (c_{uik}\log\lambda_{uik} - \log c_{uik}!) \tag{A.37}$$

Ces équations correspondent respectivement à : la distribution des données ordinales pour l'Éq. (A.35), l'augmentation ZTP pour l'Éq. (A.36), et l'augmentation multinomiale pour l'Éq. (A.37). Ainsi, pour une observation $y_{ui} > 0$, on obtient :

$$\log\, p(y_{ui}, n_{ui}, \mathbf{c}_{ui}|\mathbf{W}, \mathbf{H}) \tag{A.38}$$

$$= \log p(y_{ui}|\mathbf{W}, \mathbf{H}) + \log p(n_{ui}|y_{ui}, \mathbf{W}, \mathbf{H}) + \log p(\mathbf{c}_{ui}|n_{ui}, \mathbf{W}, \mathbf{H}) \tag{A.39}$$

$$= -\theta_{y_{ui}-1}\lambda_{ui} + n_{ui}\log\Delta_{y_{ui}} + \sum_k (c_{uik}\log\lambda_{uik} - \log c_{uik}!) \tag{A.40}$$

En introduisant H_y comme décrit en Éq. (4.34), on obtient la densité de la distribution jointe du modèle augmenté :

$$\log p(\mathbf{Y}, \mathbf{N}, \mathbf{C}|\mathbf{W}, \mathbf{H}) = \sum_{ui}\left[n_{ui}\log\Delta_{y_{ui}} + \sum_k (c_{uik}\log\lambda_{uik} - \log c_{uik}!) - H_{y_{ui}}\lambda_{ui}\right]. \tag{A.41}$$

Variables N et C. Encore une fois, on garde les variables **N** et **C** couplées. Lorsque $y_{ui} > 0$, on a $n_{ui} \in \mathbb{N}^*$ et $\sum_k c_{uik} = n_{ui}$. On obtient :

$$\begin{aligned}
\log q(\mathbf{c}_{ui}) &= \mathbb{E}_{q_{-\{\mathbf{N},\mathbf{C}\}}}(\log p(y_{ui}, \mathbf{c}_{ui}, n_{ui}|\mathbf{W}, \mathbf{H})) + cste && \text{(A.42)}\\
&= n_{ui} \log \Delta_{y_{ui}} + \sum_k (c_{uik} \log \Lambda_{uik} - \log c_{uik}!) + cste && \text{(A.43)}\\
&= n_{ui} \log \Delta_{y_{ui}} + n_{ui} \log \Lambda_{ui} - \log n_{ui}! + \log \mathrm{Mult}(\mathbf{c}_{ui}; n_{ui}, \tilde{\boldsymbol{\phi}}_{ui}) + cste && \text{(A.44)}\\
&= \log \mathrm{ZTP}(n_{ui}; \Lambda_{ui}\Delta_y) + \log \mathrm{Mult}(\mathbf{c}_{ui}; n_{ui}, \tilde{\boldsymbol{\phi}}_{ui}) + cste, && \text{(A.45)}
\end{aligned}$$

où $\Lambda_{uik} = \exp\left(\mathbb{E}_q(\log w_{uk}) + \mathbb{E}_q(\log h_{ik})\right)$ et $\tilde{\boldsymbol{\phi}}_{ui}$ est un vecteur de probabilité de taille K ayant pour coefficients $\tilde{\phi}_{uik} = \frac{\Lambda_{uik}}{\Lambda_{ui}}$. Par conséquent, on a $q(n_{ui}, \mathbf{c}_{ui}) = q(n_{ui})q(\mathbf{c}_{ui}|n_{ui})$ avec :

$$\begin{aligned}
q(n_{ui}) &= \mathrm{ZTP}(n_{ui}; \Lambda_{ui}\Delta_y) && \text{(A.46)}\\
q(\mathbf{c}_{ui}|n_{ui}) &= \mathrm{Mult}(\mathbf{c}_{ui}; n_{ui}, \tilde{\boldsymbol{\phi}}_{ui}). && \text{(A.47)}
\end{aligned}$$

Variables W et H. De la même façon que les modèles précédents, on a pour la variable **W** :

$$\begin{aligned}
\log q(\mathbf{W}) &= \mathbb{E}_{q_{-\mathbf{W}}}(\log p(\mathbf{W}|\mathbf{Y}, \mathbf{N}, \mathbf{C}, \mathbf{H})) && \text{(A.48)}\\
&= \sum_{uik} \Big[\mathbb{E}_q(c_{uik}) \log w_{uk} - H_{y_{ui}} w_{uk} \mathbb{E}_q(h_{ik})\Big] + \sum_{uk} \Big[(\alpha - 1) \log w_{uk} - \beta w_{uk}\Big] + cste
\end{aligned}$$

Par conséquent, on obtient :

$$q(w_{uk}) = \mathrm{Gamma}\left(w_{uk}; \alpha + \sum_i \mathbb{E}_q(c_{uik}), \beta + \sum_i H_{y_{ui}} \mathbb{E}_q(h_{ik})\right), \quad \text{(A.49)}$$

$$q(h_{ik}) = \mathrm{Gamma}\left(h_{ik}; \alpha + \sum_u \mathbb{E}_q(c_{uik}), \beta + \sum_u H_{y_{ui}} \mathbb{E}_q(w_{uk})\right). \quad \text{(A.50)}$$

Bibliographie

[AGZ15] A. Acharya, J. Ghosh et M. Zhou. "Nonparametric Bayesian Factor Analysis for Dynamic Count Matrices". In : *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2015 (cf. p. 85, 89).

[AK11] A. Agresti et M. Kateri. *Categorical Data Analysis*. Springer, 2011 (cf. p. 85, 86, 89).

[AK97] C. V. Ananth et D. G. Kleinbaum. "Regression Models for Ordinal Responses : A Review of Methods and Applications." In : *International journal of epidemiology* 26.6 (1997), p. 1323-1333 (cf. p. 89).

[AT05] G. Adomavicius et A. Tuzhilin. "Toward the next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions". In : *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), p. 734-749 (cf. p. 7).

[BC92] N. J. Belkin et W. B. Croft. "Information Filtering and Information Retrieval : Two Sides of the Same Coin". In : *Communications of the ACM*. 1992 (cf. p. 8).

[BE16] M. E. Basbug et B. E. Engelhardt. "Hierarchical Compound Poisson Factorization". In : *Proc. International Conference on Machine Learning (ICML)*. 2016 (cf. p. 11, 36, 40, 54, 55, 58, 59, 79, 125).

[BE17] M. E. Basbug et B. E. Engelhardt. "Coupled Compound Poisson Factorization". In : *arXiv preprint arXiv :1701.02058* (2017) (cf. p. 40, 54, 55, 81).

[Ber+11] T. Bertin-Mahieux, D. P. Ellis, B. Whitman et P. Lamere. "The Million Song Dataset". In : *Proc. International Society for Music Information Retrieval (ISMIR)*. T. 2. 2011, p. 10 (cf. p. 29, 47, 73, 118).

[BF53] C. I. Bliss et R. A. Fisher. "Fitting the Negative Binomial Distribution to Biological Data". In : *Biometrics* 9.2 (1953), p. 176-200 (cf. p. 51).

[BHK98] J. S. Breese, D. Heckerman et C. Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 1998, p. 43-52 (cf. p. 9).

[BJ06] W. Buntine et A. Jakulin. "Discrete Component Analysis". In : *Subspace, Latent Structure and Feature Selection*. Springer, 2006, p. 1-33 (cf. p. 23, 25, 39).

[BK07] R. M. Bell et Y. Koren. “Lessons from the Netflix Prize Challenge.” In : *SiGKDD Explorations* 9.2 (2007), p. 75-79 (cf. p. 13).

[BKM17] D. M. Blei, A. Kucukelbir et J. D. McAuliffe. “Variational Inference : A Review for Statisticians”. In : *Journal of the American Statistical Association* (2017) (cf. p. 17, 18).

[BL+07] J. Bennett, S. Lanning et al. “The Netflix Prize”. In : *Proc. KDD Cup and Workshop*. T. 2007. 2007, p. 35 (cf. p. 1, 13).

[BNJ03] D. M. Blei, A. Y. Ng et M. I. Jordan. “Latent Dirichlet Allocation”. In : *The Journal of Machine Learning Research* 3.Jan (2003), p. 993-1022 (cf. p. 9, 25).

[BS97] M. Balabanović et Y. Shoham. “Fab : Content-Based, Collaborative Recommendation”. In : *Communications of the ACM* 40.3 (1997), p. 66-72 (cf. p. 8).

[Bur02] R. Burke. “Hybrid Recommender Systems : Survey and Experiments”. In : *User modeling and user-adapted interaction* 12.4 (2002), p. 331-370 (cf. p. 8).

[BW74] W. E. Bleick et P. C. Wang. “Asymptotics of Stirling Numbers of the Second Kind”. In : *Proc. of the American Mathematical Society* 42.2 (1974), p. 575-580 (cf. p. 65).

[Cai+10] D. Cai, X. He, J. Han et T. S. Huang. “Graph Regularized Nonnegative Matrix Factorization for Data Representation”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2010), p. 1548-1560 (cf. p. 15).

[Can+11] L. Candillier, M. Chevalier, D. Dudognon et J. Mothe. “Diversity in Recommender Systems”. In : *Proc. International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services. CENTRIC*. 2011, p. 23-29 (cf. p. 13).

[Can04] J. Canny. “GaP : A Factor Model for Discrete Data”. In : *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*. 2004, p. 122-129 (cf. p. 23, 25, 39).

[Cel10] Ò. Celma. “Music Recommendation”. In : *Music Recommendation and Discovery*. Springer Berlin Heidelberg, 2010, p. 43-85 (cf. p. 73, 118).

[Cem09] A. T. Cemgil. “Bayesian Inference for Nonnegative Matrix Factorisation Models”. In : *Computational Intelligence and Neuroscience* (2009) (cf. p. 23-26, 39, 49).

[CF17] F. Caron et E. B. Fox. “Sparse Graphs Using Exchangeable Random Measures”. In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 79.5 (2017), p. 1295-1366 (cf. p. 32).

[CG05] W. Chu et Z. Ghahramani. “Gaussian Processes for Ordinal Regression”. In : *The Journal of Machine Learning Research* 6.Jul (2005), p. 1019-1041 (cf. p. 88).

[Cha+15] L. Charlin, R. Ranganath, J. McInerney et D. M. Blei. “Dynamic poisson factorization”. In : *Proc. ACM Conference on Recommender Systems (RecSys)*. 2015, p. 155-162 (cf. p. 128).

[CM07] G. Consonni et J.-M. Marin. “Mean-Field Variational Approximate Bayesian Inference for Latent Variable Models”. In : *Computational Statistics & Data Analysis* 52.2 (2007), p. 790-798 (cf. p. 19, 89).

[CP10] E. J. Candès et Y. Plan. “Matrix Completion with Noise”. In : *Proceedings of the IEEE* 98.6 (2010), p. 925-936 (cf. p. 14).

[CR09] E. J. Candès et B. Recht. “Exact Matrix Completion via Convex Optimization”. In : *Foundations of Computational mathematics* 9.6 (2009), p. 717 (cf. p. 14).

[CTM14] F. Caron, Y. W. Teh et B. T. Murphy. “Bayesian Nonparametric Plackett-Luce Models for the Analysis of Clustered Ranked Data”. In : *Annal of Applied Statistics* 8 (2014), p. 1145-1181 (cf. p. 32).

[DGG19] C.-E. Dias, V. Guigue et P. Gallinari. “Personalized Attention for Textual Profiling and Recommendation”. In : *Proc. of SIGIR Workshop on ExplainAble Recommendation and Search (EARS)*. 2019 (cf. p. 8).

[Du+15] N. Du, Y. Wang, N. He, J. Sun et L. Song. “Time-Sensitive Recommendation From Recurrent User Activities”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2015, p. 3492-3500 (cf. p. 54, 128).

[Dur16] G. Durif. “Multivariate Analysis of High-Throughput Sequencing Data”. Thèse de doct. Lyon, 2016 (cf. p. 51).

[Eck+08] D. Eck, P. Lamere, T. Bertin-Mahieux et S. Green. “Automatic Generation of Social Tags for Music Recommendation”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2008, p. 385-392 (cf. p. 110).

[FBD09] C. Févotte, N. Bertin et J.-L. Durrieu. “Nonnegative Matrix Factorization with the Itakura-Saito Divergence : With Application to Music Analysis”. In : *Neural computation* 21.3 (2009), p. 793-830 (cf. p. 17).

[FD15] C. Févotte et N. Dobigeon. “Nonlinear Hyperspectral Unmixing with Robust Nonnegative Matrix Factorization”. In : *IEEE Transactions on Image Processing* 24.12 (2015), p. 4810-4819 (cf. p. 38).

[Fel08] W. Feller. *An Introduction to Probability Theory and Its Applications*. T. 2. John Wiley & Sons, 2008 (cf. p. 56).

[FI11] C. Févotte et J. Idier. “Algorithms for Nonnegative Matrix Factorization with the Beta-Divergence”. In : *Neural computation* 23.9 (2011), p. 2421-2456 (cf. p. 17, 43, 116).

[FS11] Y. Fang et L. Si. “Matrix Co-Factorization for Recommendation with Rich Side Information and Implicit Feedback”. In : *Proc. International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec)*. 2011, p. 65-69 (cf. p. 110, 111).

[GCB14] P. K. Gopalan, L. Charlin et D. Blei. “Content-Based Recommendations with Poisson Factorization”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2014, p. 3176-3184 (cf. p. 111, 112).

[GH16] C. A. Gomez-Uribe et N. Hunt. “The Netflix Recommender System : Algorithms, Business Value, and Innovation”. In : *ACM Transactions on Management Information Systems (TMIS)* 6.4 (2016), p. 13 (cf. p. 7).

[GHB15] P. Gopalan, J. M. Hofman et D. M. Blei. “Scalable Recommendation with Hierarchical Poisson Factorization.” In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 2015, p. 326-335 (cf. p. 23-26, 36, 39).

[GMS95] W. Gardner, E. P. Mulvey et E. C. Shaw. “Regression Analyses of Counts and Rates : Poisson, Overdispersed Poisson, and Negative Binomial Models.” In : *Psychological bulletin* 118.3 (1995), p. 392 (cf. p. 36).

[GOF18a] O. Gouvert, T. Oberlin et C. Févotte. “Matrix Co-Factorization for Cold-Start Recommendation.” In : *Proc. International Society for Music Information Retrieval (ISMIR)*. 2018, p. 792-798 (cf. p. 5, 109).

[GOF18b] O. Gouvert, T. Oberlin et C. Févotte. “Negative Binomial Matrix Factorization for Recommender Systems”. In : *arXiv preprint arXiv :1801.01708* (2018) (cf. p. 5, 35).

[GOF19] O. Gouvert, T. Oberlin et C. Févotte. “Recommendation from Raw Data with Adaptive Compound Poisson Factorization”. In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 2019 (cf. p. 5, 53).

[Gol+92] D. Goldberg, D. Nichols, B. M. Oki et D. Terry. “Using Collaborative Filtering to Weave an Information Tapestry”. In : *Communications of the ACM* 35.12 (1992), p. 61-71 (cf. p. 8).

[Gop+14] P. Gopalan, F. J. Ruiz, R. Ranganath et D. Blei. “Bayesian Nonparametric Poisson Factorization for Recommendation Systems”. In : *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2014, p. 275-283 (cf. p. 33).

[Gut+15] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro et C. Hervas-Martinez. “Ordinal Regression Methods : Survey and Experimental Study”. In : *IEEE Transactions on Knowledge and Data Engineering* 28.1 (2015), p. 127-146 (cf. p. 84).

[Her+04] J. L. Herlocker, J. A. Konstan, L. G. Terveen et J. T. Riedl. “Evaluating Collaborative Filtering Recommender Systems”. In : *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), p. 5-53 (cf. p. 12, 13, 128).

[Her+99] J. L. Herlocker, J. A. Konstan, A. Borchers et J. Riedl. “An Algorithmic Framework for Performing Collaborative Filtering”. In : *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*. 1999, p. 230-237 (cf. p. 9).

[HHG14] J. M. Hernandez-Lobato, N. Houlsby et Z. Ghahramani. “Probabilistic Matrix Factorization with Non-Random Missing Data”. In : *Proc. International Conference on Machine Learning (ICML)*. 2014, p. 1512-1520 (cf. p. 15, 88, 103).

[Hil11] J. M. Hilbe. *Negative Binomial Regression*. Cambridge University Press, 2011 (cf. p. 36).

[HKR00] J. L. Herlocker, J. A. Konstan et J. Riedl. “Explaining Collaborative Filtering Recommendations”. In : *Proc. ACM Conference on Computer Supported Cooperative Work*. 2000, p. 241-250 (cf. p. 8).

[HKV08] Y. Hu, Y. Koren et C. Volinsky. “Collaborative Filtering for Implicit Feedback Datasets”. In : *Proc. IEEE International Conference on Data Mining (ICDM)*. 2008, p. 263-272 (cf. p. 11, 15, 35, 40, 127).

[HL04] D. R. Hunter et K. Lange. “A Tutorial on MM Algorithms”. In : *The American Statistician* 58.1 (2004), p. 30-37 (cf. p. 43).

[Hof+13] M. D. Hoffman, D. M. Blei, C. Wang et J. Paisley. “Stochastic Variational Inference”. In : *The Journal of Machine Learning Research* 14.1 (2013), p. 1303-1347 (cf. p. 21).

[Hof99] T. Hofmann. “Probabilistic Latent Semantic Analysis”. In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 1999, p. 289-296 (cf. p. 9).

[Hos+18] S. Hosseini, A. Khodadadi, K. Alizadeh, A. Arabzadeh, M. Farajtabar, H. Zha et H. R. Rabiee. “Recurrent poisson factorization for temporal recommendation”. In : *IEEE Transactions on Knowledge and Data Engineering* (2018) (cf. p. 54, 59, 128).

[JBE17] G. Jerfel, M. E. Basbug et B. E. Engelhardt. “Dynamic Collaborative Filtering with Compound Poisson Factorization”. In : *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017 (cf. p. 128).

[JK02] K. Järvelin et J. Kekäläinen. “Cumulated Gain-Based Evaluation of IR Techniques”. In : *ACM Transactions on Information Systems (TOIS)* 20.4 (2002), p. 422-446 (cf. p. 12).

[JKK05] N. L. Johnson, A. W. Kemp et S. Kotz. *Univariate Discrete Distributions*. T. 444. John Wiley & Sons, 2005 (cf. p. 56).

[Jor+99] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola et L. K. Saul. “An Introduction to Variational Methods for Graphical Models”. In : *Machine learning* 37.2 (1999), p. 183-233 (cf. p. 17).

[Jør86] B. Jørgensen. “Some Properties of Exponential Dispersion Models”. In : *Scandinavian Journal of Statistics* (1986), p. 187-197 (cf. p. 58).

[Jor87] B. Jorgensen. “Exponential Dispersion Models”. In : *Journal of the Royal Statistical Society. Series B (Methodological)* (1987), p. 127-162 (cf. p. 58).

[Jor97] B. Jorgensen. *The Theory of Dispersion Models*. CRC Press, 1997 (cf. p. 58).

[KBV09] Y. Koren, R. Bell et C. Volinsky. “Matrix Factorization Techniques for Recommender Systems”. In : *Computer* 42.8 (2009), p. 30-37 (cf. p. 13).

[KH17] A. Karatzoglou et B. Hidasi. “Deep Learning for Recommender Systems”. In : *Proc. ACM Conference on Recommender Systems (RecSys)*. 2017, p. 396-397 (cf. p. 127).

[Kho+18] A. Khodadadi, S. A. Hosseini, E. Tavakoli et H. R. Rabiee. “Continuous-Time User Modeling in Presence of Badges : A Probabilistic Approach”. In : *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12.3 (2018), p. 37 (cf. p. 54, 128).

[Kle03] J. Kleinberg. “Bursty and Hierarchical Structure in Streams”. In : *Data Mining and Knowledge Discovery* 7.4 (2003), p. 373-397 (cf. p. 54).

[KWV07] K. Kurihara, M. Welling et N. Vlassis. “Accelerated Variational Dirichlet Process Mixtures”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2007, p. 761-768 (cf. p. 33).

[Lam+08] X. N. Lam, T. Vu, T. D. Le et A. D. Duong. “Addressing Cold-Start Problem in Recommendation Systems”. In : *Proc. International Conference on Ubiquitous Information Management and Communication (IMCOM)*. 2008, p. 208-211 (cf. p. 8, 110).

[Lam92] D. Lambert. “Zero-Inflated Poisson Regression, With an Application to Defects in Manufacturing”. In : *Technometrics* 34.1 (1992), p. 1-14 (cf. p. 40).

[Law87] J. F. Lawless. “Negative Binomial and Mixed Poisson Regression”. In : *Canadian Journal of Statistics* 15.3 (1987), p. 209-225 (cf. p. 36).

[LBR17] S. Leglaive, R. Badeau et G. Richard. “Semi-Blind Student’s t Source Separation for Multichannel Audio Convolutive Mixtures”. In : *Proc. European Signal Processing Conference (EUSIPCO)*. 2017, p. 2259-2263 (cf. p. 41).

[LFF18] A. Lumbreras, L. Filstroff et C. Févotte. “Bayesian Mean-Parameterized Nonnegative Binary Matrix Factorization”. In : *arXiv preprint arXiv :1812.06866* (2018) (cf. p. 90).

[Lia+16] D. Liang, L. Charlin, J. McInerney et D. M. Blei. “Modeling User Exposure in Recommendation”. In : *Proc. International Conference on World Wide Web (WWW)*. 2016, p. 951-961 (cf. p. 29, 39, 40, 47).

[Lia+18] D. Liang, R. G. Krishnan, M. D. Hoffman et T. Jebara. “Variational Autoencoders for Collaborative Filtering”. In : *Proc. International Conference on World Wide Web (WWW)*. 2018 (cf. p. 9, 80).

[LR14] R. J. A. Little et D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014 (cf. p. 15, 85).

[LS01] D. D. Lee et H. S. Seung. "Algorithms for Non-Negative Matrix Factorization". In : *Advances in Neural Information Processing Systems (NIPS)*. 2001, p. 556-562 (cf. p. 16).

[LS99] D. D. Lee et H. S. Seung. "Learning the Parts of Objects by Non-Negative Matrix Factorization". In : *Nature* 401.6755 (1999), p. 788-791 (cf. p. 16).

[LZE15] D. Liang, M. Zhan et D. P. Ellis. "Content-Aware Collaborative Music Recommendation Using Pre-Trained Neural Networks." In : *Proc. International Society for Music Information Retrieval (ISMIR)*. 2015, p. 295-301 (cf. p. 111).

[Ma+11] H. Ma, C. Liu, I. King et M. R. Lyu. "Probabilistic Factor Models for Web Site Recommendation". In : *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*. 2011, p. 265-274 (cf. p. 23, 25, 39).

[Mar+07] B. Marlin, R. S. Zemel, S. Roweis et M. Slaney. "Collaborative Filtering and the Missing at Random Assumption". In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 2007 (cf. p. 15, 103).

[McC80] P. McCullagh. "Regression Models for Ordinal Data". In : *Journal of the Royal Statistical Society : Series B (Methodological)* 42.2 (1980), p. 109-127 (cf. p. 84, 88).

[McF+12] B. McFee, T. Bertin-Mahieux, D. P. Ellis et G. R. Lanckriet. "The Million Song Dataset Challenge". In : *Proc. International Conference on World Wide Web (WWW)*. 2012, p. 909-916 (cf. p. 118).

[MZ09] B. M. Marlin et R. S. Zemel. "Collaborative Prediction and Ranking with Non-Random Missing Data". In : *Proc. ACM Conference on Recommender Systems (RecSys)*. 2009, p. 5-12 (cf. p. 15, 102, 103).

[Ney39] J. Neyman. "On a New Class of Contagious Distributions, Applicable in Entomology and Bacteriology". In : *Annals of Mathematical Statistics* 10.1 (1939), p. 35-57 (cf. p. 54, 58, 63).

[Pan+08] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz et Q. Yang. "One-Class Collaborative Filtering". In : *Proc. IEEE International Conference on Data Mining (ICDM)*. 2008, p. 502-511 (cf. p. 11, 15).

[Paz99] M. J. Pazzani. "A Framework for Collaborative, Content-Based and Demographic Filtering". In : *Artificial intelligence review* 13.5-6 (1999), p. 393-408 (cf. p. 8).

[PC09] J. Paisley et L. Carin. "Nonparametric Factor Analysis with Beta Process Priors". In : *Proc. International Conference on Machine Learning (ICML)*. 2009, p. 777-784 (cf. p. 32).

[Per+16] V. Perrone, P. A. Jenkins, D. Spano et Y. W. Teh. “Poisson Random Fields for Dynamic Feature Models”. In : *Journal of Machine Learning Research* 18 (2016) (cf. p. 73).

[PK13] U. Paquet et N. Koenigstein. “One-Class Collaborative Filtering with Random Graphs”. In : *Proc. International Conference on World Wide Web (WWW)*. 2013, p. 999-1008 (cf. p. 11, 15, 40).

[Pól30] G. Pólya. “Sur Quelques Points de La Théorie Des Probabilités”. In : *Annales de l’Institut Henri Poincaré*. 1930, p. 117-161 (cf. p. 58, 64).

[PTW12] U. Paquet, B. Thomson et O. Winther. “A Hierarchical Model for Ordinal Matrix Factorization”. In : *Statistics and Computing* 22.4 (2012), p. 945-957 (cf. p. 88, 103).

[Que49] M. H. Quenouille. “A Relation between the Logarithmic, Poisson, and Negative Binomial Series”. In : *Biometrics* 5.2 (1949), p. 162-164 (cf. p. 54, 58, 63).

[Ren+09] S. Rendle, C. Freudenthaler, Z. Gantner et L. Schmidt-Thieme. “BPR : Bayesian Personalized Ranking from Implicit Feedback”. In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 2009, p. 452-461 (cf. p. 127).

[Rio12] J. Riordan. *Introduction to Combinatorial Analysis*. Courier Corporation, 2012 (cf. p. 65).

[Rob07] C. Robert. *The Bayesian Choice : From Decision-Theoretic Foundations to Computational Implementation*. Springer Science & Business Media, 2007 (cf. p. 66).

[RRS11] F. Ricci, L. Rokach et B. Shapira. “Introduction to Recommender Systems Handbook”. In : *Recommender Systems Handbook*. Springer, 2011, p. 1-35 (cf. p. 7).

[RS05] J. D. Rennie et N. Srebro. “Fast Maximum Margin Matrix Factorization for Collaborative Prediction”. In : *Proc. International Conference on Machine Learning (ICML)*. 2005, p. 713-719 (cf. p. 14).

[Sar+00] B. Sarwar, G. Karypis, J. Konstan et J. Riedl. *Application of Dimensionality Reduction in Recommender System – a Case Study*. 2000 (cf. p. 14).

[Sar+01] B. M. Sarwar, G. Karypis, J. A. Konstan et J. Riedl. “Item-Based Collaborative Filtering Recommendation Algorithms.” In : *Proc. International Conference on World Wide Web (WWW)* 1 (2001), p. 285-295 (cf. p. 9).

[Sch+02] A. I. Schein, A. Popescul, L. H. Ungar et D. M. Pennock. “Methods and Metrics for Cold-Start Recommendations”. In : *Proc. ACM International on Research and Development in Information Retrieval (SIGIR)*. 2002, p. 253-260 (cf. p. 8, 110).

[SCY13] U. Simsekli, A. T. Cemgil et Y. K. Yilmaz. “Learning the Beta-Divergence in Tweedie Compound Poisson Matrix Factorization Models”. In : *Proc. International Conference on Machine Learning (ICML)*. 2013, p. 1409-1417 (cf. p. 55, 58).

[Sea+13] S. Seaman, J. Galati, D. Jackson et J. Carlin. “What Is Meant by" Missing at Random" ?” In : *Statistical Science* (2013), p. 257-268 (cf. p. 15, 85).

[Sei+14] N. Seichepine, S. Essid, C. Févotte et O. Cappé. “Soft Nonnegative Matrix Co-Factorization”. In : *IEEE Transactions on Signal Processing* 62.22 (2014), p. 5940-5949 (cf. p. 111).

[Set94] J. Sethuraman. “A Constructive Definition of Dirichlet Priors”. In : *Statistica sinica* (1994), p. 639-650 (cf. p. 33).

[Sim13] M. Simchowitz. *Zero-Inflated Poisson Factorization for Recommendation Systems*. 2013. url : https://people.eecs.berkeley.edu/~msimchow/JuniorPaper.pdf (cf. p. 39, 40).

[Sin+10] V. Sindhwani, S. S. Bucak, J. Hu et A. Mojsilovic. “One-Class Matrix Completion with Low-Density Factorizations”. In : *Proc. IEEE International Conference on Data Mining (ICDM)*. 2010, p. 1055-1060 (cf. p. 11).

[SJ10] A. Simma et M. I. Jordan. “Modeling Events with Cascades of Poisson Processes”. In : *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*. 2010, p. 546-555 (cf. p. 128).

[Ske52] J. G. Skellam. “Studies in Statistical Ecology : I. Spatial Pattern”. In : *Biometrika* 39.3/4 (1952), p. 346-362 (cf. p. 58, 66).

[Sla11] M. Slaney. “Web-Scale Multimedia Analysis : Does Content Matter ?” In : *IEEE MultiMedia* 18.2 (2011), p. 12-15 (cf. p. 8).

[SM07] R. Salakhutdinov et A. Mnih. “Probabilistic Matrix Factorization.” In : *Advances in Neural Information Processing Systems (NIPS)*. T. 1. 2007, p. 2-1 (cf. p. 14).

[SRJ05] N. Srebro, J. Rennie et T. S. Jaakkola. “Maximum-Margin Matrix Factorization”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2005, p. 1329-1336 (cf. p. 9, 14).

[Ste+46] S. S. Stevens et al. “On the theory of scales of measurement”. In : *Science* (1946) (cf. p. 84).

[SV06] L. Springael et I. Van Nieuwenhuyse. *On the Sum of Independent Zero-Truncated Poisson Random Variables*. 2006. url : https://core.ac.uk/download/pdf/34379693.pdf (cf. p. 63).

[SWZ16] A. Schein, H. Wallach et M. Zhou. “Poisson-Gamma Dynamical Systems”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2016, p. 5005-5013 (cf. p. 26, 35, 54).

[TF13] V. Y. Tan et C. Févotte. “Automatic Relevance Determination in Nonnegative Matrix Factorization with the Beta-Divergence”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.7 (2013), p. 1592-1605 (cf. p. 17, 37).

[Tho49] M. Thomas. “A Generalization of Poisson’s Binomial Limit for Use in Ecology”. In : *Biometrika* 36.1/2 (1949), p. 18-25 (cf. p. 54, 58, 66).

[Tit08] M. K. Titsias. “The Infinite Gamma-Poisson Feature Model”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2008, p. 1513-1520 (cf. p. 32).

[Twe84] M. C. Tweedie. “An Index Which Distinguishes between Some Important Exponential Families”. In : *Statistics : Applications and New Directions : Proc. Indian Statistical Institute Golden Jubilee International Conference*. 1984, p. 579-604 (cf. p. 17).

[vDS13] A. van den Oord, S. Dieleman et B. Schrauwen. “Deep Content-Based Music Recommendation”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2013, p. 2643-2651 (cf. p. 8).

[WB11] C. Wang et D. M. Blei. “Collaborative Topic Modeling for Recommending Scientific Articles”. In : *Proc. International Conference on Knowledge Discovery and Data Mining (KDD)*. 2011, p. 448-456 (cf. p. 110, 112).

[WD67] S. H. Walker et D. B. Duncan. “Estimation of the Probability of an Event as a Function of Several Independent Variables”. In : *Biometrika* 54.1-2 (1967), p. 167-179 (cf. p. 84, 88).

[YC09] J. Yoo et S. Choi. “Probabilistic Matrix Tri-Factorization”. In : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2009, p. 1553-1556 (cf. p. 123).

[YC12] Y. K. Yilmaz et A. T. Cemgil. “Alpha/Beta Divergences and Tweedie Models”. In : *arXiv preprint arXiv :1209.4280* (2012) (cf. p. 17, 55, 58).

[YIG16] K. Yoshii, K. Itoyama et M. Goto. “Student’s t Nonnegative Matrix Factorization and Positive Semidefinite Tensor Factorization for Single-Channel Audio Source Separation”. In : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, p. 51-55 (cf. p. 41).

[ZC15] M. Zhou et L. Carin. “Negative Binomial Process Count and Mixture Modeling”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.2 (2015), p. 307-320 (cf. p. 32, 63, 69).

[ZCC15] M. Zhou, Y. Cong et B. Chen. “The Poisson Gamma Belief Network”. In : *Advances in Neural Information Processing Systems (NIPS)*. 2015, p. 3043-3051 (cf. p. 26).

[Zha+06] S. Zhang, W. Wang, J. Ford et F. Makedon. “Learning from incomplete ratings using non-negative matrix factorization”. In : *Proc. SIAM International Conference on Data Mining*. 2006, p. 549-553 (cf. p. 15).

[Zho+12] M. Zhou, L. Li, D. Dunson et L. Carin. “Lognormal and Gamma Mixed Negative Binomial Regression”. In : *Proc. International Conference on Machine Learning (ICML)*. 2012, p. 1343 (cf. p. 36).

[Zho17] M. Zhou. “Nonparametric Bayesian Negative Binomial Factor Analysis”. In : *Bayesian Analysis* (2017) (cf. p. 25, 32, 37, 54, 58, 59, 63, 72, 128).

Résumé

Ces quinze dernières années, les systèmes de recommandation ont fait l'objet de nombreuses recherches. L'objectif de ces systèmes est de recommander à chaque utilisateur d'une plateforme des contenus qu'il pourrait apprécier. Cela permet notamment de faciliter la navigation des utilisateurs au sein de très larges catalogues de produits. Les techniques dites de filtrage collaboratif (CF) permettent de faire de telles recommandations à partir des historiques de consommation des utilisateurs uniquement. Ces informations sont habituellement stockées dans des matrices où chaque coefficient correspond au retour d'un utilisateur sur un article. Ces matrices de retour ont la particularité d'être de très grande dimension mais aussi d'être extrêmement creuses puisque les utilisateurs n'ayant interagi qu'avec une petite partie du catalogue. Les retours dits implicites sont les retours d'utilisateurs les plus faciles à collecter. Ils peuvent par exemple prendre la forme de données de comptage, qui correspondent alors au nombre de fois où un utilisateur a interagi avec un article. Les techniques de factorisation en matrices non-négatives (NMF) consistent à approximer cette matrice de retour par le produit de deux matrices non-négatives. Ainsi, chaque utilisateur et chaque article présents dans le système sont représentés par un vecteur non-négatif correspondant respectivement à ses préférences et attributs. Cette approximation, qui correspond à une technique de réduction de dimension, permet alors de faire des recommandations aux utilisateurs.

L'objectif de cette thèse est de proposer des méthodes bayésiennes de NMF permettant de modéliser directement les données de comptage sur-dispersées rencontrées en CF. Pour cela, nous étudions d'abord la factorisation Poisson (PF) et présentons ses limites concernant le traitement des données brutes. Pour pallier les problèmes rencontrés par la PF, nous proposons deux extensions de celle-ci : la factorisation binomiale négative (NBF) et la factorisation Poisson composée discrète (dcPF). Ces deux méthodes bayésiennes de NMF proposent des modèles hiérarchiques permettant d'ajouter de la variance. En particulier, la dcPF amène à une interprétation des variables spécialement adaptée à la recommandation musicale. Nous choisissons ensuite de travailler avec des données implicites quantifiées. Cette quantification permet de simplifier la forme des données collectées et d'obtenir des données ordinales. Nous développons donc un modèle de NMF probabiliste adapté aux données ordinales et montrons qu'il peut aussi être vu comme une extension de la PF appliquée à des données pré-traitées. Enfin, le dernier travail de cette thèse traite du problème bien connu de démarrage à froid qui affecte les méthodes de CF. Nous proposons un modèle de co-factorisation de matrices permettant de résoudre ce problème.

Abstract

In recent years, a lot of research has been devoted to recommender systems. The goal of these systems is to recommend to each user some products that he/she may like, in order to facilitate his/her exploration of large catalogs of items. Collaborative filtering (CF) allows to make such recommendations based on the past interactions of the users only. These data are stored in a matrix, where each entry corresponds to the feedback of a user on an item. In particular, this matrix is of very high dimensions and extremly sparse, since the users have interacted with a few items from the catalog. Implicit feedbacks are the easiest data to collect. They are usually available in the form of counts, corresponding to the number of times a user interacted with an item. Non-negative matrix factorization (NMF) techniques consist in approximating the feedback matrix by the product of two non-negative matrices. Thus, each user and item is represented by a latent factor of small dimension corresponding to its preferences and attributes respectively.

The goal of this thesis is to develop Bayesian NMF methods which can directly model the over-dispersed count data arising in CF. To do so, we first study Poisson factorization (PF) and present its limits for the processing of over-dispersed data. To alleviate this problem, we propose two extensions of PF : negative binomial factorization (NBF) and discrete compound Poisson factorisation (dcPF). In particular, dcPF leads to an interpretation of the variables especially suited to music recommendation. Then, we choose to work on quantified implicit data. This pre-processing simplifies the data which are therefore ordinal. Thus, we propose a Bayesian NMF model for this kind of data, coined OrdNMF. We show that this model is also an extension of PF applied to pre-processed data. Finally, in the last chapter of this thesis, we focus on the well-known cold-start problem which affects CF techniques. We propose a matrix co-factorization model which allow us to solve this issue.