Modèles probabilistes et factorisation matricielle non-négative : cadre unifié pour les données textuelles

Mickaël Febrissy

le cnam

ÉCOLE DOCTORALE Informatique, Télécommunications et Electronique
Centre Borelli, Université de Paris

THÈSE DE DOCTORAT

présentée par : **Mickaël FEBRISSY**

soutenue le : **06 juillet 2021**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline : **Sciences et technologies de l'information et de la communication**

Spécialité : **Informatique**

Modèles probabilistes et factorisation matricielle non-négative : cadre unifié pour les données textuelles

THÈSE dirigée par

Monsieur Nadif Mohamed	*Professeur, Université de Paris*

RAPPORTEURS

Monsieur Couceiro Miguel	*Professeur, Université de Lorraine*
Monsieur De Assis Tenorio de Carvalho Francisco	*Professeur, Universidade Federal de Pernambuco*

PRÉSIDENT DU JURY

Monsieur Hanczar Blaise	*Professeur, Université Paris-Saclay, Université d'Évry*

EXAMINATEURS

Madame Ghazzali Nadia	*Professeure, Université du Québec à Trois-Rivières*
Madame Niang-Keita Ndèye	*Maîtresse de conférences, Conservatoire national des arts et métiers*
Monsieur Labiod Lazhar	*Maître de conférences, Université de Paris*

Still we learn.

Remerciements

Je souhaite tout d'abord remercier mon directeur de thèse, le Professeur Mohamed Nadif, pour avoir proposé ce sujet de recherche et sa contribution perpuetuelle à l'avancée de la science des données. Ses compétences, ses conseils avisés, son soutien, ainsi que sa rigeur scienfitique et ses qualités humaines furent amplement essentiels à ma réussite et vecteurs d'apprentissage pour mes projets futurs.

J'adresse ensuite mes sincères remerciements aux rapporteurs de cette thèse : Prof. Fransisco de A.T. de Carvalho et Prof. Miguel Couceiro, pour leurs lectures et rapports avisés, pour l'évaluation de mes contributions, leurs remarques pertinentes et leurs temps accordés bien gracieusement.

Je tiens également à remercier le reste des membres de mon jury à savoir : Prof. Nadia Ghazzali, Asso Prof. Ndèye Niang-Keita, Prof. Blaise Hanczar, Asso. Prof. Lazhar Labiod.

J'adresse toute ma gratitude à mes collègues de bureau, Rafika B. et Stanislas M. pour les discussions, les repas, les sorties, les encouragements et les autres moments partagés ensemble ; aux membres de l'équipe MLDS (Séverine A., Lazhar L., François R. et bien d'autres), aux doctorants du Centre Borelli, du LIPADE, du MAP5 ; aux membres de l'UFR Mathématiques et Infomatique de l'Université de Paris, du Cédric, des directions des ressources et de la recherche du CNAM et de l'UMR 1168 INSERM, ayant contribué de près ou de loin à la réalisation de ces travaux.

À titre personnel, je remercie chaleureusement ma famille et mes amis pour leur encouragement et leurs discussions scientifiques. Je tiens à remercier tout particulièrement mes parents pour leur soutien inconditionnel, leur dévouement et leurs précieux conseils qui ont su me façonner durant toutes ces années.

Résumé

Depuis l'avenènement du Big data, les techniques de réduction de la dimension sont devenues essentielles pour l'exploration et l'analyse de données hautement dimensionnelles issues de nombreux domaines scientifiques. En créant un espace à faible dimension intrinsèque à l'espace de données original, ces techniques offrent une meilleure compréhension dans de nombreuses applications de la science des données. Dans le contexte de l'analyse de textes où les données recueillies sont principalement non négatives, les techniques couramment utilisées produisant des transformations dans l'espace des nombres réels (par exemple, l'analyse en composantes principales, l'analyse sémantique latente) sont devenues moins intuitives car elles ne pouvaient pas fournir une interprétation directe. De telles applications montrent la nécessité de techniques de réduction de la dimensionnalité comme la factorisation matricielle non négative (NMF), utile pour intégrer par exemple, des documents ou des mots dans l'espace de dimension réduite. Par définition, la NMF vise à approximer une matrice non négative par le produit de deux matrices non négatives de plus faible dimension, ce qui aboutit à la résolution d'un problème d'optimisation non linéaire. Notons cependant que cet objectif peut être exploité dans le domaine du regroupement de documents/mots, même si ce n'est pas l'objectif de la NMF. En s'appuyant sur la NMF, cette thèse se concentre sur l'amélioration de la qualité du clustering de grandes données textuelles se présentant sous la forme de matrices document-terme très creuses. Cet objectif est d'abord atteint en proposant plusieurs types de régularisations de la fonction objectif originale de la NMF. En plaçant cet objectif dans un contexte probabiliste, un nouveau modèle NMF est introduit, apportant des bases théoriques pour établir la connexion entre la NMF et les modèles de mélange finis de familles exponentielles, ce qui permet d'offrir des régularisations intéressantes. Cela permet d'inscrire, entre autres, la NMF dans un véritable esprit de clustering. Enfin, un modèle bayésien de blocs latents de Poisson est proposé pour améliorer le regroupement de documents et de mots simultanément en capturant des caractéristiques de termes bruyants. Ce modèle peut être relié à

la NMTF (Nonnegative Matrix Tri-Factorization) consacrée au co-clustering. Des expériences sur des jeux de données réelles ont été menées pour soutenir les propositions de la thèse.

Mots-clés : classification croisée, factorisation, modèles des blocs latents, modèles de mélanges, text mining.

Abstract

Since the exponential growth of available Data (Big data), dimensional reduction techniques became essential for the exploration and analysis of high-dimensional data arising from many scientific areas. By creating a low-dimensional space intrinsic to the original data space, theses techniques offer better understandings across many data Science applications. In the context of text analysis where the data gathered are mainly nonnegative, recognized techniques producing transformations in the space of real numbers (e.g. *Principal component analysis*, *Latent semantic analysis*) became less intuitive as they could not provide a straightforward interpretation. Such applications show the need of dimensional reduction techniques like Nonnegative Matrix factorization (NMF) useful to embed, for instance, documents or words in the space of reduced dimension. By definition, NMF aims at approximating a nonnegative matrix by the product of two lower dimensional nonnegative matrices, which results in the solving of a nonlinear optimization problem. Note however that this objective can be harnessed to document/word clustering domain even it is not the objective of NMF. In relying on NMF, this thesis focuses on improving clustering of large text data arising in the form of highly sparse document-term matrices. This objective is first achieved, by proposing several types of regularizations of the original NMF objective function. Setting this objective in a probabilistic context, a new NMF model is introduced bringing theoretical foundations for establishing the connection between NMF and Finite Mixture Models of exponential families leading, therefore, to offer interesting regularizations. This allows to set NMF in a real clustering spirit. Finally, a Bayesian Poisson Latent Block model is proposed to improve document and word clustering simultaneously by capturing noisy term features. This can be connected to NMTF (Nonnegative Matrix factorization Tri-factorization) devoted to co-clustering. Experiments on real datasets have been carried out to support the proposals of the thesis.

Keywords : co-clustering, factorization, latent block models, mixture models, text mining.

Table des matières

Liste des tableaux

Table des figures

Introduction

In day-to-day situations, a humongous amount of data (numbers, texts, images, videos, etc) is created around the world and stored into different entities. Taken individually, the majority of this data is rather self-explanatory. However, gathered into a large collection, it may become challenging to evaluate or synthesize the information. Throughout the years, Machine Learning has significantly increased its lead into processing automatically such collections with the arriving of more powerful computational resources and improved fitting models for the data. For those large collections that can be retranscripted in the form of a highly dimensional data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, a key area referred to as dimensional reduction was to create a low-dimensional representation of the original data. For this task, one particular technique called *Singular Value Decomposition* (SVD) has led for many years the award of dimensional reduction with applications across almost every domain. In the analysis of text collections transcripted under the format of document-term matrices s.t. $\boldsymbol{X} \in \mathbb{R}_{+}^{n \times d}$, a corresponding equivalent to SVD was made by the *Latent Semantic Analysis* (LSA) [1].

While real numbers appear in many domains (e.g. finance, meteorology, etc), those collecting data such as occurrences, concentrations of substances, images, probabilities, signals or more generally any nonnegative values would vouch for retaining a similar space of definition for the low-dimensional data in order to derive a interpretable meaning. In this sense, SVD based techniques such as *Principal Component Analysis* (PCA) (which in addition required the data matrix to be centered)[2, 3] or LSA would trouble or prevent this interpretation by allowing in their factors the presence of negative values which are initially not defined in the original data space and out of sense for the considered domain. This eventually emphasized the evident need of techniques for nonnegative data such as *Nonnegative Matrix factorization* (NMF)[4].

0.1 Motivation

As its origin will suggest, NMF was not designed for clustering purposes, however, in the last decades, many applications led to observe its potential in this area [5]. Ever since, the underlying equivalence between NMF and precursors clustering techniques such as K-means [6] or powerful modeling tools such as Finite mixture model under certain conditions were often mentioned but not formally explained. In the case of text analysis, several extensions of NMF have been suggested, however, with the majority using an error function not necessarily adapted to the analysis of document-term matrices (e.g. Frobenius). In this thesis, we undertake the problem of improving NMF document clustering

using *cluster ensembles*, regularized objective functions, *Neural Word Embedding* and *Information theory*. Furthermore, we propose to cast NMF as a clustering optimization problem and describe the exact relation between NMF, Kmeans-like algorithms and Finite Mixture Models. Ultimately, several studies of the clustering performances of our proposal are achieved on highly dimensional and sparse text data. Special attention is given to document-term matrices lying in the set of unit-sphere as this normalization has been automatically applied for practicing NMF document clustering throughout the years and found dedicated distributions in some continuous mixtures models relying for instance on von Mises-Fisher distributions [7]. Several insights toward improving the clustering performance of NMF and finite Mixture Models are also given.

0.2 Thesis outlines

— Chapter 1 : "**Preliminaries**" reviews the existing techniques and algorithms for dimensional reduction and clustering as for the probabilistic knowledge required for mixture models.

— Chapter 2 : "**Nonnegative Matrix Factorization**" is an extended review of NMF, its several variants and extensions, as well as the common algorithms used in practice to obtain local minima. A comparison of NMF with respect to the Frobenius norm and the generalized Kullback-Leibler divergence using the solutions of the multiplicative updates algorithm is given. Furthermore, a study of the local minima regarding the clustering quality highlights several limitations and the need of a consensus approach to extract better document clustering partitions out of NMF.

— Chapter 3 : **Nonnegative Matrix Factorization with semantic leveraging** details approaches taken with NMF in the last decade and presents several regularizations of the original NMF objective made to improve document clustering. A first approach consists in decomposing the data matrix and a graph of semantic contextual relationships simultaneously into a shared low-dimensional subspace. A second approach aims at leveraging subordinates semantic relations (such as hyponyms) using the Kantorovich–Rubinstein (Wasserstein) metric to obtain regularization embeddings.

— Chapter 4 : "**Toward probabilistic factors for NMF and connections with Finite Mixture Models**", initiates the optimization of a more constrained objective in order to embed NMF into a straightforward clustering problem. By transforming one subspace into a set of probability distributions, and considering the class of entropy functionals from the Rényi family to regularize

the NMF objective, this optimization results in major improvements of the clustering performance. In addition, thanks to further properties of the approximation metric (Frobenius norm, I-divergence, Itakura-Saito and any Bregman divergences), exact connections between NMF, K-means and several Finite Mixture Models using exponential distributions (e.g. Gaussian, von Mises-Fisher, Poisson) are established.

— Chapter 5 : "**Gamma-Poisson Latent Block Model for noisy text data**" draws the attention toward improving clustering of document-term matrices using co-clustering techniques of Poisson mixture. By taking advantages of the parameterization offered by the Latent Block Model, we developed a coherent scheme for capturing noisy text features using Poisson mixture. The estimation of the parameters is first achieved using Maximum Likelihood. Later on, Bayesian Inference and Monte Carlo Markov Chain (MCMC) are used to estimate the parameters and address the overfitting mixture (empty cluster solution) issue arising with finite mixture model.

— **Conclusion and Perspectives** reviews the main contributions in the thesis in terms of clustering, co-clustering and data embedding contexts, and propose some perspectives.

Notation glossary

$\mathbb{C}$	field of complex numbers
$\mathbb{Q}$	field of rational numbers
$\mathbb{N}$	field of natural numbers
$\mathbb{N}^*$	field of natural numbers with 0 excluded
$\mathbb{R}$	field of real numbers
$\mathbb{R}^n$	set of real vectors of size n
$\mathbb{R}^{n\times d}$	set of real matrices of size $n \times d$
$\mathbb{R}^+$	set of nonnegative real numbers
$\mathbb{R}_+^n$	set of nonnegative real vectors of size n
$\mathbb{R}_+^{n\times d}$	set of nonnegative real matrices of dimension $n \times d$
$\mathbb{R}^-$	set of non-positive real numbers
$\mathbb{R}_-^n$	set of non-positive real vectors of size n
$\mathbb{R}_-^{n\times d}$	set of non-positive real matrices of dimension $n \times d$
$\mathbb{S}^{d-1}$	(d-1) dimensional unit-sphere embedded in $\mathbb{R}$
$U(.,.)$	transportation polytope
Δ_n	probability simplex of size n
$\equiv$	equivalent to
$\leftarrow$	assignment operator
$:=$	assignment operator
$\forall$	for all
$\iff$	if and only if
$\|.\|_F$	Frobenius norm/spectral norm (matrix)
$\|.\|_2$	Euclidean norm (vector)
$\langle .,. \rangle$	inner product
$\langle .,. \rangle_F$	Frobenius dot product/matrix inner product
$\odot$	Hadamard product (element-wise matrix multiplication)
$\frac{A}{B}$	Hadamard/element-wise division of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
A^p	Hadamard/element-wise power of matrix $\boldsymbol{A}$ of order p
$\top$	transpose operator $(.)^\top$
G	$G = (1/g, \ldots, 1/g)^\top$ is a vector of size g with all components equals to $\frac{1}{g}$
$\boldsymbol{e_k}$	unit vector s.t. $\boldsymbol{e_k} = (0, 0, \ldots, \underset{k-thposition}{1}, \ldots, 0)^\top$
$\mathbf{1}_n$	vector of all ones of size n

$\mathbf{I}_n$	identity matrix of size $n \times n$
$\boldsymbol{x}$	$\boldsymbol{x} = (x_1, \ldots, x_n)^\top$ a natural vector
$\boldsymbol{x}_i$	$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})$ is a row-vector in $\boldsymbol{X}$
$\boldsymbol{x}_j$	$\boldsymbol{x}_j = (x_{j1}, \ldots, x_{jn})^\top$ a column vector in $\boldsymbol{X}$
x_{ij}	(i,j)th entry of matrix $\boldsymbol{X}$
$x_{i.}$	i-th row marginal of $\boldsymbol{X}$, s.t. $x_{i.} = \sum_j^d x_{ij}$
$x_{.j}$	j-th column marginal of $\boldsymbol{X}$, s.t. $x_{.j} = \sum_i^n x_{ij}$
$W_p(.,.)$	Wasserstein distance
$W_p^\lambda(.,.)$	Smoothed Wasserstein distance
$H(.)$	Shannon entropy
$H_\alpha(.)$	Rényi entropy
$D_{KL}(.\|\|.)$	Kullback-Leibler divergence
$D_I(.\|\|.)$	generalized Kullback-Leibler divergence/I-divergence
$D_{IS}(.\|\|.)$	Itakura-Saito divergence
$\text{rank}(\boldsymbol{X})$	rank of matrix $\boldsymbol{X}$
$\text{ran}(\boldsymbol{X})$	range of matrix $\boldsymbol{X}$
$\dim(\boldsymbol{X})$	dimension of matrix $\boldsymbol{X}$
$\text{diag}(\boldsymbol{x})$	diagonal matrix with $\boldsymbol{x}$ on the diagonal
$\text{span}(.)$	span of a vectors set
$\text{dom}(.)$	domain of definition
$\text{int}(\mathcal{X})$	interior of a set $\mathcal{X}$
$\text{ri}(\mathcal{X})$	relative interior of a set $\mathcal{X}$
$\text{Tr}(\boldsymbol{X})$	trace of a square matrix $\boldsymbol{X}$
$\det(\boldsymbol{X})$	determinant of a square matrix $\boldsymbol{X}$
$[\boldsymbol{X}]^+$	positive orthant of $\boldsymbol{X}$
$[\boldsymbol{X}]^-$	negative orthant of $\boldsymbol{X}$
$\prod_{a,c,e}^{b,d,f} \cdots$	$\prod_{a=1}^b \prod_{c=1}^d \prod_{e=1}^f \cdots$
$\sum_{a,c,e}^{b,d,f} \cdots$	$\sum_{a=1}^b \sum_{c=1}^d \sum_{e=1}^f \cdots$
I	canonical set of sample/document indexes s.t. $I = \{1, \ldots, n\}$
J	canonical set of feature/term indexes s.t. $J = \{1, \ldots, d\}$
I_k	subset of row indexes in the cluster k
J_ℓ	subset of column indexes in the cluster ℓ
$I_k \times J_\ell$	subset of row and column indexes in the co-cluster/block of indice (k, ℓ)
$\boldsymbol{\pi}$	vector of proportions s.t. $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)^\top \in \Delta_g$
$\boldsymbol{\rho}$	vector of proportions s.t. $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c)^\top \in \Delta_c$

Acronyms

NMF	Nonnegative Matrix Factorization
NMTF	Nonnegative Tri-Matrix Factorization
SNMF	Semantic Nonnegative Matrix Factorization
SNMF	Spherical Nonnegative Matrix Factorization
WE-NMF	WE-NMF Nonnegative Matrix Factorization
`cNMF`	constrained Nonnegative Matrix Factorization
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
LRA	Low-Rank Approximation
LSA	Latent Semantic Analysis
FMM	Finite Mixture Model
PLSA	Probabilistic Latent Semantic Analysis
LBM	Latent Block Model
EM	Expectation-Maximization
VEM	Variational Expectation-Maximization
SEM	Stochastic Expectation-Maximization
CEM	Classification Expectation-Maximization
MH	Metropolis-Hasting
NSPLBM	Noisy Sparse Poisson Latent Block Model
B-NSPLBM	Bayesian Noisy Sparse Poisson Latent Block Model
MI	Mutual Information
PMI	Pointwise Mutual Information
ACC	Accuracy
NMI	Normalized Mutual Information
ARI	Adjusted Rand Index

Chapitre 1

Preliminaries

This thesis requires basic knowledge on matrix theory. The notation is made to be intuitive throughout the paper. However, if needed, the reader can refer to Appendix A where more details on the notation and reminders on matrix theory (e.g. vector space, operations, norms, distances, eigen decomposition) are given.

1.1 Data representation in text analysis

Text data naturally arise in a unstructured fashion (e.g. text collections, tweets, articles, metadata, etc) and require some sort of representation to be analyzed or process automatically. The need of processing theses large collections of text is essentially what describes the origin of Information Retrieval (IR) and therefore IR summarizes the set techniques for searching for information in documents or for documents themselves or any types of text. Three mathematical basis are essentially considered for representing and analyzing text data. Set-theoritic models based on boolean logic (e.g. the Boolean model or the extended Boolden model [8]), algebric models based on linear algebra (e.g. Vector space Model [9], [10]) and probabilistic Models. In terms of data representation, all basis tend to represent the data in the form of a vector space and differ rather according to the mathematical approach used for analyzing. A common methodology used to provide a vector space is to primarily consider a Bag-of-words. Intuitively, this method allows us to represent each text input (e.g. a sentence or a document) as an exhaustive list of terms (referred to as a bag), disregarding of grammar, punctuation, perhaps numbers or any irrelevant vocabulary inputs defined as stopwords (e.g. repetitive words or the most common words). Note that the stopwords may be carefully picked by the user or be

domain-specific. For every terms kept in the list, the model also retained its frequency (or occurrence) in the original text input. Afterwards, the list of bags results in a structured text collections from which we can construct a document-term matrix $\boldsymbol{X}$ representative of n documents and d terms where the (i,j)-th scalar denotes the occurrence of the term j in the document i as illustrated below :

$$\begin{array}{c} \\ d_1 \\ \vdots \\ d_i \\ \vdots \\ d_n \end{array} \begin{array}{c} \begin{array}{ccc} t_1 \dots & t_j & \dots t_p \end{array} \\ \begin{pmatrix} & \vdots & \\ \dots & x_{i,j} & \dots \\ & \vdots & \end{pmatrix} \end{array}.$$

Eventually, the resulting matrix is often sparse and highly dimensional. From this representation, several IR indexing techniques [11] adapted to vector spaces can take place in order to exhibit further noisy samples or key features and narrow the relevant information. Their are referred to as automatic indexing (or subject indexing) and highlight two notions of automatic indexing : *exhaustivity* which expresses how deeply the various topics of a document are reflected in the list of terms (a high exhaustivity increases the likelihood that all the relevant articles are being retrieved) and *specificity* which expresses how exactly a term characterizes a given topic (a high specificity increases the likelihood of retrieving articles that describes the topic precisely). In this thesis, we uses the later which arises in the form of a term weighted normalization called *TF-IDF* [12, 13]. Considering a document-term matrix $\boldsymbol{X} = (x_{ij})_{n\times d}$ with a set of rows $I = \{i : 1, \dots, n\}$ and columns $J = \{j : 1, \dots, d\}$, and its indicator matrix $\bar{\boldsymbol{X}} \in \{0, 1\}^{n\times d}$, TF-IDF can be stated as the following function :

$$\text{TF-IDF}(x_{ij}) = \text{TF}_{ij} \times \log \frac{n}{\sum_i \bar{x}_{ij}}, \tag{1.1}$$

where $\text{TF}_{ij} = x_{ij}$ is the *term frequency* in the document i and $\log \frac{n}{\sum_i \bar{x}_{ij}}$ the *Inverse Document Frequency* (IDF) which is a magnitude scaled by the number of documents in which the term j appears.

Further cleaning techniques such as Stemming or Lemming are frequently applied but are not discussed in this section. For in-depth information, the readers can refers to [14, 15].

Remark. In information retrieval, the use of clustering relies on the assumption that if a document is relevant to a query, then other documents in the same cluster can also be relevant. This hypothesis can be used at different stages in the information retrieval process, the two most notable being : cluster-based retrieval to speed up search, and search result clustering to help users navigate and understand

what is in the search results. The document clustering which still remains a hot topic can be tackled under different approaches.

1.2 Clustering and Co-clustering

Clustering (or cluster analysis) is the task dedicated to identifying groups inside a population such that individuals belonging to the same group are highly similar between each others. Let $\boldsymbol{X}$ be a data matrix of size $n \times d$ with a set of individuals $I = \{i : 1, \ldots, n\}$ partitioned into g groups, the partition can holds several representations :

— a labeling vector $\boldsymbol{z} = (z_1, \ldots, z_n)^\top \in \{1, \ldots, g\}^n$ where each variable z_i equals the label of the group in which i is in,

— a hard classification matrix $\boldsymbol{Z} = (z_{ik}) \in \{0, 1\}^{n \times g}$ where $z_{ik} = 1$ if i belongs to the group k otherwise $z_{ik} = 0$,

— a soft classification matrix $\boldsymbol{Z} = (z_{ik}) \in [0, 1]^{n \times g}$ where z_{ik} is a conditional probability s.t. $\sum_k z_{ik} = 1$.

In the first two representations, the partition is said to be hard since each individual can only belong to one group. Clustering techniques building these representations are referred to as hard clustering. In the third representation, where the partition is a set of probability distributions, an individual can belong to several group. Clustering producing such partitions are referred to as soft/fuzzy clustering. Let $J = \{j : 1, \ldots, d\}$ be the set of features partitioned into c groups, s.t. $\boldsymbol{w} \in \{1, \ldots, c\}^d$ is the vector representation and $\boldsymbol{W} \in [0, 1]^{d \times c}$ is the classification matrix representation. Techniques seeking to identify simultaneously two partitions for both sets are called co-clustering. Table 1.1 illustrates the various representations. Several approaches for clustering are denoted (see [16] for a review) with perhaps one of the most popular being the **Hierarchical clustering**. This technique works in a sequential fashion and aims at generating a hierarchy of nested partitions depending on a distance function. As a consequence, this approach do not require the number of clusters in input. Two types of Hierarchical clustering are denoted :

— Hierarchical Agglomerative Clustering (HAC) referred to as the "bottom-up" approach where each data sample starts in its own cluster and clusters are consecutively paired to form the hierarchy. Different agglomerative criteria can be found in the literature such as "single-linkage" [17], complete-linkage [18], "average-linkage" [19] or the Ward criterion [20]. Lance and Williams

TABLE 1.1 – Data matrix $\boldsymbol{X}$ and associated row and column partitions indicated respectively by the representations $\boldsymbol{z}$ and $\boldsymbol{Z}$, $\boldsymbol{w}$ and $\boldsymbol{W}$, with $g = 3$.

		columns ($\mathcal{J}$)					$\boldsymbol{z}_{(n\times 1)}$	$\boldsymbol{Z} = (z_{ik})_{(n\times g)}$
		1	$\cdots$	j	$\cdots$	d		
rows ($\mathcal{I}$)	$\boldsymbol{x}_1$	x_{11}	$\cdots$	x_{1j}	$\cdots$	x_{1d}	1	1 0 0
	$\vdots$			$\vdots$			$\vdots$	$\vdots$
	$\boldsymbol{x}_i$	x_{i1}	$\cdots$	x_{ij}	$\cdots$	x_{id}	3	0 0 1
	$\vdots$			$\vdots$			$\vdots$	$\vdots$
	$\boldsymbol{x}_n$	x_{n1}	$\cdots$	x_{nj}	$\cdots$	x_{nd}	2	0 1 0

$\boldsymbol{w}_{(1\times d)}$	3	$\cdots$	1	$\cdots$	2

$\boldsymbol{W}^T = (w_{kj})$ $(g \times d)$					
	0	$\cdots$	1	$\cdots$	0
	0	$\cdots$	0	$\cdots$	1
	1	$\cdots$	0	$\cdots$	0

showed in [21] that many agglomerative clusterings are variations of a recurrence formulas for which further details are given in [22, 23, 24, 25, 26]. Standard HAC algorithn have time complexity of $\mathcal{O}(n^3)$ meaning that their not easily scalable for large datasets. A lower complexity of $\mathcal{O}(n^2)$ can be reach for some cases using the "single-linkage" or the "complete-linkage" [27].

— Hierarchical Divisive Clustering (HDC) referred to as the "top-down" approach were all the data samples start in one cluster which and clusters are subsequently split to form the hierarchy. Example of HDCs are given in [28, 23, 26]. Those approaches are less popular in practice due to their higher computational costs.

Other popular clustering techniques such as Graph-Based clustering (see the review of Schaeffer [29]), Spectral clustering [30] or Density-Based clustering [31, 32] are also denoted in the literature. For a complete review of clustering techniques, the reader can refer to these comprehensive reviews [33, 16, 22, 34].

1.2.1 Metric-based approach for partitioning

In this section, we describe clustering algorithms that optimize a criterion derived directly from the notion of distance or dissimilarity in order to determine an appropriate partition of the data.

1.2.1.1 One-way clustering

Given a data matrix $\boldsymbol{X} = (x_{ij})_{n\times d}$, metric-based clustering represents a important class of partitioning clustering methods and aims at identifying a partition of g clusters of samples $\boldsymbol{x}_i$ given a chosen distance d computed between the samples and each cluster representative. The later can either be defined : (i) as a center/centroïd which refers in this case to the method of *K-means* (MacQueen et al. [6], also see Bock [35]) ; (ii) or be chosen from the set of samples which refers to the method of *K-medoids* introduced by Kaufman & Rousseeuw in [36, 28]. While K-medoids is less sensitive to outliers, K-means remains to this day one the most popular partitioning techniques. Such methods can be formulated as an optimization problem which can be solved by an iterative process with guaranty of finding a local minimum. In the case of K-means, the objective function takes the following form :

$$J(\boldsymbol{Z}, g) = \sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik} d(\boldsymbol{x}_i, \boldsymbol{\mu}_k) = \sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2, \tag{1.2}$$

where d is the squared *Euclidean* distance (or the sum of squares/SSQ), $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$ denote the i-th object, $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kd})^\top \in \mathbb{R}^d$ is the centroid of the cluster k and $z_{ik} \in \boldsymbol{Z}$ the cluster assignment defined as :

$$z_{ik} = \left\{ \begin{array}{ll} 1 & \text{if } k = \underset{k=1,\ldots,g}{\arg\min}\ d(\boldsymbol{x}_i, \boldsymbol{\mu}_k) \\ 0 & \text{otherwise} \end{array} \right\}.$$

Note that minimizing the squared *Euclidean* distance here is equivalent to minimizing the distance and preferred due to its strictly convex property and smoothness around near points [37]. Both K-means and K-medoids are special cases of a more general centroid-based clustering approach known as *méthode des nuées dynamiques* introduced by Diday in [38]. The latter makes it possible to have centroids of various forms, not necessarily vectors in $\mathbb{R}^d$. In terms of clustering of directional data (such as text data), a variant of k-means called the *Spherical K-means* [39] where d is set as the $(1 - \cos)$ dissimilarity has for objective function :

$$J(\boldsymbol{Z}, g) = \sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik}(1 - \cos(\boldsymbol{x}_i, \boldsymbol{\mu}_k)), \tag{1.3}$$

where $\cos(\boldsymbol{x}_i, \boldsymbol{\mu}_k)$ is given by the dot product since$\|\boldsymbol{x}_i\| = \|\boldsymbol{\mu}_k\| = 1$ and z_{ik} is set similarly as in K-means but according to the $(1 - \cos)$ dissimilarity.

Another variation of K-means is known as k-medians (see Jain and Dubes [34], and Bradley et al. [40]). It is based on the estimation of the median instead of the mean and minimizes the sum of absolute

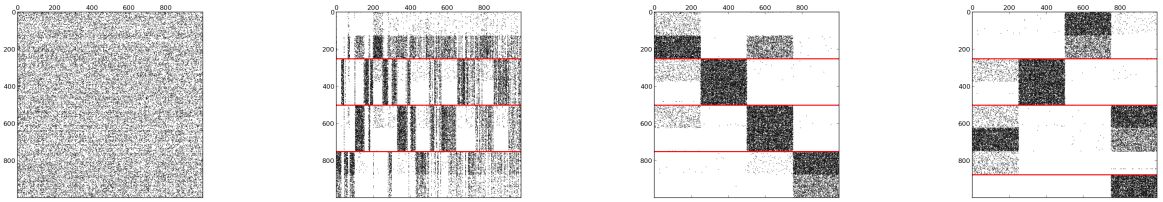
deviations (1-norm), which are equivalent to the Manhattan distance. A more recent generalization of the k-means principle to any *Bregman* divergence (for which the euclidean distance arises as a special case) has been proposed by Banerjee et al. in [41]. This relation will be of interest for the generalization of the transition from NMF to FMMs of Exponential Families, arsing in Chapter 4.

1.2.1.2 Two-way clustering (Co-clustering)

The earliest co-clustering approach that can be found in the literature is known as Direct Clustering or (Block clustering) and was proposed by Hartigan in [42]. The algorithm consists in identifying a partition of K block-clusters $\{B_1, \ldots, B_K\}$ in a data matrix $\boldsymbol{X} = (x_{ij})_{n \times d}$ by using the sum of squares between the observed data x_{ij} and the average value of x_{ij} inside a block denoted as b_k such as :

$$SSQ = \sum_{k=1}^{K} \sum_{i,j \in B_k} (x_{ij} - b_k)^2. \tag{1.4}$$

To reach a reasonable solution, Hartigan have proposed an algorithm adapted from a splitting algorithm used in one-way clustering that allows a set of rows I_k and columns J_k for a block B_k which results in the pair (I_k, J_k). A split of B_k corresponds to either a row or column "division" of B_k into two subsets B'_k and B''_k.



(a) (b) (c) (d)

FIGURE 1.1 – (a) : Original data - (b) : data reorganised according to the true row classes - (c) : data after diagonal co-clustering - (d) : data after non-diagonal co-clustering.

Other types of co-clustering have emerged, we might also mentioned the works of Govaert [43, 44, 45], Bock [46], Marcotorchino [47], Arabie and Hurbert [48], Trejos and Castillo [49] Castillo and Trejos [50], Duffy and Quiroz [51], Van mechelens and Scheppers [52], Rocci and Vichi [53], Labiod and Nadif [54] and Ailem et al. [55] who proposed a range of algorithms suitable for continuous, binary and count data (Figure 1.1). In Bio-informatics, two-way clustering approaches known as Bi-clustering have been applied in gene expression. Cheng and Church [56] proposed an algorithm called Node-deletion that

identifies partition with a *low mean squared residue* and allows block to overlap. Later, Cho et al. [57, 58] proposed a fast k-means like algorithm utilizing a similar measure and Guapta and Aggarwal [59] introduced the use of mutual information. Somes key surveys on co-clustering for biological data are proposed by Madeira and Oliveira[7], Tanay et al.[60] and Busygin et al. [61]. Several comparaisons and evaluations of Bi-clustering algorithms are proposed by Santamaría et al.[62] and Li et al[63]. More recently, Hanzcar and Nadif proposed an effective gene expression co-clustering method by developping a new bagging approach [64, 65], and also pointed later in [66] the problem of comparing biclusters obtained from several co-clustering algorithms.

1.2.2 Probabilistic modeling

In this section, we briefly review several generative models used in text mining including topic models for abstract data representation and model-based approaches primarily designed for cluster analysis. Most of the models presented in this section required the use of Exponential distributions. As a reminder, the reader can refer to Appendix B where we present several continuous and discrete Exponential distributions in terms of parameters, and their conjugate priors.

(a) FMM (b) LBM

FIGURE 1.2 – FMM and LBM as graphical models.

1.2.2.1 Mixture Models

Model-based clustering (or mixture models) is a class of generative techniques aiming at modeling a hidden distribution (the marginal) using a mixture of several known and relatively simpler distributions, e.g. Gaussian, Poisson, etc (see Figure 1.3). The distributions (referred to as a components) belong to the same family but vary in terms of parameters. The data $[\boldsymbol{x}_1|\dots|\boldsymbol{x}_n]^\top$ is assumed to be generated from that mixture of distributions. Mixture models with limited number of components

are called Finite Mixture Models (FMM) and are parametric. Models with unlimited (or unset) number of components are called Infinite Mixture Models (IMM) and are non-parametric. The marginal probability density function is described as follows :

$$f(\boldsymbol{X};\boldsymbol{\Theta}) = \prod_{i=1}^{n}\sum_{k=1}^{g}\pi_k f(\boldsymbol{x}_i;\boldsymbol{\theta}_k) \tag{1.5}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ are the components weights, $f(\boldsymbol{x}_i;\boldsymbol{\theta})$ is a probability density (or mass) function

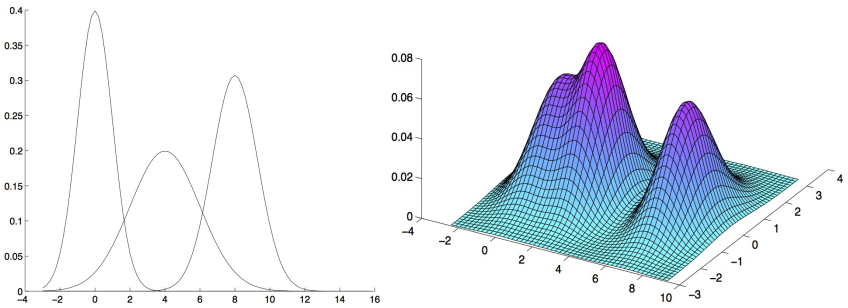


FIGURE 1.3 – Example of a mixture of 3 Gaussian distributions (univariate and multivariate).

with parameters $\in \boldsymbol{\theta}$ and $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ is the set of parameters of the model. Several optimization techniques for eq (1.5) which include Maximum Likelihood Estimation (MLE) and Bayesian Inference are detailed in the following sections (1.3.1 and 1.3.2). In the field of document clustering, we denote the work of Tantrum et al. [67] which proposed a FMM with the Gaussian distribution. However, the distribution was found to be inadequate. Dhillon and Sra [68] proposed a FMM using the von Mises-Fisher distribution (analogue of the Gaussian distribution for directional data) and demonstrated its relevance for document clustering. Subsequently, we can also mention the works of Banerjee and Gosh [69, 70] using the same distribution. More recently, Li and Zhang [71] as well as Rigouste et al [72] proposed their respective FMMs using the Multinomial distribution. Yin and Wang [73] proposed a Dirichlet Multinomial mixture model for short text clustering and later, Qiang et al. [74] proposed a Pitman-Yor process mixture model for the same application. On similar type of data (count data), we might also mention the work of Rau et al. [75] which uses Poisson mixtures to cluster digital gene expression profiles. We also denote several applications in supervised text analysis with the approach of Nigam [76, 77] which uses a Multinomial mixture model combined with a Naives Bayes, or McCallum et al. [78] which proposed to use the Multivariate Bernoulli for binarized document-term matrices.

1.2.2.2 Model-based K-means (mK-means)

Model-based K-means is a generalization of the standard k-means algorithm presented in Section 1.2.1.1 where the distance function is replaced by the log-likelihood of the observed data as follows :

$$\log p(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{x_i \in \boldsymbol{X}} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i}), \tag{1.6}$$

where $z_i = \underset{k=1,\ldots,g}{\arg\max}\ \log p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$. It can be shown that this model is a special case of the a mixture model with equal proportions where the log-likelihood is maximized using a Classification EM algorithm [79]. When $p(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i})$ is set as the Gaussian probability density function, with equal variance, mK-means collapses to the original K-means algorithm [6]. Under the same conditions, a similar deduction can be made when $p(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i})$ is the von Mises-Fisher distribution pdf which collapses to the Spherical K-means [39] algorithm presented in Section 1.2.1.1.

1.2.2.3 Latent Block Models

The *Block Mixture Model* (BMM) or *Latent Block Model* (LBM) was introduced by Govaert and Nadif in [80] for co-clustering of binary data. This model-based approach aims at identifying a pair of partitions $(\boldsymbol{z}, \boldsymbol{w})$ of $g \times c$ co/block-clusters. Let $\boldsymbol{X} = (x_{ij})$ be a data matrix of size $n \times d$ with a set of rows I and columns J, the purpose is to model a marginal distribution where each observations x_{ij} are assumed to be generated by block. Each observation x_{ij} are assumed to be iid. The marginal density is denoted as follows :

$$f(\boldsymbol{X}; \boldsymbol{\Theta}) = \sum_{(\boldsymbol{z},\boldsymbol{w}) \in \mathcal{Z} \times \mathcal{W}} p(\boldsymbol{z}) p(\boldsymbol{w}) \prod_{i,j}^{n,d} f(x_{ij}; \boldsymbol{\theta}_{z_i w_j}), \tag{1.7}$$

where $\mathcal{Z} \times \mathcal{W}$ denote the set of all possible partitions $I \times J$. $p(\boldsymbol{z}) = \prod_i^n \pi_{z_i}$ and $p(\boldsymbol{w}) = \prod_j^d \rho_{w_j}$ where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c)$ are the proportions for the row and column partitions respectively.

LBM for contingency table/two-way tables. Later, Nadif and Govaert [81] proposed an application of LBM on contingency tables using the Poisson distribution and a then several others [82, 83]. Considering $\boldsymbol{X}$ as a contingency table where the set of row and column indexes I and J are now assimilated to the modalities of two categorical variables. Each observation x_{ij} can now be described by a joint probability and the marginal distributions $\frac{x_{1\cdot}}{N}, \ldots, \frac{x_{n\cdot}}{N}$ and $\frac{x_{\cdot 1}}{N}, \ldots, \frac{x_{\cdot d}}{N}$ of the data matrix are therefore the

probability distributions of each modalities in I and J respectively. The hypothesis of independence can therefore be formulated as follows :

$$H_0 : \frac{x_{ij}}{N} = \frac{x_{i.}}{N} \times \frac{x_{.j}}{N}, \forall (i = 1, \ldots, n \text{ and } j = 1, \ldots, d). \tag{1.8}$$

Following this hypothesis, LBM can provide a parameterization taking in consideration a row and column effect along side the block parameter.

Several procedures for optimizing eq (1.7) including Maximum Likelihood Estimation (MLE) and Bayesian Inference are detailed in sections (1.3.1 and 1.3.2). The reader can refer to [84] for more details on co-clustering using LBM.

1.2.2.4 Topic Modeling

The *Topic model* is one of the most popular generative model in text analysis. Considering a document-term matrix $\boldsymbol{X} = (x_{ij})_{n \times d}$, in contrast to conventional approaches which interpret the documents according to the entire set of terms, Topic models attempt to explain the documents according to the hidden concepts (topics) embedded in the terms. Papadimitriou et al [85] described an early Topic model referred to as *probabilistic Latent Semantic Indexing* (PLSI) in which a document is seen as a probability distribution that is the convex combination of a small number of topics while a topic is as a probability distributions on terms. Another generative model inspired by LSA is *Probabilistic LSA* (PLSA) [86, 87]. As opposed to LSA which uses SVD to map the set of documents and terms in a lower dimensional vector space by using *Singular Value Decomposition*, PLSA is based on the statistical model called the *Aspect* model which defines a latent class variable (or partition) $\mathcal{Z} = (z_1, \ldots, z_K)$ over the set of all observations x_{ij}. Therefore, the joint probability of the model for one observation is expressed as :

$$p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{z \in \mathcal{Z}} p(\boldsymbol{x}_i | \boldsymbol{z}) p(\boldsymbol{x}_j | \boldsymbol{z}) p(\boldsymbol{z}). \tag{1.9}$$

Subsequently, the model parameter can be estimated using the EM algorithm (described in Section 1.3.1). By considering a Multinomial distribution denoted by $p(|z)$, the model can be reformulated in a matrix notation such as $\boldsymbol{U} = (p(\boldsymbol{x}_i|z_k))_{ik}$, $\boldsymbol{V} = (p(\boldsymbol{x}_j|z_k))_{jk}$ and $\boldsymbol{\Sigma} = \text{diag}(p(z_1), \ldots, p(z_g)) \in \mathbb{R}_+^{g \times g}$ which allows the joint probability to be expression in the following matrix product $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ for which an analogy can be established with LSA (see Section 1.6.4).

PLSA was also shown by Gaussier and Goutte [88] to be equivalent to NMF. Assuming that $\sum_{ij} x_{ij} = 1$ and denoting the approximation $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}^\top$ where $\boldsymbol{W}^{n\times g}$ and $\boldsymbol{H}^{d\times g}$ such that, $w_{ik} = p(\boldsymbol{x}_i|z_k)$, $h_{jk} = p(\boldsymbol{x}_j|z_k)$ and $[\boldsymbol{W}\boldsymbol{H}^\top]_{ij} = p(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Introducing two diagonals scaling matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{g\times g}$ such that $\sum_k w_{ik} = a_{kk}$ and $\sum_j h_{jk} = b_{kk}$. $\boldsymbol{W}\boldsymbol{H}^\top$ can be re expressed as :

$$\boldsymbol{W}\boldsymbol{H}^\top = (\boldsymbol{W}\boldsymbol{A}^\top\boldsymbol{A})(\boldsymbol{H}\boldsymbol{B}^\top\boldsymbol{B})^\top = (\boldsymbol{W}\boldsymbol{A}^\top\boldsymbol{A})(\boldsymbol{B}^\top\boldsymbol{B}\boldsymbol{H}^\top) = (\boldsymbol{W}\boldsymbol{A}^\top)(\boldsymbol{A}\boldsymbol{B}^\top)(\boldsymbol{B}\boldsymbol{H}^\top), \tag{1.10}$$

where $[\boldsymbol{W}\boldsymbol{A}^\top]_{ik} = p(\boldsymbol{x}_i|z_k)$, $[\boldsymbol{B}\boldsymbol{H}^\top]^\top_{jk} = p(\boldsymbol{x}_i|z_k)$ and $[\boldsymbol{A}\boldsymbol{B}^\top]_{kk} = p(z_k)$. In addition, the authors showed that by joining the EM formulas $(p(z_k), p(\boldsymbol{x}_i|z_k))$ to re express $p(\boldsymbol{x}_i|z_k)$ and $(p(z_k), p(\boldsymbol{x}_i|z_k))$ to re express $p(\boldsymbol{x}_i|z_k)$, PLSA could be rewritten in the form of two multiplicative updates similarly as those derived from the objective function of NMF with the I-divergence, which states that PLSA any local maximum likelihood point estimate is a local minimum point for NMF with the I-divergence. Thereafter, Ding et al. [89] showed that PLSA and NMF optimize the same objective function, but the algorithms remain different and converge in practice to different solutions even if they share that fixed point property.

Latent Dirichlet Allocation is another topic model, perhaps the more popular. Let $\mathcal{V}$ be an ordered set of terms called vocabulary (or dictionary) of size d. A term is expressed by a unit vector $\boldsymbol{t} = (t_j) \in \{0,1\}^d$ where $t_j = 1$ at the position of the term in $\mathcal{V}$ and 0 elsewhere. Therefore, each document $\boldsymbol{x}_i$ denote a sequence of N_i terms such that a document is express as the vector $\boldsymbol{x}_i = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N_i})$. Subsequently, the corpus of n documents noted $\mathcal{D}$ can be expressed as the following set $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. Following this notation, for a number of topics g, LDA is described as the subsequent generative process for each document $\boldsymbol{x}_i$ in the corpus, given by Algorithm 1.

Algorithm 1 LDA generative process

Choose $N_i \sim Poisson(\zeta)$
Choose $\boldsymbol{\theta} \sim Dirichlet(\boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_g)$
for $j' = 1, \ldots, N_i$ **do**
  Choose a topic $w_{j'} \sim Multinomial(\boldsymbol{\theta})$
  Choose a term $t_{j'}$ according to the conditional probability on the topic $p(t_{j'}|w_{j'}, \boldsymbol{W})$
end for

where $\boldsymbol{W} = (w_{jk}) \in \{0,1\}^{d\times g}$ is the matrix of the terms conditional probabilities such that $w_{jk} = p(t_j = 1|w_{j'} = k)$, $\boldsymbol{w} = (w_1, \ldots, w_d)$ is the latent class (or topic) variable of the terms in $\mathcal{V}$. Note that the N_i are independent of the generating variables $\boldsymbol{\theta}$ and $\boldsymbol{w}$. The marginal distribution of

a document is therefore equal to :

$$p(\boldsymbol{x}_i|\boldsymbol{\alpha}, \boldsymbol{W}) = \int p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \Big(\prod_{j'=1}^{N_i} \sum_{w_{j'}} p(w_{j'}|\boldsymbol{\theta}) p(t_{j'}|w_{j'}, \boldsymbol{W}) \Big) d\boldsymbol{\theta}. \tag{1.11}$$

The inference can be achieved through Laplace approximation, variational approximation or Monte Carlo Markov Chain methods.

1.3 Probabilistic modeling inference

In this section, we detail the two approaches taken in this thesis for learning the set of parameters $\boldsymbol{\Theta}$ of FMMs and BMMs arising in the next chapters. The first two sections are overviews of the common techniques, while sections 1.3.1 and 1.3.2 bring foundations for (i) the inference with BMMs and (ii) solving intractable problems arising in Bayesian statistics.

1.3.1 Maximum Likelihood Estimation (MLE)

Following the formulation of a Mixture model given in the prior Section 1.2.2.1 where the likelihood is stated as follows :

$$f(\boldsymbol{X}; \boldsymbol{\Theta}) = \prod_{i=1}^{n} \sum_{k=1}^{g} \pi_k f(\boldsymbol{x}_i; \boldsymbol{\theta}), \tag{1.12}$$

Several approaches can be taken in order to maximize the Likelihood of the model. The first approach optimizes the log-likelihood of the data defined as :

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{n} \log \Big(\sum_{k}^{g} \pi_k f(\boldsymbol{x}_i; \boldsymbol{\theta}) \Big), \tag{1.13}$$

by using the Expectation-Maximization algorithm (EM) introduced by Demptser et al. [90]. This algorithm is based upon the notion that there exists a mapping between the observed data $\boldsymbol{X}$ and a unobserved hidden variable $\boldsymbol{z}$ referred to as the complete-data s.t. :

$$p(\boldsymbol{z}|\boldsymbol{X}; \boldsymbol{\Theta}) = \frac{p(\boldsymbol{z}, \boldsymbol{X}; \boldsymbol{\Theta})}{p(\boldsymbol{X}; \boldsymbol{\Theta})} \tag{1.14}$$

with $\mathcal{L}(\boldsymbol{\Theta})$ which can be re-written as follows :

$$\mathcal{L}(\boldsymbol{\Theta}) = \log p(\boldsymbol{X}; \boldsymbol{\Theta}) = \log p(\boldsymbol{z}, \boldsymbol{X}; \boldsymbol{\Theta}) - \log p(\boldsymbol{z}|\boldsymbol{X}; \boldsymbol{\Theta}), \tag{1.15}$$

where $\mathcal{L}_c(\boldsymbol{\Theta}) = p(\boldsymbol{z}, \boldsymbol{X}; \boldsymbol{\Theta}) = p(\boldsymbol{X}|\boldsymbol{z}; \boldsymbol{\Theta}) \times p(\boldsymbol{z}; \boldsymbol{\Theta})$ is known as the complete data-likelihood. Note that the likelihood of the model can be rewritten as proposed in [80] to integrate the notion of the complete data $(\boldsymbol{z}, \boldsymbol{X})$ as follows :

$$f(\boldsymbol{X}; \boldsymbol{\Theta}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} p(\boldsymbol{z}) p(\boldsymbol{X}|\boldsymbol{z}; \boldsymbol{\theta}), \tag{1.16}$$

where $\mathcal{Z}$ denotes the set of all partitions $\boldsymbol{z}$, $p(\boldsymbol{X}|\boldsymbol{z}; \boldsymbol{\theta}) = \prod_i^n f(\boldsymbol{x}_i; \theta_{z_i})$ and $p(\boldsymbol{z}) = \prod_i^n \pi_{z_i}$. In order to maximize $\mathcal{L}(\boldsymbol{\Theta})$, EM uses a heuristic to obtain an estimate $\boldsymbol{\Theta}^*$ that maximizes the expectation of $\mathcal{L}_c(\boldsymbol{\Theta})$ given the observed data $\boldsymbol{X}$ and the current estimate $\boldsymbol{\Theta}^{(t)}$ [90]. This heuristic is closely related to the Maximum A Posteriori (MAP) principle which will defined in the next section for Bayesian inference. Note that this principle do not require derivation of $\mathcal{L}(\boldsymbol{\Theta})$. In the following, we replace $\boldsymbol{z}$ by the classification matrix format $\boldsymbol{Z}$ of conditional probabilities z_{ik}. The conditional expectation of the complete data log-likelihood is expressed by the $\mathcal{Q} - function$ as follows :

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)}) &= \mathbb{E}[\log p(\boldsymbol{Z}, \boldsymbol{X}; \boldsymbol{\Theta}) | \boldsymbol{X}, \boldsymbol{\Theta}^{(t)}] \\
&= \sum_{i,k} \mathbb{E}[z_{ik}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}] \log\left(\pi_k f(\boldsymbol{x}_i; \theta_k)\right) \\
&= \sum_{i,k} p(z_{ik} = 1|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) \log\left(\pi_k f(\boldsymbol{x}_i; \theta_k)\right) = \sum_{i,k} z_{ik}^{(t)} \log\left(\pi_k f(\boldsymbol{x}_i; \theta_k)\right)
\end{aligned} \tag{1.17}$$

where $\boldsymbol{\Theta}^{(t)}$ is the current estimate of $\boldsymbol{\Theta}$. $z_{ik}^{(t)} = \frac{\pi_k^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\theta}_k^{(t)})}{\sum_{k'} \pi_{k'}^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\theta}_{k'}^{(t)})}$ will be estimated at the Expectation phase and $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\theta}\}$ during the maximization phase such that $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$ is maximized. The EM procedure is given in Algorithm 2. The second approach is called the Classification Maximum

Algorithm 2 Expectation-Maximization (EM)

initialize $\boldsymbol{Z}^{(0)}, \boldsymbol{\Theta}^{(0)}$
repeat
  E-Step : compute $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$
  M-Step : choose $\boldsymbol{\Theta}^{(t+1)}$ which maximizes $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$
until convergence

Likelihood (CML) and was proposed by Symons in [91, 92]. This alternative consists in adding the classification labels $\boldsymbol{z}$ in the set of parameters $\boldsymbol{\Theta}$. $\boldsymbol{z}$ is estimated from the conditional probabilities computed in the EM algorithm such as $z_i = \underset{k=1,\ldots,g}{\arg\max}\, z_{ik}^{(t)}$. This optimization is referred to as the Classification EM (CEM) algorithm (introduced by Celeux and Govaert [79]) and is achieved by inserting a classification phase (C-Step) before the maximization phase (M-step) in the EM algorithm. While

this approach has several benefits in terms of scalability and convergence time, it might increase the tendency of overffiting mixture.

1.3.2 Bayesian Inference

Let θ be a parameter of our mixture model, using the maximum Likelihood allows us to obtain a point estimate θ^* that maximizes the likelihood of the model. In contrast, by setting a *prior* probability distribution for θ, Bayesian inference will allow the learning of the entire posterior distribution of the parameter conditioned on the observed data as highlighted by the Bayes' rule :

$$p(\theta|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\theta)p(\theta)}{p(\boldsymbol{X})}.$$

Note that the inference is straightforward when $p(\boldsymbol{X}|\theta)$ and $p(\theta)$ are conjugate. The likelihood $p(\boldsymbol{X}|\theta)$ is therefore augmented into a proper marginals distribution over the space of plausible parameters θ. Consequently, this prior knowledge contributes in reducing uncertainly in the model. In cases where $p(\theta)$ is chosen as a conjugate of $p(\boldsymbol{X}|\theta)$, the augmented expectation of the complete data log-likelihood $\mathcal{Q}(\theta, \theta^{(t)}) + \log p(\theta)$ has usually a similar functional form to $\mathcal{Q}(\theta, \theta^{(t)})$, an therefore, the augmented likelihood can be maximized through the EM algorithm using regularization point estimates including hyperparameters.

1.3.3 Variational approximation methods

The origin of *variational approximation* goes back to *variational calculus* which is a mathematical field focused on optimizing a functional (mapping of functions to a scalar, e.g. Shannon entropy) over a class of functions on which that functional depends. Approximate solutions arise when the class of functions is restricted [93, 94]. Often overshadowed by MCMC methods and the (analytical) Laplace approximation, variational approximations represent much faster alternatives for large models (with many parameter or hyperparameters) than MCMC, and more richer than Laplace approximation. They are however limited in terms of approximation accuracy compared to Monte Carlo methods which can be arbitrarily accurate by increasing the amount of simulations. See the reviews of Jordan et al. [95, 96] for an introduction to variational and more information on variational approximation accuracy.

Supposing that a model includes a set $\boldsymbol{x}$ of observed variables x, and unobserved (also referred to as

hidden, missing or latent) variables $\boldsymbol{\theta}$. The goal is to determine $p(\boldsymbol{\theta}|\boldsymbol{x})$ (which is might be intractable) using an arbitrary density function $q(\boldsymbol{\theta})$. From the Bayes' rule, we have

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}, \tag{1.18}$$

where $p(\boldsymbol{x})$ is the marginal likelihood (or model evidence). Therefore

$$\begin{aligned}
\log p(\boldsymbol{x}) =& \log p(\boldsymbol{x}) \int q(\boldsymbol{\theta}) d\boldsymbol{\theta} && (1.19)\\
=& \int q(\boldsymbol{\theta}) \log p(\boldsymbol{x}) d\boldsymbol{\theta} && (1.20)\\
=& \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})} d\boldsymbol{\theta} && (1.21)\\
=& \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (1.22)\\
=& \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})} d\boldsymbol{\theta} + \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (1.23)\\
=& D_{KL}(q||p) + \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} && (1.24)
\end{aligned}$$

Since $D_{KL}(q||p) \geq 0$ with equality when $q = p$, it follows that :

$$\log p(\boldsymbol{x}) \geq \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} = \mathcal{L}(q(\boldsymbol{\theta})), \tag{1.25}$$

where $\mathcal{L}(q(\boldsymbol{\theta}))$ is known as the evidence lower bound (ELBO) of the model marginal log-likelihood $\log p(\boldsymbol{x})$. Therefore, maximizing $\log p(\boldsymbol{x})$ by maximizing $\mathcal{L}(q(\boldsymbol{\theta}))$ can be considered whether the arbitrary density q manages to minimize $D_{KL}(q||p)$ properly. Tractability is achieved by restricting q to a class of "manageable" densities. Considering a partition of v disjoint groups for $\boldsymbol{\theta}$ s.t. $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_v\}$, a common restriction which takes its root in physics [97] is the mean-field approximation where $q(\boldsymbol{\theta})$ factorizes independently such that :

$$q(\boldsymbol{\theta}) = \prod_{u=1}^{v} q_u(\boldsymbol{\theta}_u). \tag{1.26}$$

Another one would be to restrict q as a member of a parametric family of density functions.

In the case of a mixture model where $\boldsymbol{z}$ is set as a labeling latent variable and $\boldsymbol{\Theta}$ as the set of parameters for the density $p(\boldsymbol{X}|\boldsymbol{\Theta})$ of the observed data $\boldsymbol{X} = [\boldsymbol{x}_1|\ldots|\boldsymbol{x}_n]^\top$:

$$p(\boldsymbol{z}|\boldsymbol{X};\boldsymbol{\Theta}) = \frac{p(\boldsymbol{X}|\boldsymbol{z};\boldsymbol{\Theta})p(\boldsymbol{z};\boldsymbol{\Theta})}{p(\boldsymbol{X};\boldsymbol{\Theta})}, \tag{1.27}$$

and

$$\log p(\boldsymbol{X};\boldsymbol{\Theta}) = \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{X};\boldsymbol{\Theta})} d\boldsymbol{z} + \int q(\boldsymbol{z}) \log \frac{p(\boldsymbol{X}|\boldsymbol{z};\boldsymbol{\Theta})p(\boldsymbol{z};\boldsymbol{\Theta})}{q(\boldsymbol{z})} d\boldsymbol{z} \tag{1.28}$$

$$= D_{KL}(q||p) + \mathcal{L}(q(\boldsymbol{z});\boldsymbol{\Theta}) \tag{1.29}$$

If $[\boldsymbol{x}_1|\dots|\boldsymbol{x}_n]^\top$ are assumed to be independent and identically distributed (iid), an example/special case of the mean-field approximation is where $\boldsymbol{z}$ form a partition of n groups, one for each variable $\boldsymbol{x}_i$ is $q(\boldsymbol{z}) = \prod_i^n q_i(z_i)$. See [98, 99] for application of the variational application in the EM context. In this thesis, we considered the Variational approximation for the Latent Block models where two hidden latent variables $\boldsymbol{z}$ and $\boldsymbol{w}$ take place for a set $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ of observed variables x_{ij},

$$p(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{X};\boldsymbol{\Theta}) = \frac{p(\boldsymbol{X}|\boldsymbol{z},\boldsymbol{w};\boldsymbol{\Theta})p(\boldsymbol{z},\boldsymbol{w};\boldsymbol{\Theta})}{p(\boldsymbol{X};\boldsymbol{\Theta})}, \tag{1.30}$$

and

$$\begin{aligned}\log p(\boldsymbol{X};\boldsymbol{\Theta}) = &\int q(\boldsymbol{z},\boldsymbol{w}) \log \frac{q(\boldsymbol{z},\boldsymbol{w})}{p(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{X};\boldsymbol{\Theta})} d\boldsymbol{z}d\boldsymbol{w} \\ &+ \int q(\boldsymbol{z},\boldsymbol{w}) \log \frac{p(\boldsymbol{X}|\boldsymbol{z},\boldsymbol{w};\boldsymbol{\Theta})p(\boldsymbol{z},\boldsymbol{w};\boldsymbol{\Theta})}{q(\boldsymbol{z},\boldsymbol{w})} d\boldsymbol{z}d\boldsymbol{w} \\ = &D_{KL}(q||p) + \mathcal{L}(q(\boldsymbol{z},\boldsymbol{w});\boldsymbol{\Theta})\end{aligned} \tag{1.31}$$

The mean-field approximation results in a structure with further independence such that the arbitrary density factorises as follows : $q(\boldsymbol{z},\boldsymbol{w}) = q_z(\boldsymbol{z}) \times q_w(\boldsymbol{w})$.

1.3.4 Markov Chain Monte Carlo methods (MCMC)

Introduced by Metropolis and Ulam [100] and generalized by Hastings [101], Markov Chain Monte Carlo methods is a class of techniques for solving intractable integration problem such as highly dimensional probability distribution/density arising in Bayesian inference. Unlike Monte Carlo methods capable of drawing independent samples, MCMC constructs a Markov Chain where the next sample is drawn dependently to the current sample. With the arriving of computational resources, these techniques have gained in popularity and are widespread in Machine Learning applications [102]. For a review of MCMC approach, the reader can refers to [103, 104, 105]. In this thesis, two MCMC approaches were used, namely, the Gibbs sampling and the Metropolis-Hastings sampling.

1.3.4.1 Gibbs sampling

The Gibbs sampling was introduced by Geman and Geman [106] for simulating high-dimensional complex distributions arising in image restorations and later popularised by Gelfand and Smith [102]. This methods is highly attractive since it requires no tuning. In addition it is a special case of the more general Metropolis-Hastings algorithm. Considering the vector of parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)$ for which the marginal joint posterior is intractable, the Gibbs sampling procedure is given in Algorithm 3 : At

Algorithm 3 Gibbs sampler

initialize $\boldsymbol{\theta}^{(0)}$
for t=1,2,3,... **do**
$\theta_1^{(t)} \sim p(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, \ldots, \theta_s^{(t-1)})$
$\theta_2^{(t)} \sim p(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \theta_4^{(t-1)}, \ldots, \theta_s^{(t-1)})$
$\theta_3^{(t)} \sim p(\theta_3|\theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t-1)} \ldots, \theta_s^{(t-1)})$
$\vdots$
$\theta_s^{(t)} \sim p(\theta_s|\theta_1^{(t)}, \theta_2^{(t)}, \theta_4^{(t)} \ldots, \theta_{s-1}^{(t)})$
end for

each time t, the Gibbs sampler replicates the sampling from the marginal/posterior joint distribution by sampling each variable θ_r knowing the full conditional distribution from all the other variables $\theta_{r' \neq r}$. Since the variables are randomly initialized, the first iterations are not representative of a sampling from true marginal (burn-in period) and samples from this period may be discarded. Note that starting values may also be supplied by maximum likelihood to reduce that effect. Raftery and Lewis also suggested an approach based on the computation of the posterior quantiles for determining the numbers of iterations required [107]. However, for t sufficiently large, the estimates can be considered to be sample as if they were using the intractable marginal/joint posterior distribution [108]. The process is ergodic (zero conditional probability should not occur). Whilst the Gibbs sampler is a powerful tool, it has several limitations whether or not the conditional probabilities can be easily derived/recognized in a known form or whether sampling from the conditional lead to "slow mixing" (that is the sampler sticks to a low density area and converges slowly to the center of the marginal distribution, see chapters 6 and 10 of [104]).

1.3.4.2 Metropolis-Hasting

The Metropolis-Hasting (HM) aims at generating samples from a probability distribution using the full marginal/joint density [101]. An example of its iterative procedure is given in Algorithm 4.

Algorithm 4 Metropolis-Hasting (MH)

initialize $\theta^{(0)}$
for t = 1,2,3,... **do**
  Draw a candidate parameter $\theta^{(c)}$ from a proposal density $\psi(.)$.
  Compute the ratio $R = \frac{f(\theta^{(c)})\psi(\theta^{(t-1)}|\theta^{(c)})}{f(\theta^{(t-1)})\psi(\theta^{(c)}|\theta^{(t-1)})}$.
  Compare R with a uniform random draw $u \sim \mathcal{U}(0,1)$. If $R > u$, set $\theta^{(t)} = \theta^{(c)}$, otherwise, set $\theta^{(t)} = \theta^{(t-1)}$.
end for

As for the Gibbs sampler, starting values must be set for θ. They can be set randomly or could be obtained for instance using maximum likelihood. MCMC theory guarantees that the stationary distribution after convergence will be the posterior marginal of interest regardless of the starting values [109]. Note that poor starting values might conduct the algorithm to reject many candidates during the first phase of the convergence and lead to substantial running time. A solution for this issue is discussed in [104] (chapter 6).

1.4 Information theory

Information theory provides an efficient framework for setting up probability distributions on the basis of partial knowledge. The theory was fully characterized by Shannon [110] to solve fundamental problem arising in communication theory (from Electrical Engineering) but finds relationships with other fields such as Computer science (*Kolmogorov complexity*), Philosophy of Science, Physics (Thermodynamics), Mathematics (Probability theory and Statistics) and Economics (Interest). Entropy, Relative entropy and Mutual information (MI) are the fundamental quantities in Information theory which aim at characterizing the behavior of long sequence random variables [111]. They are defined as functionals of probability distributions and allow us to estimate the probabilities of rare events. Entropy and MI are direct answers for certain fundamental questions of communication theory. More precisely, each of these functionals can be defined as follows :

— Considering the *ultimate Data compression* problem, Entropy is the minimum descriptive com-

plexity of a random variable. In other words, it quantifies the average uncertainty involved in the values of a random variable. The Entropy of a random variable X with probability mass function $p(X)$ is originally given as follows :

$$H(X) = -\sum_{x} p(x) \log_2 p(x). \tag{1.32}$$

— Considering the *ultimate Transmission rate* problem, MI is the communication rate in presence of noise. Specifically, it is the conditional Entropy defined as the entropy of one random variable given another random variable. Therefore, it measures the decrease of uncertainty in the presence of another variable and quantifies a dependence between both. Given two random variables X and Y, MI takes the following form :

$$MI(X,Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{1.33}$$

— MI is the relative entropy, mainly referred to as the Kullback-Leibler (KL) divergence and defined as follows :

$$D_{KL}(P(X,Y)||Q(X,Y)) = \sum_{X,Y} P(X,Y) \log \frac{P(X,Y)}{Q(X,Y)}, \tag{1.34}$$

where $Q(X,Y) = P(X)P(Y)$.

1.5 Evaluation metrics

In order to measure the clustering performance of our algorithms, we evaluate them on several benchmark datasets for which the ground-truth labels are available. In the literature, two families of direct evaluation metrics used to assess the quality of partitions provided by clustering algorithms can be distinguished :

— Internal metrics : which measure the quality of a partition according to the intrinsic properties of the initial dataset exclusively. We denote : the Davies-Bouldin index [112], the Calinski-Harabasz index [113], the BIC index of Raftery [114], the Silhouette validation model of Rousseeuw [115].

— External metrics : which compare a partition obtained by a given clustering algorithm with the ground-truth labels of the data. Hence, the more similar these partitions are, the better the clustering algorithm performs.

An empirical study of Rendon et al. that compares internal and external indexes can be found in [116]. In this section, we introduce two external metrics used in this thesis. In the following, we consider a set of objects $O = \{O_1, \ldots, 0_N\}$ for which we denote two partitions $\boldsymbol{C}$ and $\boldsymbol{T}$ where $\boldsymbol{C}$ is a partition of clusters $\{C_1, \ldots, C_g\}$ obtained from a cluster analysis and $\boldsymbol{T}$ the partition of true groups $\{T_1, \ldots, T_{g'}\}$.

1.5.1 Accuracy

Accuracy (which refers exactly to the overall accuracy) is one of the simplest statistic used to describe the correctness of a classification. The most common way to represent this measure is made across the use of an error matrix (also referred to as confusion matrix or matching matrix). This matrix takes the form of a contingency table between the predicted partition $\boldsymbol{C}$ with the actual partition $\boldsymbol{T}$. Let $\boldsymbol{C} \times \boldsymbol{T}$ be the contingency table formed by $\boldsymbol{C}$ and $\boldsymbol{T}$ as follows :

$\boldsymbol{C} \diagdown \boldsymbol{T}$	T_1	T_2	$\cdots$	$T_{g'}$	T.
C_1	n_{11}	n_{12}	$\cdots$	$n_{1g'}$	$n_{1.}$
C_2	n_{21}	n_{22}	$\cdots$	$n_{2g'}$	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_g	n_{g1}	n_{g2}	$\cdots$	$n_{gg'}$	$n_{g.}$
C.	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.g'}$	

where each count $n_{kk'}$ denotes the number of objects in common between the groups $(C_k, T_{k'})$ s.t. $n_{kk'} = |C_k \cap T_{k'}|$ and the marginals $(n_{1.}, \ldots, n_{g.})$ and $(n_{.1}, \ldots, n_{.g'})$ denote the cardinality for each group in $\boldsymbol{C}$ and $\boldsymbol{T}$ respectively. Assuming that $\boldsymbol{C} \times \boldsymbol{T}$ is ordered such that the diagonal contains the maximum number of well classified elements, the overall accuracy is therefore determine by summing the elements inside the diagonal divided by the number of elements classified $N^{(c)}$ (in our case, most of the time, all objects are classified s.t. $N^{(c)} = N$). In the ideal case where each object is correctly classified, $\boldsymbol{C} \times \boldsymbol{T}$ is a diagonal matrix. The accuracy (ACC) between two partitions can be formulated as follows :

$$ACC(\boldsymbol{C}, \boldsymbol{T}) = \frac{1}{N^{(c)}} \max \sum_{C_k, T_{k'}} |C_k \cap T_{k'}| = \frac{1}{N^{(c)}} \max \sum_{k,k'}^{g,g'} n_{kk'} \tag{1.35}$$

Aside the overall accuracy, $\boldsymbol{C} \times \boldsymbol{T}$ can be used to computed several others accuracy measures, e.g. the normalized accuracy, the KHAT statistic $\hat{\kappa}$ (see the review of Congalton [117] or Stehman [118]) the F1 score and many others. One major drawback of the accuracy is its bias relative to unbalanced partition. For instance, considering an actual partition $\boldsymbol{T}$ of $N = 100$ objects divided into two groups such that $\#T_1 = 95$ and $\#T_2 = 5$, a prediction partition $\boldsymbol{C}$ classifying all objects in one group will

achieved 95% of overall accuracy. For this reason, throughout this thesis, we will omit this metric as our proposals are evaluated on several heavily unbalanced datasets.

1.5.2 Normalized Mutual Information

The Mutual Information criterion defined in Section 1.4 is one of the external metric used in our evaluations. However, by definition, it is not bounded. To increase its interpretability, we use the Normalized Mutual Information (NMI) proposed by Strehl and Ghosh [119]. Taking advantages that the conditional entropy (MI) is inferior or equal to the minimum respective entropy (H) : $MI(\boldsymbol{C}, \boldsymbol{T}) \leq \min(H(\boldsymbol{C}), H(\boldsymbol{T}))$, several normalizations are possible. In the following, we refer to NMI using the geometric mean which implies $\text{dom}(NMI) = [0, 1]$. From a statistical point of view this normalization derives from first thinking of mutual information as an analogue to covariance and its computation is related to the Pearson correlation coefficient. Finally we have :

$$NMI(\boldsymbol{C}, \boldsymbol{T}) = \frac{MI(\boldsymbol{C}, \boldsymbol{T})}{\sqrt{H(\boldsymbol{C})H(\boldsymbol{T})}}. \tag{1.36}$$

In addition, partitions with different number of clusters can also be compared using the NMI. For details on other normalized versions of the mutual information, the reader can refer to the work of Cahill [120].

1.5.3 Adjusted Rand Index

The second evaluation criterion employed in this thesis is the Adjusted Rand Index (ARI). We explain this criterion by firstly defining the Rand Index, then subsequently its adjustment.

1.5.3.1 Rand Index

The Rand index (RI) is a intuitional approach aiming at comparing clustering by counting pairs of objects that are gathered similarly in two partitions. In the following, we give the original formulation of RI as defined by Rand in [121]. Considering the disjunctive table $D \in \{0, 1\}^{N \times N}$ for O where each element $d_{ii'}$ are set according to the partition $\boldsymbol{C}$ and $\boldsymbol{T}$ such as :

$$d_{ii'} = \begin{cases} 1 & \text{if } \exists k, k' \text{ such that } (o_i, o_{i'}) \in C_k \text{ and } (o_i, o_{i'}) \in T_{k'}, \\ 1 & \text{if } \exists k, k' \text{ such that } o_i \in (C_k, T_{k'}) \text{ and } o_{i'} \notin (C_k, T_{k'}), \\ 0 & \text{otherwise} \end{cases}$$

RI is given as follows :

$$RI(\boldsymbol{C},\boldsymbol{T}) = \sum_{i<i'}^{N} d_{ii'} / \binom{N}{2} = \frac{a+b}{a+b+c+d}, \tag{1.37}$$

where $\binom{N}{2}$ is the number of all possible pairs and the ratio given by the second equality is a more common formulation which can be proposed by defining the following quantities with respect to the elements in $\boldsymbol{O}$: (a) the number of pairs of elements that are in the same subset in $\boldsymbol{C}$ and in the same subset in $\boldsymbol{T}$; (b) the number of pairs of elements that are in different subsets in $\boldsymbol{C}$ and in different subsets in $\boldsymbol{T}$; (c) the number of pairs of elements that are in the same subset in $\boldsymbol{C}$ and in different subsets in $\boldsymbol{T}$; (d) the number of pairs of elements that are in different subsets in $\boldsymbol{C}$ and in the same subset in $\boldsymbol{T}$. Considering the confusion table given in Section 1.5.1, a more efficient computational form for RI can be achieved as follows :

$$RI(\boldsymbol{C},\boldsymbol{T}) = \Big[\binom{N}{2} - \frac{1}{2}\Big(\sum_{k}^{g}(\sum_{k'}^{g'} n_{kk'})^2 + \sum_{k'}^{g'}(\sum_{k}^{g} n_{kk'})^2\Big) + \sum_{k,k'}^{g,g'} n_{kk'}^2\Big] \Big/ \binom{N}{2}. \tag{1.38}$$

1.5.3.2 Adjusted Rand Index

The Adjusted Rand Index (ARI) was proposed by Hubert and Arabie in [122] to solve the lack of uniqueness of the solution. While $\mathrm{dom}(RI) = [0,1]$, the expected value of the RI for two random partitions does not have a constant value. The idea behind ARI is to compare RI with the expected Rand Index (ERI) under the hypothesis that two partitions are independent. Using a generalised hypergeometric distribution as the model for randomness, we obtain the following equation :

$$ARI = \frac{RI - ERI}{\max(RI) - ERI} = \frac{\sum_{k,k'}^{g,g'} \binom{n_{kk'}}{2} - \Big[\sum_{k}^{g}\binom{n_{k.}}{2}\sum_{k'}^{g'}\binom{n_{.k'}}{2}\Big] \Big/ \binom{N}{2}}{\frac{1}{2}\Big[\sum_{k}^{g}\binom{n_{k.}}{2} + \sum_{k'}^{g'}\binom{n_{.k'}}{2}\Big] - \Big[\sum_{k}^{g}\binom{n_{k.}}{2}\sum_{k'}^{g'}\binom{n_{.k'}}{2}\Big] \Big/ \binom{N}{2}}, \tag{1.39}$$

which is has 1 for upper bound and takes on the value 0 when RI = ERI. As opposed to $\mathrm{dom}(RI) = [0,1]$, $\mathrm{dom}(ARI) = (-\infty, 1]$. However, negative values of ARI have no substantive use and therefore the normalization would offer no practical benefits.

1.6 Dimensionality reduction

In this section, we briefly review several of the most prominent dimensionality reduction techniques such as the *Principal Component Analysis*. Subsequently, techniques such as *Latent Semantic Analysis*

with a high incidence in text analysis are reviewed in more detail and finally, an overview of *Nonnegative Matrix Factorization* including several extensions and variants, as well as several algorithms is presented.

1.6.1 Singular Value Decomposition

The *Singular Value Decomposition* (SVD) is an essential tools in linear algebra which generalizes the *Eigen decomposition* (see Appendix A) of a square matrix to any rectangular matrix. In contrast to the *Eigen decomposition*, the SVD exists for all matrices. It is defined by the following theorem.

Theorem 1.6.1. ***(Singular Value Decomposition)****. Let* $\boldsymbol{A} \in \mathbb{R}^{n\times d}$*, there exists* orthogonal *matrices* $\boldsymbol{U} = [\boldsymbol{u}_1|\dots|\boldsymbol{u}_n] \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{V} = [\boldsymbol{v}_1|\dots|\boldsymbol{v}_d] \in \mathbb{R}^{d\times d}$ *such that :*

$$\boldsymbol{U}^\top \boldsymbol{A}\boldsymbol{V} = \boldsymbol{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_p), \qquad p = \min(n, d), \tag{1.40}$$

where $[\boldsymbol{u}_1|\dots|\boldsymbol{u}_n]$ *are referred to as the* left singular vectors *of* $\boldsymbol{A}$*,* $[\boldsymbol{v}_1|\dots|\boldsymbol{v}_d]$ *as the* right singular vectors *and* $\sigma_1, \dots, \sigma_p$ *are called the* singular values*.*

In practice, the SVD is not unique and the *singular values* are arranged along the diagonal of $\boldsymbol{\Sigma}$ in a decreasing order $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ so that $\boldsymbol{\Sigma}$ is uniquely determined by $\boldsymbol{A}$. The number r of positive singular values (nonzero diagonal entries in $\boldsymbol{\Sigma}$) is equal to the rank of $\boldsymbol{A}$: $\text{rank}(A) = \text{rank}(\boldsymbol{\Sigma}) = r$. Also, $\boldsymbol{A} = \sum_k^r \boldsymbol{u}_k \sigma_k \boldsymbol{v}_k^\top$.

Note that $\text{diag}(\sigma_1, \dots, \sigma_p)^{\frac{1}{2}} = \text{diag}(\lambda_1, \dots, \lambda_n)$ where $\text{diag}(\lambda_1, \dots, \lambda_n)$ are the eigenvalues of $\boldsymbol{A}^\top \boldsymbol{A}$.

Proof of this theorem and the following properties are given in [123] (Chapter 2). For more insights on matrix theory, the reader can refer to [123, 124].

1.6.2 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate analysis for data matrices in which a set of samples is described by several quantitative variables. It was introduced by Pearson in [2] and later developed and titled (*Principal Component Analysis*) by Hotling in [3]. It is likely the oldest multivariate analysis and can actually be tracked back to the works of Cauchy in [125] and Jordan in [126]. Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ be a data matrix, PCA aims at producing a lower dimensional space for $\boldsymbol{X}$ of orthogonal variables (called *Principal components*) such that the variance between the samples

projections and the center is maximized. The principal components are ordered decreasingly such that the first component is the one maximizing the variance the most. This achievement results in performing the *Eigen Decomposition* of the covariance matrix or the correlation matrix where the principal components are obtained by projecting the data samples onto the eigenvectors. This can also be viewed as computing the SVD of $\boldsymbol{X}$ subject to a proper normalization. For instance, let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ be the barycenter of $\boldsymbol{X}$. After centering $\boldsymbol{X}$ such that $\boldsymbol{\mu} = \mathbf{0} \in \mathbb{R}^d$, the variance of the data sample is given by :

$$\sum_{j=1}^{d}(x_{ij} - \mu_j)^2 = \sum_{j=1}^{d} x_{ij}^2. \tag{1.41}$$

Therefore, if we normalize each variables in $\boldsymbol{X}$ to unit-length such that $[\boldsymbol{x}_1|\ldots|\boldsymbol{x}_d] = [\frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}|\ldots|\frac{\boldsymbol{x}_d}{\|\boldsymbol{x}_d\|}]$, $\boldsymbol{X}^\top\boldsymbol{X}$ becomes a correlation matrix. Consequently, as explained in Section A.1, the squared singular values of $\boldsymbol{X}$ are the eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$. It is also possible to normalize $\boldsymbol{X}$ so that $\boldsymbol{X}^\top\boldsymbol{X}$ becomes the covariance matrix instead, but most computations are achieved using the correlation matrix. For more details, a nice coverage of PCA is given by Abdi and Williams in [127].

Definition 1.6.1. (**Principal Component Analysis**) [3]. Let $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n\times d}$ be a data matrix with centered and normalized variables $[\boldsymbol{x}_1|\ldots|\boldsymbol{x}_d]$. Let $\boldsymbol{U} = [\boldsymbol{u}_1|\ldots|\boldsymbol{u}_n] \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} = [\boldsymbol{v}_1|\ldots|\boldsymbol{v}_d] \in \mathbb{R}^{d\times d}$ be the orthogonal matrices obtained by the SVD of $\boldsymbol{X}$ such that :

$$\boldsymbol{U}^\top\boldsymbol{X}\boldsymbol{V} = \boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_p), \quad p = \min(n, d), \tag{1.42}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$ are the singular values of $\boldsymbol{X}$, $\text{rank}(\boldsymbol{X}) = r$, and $\boldsymbol{X} = \sum_k^r \boldsymbol{u}_k\sigma_k\boldsymbol{v}_k^\top$. The principal components $\boldsymbol{C} = [\boldsymbol{c}_1|\ldots|\boldsymbol{c}_d] \in \mathbb{R}^{n\times d}$ also called the *factor scores* are given by :

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}. \tag{1.43}$$

This is equivalent to projecting the data samples $[\boldsymbol{x}_1|\ldots|\boldsymbol{x}_n]^\top$ onto the eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ which are given by $\boldsymbol{V}$ since :

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{X}\boldsymbol{V}. \tag{1.44}$$

1.6.3 Low-Rank Approximation

Low-Rank Approximation (LRA) is the mathematical problem of approximating a matrix by another matrix which has a lower rank g [128]. It falls into the class of dimensionality reduction techniques

and is a special of matrix nearness problems which attempt to approximate a matrix by a another matrix given a distance measure (see the work of Dhillon and Tropp for a review of matrix nearness with the class of *Bregman divergences* [129]). LRA can be formulated as the following optimization problem.

Problem 1.6.1. ***(Low-rank Approximation).*** *Given a matrix $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n\times d}$, solve :*

$$\min_{\boldsymbol{Y}\in\mathbb{R}^{n\times d},\text{rank}(\boldsymbol{Y})\leq g} \mathcal{D}(\boldsymbol{X},\boldsymbol{Y}). \tag{1.45}$$

When $\mathcal{D}$ is the Frobenius norm, the solution can be obtained using the SVD of $\boldsymbol{X}$. This consequence is the results of the following theorem.

Theorem 1.6.2. ***(Eckart-Young)*** *[128]. Let $\boldsymbol{U} = [\boldsymbol{u}_1|\ldots|\boldsymbol{u}_n] \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} = [\boldsymbol{v}_1|\ldots|\boldsymbol{v}_d] \in \mathbb{R}^{d\times d}$ be the orthogonal matrices obtained by the SVD of $\boldsymbol{X}$ such that :*

$$\boldsymbol{U}^\top\boldsymbol{X}\boldsymbol{V} = \boldsymbol{\Sigma} = \text{diag}(\sigma_1,\ldots,\sigma_p), \quad p = \min(n,d), \tag{1.46}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ are the singular values of $\boldsymbol{X}$. Let r be the number of positive singular values s.t. $\text{rank}(\boldsymbol{X}) = \text{rank}(\boldsymbol{\Sigma}) = r$ *and $\boldsymbol{X} = \sum_k^r \boldsymbol{u}_k\boldsymbol{\sigma}_k\boldsymbol{v}_k^\top$. Considering the* truncated SVD *of rank g giving $\boldsymbol{X}_g = \sum_k^g \boldsymbol{u}_k\boldsymbol{\sigma}_k\boldsymbol{v}_k^\top$ for $1 \leq g \leq r$, we have that :*

$$\min_{\text{rank}(\boldsymbol{Y})\leq g} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2 = \|\boldsymbol{X}-\boldsymbol{X}_g\|_F^2 = \sum_{k=g+1}^{r} \sigma_k \tag{1.47}$$

A proof of this theorem is given in in Chpater 2 of [123].

Remark. As mentioned in Section 1.6.1, SVD is not unique. If the singular values are not ordered, we denote $\binom{r}{g}$ possible stationary points for problem (1.47).

Since $\text{rank}(\boldsymbol{Y}) \leq g$, $\boldsymbol{Y}$ can be decomposed as the product of two matrices such that $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{W}^\top$ where $\boldsymbol{Z} \in \mathbb{R}^{n\times g}$ and $\boldsymbol{W} \in \mathbb{R}^{d\times g}$ and problem (1.45) can be rewritten as :

$$\min_{\boldsymbol{Z}\in\mathbb{R}^{n\times g},\boldsymbol{W}\in\mathbb{R}^{d\times g}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Z}\boldsymbol{W}^\top\|_F^2. \tag{1.48}$$

Let $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}$ and $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}}$ where $\boldsymbol{D} \in \mathbb{R}_+^{g\times g}$ is diagonal, $\boldsymbol{U}^\top\boldsymbol{U} = \mathbf{I}_g$ and $\boldsymbol{V}^\top\boldsymbol{V} = \mathbf{I}_g$, solving problem (1.45) is now equivalent to computing a compact SVD. More details about this optimization can be found in [130].

If follows that the dimensionality reduction technique (NMF) presented in the next chapter can be seen as special case of LRA with the nonnegativity constraint. The reader can refer to the survey of Markovsky [131] for more insights on LRA.

1.6.4 Latent Semantic Analysis

Latent Semantic Analysis (LSA) also referred to as *Latent Semantic Indexing* (LSI) is an automatic indexing technique for information retrieval, introduced by Deerwester et al in [1] to improve the detection of relevant documents given a subset of words. The concept arise due to the deficiency of term matching retrieval during queries (e.g. retrievals that omit documents referencing "automobile" when querying "car"), which makes the direct document-term relation not always reliable. The goal was therefore to represent documents throughout a hidden latent structure instead of terms. This structure is simply obtained by performing a SVD on the document-term matrix in order to obtained a lower dimensional space mapping together the terms and the documents. The queries are subsequently projected onto the lower dimensional space to return the matching documents. LSA works by retaining only the g largest singular values. This results in the exact low rank approximation technique introduced in the previous section. The lower rank g must be fixed relatively low to allow a fast retrieval but also large enough to capture the real structure.

1.7 Conclusion

We have listed all the necessary elements to describe our various contributions in terms of clustering, co-clustering and data embedding. Next, we focus on Non-negative Matrix Factorization (NMF) which is more suitable than LSA to deal with document-terms matrices both in terms of clustering and data embedding.

Chapitre 2

Nonnegative Matrix Factorization

As mentioned in the introduction, NMF which was originally designed for dimensionality reduction has received throughout the years a tremendous amount of attention for clustering purposes and showed multiple positives outcomes in several fields such as image processing or text mining. More specifically in text mining where NMF produces a meaningful interpretation for document-term matrices in comparison to methods like SVD components or LSA [1] where factors may include negative values. This chapter is fully dedicated to the presentation and evaluation of NMF toward clustering. In the first section, we present the method in its original form, then several algorithms used to find a solution. Stopping conditions as well as several initialization methods for these algorithms are also presented. Subsequently, different variants and extensions of NMF with application in document clustering and co-clustering are reviewed. The second section of this chapter is dedicated to the evaluation of NMF for the task of document clustering. A thorough study of the clustering partitions obtained from the best local minima is given and highlights the presence of better clustering partitions in lesser local minima. Thereafter, a consensus approach is elaborated in order to overcome this issue and extract the best the clustering performances from NMF.

2.1 Presentation of NMF

Given a nonnegative matrix $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}_+^{n \times d}$, Nonnegative matrix factorization is a dimensionality reduction method which aims at approximating $\boldsymbol{X}$ by the production of two lower dimensional matrices $\boldsymbol{Z} \in \mathbb{R}_+^{n \times g}$ and $\boldsymbol{W} \in \mathbb{R}_+^{d \times g}$, e.g.

$$\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{W}^\top. \tag{2.1}$$

An illustration of NMF is given in Figure 2.1 This method was first introduced by Paatero and Tapper

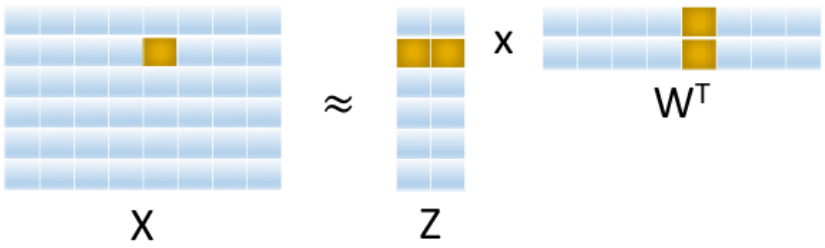


FIGURE 2.1 – NMF.

[4] under the name Positive Matrix Factorization (PMF) as a suitable and coherent alternative to PCA and SVD for nonnegative values. The method was however later popularised by Lee and Seung [5, 132]. So far NMF has found many applications in areas such as document clustering [133, 134, 135], signal processing/source separation [136, 137], computer vision such as image classification [138] or spectral unmixing [139], and others.

Due to the nonnegative constraint on the approximation factors, NMF can be viewed as a weighted sum of the original data sample $\boldsymbol{x}_i$. In this sense $\boldsymbol{Z} = [\boldsymbol{z}_1|\ldots|\boldsymbol{z}_n]^\top$ is assimilated as a coefficient or weighting matrix while $\boldsymbol{W} = [\boldsymbol{w}_1|\ldots|\boldsymbol{w}_d]^\top$ contains the basis vector, e.g. :

$$\boldsymbol{x}_i \approx \sum_{k=1}^{g} z_{ik} w_{jk} = \boldsymbol{z}_i \boldsymbol{W}^\top. \tag{2.2}$$

This interpretation is often referred to as part-based analysis and reversible (e.g. $\boldsymbol{x}_j \approx \boldsymbol{Z}\boldsymbol{w}_j$) as long as both factors are nonnegative. Several cost function are denoted in order to measure the approximation to the data matrix. The most common is the sum of squared Frobenius norm (or sum of squares) denoted as :

$$\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2 = \frac{1}{2}\sum_{i,j}^{n,d} \left(x_{ij} - [\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}\right)^2. \tag{2.3}$$

Another popular measure is the generalized Kullback-Leibler divergence also called the I-divergence, given as follows :

$$D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} \left[x_{ij} \log \frac{x_{ij}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}} - x_{ij} + [\boldsymbol{Z}\boldsymbol{W}^\top]_{ij} \right]. \tag{2.4}$$

This cost function is often acknowledged for its better results when NMF applied on directional data. The Itakura-Saito divergence is another cost function with substantial application in signal processing,

it is denoted as :

$$D_{IS}(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} \Big[\frac{x_{ij}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}} - \log \frac{x_{ij}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}} - 1 \Big]. \tag{2.5}$$

Afterwards, we formulate the problem of NMF in a form of any arbitrary cost function $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$.

Problem 2.1.1. *(**NMF**). Let $\boldsymbol{X} \in \mathbb{R}_+^{n\times d}$ and $g < \min(n, d)$, solve :*

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)\}. \tag{2.6}$$

For more details about the optimality conditions that must holds for a solution point of this problem, an in-depth review of unconstrained and constrained optimization is given in appendix C. Further connections between these conditions and the construction of the proofs of convergence for several of ours algorithms are also highlighted.

The Lagrangian function function associated with this constrained problem is stated as follows :

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) + \mathrm{Tr}(\boldsymbol{\mu}\boldsymbol{Z}^\top) + \mathrm{Tr}(\boldsymbol{\nu}\boldsymbol{W}^\top), \tag{2.7}$$

where $\boldsymbol{\mu} \in \mathbb{R}_-^{n\times g}$ and $\boldsymbol{\nu} \in \mathbb{R}_-^{d\times g}$ are the Lagrange multipliers (note that in this thesis, the inequality constraints are reversed and so the Lagrangian, e.g. $\boldsymbol{Z} \geq 0 \implies -\boldsymbol{Z} \leq 0$). Therefore according to the first-order necessary conditions for inequality constrained optimization defined in Section C and referred to as the Karush-Kuhn-Tucker (KTT) conditions, if $(\boldsymbol{Z}, \boldsymbol{W})$ is a local minimum, overall we have the following conditions for solving the constrained nmf problem using the first-order differentiation :

$$\boldsymbol{Z} \geq 0,\ \boldsymbol{W} \geq 0, \tag{2.8}$$

$$\nabla_{\boldsymbol{Z}}\mathcal{L} = 0,\ \nabla_{\boldsymbol{W}}\mathcal{L} = 0, \tag{2.9}$$

$$\boldsymbol{\mu} \odot \boldsymbol{Z} = 0,\ \boldsymbol{\nu} \odot \boldsymbol{W} = 0. \tag{2.10}$$

Assuming that the gradient of $\mathcal{F}$ has the following form :

$$\nabla\mathcal{F} = [\nabla\mathcal{F}]_+ - [\nabla\mathcal{F}]_-, \tag{2.11}$$

where $[\nabla\mathcal{F}]_+ \geq 0$ and $[\nabla\mathcal{F}]_- \geq 0$. From eq (2.9), we obtain the following expressions for $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$:

$$\boldsymbol{\mu} = -[\nabla_{\boldsymbol{Z}}\mathcal{F}]_+ + [\nabla_{\boldsymbol{Z}}\mathcal{F}]_- = -\nabla_{\boldsymbol{Z}}\mathcal{F}, \qquad \boldsymbol{\nu} = -[\nabla_{\boldsymbol{W}}\mathcal{F}]_+ + [\nabla_{\boldsymbol{W}}\mathcal{F}]_- = -\nabla_{\boldsymbol{W}}\mathcal{F}.$$

Since $\boldsymbol{\mu}, \boldsymbol{\nu} \leq 0$, these equations add another condition on the gradient $\nabla\mathcal{L}$ such that we require the following conditions for a local minimum :

$$\boldsymbol{Z} \geq 0,\ \boldsymbol{W} \geq 0, \tag{2.12}$$

$$\nabla_{\boldsymbol{Z}}\mathcal{L} = 0,\ \nabla_{\boldsymbol{W}}\mathcal{L} = 0, \tag{2.13}$$

$$\boldsymbol{\mu} \odot \boldsymbol{Z} = 0,\ \boldsymbol{\nu} \odot \boldsymbol{W} = 0, \tag{2.14}$$

$$\nabla_{\boldsymbol{Z}}\mathcal{F} \geq 0,\ \nabla_{\boldsymbol{W}}\mathcal{F} \geq 0. \tag{2.15}$$

Subsequently, (2.14) leads to the following equations :

$$\boldsymbol{Z} \odot ([\nabla_{\boldsymbol{Z}}\mathcal{F}]_- - [\nabla_{\boldsymbol{Z}}\mathcal{F}]_+) = 0, \qquad \boldsymbol{W} \odot ([\nabla_{\boldsymbol{W}}\mathcal{F}]_- - [\nabla_{\boldsymbol{W}}\mathcal{F}]_+) = 0.$$

These equations are called the stationary equations since their derivatives are indefinite at $(\boldsymbol{Z}, \boldsymbol{W}) = (\boldsymbol{0}, \boldsymbol{0})$. Similarly, $(\boldsymbol{Z}, \boldsymbol{W})$ are called stationary points due to existence of saddle points (e.g. $(\boldsymbol{Z}, \boldsymbol{0})$ if the partial derivative becomes null), since $\mathcal{D}$ is not jointly convex w.r.t. $\boldsymbol{Z}$ and $\boldsymbol{W}$ simultaneously.

2.1.1 Existing algorithms

Several algorithms used for performing NMF are review in this section. The initialization, the convergence properties as well as the stopping condition of these algorithms are also mentioned. For simplification, $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$ will be assumed to be the Frobenius norm given by eq(2.3).

2.1.1.1 Alternating Least Square

The Alternative Least Squares (ALS) algorithm was the first method proposed to solve the problem of NMF [4]. By fixing one factor alternatively, the problem of NMF can be seen as a least square problem with nonnegative constraint. The iterative procedure is stated in Algorithm 5 :

Algorithm 5 Alternating Least Squares (ALS)

initialize $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$
for t = 0,1,2,... **do**
  $\boldsymbol{Z}^{(t+1)} = \underset{\boldsymbol{Z}\geq 0}{\arg\min} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2$
  $\boldsymbol{W}^{(t+1)} = \underset{\boldsymbol{W}\geq 0}{\arg\min} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2$
end for.

2.1.1.2 Multiplicative updates

To this day, the most popular approach used for solving the problem of NMF (which will also be utilized in this thesis) remains the Multiplicative Updates (MU) algorithm proposed by Lee and Seung [132]. This algorithm is based on the Majoration-Minimization or Minorization-Maximization (MM) algorithm proposed by Ortega and Rheinboldt during the study of line search methods [140] (see an example of line search algorithm for NMF below). The methods consists in optimizing an auxiliary/surrogate function $\mathcal{G}$, in the case of NMF majorizing $\mathcal{F}$ such that :

$$\mathcal{G}(\boldsymbol{Z}, \boldsymbol{Z}^{(t)}) \geq \mathcal{F}(\boldsymbol{Z}), \forall \boldsymbol{Z} \neq \boldsymbol{Z}^{(t)}, \tag{2.16}$$

$$\mathcal{G}(\boldsymbol{Z}, \boldsymbol{Z}) \geq \mathcal{F}(\boldsymbol{Z}). \tag{2.17}$$

Minimizing $\mathcal{G}$ will therefore guarantee $\mathcal{F}$ to decrease. An example of the iterative procedure is given in Algorithm 6 :

Algorithm 6 Multiplicative Updates (MU) for NMF

initialize $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$
for t = 0,1,2,... **do**
  $\boldsymbol{Z}^{(t+1)} = \boldsymbol{Z} \odot \frac{\boldsymbol{XW}}{\boldsymbol{ZW}^\top \boldsymbol{W}}$
  $\boldsymbol{W}^{(t+1)} = \boldsymbol{W} \odot \frac{\boldsymbol{X}^\top \boldsymbol{Z}}{\boldsymbol{WZ}^\top \boldsymbol{Z}}$
end for

The multiplicative updates can be seen as a gradient descent update :

$$\boldsymbol{Z} \odot \frac{\boldsymbol{XW}}{\boldsymbol{ZW}^\top \boldsymbol{W}} = \boldsymbol{Z} - \frac{\boldsymbol{Z}}{\boldsymbol{ZW}^\top \boldsymbol{W}} \odot \nabla_{\boldsymbol{Z}} \mathcal{F}. \tag{2.18}$$

where $\frac{\boldsymbol{Z}}{\boldsymbol{ZW}^\top \boldsymbol{W}}$ is the step size.

As pointed out in several studies (Chu et al [141], Gonzalez and Zhang [142], Lin [143]), Lee and Seung showed in [132] that $\mathcal{F}$ is non-increasing under the multiplicative updates. Therefore, the algorithm does not always converge to a local minimum point and can be trapped in a saddle point. This comes down to the nature of the multiplicative update which prevent any readjustment of $\boldsymbol{Z}_{ik}^{(t+1)}$ to meet the conditions given by eqs(2.12-2.15) once $\boldsymbol{Z}_{ik}^{(t)}$ equals zero. From eq(2.18), we can see that an entry $\boldsymbol{Z}_{ik}^{(t)}$ may have its partial derivative $\nabla_{\boldsymbol{Z}_{ik}} \mathcal{F}$ either negative, positive or null in order to increase, decrease or not modify $\boldsymbol{Z}_{ik}^{(t+1)}$ (in this case, the zero entry is admissible). However, in MUs, an entry might become null (for instance due to machine precision) but has its derivative negative. Thereby,

$\boldsymbol{Z}_{ik}^{(t+1)}$ will be trapped in a non-stationary point. Several solutions arise to overcome this issue while keeping the MUs. Chi and Kolda [144] proposed to monitor the null entries in $\boldsymbol{Z}^{(t)}$ and replace them with a small constant when their partial derivative is negative. Lin [143] proposed to use the gradient update where a constant ϵ is added in the step size denominator to ensure a stationary point at convergence. This is equivalent to retain the MUs with ϵ added to the numerator and the denominator such as :

$$z_{ik} - \frac{z_{ik}}{[\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}]_{ik} + \epsilon}\nabla_{\boldsymbol{Z}_{ik}}\mathcal{F} = z_{ik}\frac{[\boldsymbol{X}\boldsymbol{W}]_{ik} + \epsilon}{[\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}]_{ik} + \epsilon}. \tag{2.19}$$

However this methods leads to non sparse solutions which remain desirable in many NMF applications. One could also use the equivalent gradient update given by eq(2.18) and applied a projection on the nonnegative orthant. In this thesis where NMF is used for achieving document clustering, sparse factors are appreciated as they convey less uncertainty for cluster assignment. Therefore, we rely on the method proposed by Chi and Kolda [144] which allows sparse factors while avoiding inadmissible zeros.

Remark. The same comments are valid for $\boldsymbol{W}$.

2.1.1.3 Projected Gradient descent

The Gradient descent (GD) algorithm a first-order conditions iterative method which estimate a local minimum point by taking a step in the direction of the negative gradient at the current point. Due to nonnegativity, practicing GD for NMF results actually in a Projected gradient algorithm. To ensure nonnegativity or the update, a projection on the nonnegative orthant is achieved by setting all nonnegative elements to zero. Let α be the step size for $\boldsymbol{Z}$ and β the step size for $\boldsymbol{W}$, a basic Projected gradient algorithm for NMF is described in Algorithm 7 : However, this algorithm is very sensitive to

Algorithm 7 Projected Gradient (PG) for NMF

initialize $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$ $\alpha^{(0)}$, $\beta^{(0)}$
for t = 0,1,2,... **do**
  compute $\nabla\mathcal{F}(\boldsymbol{Z}^{((t)})$
  choose a step size $\alpha^{(t)}$
  $\boldsymbol{Z}^{(t+1)} = [\boldsymbol{Z} - \alpha\nabla\mathcal{F}(\boldsymbol{Z})]^+$
  compute $\nabla\mathcal{F}(\boldsymbol{W}^{(t)})$
  choose a step size $\beta^{(t)}$
  $\boldsymbol{W}^{(t+1)} = [\boldsymbol{W} - \beta\nabla\mathcal{F}(\boldsymbol{W})]^+$
end for.

the step size and little can be said about its convergence without a suitable setting $\alpha^{(t)}$ and $\alpha^{(t)}$.

2.1.1.4 Projected Gradient descent with line search methods

Another popular gradient projection methods consists in using a *line search* strategy. Considering an unconstrained minimization problem of the form :

$$\min_x f(x), \tag{2.20}$$

where f is differentiable. In order to find a local minimum x^*, the *line search* strategy works by setting a direction $p^{(t)}$ at a current point estimate $x^{(t)}$ and searching along that direction a new estimate $x^{(t+1)}$ with a lower function value. The distance necessary to move along $p^{(t)}$ can be found by solving this one-dimensional minimization problem :

$$\min_\alpha \{\phi(\alpha) = f(x^{(t)} + \alpha x^{(t)})\}, \quad \alpha > 0. \tag{2.21}$$

Solving the exact distance (commonly called the step length α) is however expensive and sometimes unnecessary. In practice, a limited number of trials is set. The steepstep descent direction given by the opposite gradient $-\nabla f(x)$ is often the common direction used for the search line. A popular line search condition is the Wolfe condition which states that α should primarily provide a sufficient decrease of f which measures by this inequality :

$$f(x^{(t)} + \alpha p^{(t)}) \leq f(x^{(t)}) + \sigma\alpha\nabla f(x^{(t)})^\top p^{(t)}, \tag{2.22}$$

for some constant $\sigma \in [0, 1]$. This condition is also known as the Armijo rule, and as shown by Lin in [145], it can be adapted for projection on the nonnegative orthant for practicing NMF. Setting $x^{(t+1)} = [x^{(t)} + \alpha p^{(t)}]$, a Backtracking Line Search (BLS) algorithm which can be used to obtain α and $x^{(t+1)}$ is given afterwards (see Algorithm 8).

Algorithm 8 BLS algorithm

input : $x^{(t)}$, $\sigma \in [0, 1]$, $\beta \in [0, 1]$
$\alpha = 1$
repeat
  $\alpha = \alpha\beta$
  $x^{(t+1)} = [x^{(t)} - \alpha\nabla f(x^{(t)})]$
until $f(x^{(t+1)}) - f(x^{(t)}) \leq \sigma\nabla f(x^{(t)})^\top (x^{(t+1)} - x^{(t)})$
return α, $x^{(t+1)}$

As shown by Lin [145], by projecting the update $x^{(t+1)}$ onto the negative orthant such as $x^{(t+1)} = [x^{(t)} - \alpha \nabla f(x^{(t)})]^+$, BLS can be used to obtain the update $\boldsymbol{Z}^{(t+1)}$ and $\boldsymbol{W}^{(t+1)}$ in NMF. The iterative procedure is given in Algorithm 9.

Algorithm 9 Projected Gradient with Line Search (PGLS) for NMF

input : $\sigma \in [0, 1]$, $\beta \in [0, 1]$
initialize $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$
for t=0,1,2,... **do**
  $\boldsymbol{Z}^{(t+1)} = BLS(\boldsymbol{Z}, \sigma, \beta)$
  $\boldsymbol{W}^{(t+1)} = BLS(\boldsymbol{W}, \sigma, \beta)$
end for

A pertinent review of first-order conditions for Line search methods is proposed by Nocedal and Wright in [146].

Overall the reader can refer to the insightful reviews of Gillis [147] and Ho et al [148] for an overview of algorithms for NMF.

2.1.2 Initialization

Several initializations for NMF are proposed in the literature. Since good starting points can accelerate the convergence toward local minima points and reduce the amount of iterations, plenty of attention has been devoted to that subject.

2.1.2.1 Random seeding

Naturally, the most common initialization technique is to generate random starting points. In addition to its simplicity, this methods is actually justified since no knowledge is given about the local minima. In practice, the algorithm may be sensitive to scaling and the random values are set accordingly to the values observed in $\boldsymbol{X}$. In this thesis, we mainly achieve this initialization using a uniform distribution such that each scalar $z_{ik}^{(0)}$ and $w_{j\ell}^{(0)}$ are respectively set as equal to $\sqrt{\alpha \times \max(\boldsymbol{X})}$ where $\alpha \sim \mathcal{U}(0, 1)$. Other variants of this seeding might be employed (e.g. using the mean instead of the max) and thereby specified at the given time. We might also mention the work of Langville [149] which introduced and compared several random initializations for NMF using the ALS algorithm. We denote the *random Acol* method which initialize each column $\boldsymbol{W}_k^{(0)}$ of the basis matrix by averaging p

random columns of $\boldsymbol{X}$. However, this method might produce sparse factors which should be avoided for algorithms with multiplicative updates.

2.1.2.2 Spherical K-means seeding

Wild et al [150, 151] proposed to initialize NMF using the Spherical K-means algorithms whose objective function is recalled below :

$$J(\boldsymbol{z}, g) = \sum_{i=1}^{n} \sum_{k=1}^{g} z_{ik}(1 - \cos(\mathbf{x_i}, \boldsymbol{\mu}_k)).$$

This method is well suited for directional data such as high dimensional sparse document-term matrices. Two Spherical K-means seeding were attempted in this thesis :

— $z_{ik}^{(0)} := \sqrt{\alpha \times \max(\boldsymbol{X})}$ where $\alpha \sim \mathcal{U}(0,1)$ and $\boldsymbol{W}^{(0)} := \{\boldsymbol{\mu}_1^* | \ldots | \boldsymbol{\mu}_g^*\}$,

— $z_{ik}^{(0)} := \left\{ \begin{array}{ll} 1+\epsilon & \text{if } k = \underset{k=1,\ldots,g}{\arg\min}\ 1 - cos(\mathbf{x_i}, \boldsymbol{\mu}_k^*) \\ \epsilon & \text{otherwise} \end{array} \right\}$ and $\boldsymbol{W}^{(0)} := \{\boldsymbol{\mu}_1^* | \ldots | \boldsymbol{\mu}_g^*\}$,

where $\boldsymbol{\mu}^*$ is the centroïd obtained after convergence. Several observations regarding to the advantages and downsides of this methods when using the Frobenius norm and the I-divergence are made in the following section.

2.1.2.3 SVD-based seedings

A popular SVD-based approach for generating starting values for NMF was introduced by Boutsidis and Gallopoulos [152]. The method is referred to as *Non Negative Double Singular Value Decomposition (NNDSVD)* is based on two SVD processes. The first requires the SVD of the data matrix $\boldsymbol{X} \in \mathbb{R}_+^{n \times d}$ to create a set of unit rank matrices from the left and right singular vectors. The second successively utilizes the positive orthants of each unit rank matrix to approximate a set of positive singular vectors. In practice, the SVD is a low rank approximation produced with a truncated SVD (t-SVD). Let $\boldsymbol{U} = [\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_n] \in \mathbb{R}^{n \times n}$ and $[\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_d] \in \mathbb{R}^{d \times d}$ be the orthogonal matrices obtained by the SVD of $\boldsymbol{X}$ such that $\boldsymbol{U}^\top \boldsymbol{X} \boldsymbol{V} = \boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_p)$ where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ are the singular values and $p = \min(n, d)$. From the Eckart-Young theorem, the LRA of $\boldsymbol{X}$ is given as :

$$\underset{\text{rank}(\boldsymbol{Y}) \leq g}{\arg\min} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 = \boldsymbol{X}_g = \sum_{k=1}^{g} \boldsymbol{u}_k \sigma_k \boldsymbol{v}_k^\top = \sum_{k=1}^{g} \sigma_k \boldsymbol{C}_k, \tag{2.23}$$

where $\{\boldsymbol{C}^{(1)}, \ldots, \boldsymbol{C}^{(g)}\}$ is the set of unit rank matrices such that $\boldsymbol{C}^{(k)} = \boldsymbol{u}_k \boldsymbol{v}_k^\top \in \mathbb{R}^{n \times d}$. In the sequel, we describe the basic routine for NNDSVD, however, the authors proposed a much efficient approach results in the properties of the positive orthants $\{[\boldsymbol{C}^{(1)}]^+, \ldots, [\boldsymbol{C}^{(g)}]^+\}$ of the unit rank matrices with respect to the Frobenius norm. The procedure is described in Algorithm 10.

Algorithm 10 NNDSVD

input : $\boldsymbol{X}$, g
compute the t-SVD of $\boldsymbol{X}$ s.t. $\boldsymbol{X}_g = \sum_{k=1}^{g} \boldsymbol{u}_k \sigma_k \boldsymbol{v}_k^\top$.
$\boldsymbol{C}^{(k)} = \boldsymbol{u}_k \boldsymbol{v}_k^\top, \forall k = 1, \ldots, g$
compute $[\boldsymbol{C}^{(k)}]^+, \forall k = 1, \ldots, g$
$\boldsymbol{Z}_1^{(0)} = \sqrt{\sigma_1} \boldsymbol{u}_1$
$\boldsymbol{W}_1^{(0)} = \sqrt{\sigma_1} \boldsymbol{v}_1$
for k = 2,. . . ,g **do**
  compute the t-SVD of $[\boldsymbol{C}^{(k)}]^+$ s.t. $[\boldsymbol{C}_2^{(k)}]^+ = \sum_{l=1}^{2} \boldsymbol{u}_l{}' \sigma_l{}' \boldsymbol{v}_l{}'^\top$
  $\boldsymbol{Z}_k^{(0)} = \sqrt{\sigma_k} \boldsymbol{u}_1{}'$
  $\boldsymbol{W}_k^{(0)} = \sqrt{\sigma_k} \boldsymbol{v}_1{}'$
end for
return $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$

This algorithm produces the same initialization for a given data matrix. Two variants named NNDSVDa and NNDSVDar which replace null entries in the resulting initialization factors $(\boldsymbol{Z}^{(0)}, \boldsymbol{W}^{(0)})$ are denoted. The first replaces zeros with the average value of $\boldsymbol{X}$, the second simulate uniform variables following $\mathcal{U}(0, \text{mean}(\boldsymbol{X})/100)$.

Remark. This algorithm uses the fact that the first singular vector is positive if $\boldsymbol{X} \geq 0$.

Recently, Atif et al [153] proposed to correct the low rank approximation on the basis that the approximation error should decrease as the rank increases.

We can also mention the work of Qiao [154] which proposed to use the absolute value of the SVD singular vectors to provide starting values for the NMF factors. Moreover, this approach also consider the fixation of the rank using a "choosing" rule on the positive singular values.

2.1.2.4 Stopping conditions

Several stopping conditions are denoted for NMF algorithms. The most common are analogical to linear optimization and consist in monitoring the evolution of the objective function $\mathcal{F}$ or the optimality conditions or the estimates. Note that in practice, these conditions are usually associated with a

maximum number of iterations. Monitoring the objective function remains the most popular approach and is the one used in this thesis. It is achieved by measuring the decrease between $\mathcal{F}(\boldsymbol{Z}^{(t+1)}, \boldsymbol{W}^{(t+1)})$ and $\mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{W}^{(t)})$ such that the algorithm is stopped when :

$$\mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{W}^{(t)}) - \mathcal{F}(\boldsymbol{Z}^{(t+1)}, \boldsymbol{W}^{(t+1)}) < \epsilon, \tag{2.24}$$

where ϵ is usually a very small constant. Several approaches for optimality stopping conditions are reviewed and discussed in [147, 155].

Remark. In NMF applications for document clustering, the coefficient factor matrix $\boldsymbol{Z}$ is usually normalized to have unit-length column vectors at the end of the procedure.

2.1.3 Extensions and variants

Several extensions and variants for NMF have been proposed to scope with a large range of applications. For instance, whether the data matrix is symmetric, we denote the variant called *Symmetric NMF* which produces an approximation by the product of one matrix.

Symmetric NMF (Kuang et al. [156]). Let $\boldsymbol{X} \in \mathbb{R}_+^{n \times n}$, the problem of NMF when $\boldsymbol{X}$ is a symmetric matrix can be stated as follows :

$$\min_{\boldsymbol{H} \geq 0} \|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{H}^\top\|_F^2. \tag{2.25}$$

Orthogonal NMF. Introduced by Ding et al [157], this approach consists in adding an orthogonality constraint on one factor to produce straightforward clustering interpretation. Let $\boldsymbol{X} \in \mathbb{R}_+^{n \times d}$, $\boldsymbol{Z} \in \mathbb{R}_+^{n \times g}$ and $\boldsymbol{W} \in \mathbb{R}_+^{d \times g}$, Orthogonal NMF (ONMF) where the cost function is the Frobenius norm can be stated as the following optimization problem :

$$\min_{\substack{\boldsymbol{Z} \geq 0, \boldsymbol{W} \geq 0, \\ \boldsymbol{Z}^\top \boldsymbol{Z} = \mathbf{1}_g}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2. \tag{2.26}$$

The authors derived an optimization algorithm referred to as the Ding-Ti-Peng-Park (DTTP) algorithm. It is based on the multiplicative update rules popularised by Lee and Seung [132]. One benefit provides by the orthogonality constraints is the uniqueness of the solution. Later, Choi [158, 159] also proposed to solve the optimization using a simpler approach which consists in placing $\boldsymbol{Z}$ in the *Stiefel Manifolds* $\mathbb{V}_g(\mathbb{R}^n) = \boldsymbol{Z} \in \mathbb{R}^{n \times g} : \boldsymbol{Z}^\top \boldsymbol{Z} = \mathbf{I}$ which is the set of all orthonormal g-frames.

Another declination of NMF called *Semi-NMF* was introduced for real value data matrices. This method consists in imposing the nonnegativity on one factor while the other lies in an unconstrained space similar the one of the original data matrix.

Semi-NMF. Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ be a real value matrix, *Semi-NMF* can be stated as the following optimization problem :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\in\mathbb{R}^{d\times g}} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2. \tag{2.27}$$

This methods has been proposed by Ding et al. [160] for several clustering applications including document clustering. Another approach illustrated in the same paper was *Convex-NMF*. This method aims at adding domain constraints on the basis vector vectors $[\boldsymbol{w}_1|\ldots|\boldsymbol{w}_g]$ stored in $\boldsymbol{W}$ such that their lies in the columns space of the original data matrix. This constraint takes weighted-sum interpretation due to the nonnegativity of one factor and subsequently constraint $\boldsymbol{W}$ to be equal to $\boldsymbol{X}^\top\boldsymbol{G}$. It follows that this constraint takes places whether $\boldsymbol{X}$ is a nonnegative matrix or a real value matrix.

Convex-NMF. Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, *Convex-NMF* can be stated as the following optimization problem :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{G}\geq 0} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{G}^\top\boldsymbol{X}\|_F^2, \tag{2.28}$$

where $\boldsymbol{Z} \in \mathbb{R}_+^{n\times g}$ and $\boldsymbol{G} \in \mathbb{R}_+^{n\times g}$.

Projective NMF. This method was introduced by Yuan and Oja [161] after Lee and Seung pointed out in [5] the benefits of nonnnegative constraints for retrieving sparses representations for images. *Projective NMF* aims at improving this characteristic by learning spatially localized sparse part-based factor of visuals patterns in images. Given a matrix $\boldsymbol{P} \in \mathbb{R}_+^{n\times n}$, the projection is equivalent to solving the following minimization problem :

$$\min_{\boldsymbol{P}\geq 0} \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{X}\|_F^2. \tag{2.29}$$

Furthermore, the symmetric matrix projection $\boldsymbol{P}$ is set as the product of an orthogonal $\boldsymbol{H} \in \mathbb{R}^{n\times g}$ matrix such that $\boldsymbol{P} = \boldsymbol{H}\boldsymbol{H}^\top$ wich result in minimizing $\|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{X}\|_F^2$ subject to $\boldsymbol{H} \geq 0$. The problem with the Kullback-Leibler as a cost function was also treated in [161]. The authors highlighted the connection with PCA after removing the nonnegative constraint. In the following chapters, we will

use **Projective NMF** in our comparative studies since the performances of the method for clustering tasks were later enhanced by Yuan and Erkki Oja in [162].

Regularized NMF. Considering the initial NMF problem where $\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{W}^\top$, we encapsulate the set of NMF extensions where a regularization or penalization is added to the main objective function under the following minimization problem :

$$\min_{\boldsymbol{Z}\in\mathbb{R}_+^{n\times g},\boldsymbol{W}\in\mathbb{R}_+^{d\times g}} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2 + \alpha\mathcal{J}_1(\boldsymbol{Z},.) + \beta\mathcal{J}_2(\boldsymbol{W},.). \tag{2.30}$$

In most applications, the regularization parameters (α, β) are constants set by the user, $\mathcal{J}_1 = \mathcal{J}_2$ or the regularization is applied on only one factor. An example of regularized NMF employed in our comparative studies is called Graph Regularized NMF (GNMF) and was proposed by Cai et al. [163]. The method aims at enforcing NMF at capturing the intrinsic geometry of the original data $\boldsymbol{x}_i$ using the Laplacian of a nearest neighbors graph. The graph is defined as an unoriented simple graph and its symmetric adjacency matrix $\boldsymbol{A} = (a_{ii'}) \in \{0,1\}^{n\times n}$ is given by :

$$a_{ii'} = \left\{ \begin{array}{ll} 1 & \text{if } \boldsymbol{x}_i \in N_p(\boldsymbol{x}_{i'}) \\ 0 & \text{otherwise} \end{array} \right\},$$

where $N_p(\boldsymbol{x}_{i'})$ denotes the p nearest neighbors subset of $\boldsymbol{x}_{i'}$. The Laplacian matrix $\boldsymbol{L}$ is given by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D} = (d_{ii'}) \in \mathbb{R}_+^{n\times n}$ is the diagonal degree matrix computed from $\boldsymbol{A}$ such that $d_{ii} = \sum_{i'} a_{ii'}$. The optimization problem takes the following from :

$$\min_{\boldsymbol{Z}\in\mathbb{R}_+^{n\times g},\boldsymbol{W}\in\mathbb{R}_+^{d\times g}} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2 + \alpha\,\mathrm{Tr}(\boldsymbol{Z}^\top\boldsymbol{L}\boldsymbol{Z}). \tag{2.31}$$

Besides, Shang et al. [164] proposed graph dual regularized NMF, which extends GNMF to model both the data manifold and feature manifold simultaneously. In order to reduce the sensitivity of GNMF to the nearest neighbor graph's parameters, the authors in [165] developed multiple graph regularized NMF where the the data manifold is approximated by a linear combination of several nearest neighbor graphs having different parameters. In the same vein, more robust extensions of NMF, which can handle data points lying in complex manifolds, have been recently proposed [166, 167].

For more NMF extensions and variants, the reader can refer to [168, 135] or [169] for a wider perspective of NMF. Also, a series of works [170, 171, 160] established theoretical connections of NMF with k-means and spectral clustering, which strengthen foundations for NMF-based clustering.

2.1.3.1 Nonnegative Matrix Tri-Factorization

Given a nonnegative matrix $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}_+^{n \times d}$, Nonnegative Matrix Tri-factorization (NMTF) is a dimensionality reduction method which aims at approximating $\boldsymbol{X}$ by the production of three lower dimensional matrices $\boldsymbol{Z} \in \mathbb{R}_+^{n \times g}$, $\boldsymbol{S} \in \mathbb{R}_+^{g \times c}$ and $\boldsymbol{W}_+^{d \times c}$, e.g.

$$\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{S}\boldsymbol{W}^\top. \tag{2.32}$$

An illustration of NMTF is given in Figure 2.2 This method was introduced by Long et al [172] under

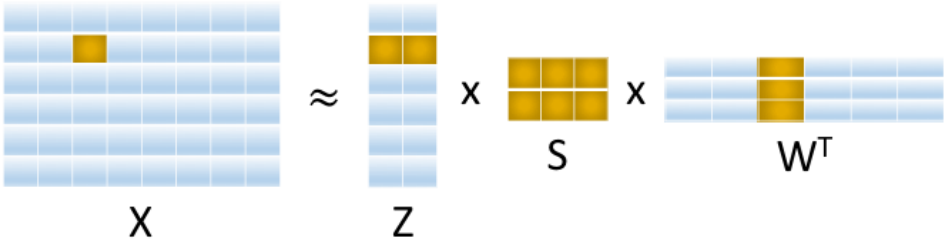


FIGURE 2.2 – NMTF.

the name Nonnegative Block Value Decomposition (NBVD) as an alternative to NMF for clustering of dyadic data (e.g count data) arising an underlying two way structure. As for NMF, NMTF can be expressed in a form of an optimization problem with the Frobenius norm remaining one of the most common cost function.

Following the weighting-sum interpretation of NMF, the co-clusters is deduced from the coefficient matrices $\boldsymbol{Z}$ and $\boldsymbol{W}$ whilst $\boldsymbol{S}$ can be seen as a summary (or block decomposition) of the original data matrix $\boldsymbol{X}$.

Problem 2.1.2. *(**NMTF**). Let $\boldsymbol{X} \in \mathbb{R}_+^{n \times d}$, $(g, c) < \min(n, d)$, solve :*

$$\min_{\boldsymbol{Z} \geq 0, \boldsymbol{S} \geq 0, \boldsymbol{W} \geq 0} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{S}\boldsymbol{W}^\top\|_F^2. \tag{2.33}$$

Similarly to NMF, several extensions and variants for NMTF including additional constraints such as orthogonality or a regularization in the objective are denoted.

Orthogonal Nonnegative Matrix Tri-Factorization (ONM3F). Ding et al.[157] introduced a 3 factors NMF with bi-orthogonality constraints on 2 factors for practicing co-clustering with NMTF. Conside-

ring $\boldsymbol{X} \in \mathbb{R}_+^{n\times d}$, $\boldsymbol{Z} \in \mathbb{R}_+^{n\times g}$, $\boldsymbol{S} \in \mathbb{R}_+^{g\times c}$ and $\boldsymbol{W} \in \mathbb{R}_+^{d\times c}$, where $\boldsymbol{Z}$ and $\boldsymbol{W}$ are orthogonal matrices and $\boldsymbol{S}$ acts as a relaxation factor, ONM3F can be stated as the following optimization problem :

$$\min_{\substack{\boldsymbol{Z}\geq 0,\boldsymbol{W}\geq 0,\\ \boldsymbol{Z}^\top\boldsymbol{Z}=\mathbf{1}_g,\boldsymbol{W}^\top\boldsymbol{W}=\mathbf{1}_g}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Z}\boldsymbol{S}\boldsymbol{W}^\top\|_F^2. \tag{2.34}$$

Similarly to its NMF counterpart, the optimization procedure was achieved using a set of multiplicative update rules. As with NMF, it is possible to consider the *Stiefel manifold* for achieving orthogonality on the NMTF outer factors. The method was suggested by yoo and Choi [173] and is referred subsequently as $ONMTF\_SM$.

Symmetric NMTF. As with NMF, a symmetric variant was considered by several authors. Long et al [172] proposed a method called SNBVD for approximating a symmetric data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ as follows : $\boldsymbol{X} \approx \boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^\top$, where $\boldsymbol{S} \in \mathbb{R}_+^{g\times g}$ and $\boldsymbol{H} \in \mathbb{R}_+^{n\times g}$. This method can be stated as the subsequent minimization problem :

$$\min_{\boldsymbol{H}\geq 0,\boldsymbol{S}\geq 0} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{H}\boldsymbol{S}\boldsymbol{H}^\top\|_F^2. \tag{2.35}$$

Ding et al.[157] also proposed a variant called SONM3F which differs from the later by adding an orthogonality constraint on $\boldsymbol{H}$ such that $\boldsymbol{H}^\top\boldsymbol{H} = \mathbf{I}$.

We also denote several regularizations including the work of Gu et al. [174] which proposed a Dual Regularized Co-clustering (DRCC) method based on semi-NMTF and a nearest neighbors graph regularization. The proposals of Wang et al [175] referred to as Fast NMTF (FNMTF) which constrains the factor matrices ($\boldsymbol{Z}$,$\boldsymbol{W}$) to be cluster indicators and Locally Preserved FNMTF (LP_FNMTF) which adds a Graph regularization to the problem of FNMTF.

We can also mention the work of Wang [175] which focuses on High-Order co-clustering with NMTF and decomposes conjointly multiple type of data. We summarizes in table 2.1 the objective functions of several NMTF methods and evaluate in a comparative study of Febrissy [176] regarding the task of document clustering.

TABLE 2.1 – NMTF variants & extentions.

Methods	Objective functions	References
SONM3F	$\|\boldsymbol{X} - \boldsymbol{HSH}^\top\|_F^2$ s.t. $\boldsymbol{H} \geq 0, \boldsymbol{S} \geq 0$ and $\boldsymbol{H}^\top\boldsymbol{H} = \boldsymbol{I}$	[157]
SNBVD	$\|\boldsymbol{X} - \boldsymbol{HSH}^\top\|_F^2$ s.t. $\boldsymbol{H} \geq 0, \boldsymbol{S} \geq 0$	[172]
ONM3F	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2$ s.t. $\boldsymbol{Z} \geq 0, \boldsymbol{S} \geq 0, \boldsymbol{W} \geq 0$ and $\boldsymbol{Z}^\top\boldsymbol{Z} = \boldsymbol{I}, \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}$	[157]
NBVD	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2$ s.t. $\boldsymbol{Z} \geq 0, \boldsymbol{S} \geq 0, \boldsymbol{W} \geq 0$	[172]
ONMTF_SM	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2$ s.t. $\boldsymbol{Z} \geq 0, \boldsymbol{S} \geq 0, \boldsymbol{W} \geq 0$ and $\boldsymbol{Z}^\top\boldsymbol{Z} = \boldsymbol{I}, \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}$	[158]
DRCC	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2 + \lambda \operatorname{Tr}(\boldsymbol{W}^\top\boldsymbol{L}_W\boldsymbol{W}) + \mu \operatorname{Tr}(\boldsymbol{Z}^\top\boldsymbol{L}_Z\boldsymbol{Z})$ s.t. $\boldsymbol{Z} \geq 0, \boldsymbol{S} \in \mathbb{R}^{n\times g}\boldsymbol{W} \geq 0$	[174]
FNMTF	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2$ s.t. $\boldsymbol{Z} \in \Psi^{n\times g}$, $\boldsymbol{W} \in \Psi^{d\times c}$	[175]
LP_FNMTF	$\|\boldsymbol{X} - \boldsymbol{ZSW}^\top\|_F^2 + \alpha\|\boldsymbol{W} - \boldsymbol{B}_d\boldsymbol{Q}_d\|^2 + \beta\|\boldsymbol{Z} - \boldsymbol{B}_f\boldsymbol{Q}_f\|^2$ s.t. $\boldsymbol{Q}_d^\top\boldsymbol{Q}_d = \boldsymbol{I}, \boldsymbol{Q}_f^\top\boldsymbol{Q}_f = \boldsymbol{I}$	[175]
O-NMTF	$\|\boldsymbol{R} - \boldsymbol{GVG}^\top\|_F^2 + \lambda \operatorname{Tr}(\boldsymbol{G}^\top\boldsymbol{LG})$ s.t. $\boldsymbol{G} \geq 0, \boldsymbol{G}^\top\boldsymbol{DG} = \boldsymbol{I}$	[177]

where $\boldsymbol{R} = \begin{bmatrix} 0_{n\times n} & \boldsymbol{X}_{n\times d} \\ \boldsymbol{X}^\top_{d\times n} & 0_{d\times d} \end{bmatrix}$, $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{Z}_{n\times g} & 0_{n\times c} \\ 0_{d\times g} & \boldsymbol{W}_{d\times c} \end{bmatrix}$, $\boldsymbol{V} = \begin{bmatrix} 0_{g\times g} & \boldsymbol{S}_{g\times c} \\ \boldsymbol{S}^\top_{c\times g} & 0_{c\times c} \end{bmatrix}$, $\boldsymbol{I} \leftrightarrow$ identity matrix, $\Psi = \{0, 1\}$.

2.2 A consensus approach to improve NMF document clustering

2.2.1 Motivations

Despite its mathematical elegance and simplicity, NMF has exposed a main issue which is its strong sensitivity to starting points, resulting in NMF struggling to converge toward an optimal solution. On another hand, we came to explore and discovered that even after providing a meaningful initialization, selecting the solution with the best local minimum was not always leading to the one having the best clustering quality, but somehow a better clustering could be obtained with a solution slightly off in terms of criterion. Therefore in this section, we undertake to study the clustering characteristics and quality of a set of NMF best solutions and provide a method delivering a better partition using a consensus made of the best NMF solutions.

Unlike supervised learning, the evaluation of clustering algorithms - unsupervised learning - remains a difficult problem. When relying on generative models, it is easier to evaluate the performance of a given clustering algorithm based on the simulated partition. On real data already labeled, many papers evaluate the performance of clustering algorithms by relying on indices such as Accuracy (ACC),

Normalized Mutual Information (NMI) [119] and Adjusted Rand Index (ARI) [122]. However, the algorithms commonly used which are of type k-means, EM [90], Classification EM [79], NMF [132] etc. are iterative and require several initializations ; the resulting partition is the one optimizing the objective function. Sometimes in these works, we observe comparative studies between methods on the basis of maximum ACC/NMI/ARI measures obtained after several initializations and not optimizing the criterion used in the algorithm. Such a comparison is thereby not accurate, because in fact these measures cannot be calculated in practice and cannot be used in this way to evaluate the quality of a clustering algorithm.

A fair comparison can only be made on the basis of objective functions considered in a clustering purpose ; for example, within-cluster inertia, likelihood, classification likelihood for mixture models, factorization, etc. Nonetheless, in our experiences, we realized that while the clustering results become better in terms of ACC/NMI/ARI when the objective function value increases, the best value is not necessarily associated with the best results. However, by ranking the objective values, the best partition tends to be among those leading to the first best scores. We illustrate this behavior in Figure 2.6. This remark leads us to consider an *ensemble method* that is widely used in supervised learning [178, 179] but a little less in unsupervised learning [119]. If this approach, referred to as *consensus clustering*, is often used in the context of comparing partitions obtained with different algorithms, it is less studied considering the same algorithm.

In the following, the algorithm used to provide a solution for NMF is the multiplicative updates (MU) and the optimization of NMF (problem 2.6) is considered when $\mathcal{D}$ is respectively equal to the Frobenius norm and the KL divergence. The MU algorithm accordingly to each objective is given by algorithm 11 and algorithm 12.

Algorithm 11 (`NMF-F`).

Input : $\boldsymbol{X}$, g, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
1. $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{X}\boldsymbol{W}}{\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}}$;
2. $\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}^\top\boldsymbol{Z}}{\boldsymbol{W}\boldsymbol{Z}^\top\boldsymbol{Z}}$;

until convergence
5. Normalize $\boldsymbol{Z}$ so as it has unit-length column vectors.

Algorithm 12 (`NMF-KL`).

Input : $\boldsymbol{X}$, g, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
1. $z_{ik} \leftarrow z_{ik}\big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\big)_{ik} / \sum_j^d w_{jk}$;
2. $w_{jk} \leftarrow w_{jk}\big(\frac{\boldsymbol{X}^\top}{\boldsymbol{W}\boldsymbol{Z}^\top}\boldsymbol{Z}\big)_{jk} / \sum_i^n z_{ik}$;

until convergence
5. Normalize $\boldsymbol{Z}$ so as it has unit-length column vectors.

2.2.2 Cluster ensembles (CE)

In machine learning, the idea of utilizing multiple sources of data partitions firstly occurred with multi-learner systems where the output of several classifier algorithms where used together in order to improve the accuracy and robustness of a classification or regression, for which strong performances were acknowledged [119, 180, 178]. At this stage, very few approaches have worked toward applying a similar concept to unsupervised learning algorithms. In this sense, we denote the work of [181] who tried to combine several clustering partitions according to the combination of the cluster centers. In the early 2000, [119] were the first to consider an idea of combining several data partitions however, without accessing any original sources of information (features) or led computed centers. This approach is referred to as *cluster ensembles*. At the time, their idea was motivated by the possibility of taking advantage of existing information such as a prior clustering partitions or an expert categorization (all regrouped under the terms Knowledge Reuse), which may still be relevant or substantial for a user to consider in a new analysis on the same objects, whether or not the data associated with these objects may also be different than the ones used to define the prior partitions. Another motivation was *Distributed computing*, referring to analyzing different sources of data (which might be complicated to merge together for instance for privacy reasons) stored in different locations. In our concept, we will use *cluster ensembles* to improve the quality of the final partition (as opposed to selecting a unique one) and therefore extract all the possibilities offered by the miscellaneous best solutions created by NMF.

In [119], the authors introduced three consensus methods that can produce a partition. All of them consider the consensus problem on a hypergraph representation $\boldsymbol{H}$ of the set of partitions $\boldsymbol{H}^r$. More specifically, each partition $\boldsymbol{H}^r$ equals a binary classification matrix (with objects in rows and clusters in columns) where the concatenation of all the set defines the hypergraph $\boldsymbol{H}$.

— The first one is called Cluster-based Similarity Partitioning Algorithm (**CSPA**) and consists in performing a clustering on the hypergraph according to a similarity measure.

— The second is referred to as HyperGraph Partitioning Algorithm (**HGPA**) and aims at optimizing a minimum cut objective.

— The third one is called Meta-CLustering Algorithm (**MCLA**) and looks forward to identifying and constructing groups of clusters.

Furthermore, in [119] the authors proposed an objective function to characterize the *cluster ensembles*

problem and therefore allowing a selection of the best consensus algorithm among the three to deliver its ensemble partition. Let $\Lambda = \{\lambda^{(q)} | q \in \{1, \ldots, r\}\}$ be a given set of r partitions $\lambda^{(q)}$ represented as labels vectors. The ensemble criterion denoted as $\lambda^{(k-opt)}$ is called the optimal combine clustering and aims at maximizing the Average Normalized Mutual Information (ANMI). It is defined as follows :

$$\lambda^{(k-opt)} = \arg\max_{\widetilde{\lambda}} \sum_{q=1}^{r} \mathrm{NMI}(\widetilde{\lambda}, \lambda^{(q)}). \tag{2.36}$$

The ANMI is simply the average of the normalized mutual information of a labels vector $\widetilde{\lambda}$ with all labels vectors $\lambda^{(q)}$ in Λ :

$$\mathrm{ANMI}(\Lambda, \widetilde{\lambda}) = \frac{1}{r} \sum_{q=1}^{r} \mathrm{NMI}(\widetilde{\lambda}, \lambda^{(q)}). \tag{2.37}$$

To cast with cases where the vector labels $\lambda^{(q)}$ have missing values, the authors have proposed a generalized expression of (2.36) not substantially different that viewers can refer to in the original paper [119].

2.2.3 Experiments

We conduct several experiences leading to emphasise the behavior of NMF regarding a clustering task compared to a dedicated clustering algorithm such as Spherical K-means referred to as `S-Kmeans` [39] which was introduced for clustering large sets of sparse text data (or directional data) and remains appealing for its low computational cost beside its good performances. It was also retained along side the random starting points (generated according to an uniform distribution $\mathcal{U}(0,1) \times mean(\boldsymbol{X})$) as initialization for NMF. We use two error measures frequently employed for NMF : the Frobenius norm (which will be referred to as `NMF-F`) and the Kullback-Leibler divergence (`NMF-KL`). Eventually, we compute the consensus partition by using the Cluster Ensemble Python package [1] which utilizes the consensus methods defined earlier [119].

2.2.3.1 Datasets

We apply NMF on 5 bench-marking document-term matrices for which the detailed characteristics are available in Table 2.2 where nz indicates the percentage of values other than 0 and the *balance* coefficient is defined as the ratio of the number of documents in the smallest class to the number

1. https ://pypi.org/project/Cluster_Ensembles/

of documents in the largest class. These datasets highlight several varieties of challenging situations such as the amount of clusters, the dimensions, the clusters balance, the degree of mixture of the different groups and the sparsity. We normalized each data matrix with TF-IDF and their respective documents-vectors to unit L_2-norm to remove the bias introduced by their length.

TABLE 2.2 – Datasets description : # denotes the cardinality.

Datasets	Characteristics				
	#Documents	#Words	#Clusters	$nz(\%)$	Balance
CSTR	475	1000	4	3.40	0.399
CLASSIC4	7095	5896	4	0.59	0.323
RCV1	6387	16921	4	0.25	0.080
NG5	4905	10167	5	0.92	0.943
NG20	18846	14390	20	0.59	0.628

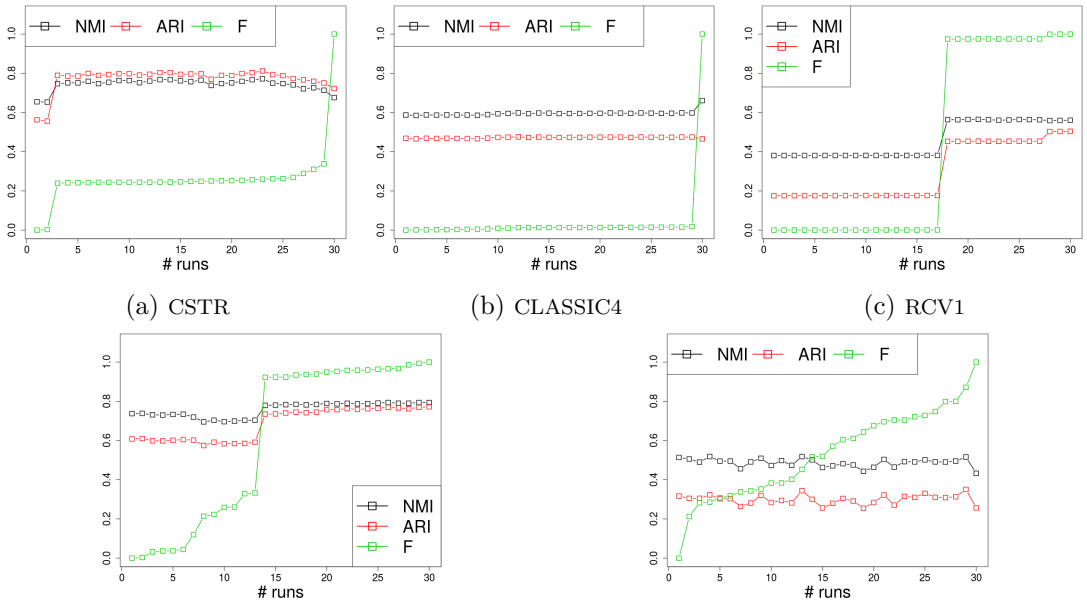
2.2.3.2 NMF raw performances and initialization

The results obtained by `NMF-F` and `NMF-KL` according to `S-Kmeans` and the random starting points are available in Table 2.3. The clustering quality of the `S-Kmeans` partitions given as entry to both algorithms are also displayed. We make use of two relevant measures to quantify and assess the clustering quality of each algorithm. The first one is the NMI [119] which quantifies how much information the clustering partition shares with the true partition, the second is the ARI [122], sensitive to the clusters proportions and measures the degree of agreement between the clustering and the true partition. To replicate a relevant user experience achieving an unsupervised task, we refer to the criterion of each algorithm in order to select the 10 first best solutions (out of 30 runs) and report their average NMI and ARI with the true partition.

One can clearly see that `NMF-F` and `NMF-KL` do not react similarly to the different initializations. While `NMF-F` substantially benefits from the `S-kmeans` initialization on every datasets compared to the random initialization, `NMF-KL` does not seem to accommodate `S-kmeans` entries. In fact, `S-Kmeans` as starting values seems to worsen `NMF-KL` solutions, especially on CLASSIC4 and NG5. For this reason, we will avoid this initialization strategy for `NMF-KL` in the future although it improves on RCV1. Also, `NMF-KL` with a random initialization provides much better results than the other algorithms on almost all datasets. We reported in Figures 2.3-2.6 the clustering quality of the algorithm's solutions ranked from the best one in terms of criterion to the poorest one. The respective criterion of each algorithm is normalized to belong to $[0, 1]$.

TABLE 2.3 – Mean and standard deviation of NMI and ARI computed over the 10 best solutions.

Datasets	Metrics	`Skmeans`	`NMF-F (Random)`	`NMF-F (Skmeans)`	`NMF-KL (Random)`	`NMF-KL (Skmeans)`
CSTR	NMI	0.76±0.007	0.65±0.002	0.73±0.04	0.73±0.03	0.76±0.006
	ARI	0.80±0.007	0.55±0.002	0.75±0.10	0.77±0.04	0.80±0.006
CLASSIC4	NMI	0.60±0.001	0.53±0.003	0.59±0.002	0.71±0.02	0.61±0.03
	ARI	0.47±0.0009	0.45±0.003	0.47±0.002	0.65±0.06	0.47±0.004
RCV1	NMI	0.38±0.0003	0.35±0.0005	0.38±0.0002	0.47±0.02	0.53±0.002
	ARI	0.18±0.0004	0.13±0.0008	0.18±0.0003	0.42±0.02	0.46±0.02
NG5	NMI	0.72±0.02	0.56±1.0e-05	0.72±0.02	0.80±0.03	0.79±0.003
	ARI	0.60±0.01	0.33±2.5e-05	0.60±0.01	0.82±0.04	0.76±0.005
NG20	NMI	0.49±0.02	0.41±0.01	0.49±0.02	0.48±0.02	0.51±0.01
	ARI	0.30±0.02	0.23±0.01	0.30±0.02	0.34±0.02	0.32±0.02



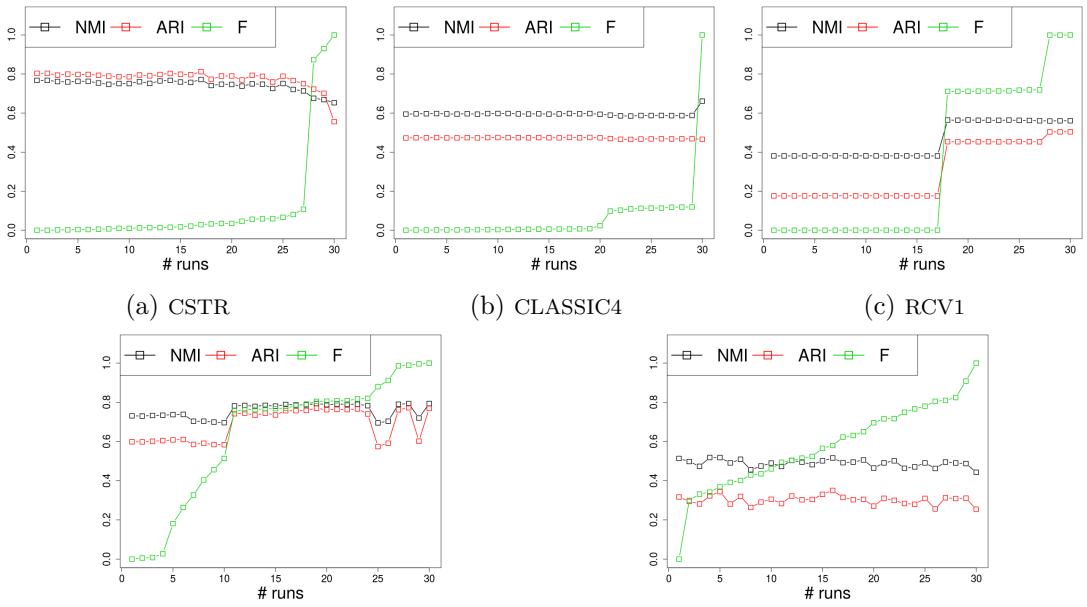
(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) NG20

FIGURE 2.3 – `NMF-F` : NMI/ARI behaviour according to the objective function F (initializations by `S-Kmeans`).

When one does have the real partition, a common practice to evaluate the clustering result, one relies on the best solution obtained by optimizing the objective function. Figures 2.3 and 2.5 highlight a critical behavior of `NMF-F` which tends to produce solutions with the lowest minima that do not fulfil the best clustering partitions, sometimes with a substantial gap (see CSTR, RCV1, NG5 in Figure 2.3). Moreover, a surprising lesser but still similar behavior is delivered by `S-Kmeans` which compared to `NMF`, optimizes a clustering objective by definition. The results are displayed in Figure 2.4. In reality,

this behavior can be observed with several types of what we refer to clustering algorithms hosting an optimization procedure. Initializing `NMF-F` randomly as shown in Figure 2.5 seems to lighten this effect (on CSTR, Classic4 and RCV1). On another hand, `NMF-KL` which to this day remains recognized as a relevant method for document clustering [134] seems to consistently deliver solutions with the lowest criteria aligned with the goodness of their clustering, sustaining the use of NMF for clustering purposes. Furthermore, compared to all, `NMF-KL` is the only method emphasizing a wide variety of solutions and therefore seems to explore way more possibilities than `NMF-F` or `S-Kmeans`. Its better behavior might almost comfort the idea of selecting the best partition in terms of criterion as the one to keep. However, it still fails on RCV1 which is the toughest dataset to partition mainly because of its scant density. Eventually, it remains concerning to select the best partition just based on the fact that, even with `NMF-KL`, the solution among the best ones providing the best clustering, is not necessarily the first one (see on CSTR, CLASSIC4 and NG5).



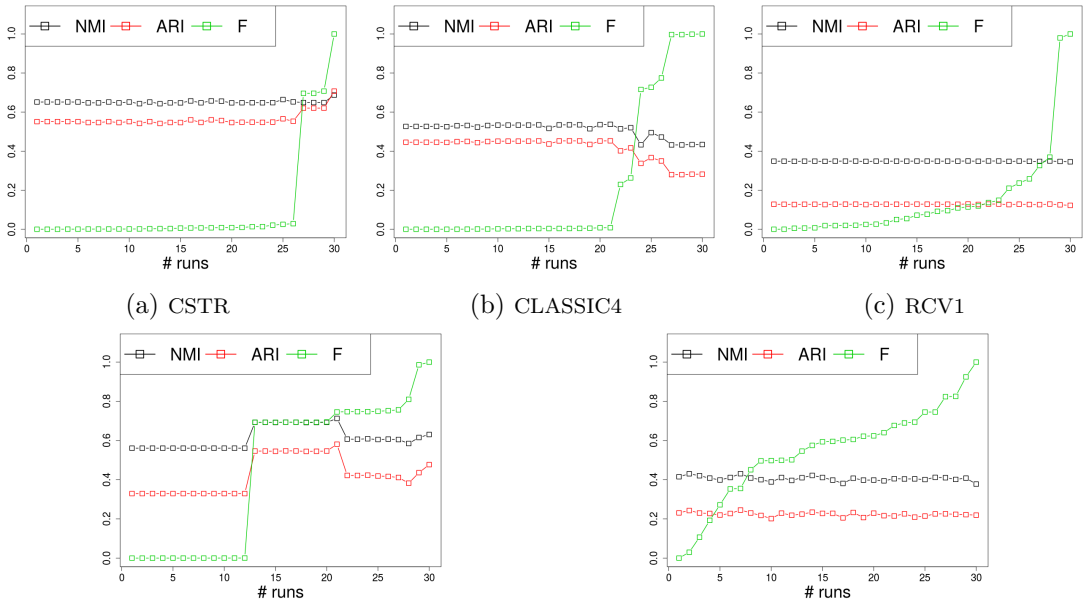
(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) NG20

FIGURE 2.4 – `S-Kmeans` : NMI/ARI behaviour according to the objective function F (Random initializations).

In addition, while the best solutions possibly share a similar amount of information with the true partition, they could be fairly distinct from each other, making their use appealing to deduce an even more exhaustive solution. Figure 2.7 shows results of pairwise NMI and ARI between the top 10 partitions (criterion-wise) of each algorithm. `NMF-KL`'s best solutions appear to be fairly different

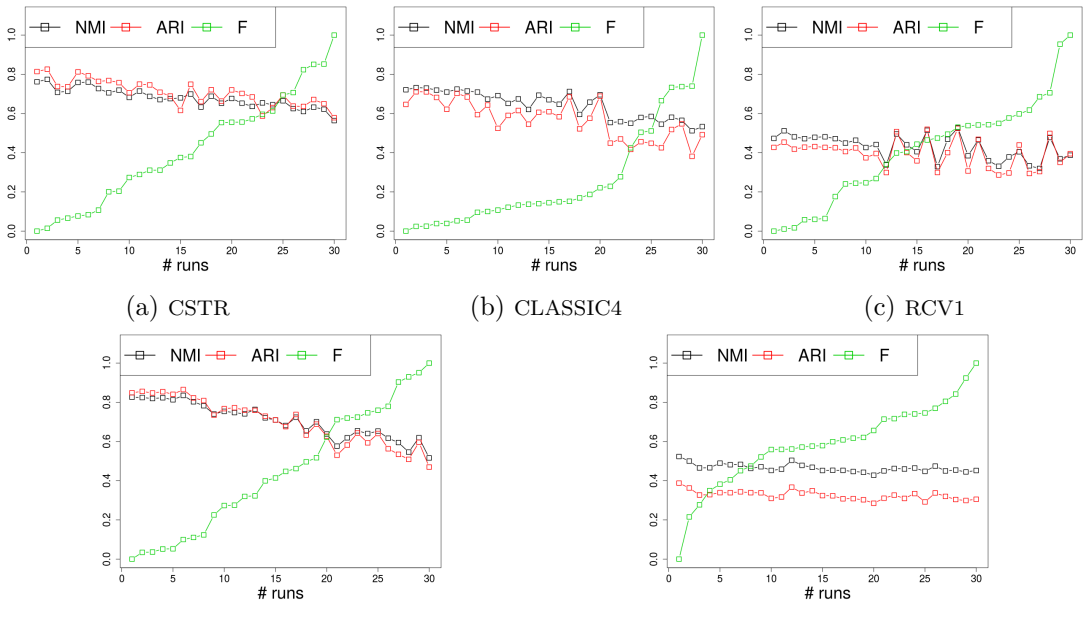
among each other.



(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) NG20

FIGURE 2.5 – `NMF-F` : NMI/ARI behaviour according to the objective function F (Random initializations).



(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) NG20

FIGURE 2.6 – `NMF-KL` : NMI/ARI behaviour according to the objective function F (Random initializations).

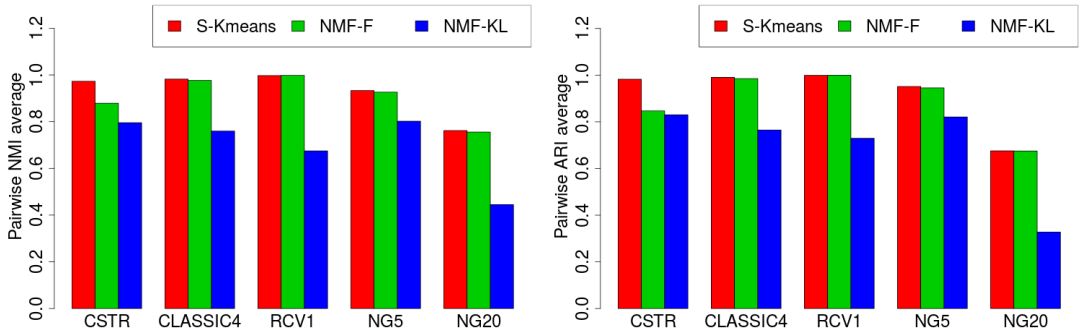


FIGURE 2.7 – Pairwise NMI & ARI averages between the top 10 solutions.

2.2.3.3 Consensus clustering

TABLE 2.4 – Mean and standard deviation, first best result and CE consensus computed over the 10 best solutions.

Datasets	Metrics	`NMF-F (Skmeans)`			`Skmeans`			`NMF-KL (Random)`		
		Mean±SD	(best)	CE	Mean±SD	(best)	CE	Mean±SD	(best)	CE
CSTR	NMI	0.73±0.04	(0.65)	(0.76)	0.76±0.007	(0.77)	(0.77)	0.73±0.03	(0.76)	**(0.80)**
	ARI	0.75±0.10	(0.56)	(0.80)	0.80±0.007	(0.80)	(0.80)	0.77±0.04	(0.81)	**(0.83)**
CLASSIC4	NMI	0.59±0.002	(0.59)	(0.59)	0.60±0.001	(0.59)	(0.60)	0.71±0.02	(0.72)	**(0.74)**
	ARI	0.47±0.002	(0.47)	(0.47)	0.47±0.0009	(0.47)	(0.47)	0.65±0.06	(0.65)	**(0.72)**
RCV1	NMI	0.38±0.0002	(0.38)	(0.35)	0.38±0.0003	(0.38)	(0.35)	0.47±0.02	(0.47)	**(0.52)**
	ARI	0.18±0.0003	(0.18)	(0.26)	0.18±0.0004	(0.18)	(0.26)	0.42±0.02	(0.43)	**(0.46)**
NG5	NMI	0.72±0.02	(0.74)	(0.76)	0.72±0.02	(0.73)	(0.75)	0.80±0.03	(0.83)	**(0.86)**
	ARI	0.60±0.01	(0.61)	(0.60)	0.60±0.01	(0.60)	(0.64)	0.82±0.04	(0.85)	**(0.88)**
NG20	NMI	0.49±0.02	(0.51)	(0.50)	0.49±0.02	(0.51)	(0.50)	0.48±0.02	(0.50)	**(0.61)**
	ARI	0.30±0.02	(0.32)	(0.34)	0.30±0.02	(0.32)	(0.34)	0.34±0.02	(0.36)	**(0.49)**

Following the previous statement, we went ahead and computed a cluster ensemble (CE) for `NMF-F` and `NMF-KL` according to their best initialization strategy as well as for `S-Kmeans` due to its pertinence for initializing `NMF-F` and the method being widely known as relevant for document clustering. The results are reported in Table 2.4. It appears that the consensus obtained with the top 10 results of each method generally outperforms the best solution. This result is even stronger for `NMF-KL` where the ensemble clustering increases the NMI and ARI by respectively 11 and 13 points on NG20. Note that NG20 is the dataset where the average pairwise NMI and ARI between the 10 top partitions

are the lowest, meaning the most different (see Figure 2.7). Furthermore, it is interesting to note that these performances are obtained from solutions giving an average NMI and ARI smaller than the best solution itself.

2.2.3.4 Consensus multinomial

Following the cluster-based consensus approach which implies a similarity-based clustering algorithm, we decided to make use of a model-based clustering to go and try to obtain a better final partition than the one delivered by *cluster ensembles*. In [182], the authors have used the Multinomial mixture approach to propose a consensus function. In model-based clustering, it is assumed that the data are generated by a mixture of underlying probability distributions, where each component k of the mixture represents a cluster.

Let $\Lambda \in \mathbb{N}_0^{n \times r}$ be the data matrix of labels vectors from the top r solutions. Our data being categorical, we used a Multinomial Mixture Model (MMM) in order to partition the elements λ_i. Categorical data being a generalization of binary data ; assuming a perfect scenario where there is no partition with an empty cluster, a disjunctive matrix $\boldsymbol{M} \in \{0,1\}^{n \times rg}$ is usually used instead of Λ with value $m_{iq}^{(h)}$ where $h \in \{1, \ldots, g\}$ is a cluster label. Therefore, the data values $m_{iq}^{(h)}$ are assumed to be generated from a Multinomial distribution of parameter $\mathcal{M}(m_{iq}^{(h)}; \alpha_{kq}^{(h)})$ where $\alpha_{kq}^{(h)}$ is the probability that an element m_i in the group k takes the category h for the partition/variable λ_q. The density probability function of the model can be stated as :

$$f(\boldsymbol{M}; \boldsymbol{\Theta}) = \prod_{i=1}^{n} \sum_{k=1}^{g} \pi_k \prod_{q=1}^{r} \prod_{h=1}^{g} (\alpha_{kq}^{(h)})^{m_{iq}^{(h)}}, \tag{2.38}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha})$ are the parameters of the model with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ being the proportions and $\boldsymbol{\alpha}$ the vector of the components parameters.

The Rmixmod package [2] is used to achieve our analysis. We employ the default settings to compute the clustering, allowing the selection between 10 parsimonious models according to the Bayesian information Criterion (BIC) [183]. With CSTR, the model mainly selected is the one keeping the proportions π_k free with the model also independent from the variables (labels vectors), meaning $\mathcal{M}(m_{iq}^{(h)}; \alpha_k)$. CSTR is the dataset with the highest pairwise NMI and ARI therefore with the most similar best solutions. On CLASSIC4 and RCV1 where the pairwise NMI & ARI are a little bit lower,

2. https ://cran.r-project.org/web/packages/Rmixmod/Rmixmod.pdf

it is the model with free proportions and parameters $\boldsymbol{\alpha}$ depending on distinct components and labels vectors ($\mathcal{M}(m_{iq}^{(h)}; \alpha_{kq}^{(h)})$) which is mainly chosen. On NG5 where the best solutions are fairly similar (high pairwise NMI & ARI), it is the model depending on the components and the labels vectors which has been retained. However, the proportions here were kept equal. For NG20 where the best solutions were fairly distinct, the model selected is the one depending on the components and the variables. As previously, the proportions π_k are kept equal. Following the characteristics in Table 2.2, it is notable to see that the datasets where the proportions are kept equal are actually those with the more balanced real clusters proportions. The results of the obtained consensus are displayed in Table 2.5 which only retains prior results of `NMF-KL` top 10 solutions and CE consensus, as they were the best overall. Apart from CSTR, we can see that MMM does a better job at computing a better partition from the top 10 solutions than CE.

TABLE 2.5 – MMM consensus results over the 10 best solutions.

Datasets	Metrics	`NMF-KL (Random)`			
		Mean±SD	(best)	CE	MMM
CSTR	NMI	0.73±0.03	(0.76)	**(0.80)**	(0.77)
	ARI	0.77±0.04	(0.81)	**(0.83)**	(0.82)
CLASSIC4	NMI	0.71±0.02	(0.72)	(0.74)	**(0.77)**
	ARI	0.65±0.06	(0.65)	(0.72)	**(0.75)**
RCV1	NMI	0.47±0.02	(0.47)	**(0.52)**	**(0.52)**
	ARI	0.42±0.02	(0.43)	**(0.46)**	**(0.46)**
NG5	NMI	0.80±0.03	(0.83)	**(0.86)**	**(0.86)**
	ARI	0.82±0.04	(0.85)	(0.88)	**(0.89)**
NG20	NMI	0.48±0.02	(0.50)	(0.61)	**(0.63)**
	ARI	0.34±0.02	(0.36)	(0.49)	**(0.50)**

2.3 Conclusion

We have presented NMF and several possible extensions to narrow its objective toward a clustering point of view. We proposed to use the MU algorithm to solve the NMF problem and studied its solutions from a clustering perspective. By using *cluster ensembles*, we have proposed a simple method to avoid poor clustering results using the best local minimum and improved the overall clustering performances. From its gathering nature, this process should also alleviate the uncertainty based around the overall

quality of the final partition compared to other selection practices such as keeping an unique solution according to the best criterion. Furthermore, we have shown that it was possible to improve the consensus quality through the use of finite mixture models, allowing more powerful underlying settings than cluster-based consensus involving plain similarities or distances.

In the next chapter, we shall consider this approach along side our new NMF extensions and perhaps, investigate the use of *cluster ensembles* for other recent clustering algorithms [184, 185, 186, 187, 188].

Chapitre 3

Nonnegative Matrix Factorization with semantic leveraging

NMF and its variants have been successfully used for clustering text documents. However, in its original formulation, NMF do not explicitly account for the contextual dependencies between words. To remedy this limitation, we propose in this chapter two regularizations for the NMF objective considering respectively the Frobenius norm, and the generalized Kullback-Leibler divergence as cost functions. The first approach draws inspiration from neural word embedding and posits that words that frequently co-occur within the same context (e.g., sentence or document) are likely related to each other in some semantic aspect. We then propose to jointly factorize the document-word and word-word co-occurrence matrices. Due to to the low computational cost of its gradient, the Frobenius norm is set a the cost function and the decomposition of the latter matrix encourages frequently co-occurring words to have similar latent representations to reflect their relationships. Empirical results, on several real-world datasets, provide strong support for the benefits of our approach and illustrates improvement of the clustering performance of NMF. This approach is referred to as `SNMF` and presented in the first section of this chapter. Following the results obtained in the previous chapter, *cluster ensembles* and finite mixture models are also employed to enhance and validate the potential of a consensus approach for NMF regularized objectives.

The second approach aimed at leveraging subordinates semantic relations (such as hyponyms) using the Wasserstein [1] metric to obtain regularization embeddings. In the field of document clustering (or dictionary learning), the Wasserstein distance showed some advantages for measuring the approxima-

1. In this paper, we use "Wasserstein", "Earth Mover's", "Kantorovich–Rubinstein" interchangeably

tion of the original data. Further, It is able to capture redundant information, for instance synonyms in bag-of-words, which in practice cannot be retrieved using classical metrics. However, despite the use of smoothed approximation allowing faster computations, this distance suffers from its high computational cost and remains uneasy to handle with a substantial amount of data. To circumvent this issue, we propose a different scheme of NMF relying on the generalized Kullback-Leibler divergence for the term approximating the original data and a regularization term consisting in the approximation of the Wasserstein embeddings in order to leverage more semantic relations. With experiments on benchmark datasets, the results show that our proposal achieves good clustering and support for visualizing the clusters. We refer to this approach as $\texttt{WE} - \texttt{NMF}$ and present it in the second section of this chapter.

3.1 Improving NMF Clustering by Leveraging Contextual Relationships Among Words

3.1.1 Motivations

Words having a common meaning—synonyms—or more generally words that are about the same topic are not guaranteed to be mapped in the same direction in the latent space. This is simply due to the fact that words with similar meanings are not necessarily used exactly in the same documents. Consequently, similar embeddings are not guaranteed even for closely related documents using words with similar meanings. Hence, our intuition is that, if we are successful in capturing the semantic relationships among words in an NMF model we can expect document factors which are even better for clustering.

The research question is how to capture and leverage the relationships among words in an NMF model ? In this section, we draw inspiration from neural word embedding and rely on the distributional hypothesis [189], which states that words in similar contexts have similar meanings. The context is a modeling choice that could be data- or problem-specific. For instance, a document or a sentence is a context in which words co-occur. Note that other definitions of "contexts" are possible [190]. An early application of that hypothesis in Matrix Factorization is the *Hyperspace Analogue to Language* (HAL) [191] framework. It employs a word-word co-occurrence matrix whose entries encode the number of times each pair of words has occurred in the same context. Thus, following the distributional hypothesis, we assume that words which frequently co-occur in the same context are likely related to

each other in some semantic aspect. We then, propose a new NMF model which jointly decomposes the document-word and word-word co-occurrence matrices into two separate products that share one factor. The intuition behind the decomposition of the latter matrix is to make the representations of frequent co-occurring words closer to each other in the latent space so as to reflect the relationships among them. We further consider a non-linear transformation of the word co-occurrences, based on the Pointwise Mutual Information (PMI), for effectiveness and efficiency purposes.

In order to infer the factor matrices, we propose a scalable alternating optimization procedure based on a set of multiplicative update rules, similar to original NMF, which guarantees to decrease monotonically our objective function at each iteration, until convergence. We conduct extensive experiments to illustrate the benefits of our model and better characterize the circumstances in which it offers the most significant improvements. Our main finding is that, we can drastically improve the clustering performance of NMF by leveraging explicitly the contextual relationships among words [2].

3.1.2 Related Works

Below we try to provide a brief review of works that are most closely related to our contribution.

In order to leverage the relationships among words in NMF, we draw inspiration from neural word embedding. These approaches, seek continuous representations of words that reflect various linguistic regularities between them [192, 193, 194]. To achieve their objective, most neural word embedding methods rely on the distributional hypothesis of Harris [189]. For instance, the recently proposed skip-gram model with negative sampling aims to maximize the dot-product between the vectors of frequently occurring word-context pairs, and minimize it for random word-context pairs. For more details please refer to [194]. What makes these models particularly appealing is their ability to learn word vectors that are good at capturing meaningful semantic and syntactic regularities between words [195]. Similar to word embedding techniques, the model we propose relies on the distributional hypothesis to capture the semantic relationships between words in NMF. Our preliminary investigation of infusing NMF with contextual relationships among words has appeared recently as a short paper [196]. In the present manuscript, we delve in-depth into this idea and present several new theoretical and empirical results.

2. We use « contextual relationships » and « semantic relationships » interchangeably. The former relationships underlie the latter ones, and our objective is to rely on the words' context to capture the semantic relationships among them

3.1.3 Preliminaries

The word co-occurrence matrix is represented by $\boldsymbol{C} = (c_{jj'}) \in \mathbb{R}_+^{d \times d'}$, following the nomenclature in neural word embedding, row $j \in \mathcal{J}$ corresponds to word w_j, column $j' \in \mathcal{J}'$ denotes context word $w_{j'}$, and each entry $c_{jj'}$ denotes the number of times the word-context pair (w_j,$w_{j'}$) occurred in the same context (e.g., a sentence or a document). The word and context word vocabularies, $\mathcal{J}$ and $\mathcal{J}'$ might be different. The PMI is an information theoretic measure widely used to quantify the association between pairs of outcomes coming from discrete random variables. Formally, the PMI between word w_j and its context word $w_{j'}$ is given by

$$\text{PMI}(w_j, w_{j'}) = \log \frac{p(w_j, w_{j'})}{p(w_j)p(w_{j'})}. \tag{3.1}$$

Given the word co-occurrence matrix $\boldsymbol{C}$ defined above, the PMI between w_j and $w_{j'}$ can be empirically estimated as follows

$$\text{PMI}(w_j, w_{j'}) = \log \frac{c_{jj'} \times c_{..}}{c_{j.} \times c_{.j'}}, \tag{3.2}$$

where $c_{..} = \sum_{j=1}^{d} \sum_{j'=1}^{d'} c_{jj'}$, $c_{j.} = \sum_{j'=1}^{d'} c_{jj'}$ and $c_{.j'} = \sum_{j=1}^{d} c_{jj'}$.

The expected value of the PMI across all the possible events is the Mutual Information (MI) that is positive. A null PMI indicates that the events are independent, negative values of PMI indicate that those events occur less frequently than expected. Therefore a useful variation called Positive PMI (PPMI) is to set all negative PMI values to zero. This transformation has been shown to produce good semantic representations [197].

3.1.4 Method

3.1.4.1 Formulation

In this section, we describe our model, Semantic-NMF, which jointly performs NMF on the document-word matrix and word-word PPMI matrix, with shared word factors, to better capture and leverage the semantic relationships among words. Formally the objective function of Semantic-NMF, to be minimized, is given by

$$\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{Q}) = \underbrace{\mathcal{D}_1(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)}_{\text{NMF}} + \lambda \underbrace{\mathcal{D}_2(\boldsymbol{M}, \boldsymbol{W}\boldsymbol{Q}^\top)}_{\text{word embedding}}, \tag{3.3}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$, are cost functions for measuring the divergence between non-negative matrices, λ is a regularization parameter, and following the nomenclature in neural word embedding, we refer to $\boldsymbol{Q} \in \mathbb{R}_+^{d' \times g}$ as the context factor matrix. The above objective function can be viewed as regularizing the word factors in NMF beyond usual regularization schemes (e.g., L_2 norm). Note that, both terms in (3.3) infer low dimensional representations of words. In the NMF term, word factors encode how words are used in documents, while in the word embedding term, word representations encode word co-occurrence patterns. Semantic-NMF seeks to leverage both of the above information, simultaneously. Additionally, whilst $d' = d$ due to $\boldsymbol{M}$ defined as a word-word PPMI matrix, Semantic-NMF can easily accommodate the definition of $\boldsymbol{M}$ as a word embedding matrix where $d' \neq d$ (favorably $d' \leq d$). Figure 3.1 provides a graphical illustration of Semantic-NMF.

3.1.4.2 Inference

In this section, we shall investigate the case where both $\mathcal{D}_1$ and $\mathcal{D}_2$ are the square of the Frobenius norm, and derive an iterative optimization procedure to infer the latent factor matrices. In this case, (3.3) takes the following form :

$$\begin{aligned}
\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{Q}) =& \frac{1}{2}||\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top||_F^2 + \frac{\lambda}{2}||\boldsymbol{M} - \boldsymbol{W}\boldsymbol{Q}^\top||_F^2 \\
=& \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top)(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top)^\top\right) \\
&+ \frac{\lambda}{2}\operatorname{Tr}\left((\boldsymbol{M} - \boldsymbol{W}\boldsymbol{Q}^\top)(\boldsymbol{M} - \boldsymbol{W}\boldsymbol{Q}^\top)^\top\right) \\
=& \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{X}\boldsymbol{X}^\top - 2\boldsymbol{X}\boldsymbol{W}\boldsymbol{Z}^\top + \boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{Z}^\top\right) \\
&+ \frac{\lambda}{2}\operatorname{Tr}\left(\boldsymbol{M}\boldsymbol{M}^\top - 2\boldsymbol{M}\boldsymbol{Q}\boldsymbol{W}^\top + \boldsymbol{W}\boldsymbol{Q}^\top\boldsymbol{Q}\boldsymbol{W}^\top\right).
\end{aligned} \tag{3.4}$$

In the following, we derive a set of multiplicative update rules in order to minimize $\mathcal{F}$ under the constraints of positivity of $\boldsymbol{Z}$, $\boldsymbol{W}$ and $\boldsymbol{Q}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^{n \times g}$, $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$, $\boldsymbol{\gamma} \in \mathbb{R}^{d' \times g}$ be the Lagrange multipliers for the constraints, the Lagrange function $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{Q}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \mathcal{L}$ is given by

$$\mathcal{L} = \mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{Q}) + \operatorname{Tr}(\boldsymbol{\alpha}\boldsymbol{Z}^\top) + \operatorname{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top) + \operatorname{Tr}(\boldsymbol{\gamma}\boldsymbol{Q}^\top).$$

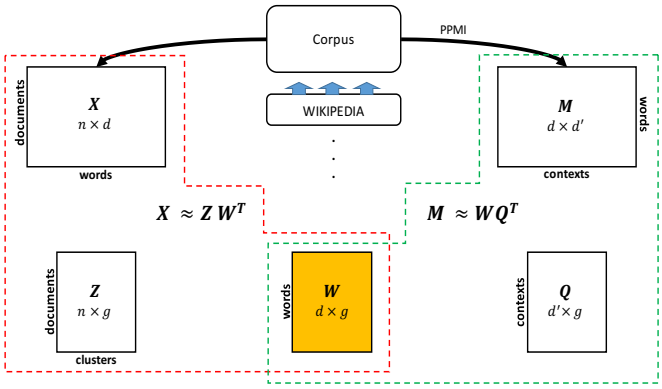


FIGURE 3.1 – Illustrative scheme of the proposed Semantic-NMF model. $\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{W}^\top$ and $\boldsymbol{M} \approx \boldsymbol{W}\boldsymbol{Q}^\top$

The derivatives of $\mathcal{L}$ with respect to $\boldsymbol{Z}$, $\boldsymbol{W}$ and $\boldsymbol{Q}$ are

$$\nabla_{\boldsymbol{Z}}\mathcal{L} = -\boldsymbol{X}\boldsymbol{W} + \boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{\alpha}, \tag{3.5a}$$

$$\nabla_{\boldsymbol{W}}\mathcal{L} = -(\boldsymbol{X}^\top\boldsymbol{Z} + \lambda\boldsymbol{M}\boldsymbol{Q}) + \boldsymbol{W}(\boldsymbol{Z}^\top\boldsymbol{Z} + \lambda\boldsymbol{Q}^\top\boldsymbol{Q}) + \boldsymbol{\beta}, \tag{3.5b}$$

$$\nabla_{\boldsymbol{Q}}\mathcal{L} = -\lambda\boldsymbol{M}^\top\boldsymbol{W} + \lambda\boldsymbol{Q}\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{\gamma}. \tag{3.5c}$$

Setting these gradients to zero and making use of the Kuhn-Tucker conditions

$$\begin{cases} \boldsymbol{\alpha} \odot \boldsymbol{Z} &= 0 \\ \boldsymbol{\beta} \odot \boldsymbol{W} &= 0 \\ \boldsymbol{\gamma} \odot \boldsymbol{Q} &= 0 \end{cases}$$

we obtain the following stationary equations :

$$-(\boldsymbol{X}\boldsymbol{W}) \odot \boldsymbol{Z} + (\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}) \odot \boldsymbol{Z} = 0,$$

$$-(\boldsymbol{X}^\top\boldsymbol{Z} + \lambda\boldsymbol{M}\boldsymbol{Q}) \odot \boldsymbol{W} + \boldsymbol{W}(\boldsymbol{Z}^\top\boldsymbol{Z} + \lambda\boldsymbol{Q}^\top\boldsymbol{Q}) \odot \boldsymbol{W} = 0,$$

$$-(\boldsymbol{M}^\top\boldsymbol{W}) \odot \boldsymbol{Q} + (\boldsymbol{Q}\boldsymbol{W}^\top\boldsymbol{W}) \odot \boldsymbol{Q} = 0.$$

Based on the above equations we derive the following multiplicative update rules

$$\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{X}\boldsymbol{W}}{\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W}}, \tag{3.6a}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{(\boldsymbol{X}^\top\boldsymbol{Z} + \lambda\boldsymbol{M}\boldsymbol{Q})}{\boldsymbol{W}(\boldsymbol{Z}^\top\boldsymbol{Z} + \lambda\boldsymbol{Q}^\top\boldsymbol{Q})}, \tag{3.6b}$$

$$\boldsymbol{Q} \leftarrow \boldsymbol{Q} \odot \frac{\boldsymbol{M}^\top\boldsymbol{W}}{\boldsymbol{Q}\boldsymbol{W}^\top\boldsymbol{W}}. \tag{3.6c}$$

These update rules are analogous to those of NMF [132]. The difference is in how we update the word factors in Semantic-NMF. In the latter, the update of $\boldsymbol{W}$ depends on two sources of data (i) the document-word matrix and (ii) the PPMI co-occurence matrix $\boldsymbol{M}$.

Theorem 1. *The objective function of Semantic-NMF is non-increasing under the update formulas (3.6a), (3.6b) and (3.6c).*

Proof. Equations (3.6a) and (3.6c) are similar to those of NMF [132], therefore based on the proof of [132] the objective function of Semantic-NMF is non-increasing under these two equations. Hence, we only need to demonstrate that $\mathcal{F}$ is non-increasing under the update rule (3.6b), given $\boldsymbol{Z}$ and $\boldsymbol{Q}$. To this end, we follow a similar approach to the one described in [132], which is inspired by the Expectation-Maximization (EM) algorithm [90] and consists in using an auxiliary function.

Definition. $\mathcal{G}(w, w')$ *is an auxiliary function for* $\mathcal{F}(w)$ *if the following conditions are satisfied* $\mathcal{G}(w, w') \geq \mathcal{F}(w)$ *and* $\mathcal{G}(w, w) = \mathcal{F}(w)$.

A key point to the auxiliary function is described by the following lemma.

Lemma 1. If $\mathcal{G}$ *is an auxiliary function for* $\mathcal{F}$*, then* $\mathcal{F}$ *is non-increasing under the update*

$$w^{(t+1)} = \arg\min_{w} \mathcal{G}(w, w^{(t)}). \tag{3.7}$$

Proof.
$\mathcal{F}(w^{(t+1)}) \leq \mathcal{G}(w^{(t+1)}, w^{(t)}) \leq \mathcal{G}(w^{(t)}, w^{(t)}) = \mathcal{F}(w^{(t)}).\square$

Now we will make use of an appropriate auxiliary function to demonstrate that our objective function $\mathcal{F}$ is non-increasing under the update rule (3.6b). Let w_{jk} denote any element in $\boldsymbol{W}$, and let $\tilde{\mathcal{F}}(w_{jk})$ denote the part of $\mathcal{F}$ containing w_{jk}. As the update (3.6b) is element-wise, it is sufficient to show that $\tilde{\mathcal{F}}$ is non-increasing under the update of w_{jk} based on equation (3.6b). The first and second partial derivatives of $\tilde{\mathcal{F}}$ noted $\tilde{\mathcal{F}}'$, $\tilde{\mathcal{F}}''$ are respectively given by

$$\tilde{\mathcal{F}}'(w_{jk}) = \left(-\boldsymbol{X}^\top \boldsymbol{Z} - \lambda \boldsymbol{M}\boldsymbol{Q} + \boldsymbol{W}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q})\right)_{jk},$$

$$\tilde{\mathcal{F}}''(w_{jk}) = \left(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q}\right)_{kk}.$$

The following lemma yields an auxiliary function for $\tilde{\mathcal{F}}$.

Lemma 2. The function $\mathcal{G}$ defined as follows

$$\begin{aligned}\mathcal{G}(w_{jk}, w_{jk}^{(t)}) = \tilde{\mathcal{F}}(w_{jk}^{(t)}) + \tilde{\mathcal{F}}'(w_{jk}^{(t)})(w_{jk} - w_{jk}^{(t)}) \\ + \frac{\left(\boldsymbol{W}\left(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q}\right)\right)_{jk}}{2w_{jk}^{(t)}}(w_{jk} - w_{jk}^{(t)})^2\end{aligned} \tag{3.8}$$

is an auxiliary function for $\tilde{\mathcal{F}}$.

Proof. Based on Lemma 2 it straightforward to verify that $\mathcal{G}(w_{jk}, w_{jk}) = \tilde{\mathcal{F}}(w_{jk})$. We will now show that $\mathcal{G}(w_{jk}, w_{jk}^{(t)}) \geq \tilde{\mathcal{F}}(w_{jk})$, by making use of the second order Taylor expansion of $\tilde{\mathcal{F}}$ about $w_{jk}^{(t)}$ given by

$$\begin{aligned}\tilde{\mathcal{F}}(w_{jk}) = \tilde{\mathcal{F}}(w_{jk}^{(t)}) + \tilde{\mathcal{F}}'(w_{jk}^{(t)})(w_{jk} - w_{jk}^{(t)}) \\ + \frac{\left(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q}\right)_{kk}}{2}(w_{jk} - w_{jk}^{(t)})^2.\end{aligned} \tag{3.9}$$

Since

$$\left(\boldsymbol{W}\boldsymbol{Z}^\top \boldsymbol{Z}\right)_{jk} = \sum_{k'=1}^{g} w_{jk'}^{(t)} \left(\boldsymbol{Z}^\top \boldsymbol{Z}\right)_{k'k} \geq w_{jk}^{(t)} \left(\boldsymbol{Z}^\top \boldsymbol{Z}\right)_{kk}$$

and similarly

$$\left(\boldsymbol{W}\boldsymbol{Q}^\top \boldsymbol{Q}\right)_{jk} \geq w_{jk}^{(t)} \left(\boldsymbol{Q}^\top \boldsymbol{Q}\right)_{kk},$$

we have $\frac{\left(\boldsymbol{W}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q})\right)_{jk}}{w_{jk}^{(t)}} \geq \left(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q}\right)_{kk}$. Thereby, from (3.8) and (3.9), $\mathcal{G}(w_{jk}, w_{jk}^{(t)}) \geq \tilde{\mathcal{F}}(w_{jk})$ holds. □

Thus, to prove Theorem 1 it is sufficient to show that equation (3.6b) for w_{jk} satisfies *Lemma 1* where the auxiliary function $\mathcal{G}$ is given by *Lemma 2*. Substituting equation (3.8) to $\mathcal{G}(w_{jk}, w_{jk}^{(t)})$ in *Lemma 1* leads to solve $\frac{\partial \mathcal{G}(w_{jk}, w_{jk}^{(t)})}{\partial w_{jk}} = 0$ or,

$$\tilde{\mathcal{F}}'(w_{jk}^{(t)}) + \frac{\left(\boldsymbol{W}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q})\right)_{jk}}{2w_{jk}^{(t)}}(2w_{jk} - 2w_{jk}^{(t)}) = 0.$$

Then $w_{jk}^{(t+1)} = \arg\min_w \mathcal{G}(w_{jk}, w_{jk}^{(t)})$ leads to

$$\begin{aligned} w_{jk}^{(t+1)} &= -w_{jk}^{(t)} \frac{\tilde{\mathcal{F}}'(w_{jk}^{(t)})}{\left(\boldsymbol{W}\left(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q}\right)\right)_{jk}} + w_{jk}^{(t)} \\ &= w_{jk}^{(t)} \frac{\left(\boldsymbol{X}^\top \boldsymbol{Z} + \lambda \boldsymbol{M}\boldsymbol{Q}\right)_{jk}}{\left(\boldsymbol{W}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q})\right)_{jk}}. \end{aligned}$$

It follows from the latter result and *Lemma 1* that $\tilde{\mathcal{F}}$ is non-increasing under the update of w_{jk} in equation (3.6b), $\forall j, k$. Given that (3.6b) is element-wise, the objective function of Semantic-NMF is non-increasing under the update rule (3.6b). ■

Thereby, based on Theorem 1, the fact that (3.6a), (3.6b) and (3.6c) satisfy the KKT conditions at convergence and $\mathcal{F}$ is bounded from below by 0, iteratively alternating the application of (3.6a), (3.6b) and (3.6c) will monotonically decrease criterion (3.4) and converge to a locally optimal solution. Our optimization procedure is depicted in Algorithm 13.

Algorithm 13 Semantic-NMF (`SNMF`).

Input : $\boldsymbol{X}$, $\boldsymbol{M}$, λ and g the dimension of the latent factors.
Output : $\boldsymbol{Z}$, $\boldsymbol{W}$ and $\boldsymbol{Q}$.
1. Initialization : $\boldsymbol{Z} \leftarrow \boldsymbol{Z}^{(0)}$; $\boldsymbol{W} \leftarrow \boldsymbol{W}^{(0)}$ and $\boldsymbol{Q} \leftarrow \boldsymbol{Q}^{(0)}$;
repeat
  2. $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \odot \frac{\boldsymbol{X}\boldsymbol{W}}{\boldsymbol{Z}\boldsymbol{W}^\top \boldsymbol{W}}$;
  3. $\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{(\boldsymbol{X}^\top \boldsymbol{Z} + \lambda \boldsymbol{M}\boldsymbol{Q})}{\boldsymbol{W}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{Q}^\top \boldsymbol{Q})}$;
  4. $\boldsymbol{Q} \leftarrow \boldsymbol{Q} \odot \frac{\boldsymbol{M}^\top \boldsymbol{W}}{\boldsymbol{Q}\boldsymbol{W}^\top \boldsymbol{W}}$;
until convergence
5. Normalize $\boldsymbol{Z}$ so as it has unit-length column vectors.

3.1.4.3 Computational Complexity Analysis

The following Proposition shows that the computational complexity of the `SNMF` algorithm scales linearly with the number of non-zero entries in the document-word and PPMI matrices. In practice $\boldsymbol{X}$ and $\boldsymbol{M}$ are very sparse, i.e., $nz_X \ll n \times d$ and $nz_M \ll d \times d$. Furthermore, multiplicative update rules (3.6a), (3.6a) and (3.6c) are parallelizable across documents and words, thereby Semantic-NMF can easily scale to large datasets.

Proposition 1. *Let nz_X and nz_M denote respectively the number of non-zero entries in $\boldsymbol{X}$ and $\boldsymbol{M}$, and let it be the number of iterations. The computational complexity of Semantic-NMF is given in* $O(it \cdot g \cdot (nz_X + nz_M) + it \cdot g^2 \cdot (n + d))$.

Proof. The computational bottleneck of `SNMF` is with the multiplicative update formulas (3.6a), (3.6b) and (3.6c). Equations (3.6a) and (3.6c) are similar to those of NMF, and their respective complexities are $O(nz_X \cdot g + (n + d) \cdot g^2)$ and $O(nz_M \cdot g + d \cdot g^2)$. The number of operation in (3.6b), including multiplications, additions and divisions, is $g(2nz_X + 3nz_M + 3d + g(4d + 2n + 1))$, where we used $d' = d$. The complexity of (3.6b) is thereby given in $O(g \cdot (nz_X + nz_M) + (n + d) \cdot g^2)$. Therefore, the total computational complexity of Semantic-NMF is

$$O(it \cdot g \cdot (nz_X + nz_M) + it \cdot g^2 \cdot (n + d)).\blacksquare$$

3.1.5 Experimental study

Our objective is to investigate the effect of the contextual relationships between words on NMF models. To this end, we conduct extensive experiments in which we benchmark our model, Semantic-NMF (`SNMF`), against several state-of-the-art algorithms (including NMF models and clustering algorithms) on several real-world datasets. Furthermore, we also challenge the choice of the PPMI for $\boldsymbol{M}$ by considering another transformation arising from the word-word co-occurrence matrix, namely the Global Vectors for Word Representation (GloVe) [198]. Note that, the Hellinger PCA (HPCA) [199] was also tested but did not demonstrated good enough performances to be considered in our proposal.

3.1.5.1 Datasets

We use six popular benchmark datasets, described in Table 3.1, namely **CSTR** [200], **CLASSIC4** [3], **RCV1** containing the four largest classes of the Reuters corpus [4], the **SPORTS** dataset (from the CLUTO toolkit [201]) containing documents relating to seven different sports, the 20-newsgroups dataset **NG20**[3], and the **NG5** dataset consisting of five classes [5] of NG20. These datasets are carefully selected so as to represent various particular challenging situations : different numbers of clusters, different sizes, different degrees of cluster overlap and different degrees of cluster balance (the *Balance*

3. http ://www.dataminingresearch.com/
4. http ://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html
5. rec.sport.baseball, soc.religion.christian, talk.politics.mideast, sci.electronics and sci.med

coefficient being the ratio of the minimum cluster size to the maximum cluster size). For each dataset, we apply the TF-IDF weighting scheme and normalize each document to unit L_2 norm so as to remove the biases induced by the length of documents.

TABLE 3.1 – Description of Datasets, # denotes the cardinality.

Datasets	Characteristics				
	#Documents	#Words	#Clusters	nz_X (%)	Balance
CSTR	475	1000	4	3.40	0.399
CLASSIC4	7095	5896	4	0.59	0.323
RCV1	6387	16921	4	0.25	0.080
NG5	4905	10167	5	0.92	0.943
SPORTS	8580	14870	7	0.86	0.0358
NG20	18846	14390	20	0.59	0.628

3.1.5.2 Competing methods

Without the word embedding term in (3.4), when $\lambda = 0$, the proposed `SNMF` degenerates to the original NMF (`NMF`) [202]. Hence, we can achieve our objective of studying the effects of the word relationships on NMF, most effectively by comparing `SNMF` to `NMF`. Moreover, in order to show that leveraging the contextual relationships among words in NMF is effective for text document clustering, we also consider three strong NMF variants, namely orthogonal NMF (`ONMF`) [158], Projective NMF (`PNMF`) [162] and graph regularized NMF (`GNMF`) [163]. All the above models have been found to perform very well and better than several other approaches in terms of text document clustering. A Deep-Learning algorithm, namely Deep Clustering Network (`DCN`) [203] is also considered in our comparison ; it outperforms several clustering (k-means, Spectral Clustering), NMF based method such as (`LCCF`) [204] and Deep Learning algorithms (e.g. `SAE` [205]). The Spherical k-means algorithm `Skmeans` [39], which to this day, remains popular for the task document clustering is also included rather than k-means that is not suitable for sparse data.

3.1.5.3 Evaluation metrics

We retain two widely used measures to assess the quality of clustering, namely the Normalized Mutual Information (NMI) [119] and the Adjusted Rand Index (ARI) [122]. Intuitively, NMI quantifies how much the estimated clustering is informative about the true clustering, while the ARI measures the degree of agreement between an estimated clustering and a reference clustering ; both NMI and

ARI are equal to 1 if the resulting clustering is identical to the true one.

3.1.5.4 Settings

For each dataset, g is the true number of clusters. To produce a fair comparison, the same initialization (namely Skmeans) was used across the NMF-like algorithms. Similar settings to the ones used to in [198] are employed for producing the GloVe embeddings ; note that any other type of *word-embedding* can be used for the matrix $\boldsymbol{M}$. Therefore, the GloVe embeddings dimension (in our case d') was set to 100, x_{max} to 100, α to 3/4. A stochastic gradient descent algorithm with a learning rate of 0.15 was used to train the model. Subsequently, all negative entries in the GloVe embeddings are set to zero. In the following, this transformation is referred to as PGLOVE. The setting of the regularization parameter λ is achieved empirically and established w.r.t. the PPMI and PGLOVE matrices.

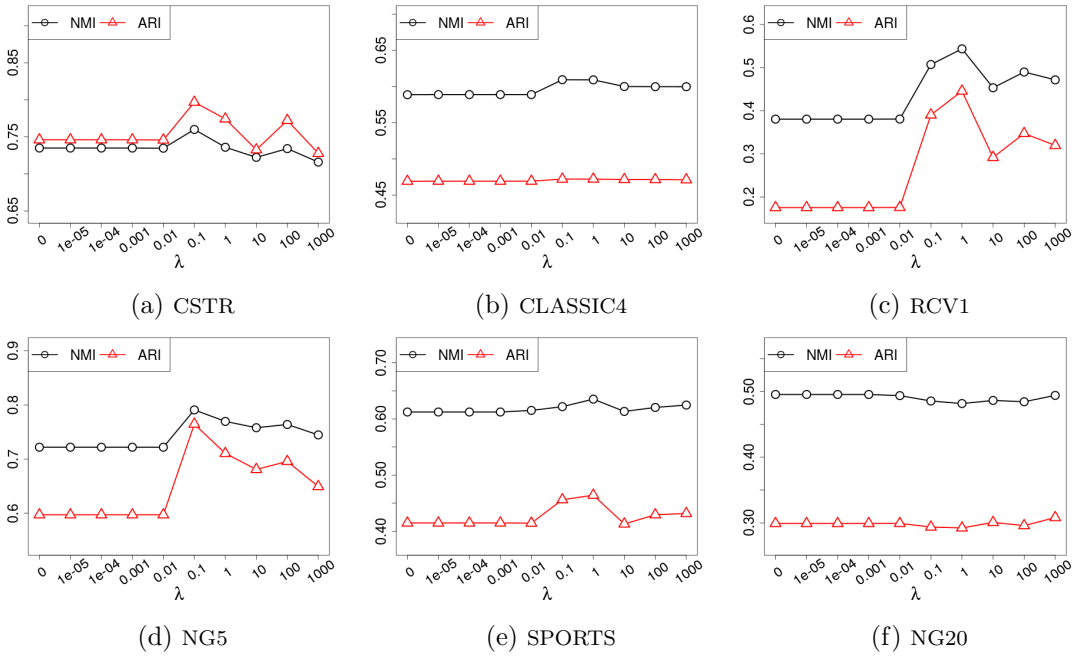
3.1.5.5 Empirical results

Below we comment on the results of our experiments and answer several questions related to our proposal.

What is the impact of the regularization parameter on the performances of SNMF ?

Figure 3.2 and 3.3 display the behaviors of `SNMF` w.r.t. the PGLOVE and PPMI matrices respectively. The results are shown in terms of NMI and ARI scores for several values of λ going from 0 to 10^3. In the case PGLOVE (see Figure 3.2), the variations of the NMI and ARI scores are unfortunately inconsistent across the range of λ values (see CSTR, RCV1, NG5, SPORTS) making the setting of λ quite difficult and unreliable. However, a good trade off would be $\lambda = 0.1$. On the other hand, using the PPMI (see Figure 3.2), the variations of the NMI and ARI scores are consistent and linear once a jump is observed. In this case, setting λ is much trivial and reliable and we recommend to set λ to 0.1 since we observe good performance scores even for higher values of λ on all the datasets. For these reasons, using the PPMI appears as safer alternative.

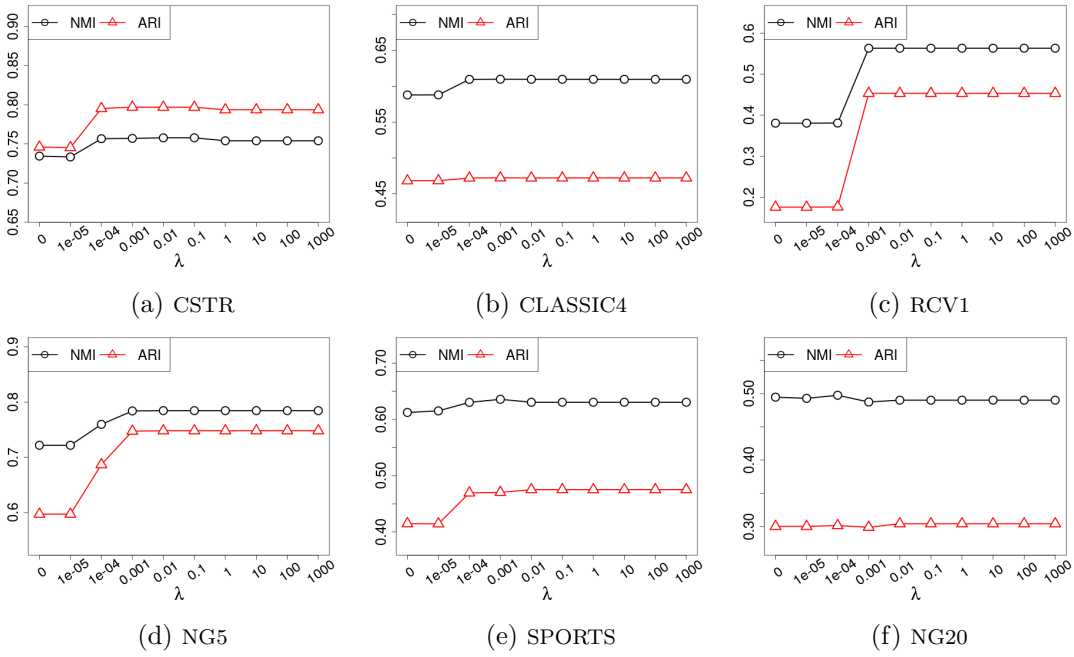
Table 3.2, summarizes the results of the different methods in terms of NMI and ARI, over all datasets. All the scores are averages considering the 10 best solutions (in terms of criterion) among a set of fifty different trials. As this table clearly shows, both versions of our model `SNMF` outperform the other competing methods by an important margin, in most cases. Recalling that `SNMF` corresponds to



(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) SPORTS (f) NG20

FIGURE 3.2 – Impact of the regularization parameter λ (PGLOVE).



(a) CSTR (b) CLASSIC4 (c) RCV1

(d) NG5 (e) SPORTS (f) NG20

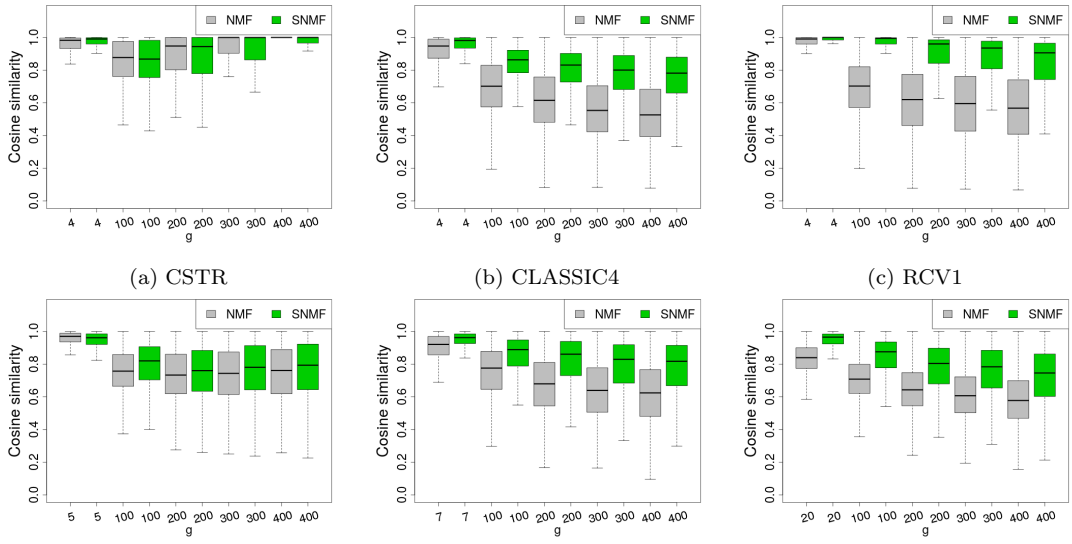
FIGURE 3.3 – Impact of the regularization parameter λ (PPMI).

NMF with an extra term encoding word co-occurrences. We can therefore attribute the improvement of SNMF upon the performance of NMF to the additional factorization of the PGLOVE or PPMI matrix.

In addition, between our two versions (PGLOVE, PPMI), using the PPMI appears to offer better performance overall and will be the version considered in the rest of the paper.

TABLE 3.2 – Mean ± SD of NMI and ARI over different datasets.

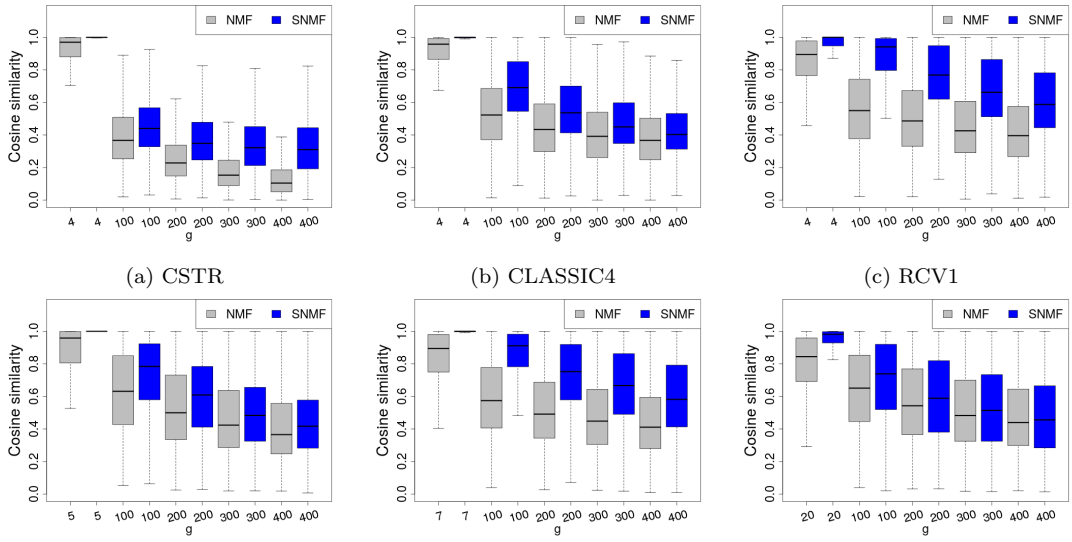
Datasets	Metrics	`Skmeans`	`NMF`	`ONMF`	`PNMF`	`GNMF`	`DCN`	`SNMF` (*PGLOVE*)	`SNMF` (*PPMI*)
CSTR	NMI	0.76±0.00	0.73±0.04	0.65±0.00	0.72±0.04	0.69±0.00	0.63±0.024	0.76±0.00	0.76±0.01
	ARI	0.80±0.00	0.75±0.10	0.60±0.03	0.73±0.09	0.75±0.02	0.53±0.03	0.80±0.00	0.80±0.01
CLASSIC4	NMI	0.60±0.00	0.59±0.00	0.49±0.02	0.51±0.00	0.62±0.00	0.57±0.01	0.61±0.02	0.61±0.03
	ARI	0.47±0.00	0.47±0.00	0.41±0.01	0.42±0.00	0.45±0.00	0.42±0.01	0.47±0.00	0.47±0.00
RCV1	NMI	0.38±0.00	0.38±0.00	0.35±0.00	0.36± 0.00	0.34±0.00	0.34±0.00	0.51±0.08	0.56±0.00
	ARI	0.18±0.00	0.18±0.00	0.14±0.00	0.16±0.00	0.12±0.00	0.12±0.00	0.39±0.15	0.45±0.00
NG5	NMI	0.72±0.02	0.72±0.02	0.52±0.01	0.69± 0.00	0.58±0.04	0.62±0.02	0.79±0.00	0.78±0.00
	ARI	0.60±0.01	0.60±0.01	0.29±0.00	0.54±0.00	0.50±0.07	0.47±0.02	0.76±0.00	0.75±0.01
SPORTS	NMI	0.62±0.02	0.61±0.03	0.55±0.02	0.56±0.00	0.55±0.00	0.59±0.01	0.62±0.05	0.63±0.04
	ARI	0.40±0.04	0.41±0.04	0.28±0.01	0.28±0.00	0.28±0.00	0.37±0.03	0.46±0.07	0.48±0.05
NG20	NMI	0.49±0.02	0.49±0.02	0.38±0.01	0.43±0.03	0.00±0.00	0.43±0.01	0.49±0.02	0.49±0.02
	ARI	0.30±0.02	0.30±0.02	0.20±0.00	0.22±0.02	0.00±0.00	0.17±0.01	0.29±0.01	0.33±0.03



(a) CSTR (b) CLASSIC4 (c) RCV1

(e) NG5 (f) SPORTS (f) NG20

FIGURE 3.4 – Distribution of cosine similarities between pairs of documents belonging to the same class, computed using the documents' embeddings obtained by `NMF` and `SNMF`. The documents of the same class tend to have more similar embeddings under `SNMF` than `NMF`.



(a) CSTR (b) CLASSIC4 (c) RCV1

(e) NG5 (f) SPORTS (f) NG20

FIGURE 3.5 – Distribution of cosine similarities between the top 30 words characterizing each document class, computed using the words' embeddings obtained by NMF and SNMF. The top words of the same class tend to have more similar embeddings under SNMF than NMF.

To gain further insights into the performances of SNMF and characterize the circumstances in which it provides the most significant improvements, we investigate several research questions below.

What happens with document embeddings ? Figure 3.4 shows the distribution of the cosine similarities between pairs of documents belonging to the same « true » class, computed using the document embeddings produced by NMF (grey boxplots) and SNMF (green boxplots). We observe that documents from the same class (topic) tend to have more similar embeddings under SNMF than NMF. This provides empirical evidence that accounting for the semantic relationships among words yields document factors that encode the clustering structure even better.

Is SNMF actually capturing the semantic relationships between words ? Based on the document-word matrix, we select the top thirty words of each true class. In Figure 3.5, we report the distribution of the cosine similarities between pairs of top words of the same class, computed using the word vectors inferred by NMF (grey boxplots) and SNMF (blue boxplots). Because the cosine similarity is likely to be high between low dimensional vectors (e.g. $g = 4$), we vary g from the real number of clusters to 400 for each dataset. As this figure shows clearly, the top words of each class have more similar embeddings under SNMF than NMF. This confirms that SNMF does a better job than

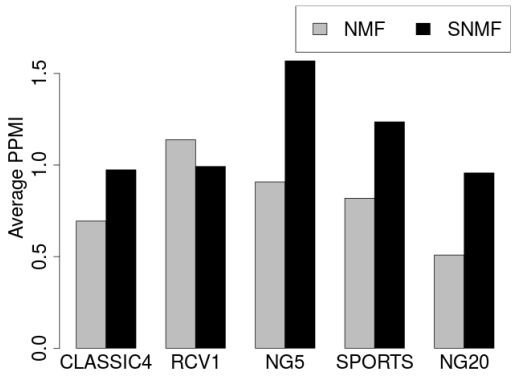


FIGURE 3.6 – Cluster interpretability : Average PMI score. Semantic-NMF leads more interpretable document clusters than NMF.

NMF in capturing semantics, by making the representations of words which are about the same topic (class) closer to each other in the latent space.

We also investigate the effect of the contextual relationships between words by comparing SNMF with NMF in terms of cluster interpretability. To human subjects, interpretability is closely related to coherence [206], i.e., how much the top words of each cluster are « associated » with each other. For each cluster k, we select its top 30 words based on the kth column of $\boldsymbol{W}$. We use the PMI, which is highly correlated with human judgments [207, 208], to measure the degree of association between top word pairs. For each cluster we average the PMI's among its top words, and for a model we average PMI across clusters. Because SNMF already exploits the PMI estimated from word co-occurrences in each dataset, we propose to use an external corpus to estimate the PMI in this experiment. Following Newman et al. [207], we use the whole English WIKIPEDIA corpus, that consists of approximately 4 millions of documents and 2 billions of words. Hence, $p(w_j)$ is the probability that word w_j occurs in WIKIPEDIA, and $p(w_j, w_{j'})$ is the probability that words w_j and $w_{j'}$ co-occur in a 5-word window in any WIKIPEDIA document.

Figure 3.6 shows the average PMI obtained by SNMF and NMF, over the different datasets ; it is clear that SNMF successes in capturing more semantics and inferring more interpretable clusters than NMF.

3.1.5.6 Cluster Ensembles

Throughout our experiments, Skmeans has proved to be a good initialization for gaining better NMF solutions with text data. However, we noticed that random starting values could sometimes lead to better solutions. Table 3.3 reports results of `SNMF` initialized with Skmeans and randomly. We can see that with RCV1, `SNMF (Random)` provides better partitions than `SNMF (Skmeans)`. While this improvement only appears with one dataset (other encountering losses, see CLASSIC4 and NG5), we tried to benefit from that infrequent/inconsistent behavior by using the `SNMF (Random)` solutions along side those obtained with a Skmeans initialization. Furthermore, in unsupervised learning, selecting an unique partition among the set of trials has also been a reluctant problem which to this day remains unclearly addressed. As with NMF, the objective function of Semantic-NMF is not defined as a clustering problem, therefore, it often happens that the selection of the best run (criterion-wise) among several does not account for getting the best clustering. However the best clustering could be among a set of lead solutions (for instance the 10 first ones). In other words, a consensus approach similar to the one introduced in the previous chapter will also help us to overcome this issue. Therefore, in the following, we evaluate the performance gain for `SNMF` utilizing *cluster ensembles* (CE) and the consensus obtained from the Multinomial Mixture Model (MMM).

Consensus results

Following the previous statements, we believe that using `SNMF (Random)` solutions could potentially improve the quality of the final partition. While they look unattractive compared to those of `SNMF (Skmeans)` due to their lower performance (see Table 3.3 where overall, SNMF (Random) appears to be a bad initialization strategy except for RCV1), these solutions still lead to minima which in an unsupervised situation, could benefit to other groups of individuals. More specifically, clusters could be different to the ones captured by `SNMF (Skmeans)` and therefore might bring another source of information to get closer to the actual partition. Our proposition referred to as `SNMF (Skmeans & Random)` consists in retrieving the 5 top SNMF solutions given by each initialization strategy (Skmeans and Random) and performing a consensus using the ensemble methods defined earlier. For comparison, we also provide a consensus for `SNMF (Skmeans)` and `SNMF (Random)` individually. Table 3.3 also reports the average performances of the mix of solutions of SNMF (Skmeans & Random). Consensus

results obtained with CE and MMM for each strategy are also available.

Figure 2.7 displays the pairwise NMI and ARI between the top partitions of each strategy : SNMF (Skmeans) denoted "SNMF Sk", SNMF (Random) denoted "SNMF Ra" and SNMF (Skmeans& Random) denoted "SNMF Sk & Ra". This allow us to assess how similar/related the respective partitions of each strategy are among each other. For instance SNMF Sk & Ra will translate how different SNMF (Random) solutions are from SNMF (Skmeans), while SNMF Sk relates how different SNMF (Skmeans solutions) are between each other. The closer we are to 1, the less diversity there is in the set of partitions.

Through our experiments, one can wonder what strategy should we use to improve clustering performance ? As we are in an unsupervised context, this question is difficult but through our obtained results we can nevertheless make some useful recommendations for the user.

1. First, it is clear that the MMM approach is undoubtedly superior to the CE approach [119] (see Table 3.3).

2. Between the two approaches Skmeans and Random, the former seems more often better than the latter. This can be due to the diversity it offers ; see for example SPORTS and NG20.

3. In the absence of diversity, the MMM approach does not bring improvement whatever the strategy used (Skmeans or Random). In this case combining them (Skmeans & Random) can even degrade the result as is the case with RCV1. Otherwise, with a great diversity of the two strategies one can expect an improvement ; this is the case of NG20.

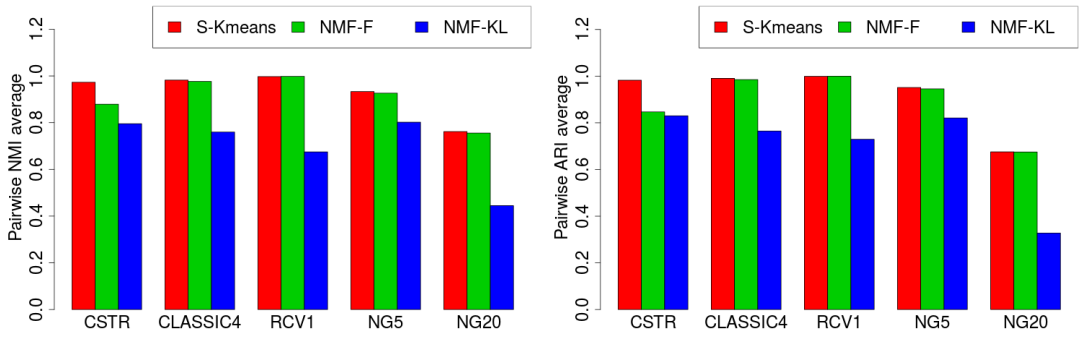


FIGURE 3.7 – Pairwise NMI & ARI averages between the top 10 solutions.

TABLE 3.3 – Mean ± SD of NMI and ARI & consensus over different datasets using CE and the Multinomial Mixture Model (MMM).

Datasets	Metrics	`SNMF (Skmeans)`			`SNMF (Random)`			`SNMF (Skmeans & Random)`		
		Mean±SD	CE	MMM	Mean±SD	CE	MMM	Mean±SD	CE	MMM
CSTR	NMI	0.76±0.01	(0.76)	(0.76)	0.75±0.00	(0.75)	(0.75)	0.75±0.00	(0.75)	(**0.77**)
	ARI	0.80±0.01	(0.80)	(0.80)	0.80±0.00	(0.80)	(0.80)	0.80±0.00	(0.80)	(**0.81**)
CLASSIC4	NMI	0.61±0.03	(0.60)	(0.60)	0.54±0.00	(0.49)	(0.54)	0.58±0.05	(0.57)	(**0.65**)
	ARI	0.47±0.00	(0.47)	(0.47)	0.38±0.00	(0.31)	(0.38)	0.48±0.05	(0.40)	(**0.47**)
RCV1	NMI	0.56±0.00	(0.56)	(0.56)	0.61±0.00	(0.51)	(**0.61**)	0.59±0.03	(0.51)	(0.52)
	ARI	0.45±0.00	(0.45)	(0.45)	0.63±0.00	(0.38)	(**0.63**)	0.54±0.04	(0.45)	(0.45)
NG5	NMI	0.78±0.00	(**0.78**)	(**0.78**)	0.67±0.00	(0.67)	(0.67)	0.73±0.06	(0.67)	(0.77)
	ARI	0.75±0.01	(0.75)	(0.74)	0.64±0.00	(0.64)	(0.64)	0.69±0.06	(0.60)	(**0.79**)
SPORTS	NMI	0.63±0.04	(0.63)	(**0.66**)	0.43±0.00	(0.43)	(0.43)	0.54±0.12	(0.53)	(0.57)
	ARI	0.48±0.05	(0.48)	(**0.54**)	0.32±0.00	(0.32)	(0.32)	0.41±0.10	(0.40)	(0.46)
NG20	NMI	0.49±0.02	(0.50)	(0.50)	0.47±0.01	(0.47)	(0.47)	0.48±0.02	(0.50)	(**0.52**)
	ARI	0.33±0.03	(0.33)	(0.30)	0.32±0.02	(0.33)	(0.33)	0.32±0.02	(0.34)	(**0.37**)

3.1.6 Discussion

In this section, we discuss some directions that we have already investigated since we developed Semantic-NMF. We also discuss some weaknesses and possible improvements of Semantic-NMF.

3.1.6.1 The orthogonality constraint

The orthogonality constraint on $\boldsymbol{Z}$ is almost always adopted for the clustering task [157, 158]. With this constraint NMF is equivalent to k-means clustering, and several work empirically demonstrated that such constrain improves the clustering performance of NMF, in most situations. In our case, we found that the orthogonality constraint on $\boldsymbol{Z}$ has only a slight impact on the performances of Semantic-NMF. Since this constraint adds a little computational overhead, we have chosen not to consider it for efficiency purposes. Note that, introducing the orthogonality constraint into Semantic-NMF is trivial as we only need to replace the update rule of $\boldsymbol{Z}$ (7a) by the one of Orthogonal NMF [157, 158].

3.1.6.2 Regularizing document factors using document-document co-occurrences

A natural extension of Semantic-NMF is to regularize the document factors using the document-document co-occurrence information. While such an extension is expected to yield further improvements, our first results show that in some cases adding this regularization declines the clustering performance of Semantic-NMF. We believe that this is might be due to the fact that even the most closely related documents do not necessarily use exactly the same words. We are currently performing further investigations and try to figure out what is causing this issue.

3.1.6.3 Weaknesses and possible improvements

Although we have shown that Semantic-NMF improves the performances of NMF models by a noticeable amount, Semantic-NMF has two potential weaknesses : (i) as in most NMF models, the dimensionality, g, of the latent space is the same for both documents and words. For the clustering task, g also denotes the number of clusters. When the latter is small (< 10), this may not be enough to learn high quality word representations that capture finer linguistic regularities and patterns between words. A better alternative, is to make the dimensionality of the word embeddings independent from the number of clusters. This is possible using Non-Negative Matrix Tri-factorization [157]. (ii) In some situations, when the PPMI matrix, $\boldsymbol{M}$, is defined deterministically from the local corpus of each dataset—as this is the case in this paper—, Semantic-NMF does not have a clear generative interpretation, which could limit the scope of its use. We can overcome this weakness by using a huge external corpora such as `WIKIPEDIA` and `GOOGLE` to build the PPMI matrix. In this case, not only Semantic-NMF has a clear generative interpretation and can be embedded in a well defined probabilistic model [209], but also the PPMI matrix encodes richer and more accurate semantic regularities between words. Leveraging a huge external corpora, such as the aforementioned ones, so as to preserve semantics in NMF, constitutes our main focus for a future extension of Semantic-NMF.

3.2 Wasserstein Embeddings for Nonnegative Matrix Factorization

3.2.1 Motivations

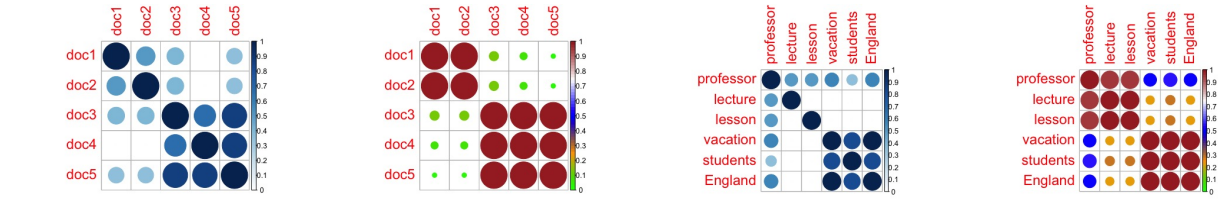
Despite all the notable efforts highlighting the potential of NMF for document clustering [135], these approaches still exhibit some limitations, namely they do not explicitly account for the semantic relationships between words as taken into account, for instance, by integrating a word embedding model into NMF [196, 187, 188]. Therefore, words having a common *meaning, synonyms* or more generally words that are about the same topic are not guaranteed to be mapped in the same direction within the lower dimensional space produces by NMF. This is simply due to the fact that words with similar meanings are not exactly used in similar documents (note that this issue reminds the foundation of LSA). Consequently, the document embeddings resulting from the approximation are also not guaranteed to share all potential similarities when the documents are actually from the same topic. We illustrate our idea in the following example : Taking two groups of documents *group*1={"The professor is doing a lecture" (doc1), "The professor is giving a lesson" (doc2)} and *group*2={"The professor is on vacation in England" (doc3), "The students are on vacation in England" (doc4), "The students and their professor are on vacation in England" (doc5)} in terms of meaning but different regarding the words shared between each other. Considering a bag-of-word representation of these sentences (Table 3.4). In this example, we recognize that *lecture* and *lesson* are synonym. Nonetheless,

TABLE 3.4 – Document×term matrix.

	professor	lecture	lesson	vacation	students	England
doc1	1	1	0	0	0	0
doc2	1	0	1	0	0	0
doc3	1	0	0	1	0	1
doc4	0	0	0	1	1	1
doc5	1	0	0	1	1	1

if we compute the cosine similarity between those terms from $\boldsymbol{X}$ as shown in Figure 3.8, they would not be related. In order to leverage this relation, we draw inspiration from several NMF algorithms [210, 211, 212] aiming to overcome this issue (also regularly encountered in image processing) by using the Wasserstein distance [213]. This distance which aims to measure the gap between probability distributions/histograms is arguably less sensitive to these types of redundant representations. However, computing the distance between two histograms of dimension n is expensive and requires to solve a

linear program in $\mathcal{O}(n^3 \log(n))$. Multiple works have shown that depending on the ground metric, it could be computed in $\mathcal{O}(n^2)$ time for instance using the L_1 ground distance [214], or several orders of magnitude faster using threshold ground distances [215]. But in a NMF learning process using large-



Documents similarities in $\boldsymbol{X}$ Documents similarities in $\boldsymbol{H}$ Terms similarities in $\boldsymbol{X}$ Terms similarities in $\boldsymbol{G}$

FIGURE 3.8 – Cosine similarity between documents or terms. The color and size indicate the binding force between the documents and the words in $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, $\boldsymbol{H} \in \mathbb{R}^{g \times n}$ and $\boldsymbol{G} \in \mathbb{R}^{g \times d}$.

scale histograms for which the distance is computed between more than one pair, the process remains very expensive and time consuming. Therefore, to go further and take advantages of the Optimal Transport at a lower computational cost with NMF, we use the Wasserstein embeddings obtained from the Wasserstein distance computed between the two probability marginals of $\boldsymbol{X}$. The model consists in transporting the weights of each marginal living in their respective simplex Δ_n and Δ_d into a respective lower dimensional simplex Δ_g using the data $\boldsymbol{X}$ and the lower dimensional factors data $\boldsymbol{Z}$ and $\boldsymbol{W}$. Subsequently, a regularization of $\boldsymbol{Z}$ and $\boldsymbol{W}$ according to those embeddings is achieved. `WE-NMF` implies the computation of $g(n+d)$ Wasserstein parameters that can be stored inside two matrices $\boldsymbol{G} \in \mathbb{R}_+^{g \times d}$ and $\boldsymbol{H} \in \mathbb{R}_+^{g \times n}$. These parameters deliver the optimal transportation for shifting the mass of documents (resp. terms) into the mass of the latent factors $\boldsymbol{w}_k$ with $k \in \{1, \ldots, g\}$ (resp. $\boldsymbol{z}_k$). As shown in Figure 3.8, we can see that this distance allows to highlight the relation between the two synonyms leading to a better understanding of relations between the documents citing those terms. Overall, we believe that this distance will be also relevant to leverage relations such as hyponyms (for instance : bus and car are hyponym of vehicle) which might be subject to reveal more proximities between documents.

3.2.2 Optimal transport and Wasserstein distance

Let $\mu = \sum_{i=1}^n a_i \delta_{x_i}$ be an empirical measure with a family of points $\mathcal{X} = (x_1, \ldots, x_n) \in \Omega^n$ and weights $\boldsymbol{a} = (a_1, \ldots, a_n)$ living in the probability simplex $\Delta_n = \{\forall \boldsymbol{a} \in \mathbb{R}_+^n : \sum_{i=1}^n a_i = 1\}$ (where Ω is an arbitrary space and δ_{x_i} the Dirac unit mass on x_i). Let ν be another empirical measure with

family $\mathcal{Y} = (y_1, \ldots, y_m) \in \Omega^m$ and weights $\boldsymbol{b} = (b_1, \ldots, b_m)$ living in the simplex Δ_m. The Wasserstein distance between μ and ν, also known as the transportation problem is defined as the optimization of the following problem :

$$W_p(\mu, \nu) = \mathbf{p}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}}) = \min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{T}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}} \rangle_F \tag{3.10}$$

where $U(\boldsymbol{a}, \boldsymbol{b})$ is the transportation polytope acting as the feasible set of all matrices $\boldsymbol{T} = (t_{ij}) \in \mathbb{R}_+^{n \times m}$ with the row and column marginals respectively equal to $\boldsymbol{a}$ and $\boldsymbol{b}$ s.t.

$$U(\boldsymbol{a}, \boldsymbol{b}) = \{\boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \sum_{i=1}^{n} t_{ij} = a_i, \sum_{j=1}^{m} t_{ij} = b_j\},$$

$\boldsymbol{M}_{\mathcal{X}\mathcal{Y}} = (m_{ij})$ is the matrix of pairwise distances (also called the cost parameter) between elements of $\mathcal{X}$ and $\mathcal{Y}$, $\mathbf{p}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}})$ is the Wasserstein distance in a form of the optimum of a linear program on $n \times m$ variables and parameter $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{M}_{\mathcal{X}\mathcal{Y}}$; $\langle \boldsymbol{T}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}} \rangle_F = \mathrm{Tr}(\boldsymbol{T}^\top \boldsymbol{M}_{\mathcal{X}\mathcal{Y}}) = \sum_{i,j}^{n,m} t_{ij} m_{ij}$ is the Frobenius dot-product.

3.2.3 Cuturi regularized Optimal Transport (Discrete)

$W_p(\mu, \nu)$ is a linear function with a cubic complexity $\mathcal{O}(n^3 \log(n))$ (when computed between two histograms of dimension n). Moreover, when n is large, $W_p(\mu, \nu)$ does not have a unique solution. In order to leverage these difficulties, [216] introduced a penalized version of the criterion using Shannon's entropy which has for effects to smooth the linear problem and turns it into a strictly convex problem which can be solved faster. The regularized criterion $W_p^\lambda(\mu, \nu)$ takes the following form :

$$\mathbf{p}_\lambda(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}}) = \min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{T}, \boldsymbol{M}_{\mathcal{X}\mathcal{Y}} \rangle_F - \lambda^{-1} H(\boldsymbol{T}) \tag{3.11}$$

where $H(\boldsymbol{T}) = -\sum_{i,j}^{n,m} t_{ij} \log(t_{ij})$ is the Shannon entropy and $\lambda \in [0, \infty]$ the regularization parameter. Depending on the value of λ, the smooth criterion converges toward the classical Wasserstein distance. If $\lambda \longrightarrow \infty$, $H(\boldsymbol{T})$ decreases and leans toward $W_p(\mu, \nu)$ (Deterministic coupling). In this case, $W_p^\lambda(\mu, \nu)$ becomes as or even more difficult to solve than the classical problem using an efficient linear solver. If $\lambda \longrightarrow 0$, $H(\boldsymbol{T})$ increases and pulls away $W_p^\lambda(\mu, \nu)$ from $W_p(\mu, \nu)$ (Independent coupling where μ and ν are assumed to be more independent). To solve $W_p^\lambda(\mu, \nu)$, $\boldsymbol{T}$ can be formulated as the solution of a scaling problem such as :

$$\boldsymbol{T} = \mathrm{diag}(\boldsymbol{a}) \boldsymbol{K} \mathrm{diag}(\boldsymbol{b}) \tag{3.12}$$

where $\boldsymbol{K}$ is the Gibbs kernel s.t. $\boldsymbol{K} = e^{-\lambda \boldsymbol{M}_{\mathcal{XY}}}$. To obtain $\boldsymbol{T}$, solution of $W_p^\lambda(\mu, \nu)$, the Sinkhorn-Knopp's algorithm which has a complexity $\mathcal{O}(nm)$ is commonly used. It involves matrix/vector multiplications and converges with a speed of several orders of magnitude faster than the regular EMD (Earth Mover's Distance) solvers. A version of the algorithm adapted for the Wasserstein distance can be found in [216] as well as an updated version in [217] which also solves the dual problem of eq(3.11). In the following, we will refer to this algorithm as SD for Sinkhorn Distance and its optimal solution for $\boldsymbol{T}$ as $\boldsymbol{T}^*$. It is also notable to note that eq(3.11) can be seen as a relative entropy and becomes a projection problem similar to the one encountered in `NMF-KL`; eq(3.11) is equivalent to $\min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} D_{KL}(\boldsymbol{T}||\boldsymbol{K})$ where D_{KL} is the Kullback-Leibler divergence. The Sinkhorn-Knopp's algorithm can easily be adapted to matrix/matrix multiplications to allow the computation of the Wasserstein distance between one histogram and a set of histograms.

3.2.4 Wasserstein Embeddings NMF (`WE-NMF`)

Let $\boldsymbol{X} \in \mathbb{R}_+^{n \times d}$ be a document-term matrix. NMF using the Wasserstein distance as an error for approximating several histograms $\boldsymbol{x}_j \in \mathbb{R}_+^n$ in $\boldsymbol{X}$ can be stated as minimizing $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) = \sum_j^d W_p(\boldsymbol{x}_j, [\boldsymbol{Z}\boldsymbol{W}^\top]_j)$ subject to $\boldsymbol{Z}\boldsymbol{W}^\top \in (\Delta_n)^d$, $\boldsymbol{Z} \geq 0$ and $\boldsymbol{W} \geq 0$. This implies a number of d intermediate linear calculus of complexity $\mathcal{O}(n^3 \log(n))$ using W_p, or d matrix scaling problems of complexity $\mathcal{O}(n^2)$ using W_p^λ. Both methods, `EMD-NMF` [210] and `W-NMF` [211] propose solutions to speed up the computational time. `EMD-NMF` uses the wavelet EMD approximation [218] while `W-NMF` uses the Legendre-Fenchel conjugate of W_p^λ which has a closed-form gradient and benefits from GPU parallelization. However with both methods, the overall computational time remains substantial for high dimensional data. With `WE-NMF` that we propose, we initiate a different approach aiming at reducing the amount of intermediate matrix scaling problems by considering only 4 histograms : the respective marginals of $\boldsymbol{X}$ of sizes d and n and their respective representations for the latent factors $\boldsymbol{z}_k \in \mathbb{R}_+^d$ and $\boldsymbol{w}_k \in \mathbb{R}_+^d$ of size g. Therefore, in `WE-NMF` we have the computation of two Wasserstein distances : $W_p^\lambda(\mu, \mu_{bis})$ (with the cost computed between the column vectors $\boldsymbol{x}_j$'s and the factors $\boldsymbol{z}_k$'s) and $W_p^\lambda(\nu, \nu_{bis})$ (with the cost computed between the row vectors $\boldsymbol{x}_i \in \mathbb{R}_+^d$ and the factors $\boldsymbol{w}_k$'s). Let $x_{i.} = \sum_j^d x_{ij}$, $x_{.j} = \sum_i^n x_{ij}$ and $\overline{Z} \in \{0,1\}^{n \times g}$ (resp. $\overline{W} \in \{0,1\}^{d \times g}$) be the classification matrix deduced from $\boldsymbol{Z}$ (resp. $\boldsymbol{W}$). We denote the respective weights for μ and μ_{bis} with $\boldsymbol{a} = (\frac{x_{1.}}{n}, \ldots, \frac{x_{n.}}{n}) \in \Delta_n$ and $\boldsymbol{a}_{bis} = (\sum_i^n a_i \overline{z}_{i1}, \ldots, \sum_i^n a_i \overline{z}_{ig}) \in \Delta_g$; the respective weights for ν and ν_{bis}

with $\boldsymbol{b} = (\frac{x.1}{d}, \ldots, \frac{x.d}{d}) \in \Delta_d$ and $\boldsymbol{b}_{bis} = (\sum_j^d b_j \overline{w}_{j1}, \ldots, \sum_j^d b_j \overline{w}_{jg}) \in \Delta_g$. Let $\boldsymbol{T}$ be the transportation matrix in the polytope $U(\boldsymbol{b}, \boldsymbol{b}_{bis})$ associated with the cost matrix $\boldsymbol{M_{ZX}} = \left[D(\boldsymbol{z}_k, \boldsymbol{x}_j)^p\right]_{kj} \in \mathbb{R}_+^{g\times d}$, $\boldsymbol{S}$ the transportation matrix in $U(\boldsymbol{a}, \boldsymbol{a}_{bis})$ associated with the cost matrix $\boldsymbol{M_{WX}} = \left[D(\boldsymbol{w}_k, \boldsymbol{x}_i)^p\right]_{ki} \in \mathbb{R}_+^{g\times n}$ and D is the ground metric. Thereby, we define the Wasserstein embedding matrices as :

$$\boldsymbol{G} \overset{def}{=} \boldsymbol{T} \odot \boldsymbol{M_{ZX}} \quad \text{and} \quad \boldsymbol{H} \overset{def}{=} \boldsymbol{S} \odot \boldsymbol{M_{WX}}, \tag{3.13}$$

where $\odot$ refers to the Hadamard product. The parameter b_{bis} denotes the samples weights detained per each cluster of samples while a_{bis} denotes the features weights per cluster of features. Both are respectively updated at each iteration of the algorithm. In the sequel, we aim to solve the following problem which consists in minimizing the objective function $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ taking the following form :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) + \gamma(D_I(\boldsymbol{H}||\boldsymbol{Z}^\top) + D_I(\boldsymbol{G}||\boldsymbol{W}^\top))\}, \tag{3.14}$$

where $\gamma \in \mathbb{R}_+$ is a regularization parameter. Solving problem(3.14) can be achieved through a set of multiplicative update rules. Let $\boldsymbol{\alpha} \in \mathbb{R}^{n\times g}$, $\boldsymbol{\beta} \in \mathbb{R}^{d\times g}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{d\times g}$ be the Lagrange multipliers, the Lagrangian function $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is equal to $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) + \mathrm{Tr}(\boldsymbol{\alpha}\boldsymbol{Z}^\top) + \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top)$. The resulting gradients are

$$\nabla_{z_{ik}}\mathcal{L} = -\Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\Big)_{ik} + \sum_j^d w_{jk} - \gamma\frac{h_{ki}}{z_{ik}} + \gamma + \alpha_{ik},$$

and

$$\nabla_{w_{jk}}\mathcal{L} = -\Big(\frac{\boldsymbol{X}^\top}{\boldsymbol{W}\boldsymbol{Z}^\top}\boldsymbol{Z}\Big)_{jk} + \sum_i^n z_{ik} - \gamma\frac{g_{kj}}{w_{jk}} + \gamma + \beta_{jk}.$$

Making use of the Karush-Kuhn-Tucker conditions, we obtain the stationary equations :

$$z_{ik}\Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W} + \gamma\frac{\boldsymbol{H}^\top}{\boldsymbol{Z}}\Big)_{ik} - z_{ik}\Big(\sum_j^d w_{jk} + \gamma\Big) = 0,$$

and

$$w_{jk}\Big(\frac{\boldsymbol{X}^\top}{\boldsymbol{W}\boldsymbol{Z}^\top}\boldsymbol{Z} + \gamma\frac{\boldsymbol{G}^\top}{\boldsymbol{W}}\Big)_{jk} - w_{jk}\Big(\sum_i^n z_{ik} + \gamma\Big) = 0,$$

which lead to the following update rules :

$$z_{ik} \leftarrow \frac{\left(\boldsymbol{Z} \odot \frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W} + \gamma\boldsymbol{H}^\top\right)_{ik}}{\sum_j^d w_{jk} + \gamma} \tag{3.15}$$

$$w_{jk} \leftarrow \frac{\left(\boldsymbol{W} \odot \frac{\boldsymbol{X}^\top}{\boldsymbol{W}\boldsymbol{Z}^\top}\boldsymbol{Z} + \gamma\boldsymbol{G}^\top\right)_{jk}}{\sum_i^n z_{ik} + \gamma}. \tag{3.16}$$

Algorithm 14 Wasserstein Embeddings NMF (`WE-NMF`), $\mathcal{O}(ngd + gN)$.

Input : $\boldsymbol{X}$, γ, λ, p, $\boldsymbol{a} \in \Delta_n$, $\boldsymbol{b} \in \Delta_d$ and g.
Output : $\boldsymbol{Z}$, $\boldsymbol{W}$, $\boldsymbol{G}$, and $\boldsymbol{H}$.
Initialization : $\boldsymbol{Z} \leftarrow \boldsymbol{Z}^{(0)}$; $\boldsymbol{W} \leftarrow \boldsymbol{W}^{(0)}$
repeat
1. $\boldsymbol{M_{ZX}} = \left[D(\boldsymbol{z}_k, \boldsymbol{x}_j)^p\right]_{kj}$, update $\boldsymbol{b}_{bis}$
1'. $\boldsymbol{T} \leftarrow \boldsymbol{T}^*$ using $SD(\boldsymbol{M_{ZX}}, \lambda, \boldsymbol{b}, \boldsymbol{b}_{bis})$, $\boldsymbol{G} \leftarrow \boldsymbol{T} \odot \boldsymbol{M_{ZX}}$
2. $\boldsymbol{M_{WX}} = \left[D(\boldsymbol{w}_k, \boldsymbol{x}_i)^p\right]_{ki}$, update $\boldsymbol{a}_{bis}$
2'. $\boldsymbol{S} \leftarrow \boldsymbol{S}^*$ using $SD(\boldsymbol{M_{WX}}, \lambda, \boldsymbol{a}, \boldsymbol{a}_{bis})$, $\boldsymbol{H} \leftarrow \boldsymbol{S} \odot \boldsymbol{M_{WX}}$
3. update $\boldsymbol{Z}$ with eq(3.15)
4. update $\boldsymbol{W}$ with eq(3.16)
until convergence
5. Normalize each $\boldsymbol{z}_k$ to unit-norm.

In this case, D is the $(1 - \cos)$ dissimilarity and $p = 2$. SD stands for Sinkhorn Distance. Note that steps (1,1') and steps (2,2') are independent and can be parallelized.

3.2.4.1 Convergence analysis

Recalling the optimization problem of `WE-NMF` as follows :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) + \gamma(D_I(\boldsymbol{H}||\boldsymbol{Z}^T) + D_I(\boldsymbol{G}||\boldsymbol{W}^T))\}. \tag{3.17}$$

Theorem 3.2.1. *$\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ is non-increasing under the update of $\boldsymbol{Z}$ and $\boldsymbol{W}$.*

Definition 3.2.1. Let $(\boldsymbol{z}, \boldsymbol{z}') \subseteq \mathbb{R}_+^g \times \mathbb{R}_+^g$, $\mathcal{G}(\boldsymbol{z}, \boldsymbol{z}')$ is an auxiliary function for $\mathcal{F}(\boldsymbol{z})$ if the following conditions are satisfied :

$$\forall \boldsymbol{z}, \mathcal{G}(\boldsymbol{z}, \boldsymbol{z}') \geq \mathcal{F}(\boldsymbol{z}) \quad \text{and} \quad \mathcal{G}(\boldsymbol{z}, \boldsymbol{z}) = \mathcal{F}(\boldsymbol{z}).$$

A key point to the auxiliary function is the following lemma :

Lemma 3.2.2. *If $\mathcal{G}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{F}(\boldsymbol{z})$, $\mathcal{F}(\boldsymbol{z})$ is non-increasing under the update*

$$\boldsymbol{z}^{(t+1)} = \arg\min_{\boldsymbol{z}} \mathcal{G}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$$

Proof. $\mathcal{F}(\boldsymbol{z}^{(t+1)}) \leq \mathcal{G}(\boldsymbol{z}^{(t+1)}, \boldsymbol{z}^{(t)}) \leq \mathcal{G}(\boldsymbol{z}^{(t)}, \boldsymbol{z}^{(t)}) = \mathcal{F}(\boldsymbol{z}^{(t)})$.■ □

Re-writting $\mathcal{F}(\boldsymbol{Z})$ in a vector coordinates format and as a sum of convex functions. We denote

$\mathcal{F}(\boldsymbol{z}) = \mathcal{F}_1(\boldsymbol{z}) + \mathcal{F}_2(\boldsymbol{z})$ where

$$\begin{cases} \mathcal{F}_1(\boldsymbol{z}) \overset{def}{=} \sum_j^d \Big[x_j \log \frac{x_j}{\sum_k^g z_k w_{jk}} - x_j + \sum_k^g z_k w_{jk} \Big], & \text{(3.18a)} \\ \mathcal{F}_2(\boldsymbol{z}) \overset{def}{=} \gamma \Big[\sum_k^g h_k \log \frac{h_k}{z_k} - h_k + z_k + \sum_{k,j}^{g,d} h_{kj} \log \frac{h_{kj}}{w_{jk}} - h_{kj} + w_{jk} \Big]. & \text{(3.18b)} \end{cases}$$

Proposition 3.2.1. $\mathcal{G}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \mathcal{G}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{F}_2(\boldsymbol{z})$ is an auxiliary function for $\mathcal{F}(\boldsymbol{z})$ where

$$\begin{aligned} \mathcal{G}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \overset{def}{=} & \sum_j^d (x_j \log x_j - x_j) + \sum_{j,k}^{d,g} z_k w_{jk} \\ & - \sum_{j,k}^{d,g} x_j \frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{j\ell}} \Big[\log(z_k w_{jk}) - \log \Big(\frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{j\ell}} \Big) \Big]. \end{aligned} \tag{3.19}$$

Lemma 3.2.3. $\mathcal{G}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{F}_1(\boldsymbol{z})$.

Proof. Using the convexity of the negative logarithm, we derive the following inequality :

$$-\log \Big(\sum_k^g z_k w_{jk} \Big) = -\log \Big(\sum_k^g \mu_k \frac{z_k w_{jk}}{\mu_k} \Big) \overset{Jensen}{\leq} -\sum_k^g \mu_k \log \Big(\frac{z_k w_{jk}}{\mu_k} \Big), \tag{3.20}$$

where $\mu_k = \frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{j\ell}}$ and $\sum_k^g \mu_k = 1$. From the inequality (3.20), it follows that $\mathcal{G}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{F}_1(\boldsymbol{z})$.□

□

Proof of Theorem 3.2.1. Finding the minimum of $\mathcal{G}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ with respect to z_k gives :

$$\nabla_{z_k} \mathcal{G}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \sum_j^d w_{jk} - \sum_j^d x_j \frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{j\ell}} \frac{1}{z_k} + \gamma \Big[1 - \frac{h_k}{z_k} \Big]. \tag{3.21}$$

Setting the gradient to zero leads to :

$$z_k^{(t+1)} = \frac{z_k^{(t)} \sum_j^d \frac{x_j}{\sum_\ell^g z_\ell^{(t)} w_{j\ell}} w_{jk} + \gamma h_k}{\sum_j^d w_{jk} + \gamma}. \tag{3.22}$$

Rewritten in a matrix format, this is equivalent to the original update rule. Therefore Lemma 3.2.2 is approved and $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ is non-increasing under the update of $\boldsymbol{Z}$. The same process can be reversed to show that $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ is non-increasing under update of $\boldsymbol{W}$. ■

3.2.4.2 Complexity analysis

In the worst case scenario, the complexities of the multiplicative updates(3.15,3.16) remain the same as for NMF which is $\mathcal{O}(ngd)$. However the computational cost of updates(3.13) becomes the main bottleneck as the complexity of `WE-NMF` depends directly on the complexity of the chosen algorithm used to compute $\boldsymbol{T}$ and $\boldsymbol{S}$. Using W_p^λ of complexity $\mathcal{O}(gd)$ for $\boldsymbol{G}$ and $\mathcal{O}(gn)$ for $\boldsymbol{H}$, the complexity for Eqs.(3.13) is then $\mathcal{O}(gN)$ where $N = \max(d, n)$, leading to the overall complexity for one iteration at $\mathcal{O}(ngd + gN)$.

3.2.5 Experiments

To assess the performance of our model, we compare it with several NMF models commonly used for document clustering as well as Sinkhorn Distance/Earth Mover's Distance clustering methods. The list includes : orthogonal NMF (ONMF) [158], Projective NMF (PNMF) [161], Graph Regularized NMF (GNMF) [163], Spherical K-means [39], and Variational Wasserstein Clustering (VWC) [219], which is equivalent to a Wasserstein Spherical-Kmeans with the $(1 - \cos)$ dissimilarity as the ground metric. Moreover, `WE-NMF` collapses to the original NMF when $\gamma = 0$ which will be our baseline for comparing the direct gain of our model.

Five benchmarking document-term datasets highlighting several varieties of challenging situations were selected for these experiments. Their characteristics are displayed in Table 3.5. They differ in terms of amount of clusters, dimension, clusters balance (coefficient defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class), degree of mixture of the different groups and sparsity (where nz indicates the percentage of non-zero values). We normalized each data matrix with TF-IDF and their respective documents to unit L_2-norm to remove the bias introduced by their length. Two measures widely used to quantify the clustering performance of an

TABLE 3.5 – Datasets description (# denotes the cardinality).

Datasets	Characteristics				
	#Documents	#Words	#Clusters	nz (%)	Balance
CSTR	475	1000	4	3.40	0.399
CLASSIC4	7095	5896	4	0.59	0.323
NG5	4905	10167	5	0.92	0.943
NG20	18846	14390	20	0.59	0.628
SPORTS	8580	14870	7	0.86	0.0358

algorithm were employed, namely the Normalized Mutual Information (NMI) [119] and the adjusted Rand Index (ARI) [122]. Both criteria reach a value of 1 when the clustering is identical to the ground truth.

3.2.5.1 Settings

As defined earlier, matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ can be recovered after solving problem(3.10) by using a Sinkhorn-Distance algorithm in order to obtain the optimal transportation matrices $\boldsymbol{T}$ and $\boldsymbol{S}$. The results showcased here were obtained using algorithm 3 of [217] (Smooth Primal and Dual Optima) in our algorithm. The number of clusters was set to the original number of classes for each dataset. The results of each respective algorithm were obtained over an average of 30 random runs. Their respective parameters (if required) were set according to the recommended settings ; for instance $\gamma = 100$ for GNMF.

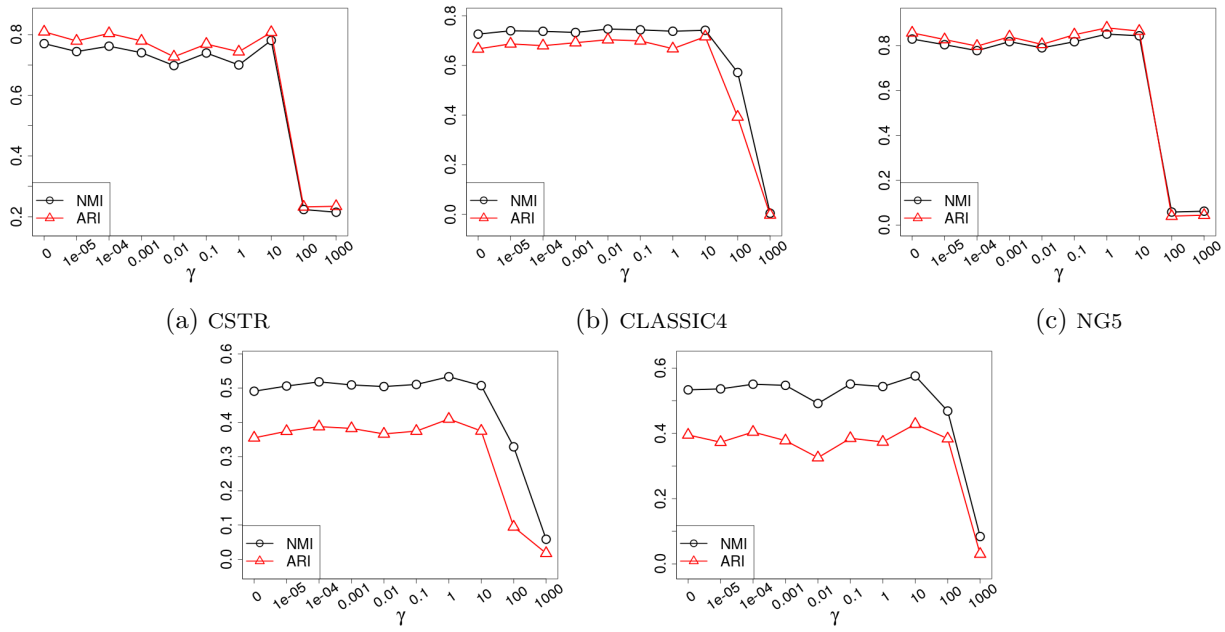
3.2.5.2 Other Optimal Transport algorithms

Algorithms such as SO-TROT (Second Order Row-Tsallis regularized Optimal Transport), KL-TROT [220] and SAG (Stochastic Average Gradient) for discrete OT [221] were also tested. In our model, $\boldsymbol{S}$ and $\boldsymbol{T}$ are quite small as the rank defined by the user remains low for clustering application. Thereby most of the time, the use of stochastic methods become unnecessary as conventional algorithms converge in a decent time. Nevertheless, they can become handy whether the user specifies a large amount of clusters. After testing, we denote very similar results with SAG for discrete OT compared to the ones obtained with SD. The results with SO-TROT and KL-TROT were similar on CSTR. The TROT distance is appealing since it generalizes Optimal transport and [216] approach as well as involving the escort distribution. Unfortunately, its very high computational cost makes it unsuitable on larger dataset.

3.2.5.2.1 Ground metric. We chose to retain the $(1-\cos)$ dissimilarity as the ground metric function to map elements $(\boldsymbol{x}_j, \boldsymbol{z}_k)$ and $(\boldsymbol{x}_i, \boldsymbol{w}_k)$. Despite its limitations for advanced semantic relations, this measure is widely acknowledged as a referenced for text mining and remains relevant in most situations. Indeed, It is particularly appealing when we are dealing with large amount of directional sparse data ; it does not affect therefore the computational cost of this dissimilarity.

3.2.5.2.2 λ setting. As λ increases, the objective of problem(3.11) is expected to converge toward the classical Optimal Transportation distance. However, [216] has reported in his experiments that (3.11) tends *to hover* above the classical OT distance by about 10% and that practical value of λ were not necessarily the highest. Moreover, fixing λ has to be in regard to the order of magnitude of the cost matrix. While in our case, $m_{ij} \in [0, 1]$, other continuous function may attribute larger values of m_{ij} which may result in constraining λ in a reduced interval to avoid numerical overflow.

3.2.5.2.3 γ setting. The regularization parameter γ has been studied across each data matrix along a range going from 10^{-5} to 10^{3}. Figure 3.9 showcases variations of NMI and ARI with `WE-NMF` according to the evolution of γ; taking $\gamma = 10$ seems to be good trade-off. NG5 and NG20 are the only datasets where $\gamma = 1$ will be better, however the performance at $\gamma = 10$ remains equal or superior to the one of NMF. For $\gamma > 10$, the algorithm fails due to numerical overflow.



(a) CSTR (b) CLASSIC4 (c) NG5

(d) NG20 (e) SPORTS

FIGURE 3.9 – Impact of the regularization parameter γ of `WE-NMF` (SD).

3.2.5.3 Empirical results

Table 3.6 summarizes the different results. As we can see, `WE-NMF` provides better performances overall. Notice also that `NMF-KL` gives similar achievements in terms of ARI on CSTR and NG5. Ack-

nowledging the abilities of the Wasserstein embeddings to improve the clustering performances, we decided to use them as supports for visualizing the data (samples and features) with respect to the clustering for the set of terms and the available original partition for the documents. The different groups are depicted in color. We observed that the embeddings matrices $\boldsymbol{H}$ and $\boldsymbol{G}$ respectively provide better representations for the actual document clusters and even more significant ones for the term clusters with a soaring separability. Figure 3.10 highlights different visualizations of the documents of each

TABLE 3.6 – Mean and standard deviation of NMI and ARI over different datasets.

Datasets	Metrics	NMF-KL	ONMF	PNMF	GNMF	S-Kmeans	VWC	WE-NMF(SD)
CSTR	NMI	0.77±0.02	0.65±0.05	0.66±0.01	0.57±0.08	0.76±0.01	0.55±0.03	**0.78**±0.03
	ARI	**0.81**±0.02	0.56±0.04	0.56±0.01	0.53±0.11	0.79±0.01	0.50±0.03	**0.81**±0.03
CLASSIC4	NMI	0.72±0.09	0.55±0.09	0.59±0.05	0.65±0.04	0.60±0.001	0.54±0.02	**0.74**±0.01
	ARI	0.65±0.10	0.39±0.09	0.44±0.01	0.49±0.05	0.47±0.001	0.45±0.01	**0.72**±0.01
NG5	NMI	0.83±0.01	0.65±0.04	0.65±0.05	0.63±0.07	0.74±0.03	0.68±0.04	**0.84**±0.01
	ARI	**0.86**±0.01	0.48±0.08	0.47±0.09	0.62±0.09	0.64±0.07	0.68±0.06	**0.86**±0.01
NG20	NMI	0.49±0.01	0.44±0.02	0.45±0.02	0.50±0.01	0.50±0.02	0.41±0.01	**0.51**±0.01
	ARI	0.35±0.01	0.22±0.02	0.24±0.02	0.35±0.05	0.31±0.02	0.27±0.01	**0.38**±0.02
SPORTS	NMI	0.53±0.01	0.55±0.02	0.56±0.001	0.55±0.001	**0.62**±0.02	0.55±0.02	0.58±0.01
	ARI	0.40±0.01	0.28±0.01	0.28±0.001	0.28±0.001	0.40±0.04	0.39±0.01	**0.43**±0.01

dataset using UMAP's (Uniform Manifold Approximation and Projection for Dimension Reduction) components [222] obtained on $\boldsymbol{X}$, $\boldsymbol{Z}$ and $\boldsymbol{H}^\top$, where the true classes are projected. Figure 3.11 shows similar projections for the set of terms where the depicted groups are the term clusters obtained from the solution with the best criterion according to NMF-KL (for $\boldsymbol{X}^\top$ and $\boldsymbol{W}$) and WE-NMF (for $\boldsymbol{G}^\top$). In Figure 3.10, UMAP does not always provide a meaningful visualization of the data samples $\boldsymbol{x}_i$ (neither the features $\boldsymbol{x}_j$, see line 1 in Figure 3.11). Several setups made according to the crucial parameters (*min_dist* and *n_neighbors*) emphasized by the authors were tested with $n\_neighbors \in \{15, 80, 320\}$; neither of them was successful to circumvent these issues. Also, we observed that the use of different metrics such as cosine similarity instead of the euclidean distance did surprisingly not improved the visualization. Therefore, we conducted the rest of our experiments with the defaults parameters. CLASSIC4 and NG20 are the datasets with the highest sparsity rates which might be the reason leading UMAP to fail.

Furthermore, Figure 3.10 shows in some cases that a better separability between document clusters can be observed from the Wasserstein Embeddings $\boldsymbol{H}^\top$ of WE-NMF as opposed to NMF-KL factor $\boldsymbol{Z}$.

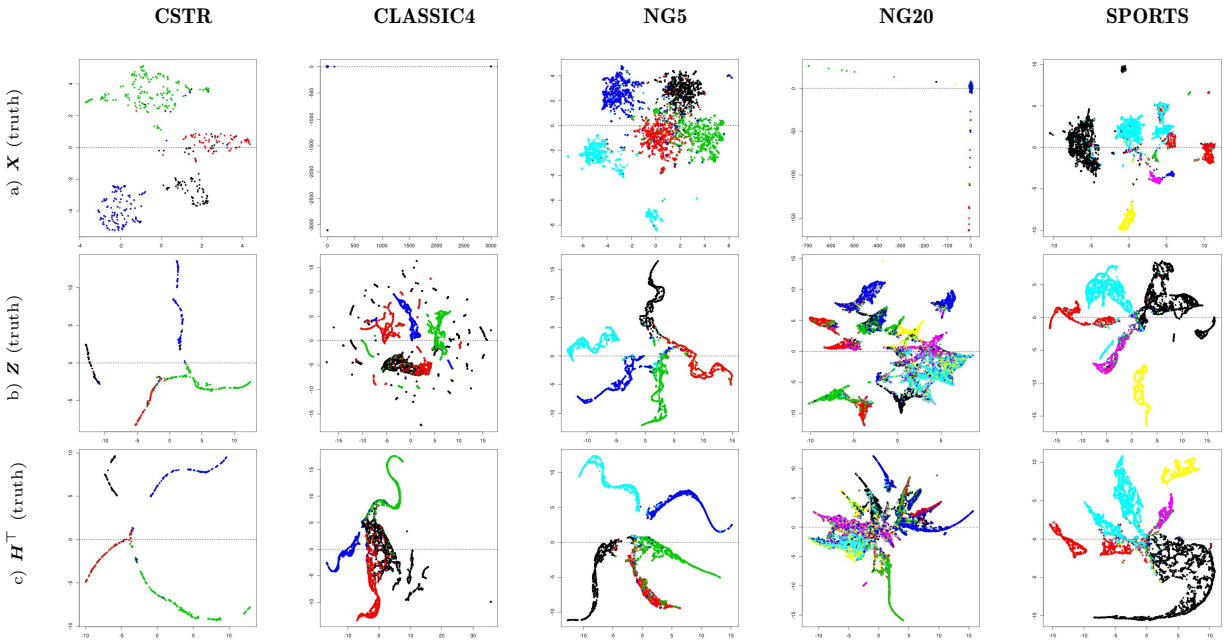


FIGURE 3.10 – Visualizations of true document classes by UMAP applied on a) $\boldsymbol{X}$, b) $\boldsymbol{Z}$ obtained by `NMF-KL`, and c) $\boldsymbol{H}^\top$ by `WE-NMF`.

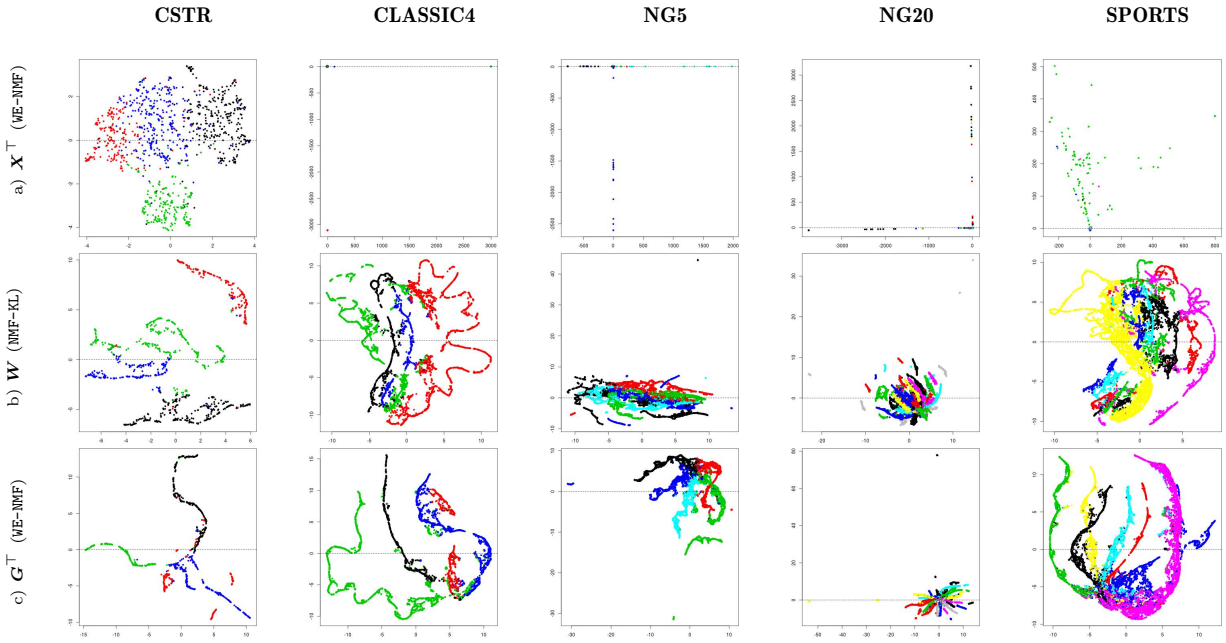


FIGURE 3.11 – Visualizations of term clusters by UMAP applied on a) $\boldsymbol{X}^\top$ using `WE-NMF` clusters, b) $\boldsymbol{W}$ using `NMF-KL` clusters, and c) $\boldsymbol{G}^\top$ using `WE-NMF` clusters.

It is highlighted with CSTR and CLASSIC4 which in addition is the dataset where WE-NMF improves the most in terms of NMI and ARI. With NG20 and SPORTS were we also see major improvements, the document visualizations from $\boldsymbol{H}^\top$ are substantially different from those of $\boldsymbol{Z}$. In Figure 3.11, although the true groups of terms are not available, our method seems to be more suitable by allowing representations with substantial clusters separability compared to NMF-KL's factor $\boldsymbol{W}$ and $\boldsymbol{X}^\top$. UMAP which builds its components according to the samples instead of the features does not provide any meaningful visualizations for the terms on the transposed data matrix $\boldsymbol{X}^\top$ although it could detect groups of documents on $\boldsymbol{X}$ (Figure 3.10). In practice, terms are more difficult to classify. They can appear in several contexts, be used in different topics or even be considered as noisy depending on the pre-processing applied to build the document-term matrix. From these observations, dimensionality reduction seems to be beneficial and WE-NMF which allows the decomposition matrices to be approximated by not only $\boldsymbol{X}$ but also by $\boldsymbol{H}$ and $\boldsymbol{W}$ might already gain an advantage for this type of data.

3.3 Conclusion

In this chapter, we have described two novel NMF approaches which explicitly account for the semantic relationships among words. In SNMF, the joint decomposition of the data matrix and the word-context (or word-word co-occurrence) matrix simultaneously into a shared factor proved its efficiency in leveraging more relationships between words that were not emerging in the data matrix. The low dimensional latent space is therefore enriched and the shared factor is able to transport additional information for finding a decomposition of the data matrix more faithful to the original partitions of clusters. Moreover, we identified in which situations Semantic-NMF does provide the most significant improvements, which allows us to gain further insights into the benefits of leveraging the word relationships. In addition, our approach does not necessarily require an additional source of data but is versatile and can allow contexts from other resources. It is subject to many more improvements using a consensus approach. With $\texttt{WE}-\texttt{NMF}$, we introduced a novel NMF model to take in consideration semantic relationships such as synonyms and hyponyms (non linear relations). Using the embeddings formed by the transportation matrix and the ground metric matrix of the Wasserstein distance between the marginals of the data matrix and the decomposition factors, $\texttt{WE}-\texttt{NMF}$ is able to regularize the later according to the non linear relations capture by the Wasserstein distance.

Furthermore, the low dimensionality of these embeddings and the regularization scheme allows to utilize this distance in NMF at a much lower computational cost. The regularized factors were effective to boost the clustering performances of the Kullback-Leibler NMF and the separability offers by the Wasserstein embedings provides strong supports for observing the data within the lower dimensional space.

Overall, while `SNMF` outperforms the results of several proven NMF extensions, it does not consistently overcome the results of NMF with the I-divergence, let alone $\texttt{WE} - \texttt{NMF}$. This confirms the acknowledgement of the I-divergence as a better cost function for achieving NMF document clustering. The higher complexity of its gradient [223] has often led to more efforts being devoted to enhance the clustering performance through the use of less expensive error measures such as the Frobenius norm. However using this measure blurs the real potential of NMF for document clustering and therefore, in the following, our focus will be mainly on the I-divergence.

Chapitre 4

Toward probabilistic factors for NMF and connections with Finite Mixture Models

In contrast to the previous proposals where several types of co-variables where introduced to regularize the NMF factors, in this chapter, we present an approach which avoids adding any more computational complexity to the problem of NMF with the I-divergence. In addition, this approach offers a characterization of NMF as a direct clustering algorithm.

Thereby, in the first section, we achieve a regularization of the main NMF objective to simultaneously maximize the Shannon Entropy along side the distance function. It results that this optimization coincides with the well known maximum entropy principle and implies one factor (namely the document factor matrix) to be constrained as a set of probability distributions. Furthermore, the uncertainty of the latter is shown to be modeled by the setting of a Lagrange multiplier and the characteristic of the underlying probability distribution associated with our distance function. To emphasize the impact of the Shannon Entropy, the class of entropy functionals from the Rényi family is considered, which included several well know entropies such as Shannon's or Hartley's. Our algorithm, that its convergence is guaranteed, is evaluated as similarly and demonstrates significant major improvements compared to the state-of-the-art methods.

The subject described in the second section of this chapter originally takes its roots from our desire to reduce the computational cost of the gradient of the I-divergence, to speed up the convergence. Therefore, considering that one factor is now a set of probabilities, using the Jensen inequality, we highlight the transition from NMF to mixture models and shows that solving NMF is equivalent to maximizing a surrogate function of the complete log-likelihood. A common practice in NMF of text

data is to normalize the data in the unit-sphere. Therefore, in addition, we study the impact of the normalization of sparse random variables in the unit-sphere for NMF and FMMs and emphasize that the discrete Poisson distribution with variables in the unit-sphere has minimum entropy. To complete our comparison of NMF and FMMs, we also derive a new NMF method using the $(1-\cos)$ dissimilarity referred to as Spherical NMF. Furthermore, several NMF regularizations (in which the first proposal appears to be a special case) channeled from FMMs are also highlighted.

4.1 Constrained NMF with entropic regularization

4.1.1 Motivations

We focus on improving the clustering performance of $\texttt{NMF}-\texttt{KL}$ by considering an optimization procedure maximizing the non-linear entropy of a set of distributions $\boldsymbol{Z}$ subject to NMF's data approximation $\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{W}^\top$. The information measure $H(.)$ (known as Entropy) introduced by Shannon [110] is a useful and popular concept for measuring the amount of information for an observed random variable [111]. Let $\boldsymbol{z} = (p(w_1), \ldots, p(w_g))$ be the discrete probability distribution of a random variable $W = (w_1, \ldots, w_g)$:

$$H(\boldsymbol{z}) = -\sum_{k=1}^{g} z_k \log z_k. \tag{4.1}$$

For $\boldsymbol{z} = (1/g, \ldots, 1/g)^\top$, the uncertainty among all events is total and $H(\boldsymbol{z})$ is maximized. Therefore, as enumerated by Jaynes [224], maximizing $H(\boldsymbol{z})$ is established as what sounds logical to avoid bias assumptions on the available data and produce unbiased inferences. On the other hand, as stated in [225], the minimization of $H(\boldsymbol{z})$ suggests *clustering validity* since the probability would tend toward 0 or 1. Following these ideas, our challenge was set to maximize $H(\boldsymbol{z})$ accordingly to obtaining of high quality approximation.

4.1.2 Related Works

The inspiration for our contribution can be traced back to the work of Jaynes [224, 226, 111] on maximum-entropy distributions where $H(.)$ is maximized given a probability marginal constraint and

one or several moment constraints c_j :

$$\max_{\boldsymbol{z}} H(\boldsymbol{z}) + \gamma\Big(\sum_{k}^{g} z_k - 1\Big) + \sum_{j}^{d} \lambda_j \Big(\sum_{k}^{g} z_k f_j(W) - c_j\Big), \tag{4.2}$$

where γ and λ_j are the associated Lagrange multipliers. The problem is often referred to as the maximum entropy principle and collapses to many common probability distributions (e.g. uniform with no moment constraint ; Gaussian with variance constraint ; exponential with the mean constraint).

In our case, we consider the non-linear version of this problem for a set of probabilities distribution arranged in $\boldsymbol{Z}^\top \in (\Delta_g)^n$ such that $H(\boldsymbol{Z}) = -\sum_{i,k}^{n,g} z_{ik} \log z_{ik}$.

Subsequently, we undertake to study the amount of uncertainty arising in $\boldsymbol{Z}$ while maximizing $H(\boldsymbol{Z})$ under nonnegative matrix factorization constraint $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{W}^\top$. For instance, following the exact low-rank reconstruction of the observed data where each observations is approximated as an expectation such that $x_{ij} = \sum_k^g z_{ik} w_{jk} = \mathbb{E}_{z_i}[w_j]$, the maximum entropy principle could be stated as follows :

$$\max_{\boldsymbol{Z}^\top \in (\Delta_g)^n, \boldsymbol{W} \geq 0} H(\boldsymbol{Z}) + \sum_{i}^{n} \gamma_i \Big(\sum_{k}^{g} z_{ik} - 1\Big) + C, \tag{4.3}$$

where $C = \sum_{i,j}^{n,d} \lambda_{ij}\Big(\sum_k^g z_{ik} w_{jk} - x_{ij}\Big)$. While this exact fitting would be beneficial in the search of simpler solutions, the amount of observations implies the proliferation of $n \times d$ constraints and Lagrange multipliers. Furthermore, decreasing the amount of parameters (to n) could be achieved by considering a moment constraint defined in terms of intra-variance reminiscent of a clustering criterion. In this case, we would have $C = \sum_i^n \lambda_i\Big(\sum_k^g z_{ik} d(\boldsymbol{x}_i, \boldsymbol{w}_k) - S\Big)$ in problem (4.3). A further reduction of the amount of parameters can be formulated as a constraint in terms of matrix approximation involving the estimation of one Lagrange multiplier, such that $C = \lambda(\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) - S) \leq 0$. Whether $\mathcal{D}(.) = \frac{1}{2}\|.\|_F^2$ is the sum of squares (SSQ), it has a χ^2 distribution which facilitates the setting of λ. Otherwise, due to non-linearity, setting λ requires additional numerical computations.

Additionally, our work can be related to the subject of ill-posed problems such as Compressed sensing [227, 228], maximum entropy inversion [229, 230, 231] or sparse recovery [232] in a linear span where several attempts to recover sparse solutions using penalty functions such as the $l_1 - norm$ (reminiscent of the Lasso technique in regression) or the Shannon entropy have been suggested. Maximum entropy methods associated with maximum likelihood [233, 234] can also so be handful. More generally, the reader can also refer to [235] for an application of ME with non-linear programming

and [236, 237] for ME applied in Natural Language Processing.

4.1.3 Maximum-Entropy Inference

We reformulate our maximum-entropy problem as a non-linear minimization problem defined as :

$$\min_{\boldsymbol{Z}^\top \in (\Delta_g)^n} -H(\boldsymbol{Z}), \quad s.t.\ \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) \leq S. \tag{4.4}$$

Note that since $H(\boldsymbol{Z})$ is Schur-concave, problem (4.4) remains convex in $\boldsymbol{Z}$ and $\boldsymbol{W}$ separately but not jointly. Note that $x \log x$ has an infinite positive slope at 0 which by definition of the Shannon entropy, is handled by replacing $0 \log(0)$ with 0. The associated Lagrangian is therefore :

$$\begin{aligned}\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \lambda) = &-H(\boldsymbol{Z}) + \sum_i^n \gamma_i \Big(\sum_k^g z_{ik} - 1 \Big) \\ &+ \lambda(\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) - S).\end{aligned} \tag{4.5}$$

$\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$ is set as the I-divergence defined as :

$$D_I(\boldsymbol{X} || \boldsymbol{Z}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} x_{ij} \log \left(\frac{x_{ij}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}} \right) - x_{ij} + [\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}, \tag{4.6}$$

and therefore, unlike the sum of squares, no information is available on the statistic S. Naturally, differentiation w.r.t. γ_i and λ produce the constraint equations :

$$\nabla_{\gamma_i} \mathcal{L} = \sum_k^g z_{ik} - 1 = 0, \tag{4.7}$$

$$\nabla_\lambda \mathcal{L} = \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) - S = 0. \tag{4.8}$$

Differentiation w.r.t. z_{ik} gives the following gradient :

$$\nabla_{z_{ik}} \mathcal{L} = 1 + \log z_{ik} + \gamma_i + \lambda \partial \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}. \tag{4.9}$$

Setting this gradient to zero and substituting the resulting solution of z_{ik} in $\nabla_{\gamma_i} \mathcal{L}$ to obtain γ_i produce the following solution :

$$z_{ik} = \frac{e^{-\lambda \partial \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}}}{\sum_k^g e^{-\lambda \partial \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}}}. \tag{4.10}$$

Substituting this expression into $\nabla_\lambda \mathcal{L}$ leads to no closed-form solution for λ. In practice, λ is usually obtained using root-finding algorithm (e.g. Newton-Raphson). However, in our case, the resulting

equation is highly non-linear and finding a solution for λ is almost impossible despite a "supervised" fixing of S. $\boldsymbol{W}$ can be estimated using Projected gradient coordinate such as :

$$\boldsymbol{W} \leftarrow \left[\boldsymbol{W} - \eta \partial \nabla_{\boldsymbol{W}} \mathcal{L}\right]^{+}, \tag{4.11}$$

where η is the step parameter and $[.]^{+}$ designates the positive orthant for the nonnegativity purpose. Since problem (4.4) can be recast as the following minimization problem (where the nonnegativity constraint on $\boldsymbol{W}$ can also be added) :

$$\min_{\boldsymbol{Z}^{\top} \in (\Delta_g)^n, \boldsymbol{W} \geq 0} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = -H(\boldsymbol{Z}) + \lambda \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^{\top})\}, \tag{4.12}$$

we consider this formulation for a manual setting of λ. Note that in this formulation, S is set to zero (which is equivalent to an exact reconstruction constraint $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^{\top}) = 0$ in problem (4.4)). The associated Lagrangian $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \lambda)$ is given by

$$-H(\boldsymbol{Z}) + \sum_{i}^{n} \gamma_i \Big(\sum_{k}^{g} z_{ik} - 1 \Big) + \lambda \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^{\top}) - \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^{\top})$$

and differentiation w.r.t. w_{jk} leads therefore to :

$$\nabla_{w_{jk}} \mathcal{L} = -\lambda \Big(\frac{\boldsymbol{X}^{\top}}{\boldsymbol{W}\boldsymbol{Z}^{\top}} \boldsymbol{Z} \Big)_{jk} + \lambda \sum_{i}^{n} z_{ik} + \boldsymbol{\beta}_{jk}. \tag{4.13}$$

Setting this gradient to zero and making use of the Karush-Kuhn-Tucker (KKT) conditions $\boldsymbol{\beta} \odot \boldsymbol{W} = 0$ to obtain the stationary equation, the multiplicative update rule (which can also be rewritten as a gradient descent update) is stated as :

$$w_{jk} \leftarrow w_{jk} \frac{\left(\frac{\boldsymbol{X}^{\top}}{\boldsymbol{W}\boldsymbol{Z}^{\top}} \boldsymbol{Z}\right)_{jk}}{\sum_{i}^{n} z_{ik}} = w_{jk} - \eta \nabla_{w_{jk}} \mathcal{F}, \tag{4.14}$$

where $\eta = \frac{w_{jk}}{\sum_{i}^{n} z_{ik}}$. Differentiation w.r.t. z_{ik} gives the same solution for z_{ik} as in problem (4.4). Algorithm 15 details the procedure for maximizing entropy under non-negative matrix factorization constraint ($\texttt{ME} - \texttt{NMF}$).

Algorithm 15 $\texttt{ME} - \texttt{NMF}$

Input : $\boldsymbol{X}$, g, λ, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
1. update $\boldsymbol{Z}$ with Eq(4.10)
2. update $\boldsymbol{W}$ with Eq(4.14) ;

until convergence

Note that whether η is a fixed step or set to equal the multiplicative update, $\mathcal{F}$ has a quadratic convergence, or is non-increasing.

As mentioned in the introduction, $H(\boldsymbol{Z})$ should be maximized for unbiased parameter estimations. However minimum entropy also reduces uncertainty which increases clustering validity. From Eq(4.10), it is obvious that :

$$\lim_{\lambda \to 0} z_{ik} = \frac{e^0}{g e^0} = \frac{1}{g}, \tag{4.15}$$

and setting λ closed to 0 will favor solutions with maximum uncertainty, while increasing λ lead to more complex maximum-entropy distributions w.r.t. the constraint. Setting λ is therefore a key requirement for finding a maximum-entropy distribution not drawn solely by a low setting of λ. Normally, this could be achieved empirically. However, studying various values of λ with $\texttt{ME}-\texttt{NMF}$ fell short of expectations. Due to the non-linearity of $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$, $\partial \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}$ implies summations over d dimensions. Therefore, because of the exponential, numerical overflow is rapidly observed with Eq(4.10) as λ increases. For solving this issue, we derive a new estimate for z_{ik} by explicitly specifying the nonnegative constraint to allow the definition of another stationary equation using the KKT conditions. Since nonnegativity constraints are familiar to NMF, we refer to this maximum entropy optimization problem as "constrained NMF with entropic regularization ($\texttt{cNMF}_\texttt{H}$)".

4.1.4 Constrained NMF and Discrete Entropic regularization

The problem of $\texttt{cNMF}_\texttt{H}$ is expressed as follows :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0, \boldsymbol{Z}\mathbf{1}_g = \mathbf{1}_n} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = -H(\boldsymbol{Z}) + \lambda D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)\}. \tag{4.16}$$

For distinction, the probability simplex constraint ($\boldsymbol{Z}^\top \in (\Delta_g)^n$) is now expressed using two constraints : the nonnegativity constraint ($\boldsymbol{Z} \geq 0$) and the summation to unity constraint ($\boldsymbol{Z}\mathbf{1}_g = \mathbf{1}_n$). Solving (4.16) w.r.t. $\boldsymbol{W}$ leads to the same update rules as in problem (4.4) and the original $\texttt{NMF}-\texttt{KL}$. Therefore, we only derive the solution of $\boldsymbol{Z}$. The associated Lagrangian to this problem is defined as follows :

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) + \sum_i^n \gamma_i \Big(\sum_k^g z_{ik} - 1 \Big) - \mathrm{Tr}(\boldsymbol{\epsilon}\boldsymbol{Z}^\top) - \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \tag{4.17}$$

where $\boldsymbol{\gamma} \in \mathbb{R}_+^n$, $\boldsymbol{\epsilon} \in \mathbb{R}_+^{n\times g}$, and $\boldsymbol{\beta} \in \mathbb{R}_+^{d\times g}$ are the Lagrange multipliers. In the following, we define the Lagrangian multipliers γ_i in terms of their respective positive and negative orthant as follows

$\gamma_i = [\gamma_i]^+ - [\gamma_i]^-$ where $[\gamma_i]^+ \geq 0$ and $[\gamma_i]^- \geq 0$. The gradient w.r.t. z_{ik} is defined as follows :

$$\nabla_{z_{ik}} \mathcal{L} = 1 + \log z_{ik} - \lambda \Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W} \Big)_{ik} + \lambda \sum_j^d w_{jk} + [\gamma_i]^+ - [\gamma_i]^- - \epsilon_{ik}. \tag{4.18}$$

Setting this gradient to zero and making use of the KKT conditions $\boldsymbol{\epsilon} \odot \boldsymbol{Z} = 0$ leads to the following stationary equation :

$$z_{ik} \left[1 + \log z_{ik} - \lambda \Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W} \Big)_{ik} + \lambda \sum_j^d w_{jk} + [\gamma_i]^+ - [\gamma_i]^- \right] = 0. \tag{4.19}$$

Several multiplicative update rules can be derived from this equation. The first takes into account the negativity of $\log z_{ik}$ (since $z_{ik} \in [0, 1]$) :

$$z_{ik} \leftarrow z_{ik} \frac{\lambda (\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W})_{ik} + [\gamma_i]^-}{\lambda \sum_j^d w_{jk} + 1 + \log z_{ik} + [\gamma_i]^+}. \tag{4.20}$$

Substituting Eq(4.20) into $\nabla_{\gamma_i} \mathcal{L}$ gives :

$$[\gamma_i]^- = \frac{1 - \sum_k^g z_{ik} \frac{\lambda \left(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W} \right)_{ik}}{B_{ik}}}{\sum_k^g \frac{z_{ik}}{B_{ik}}}, \tag{4.21}$$

where $B_{ik} = \lambda \sum_j^d w_{jk} + 1 + \log z_{ik} + [\gamma_i]^+$. From Eq(4.21), we have that $[\gamma_i]^-$ depends on $[\gamma_i]^+$. Using the numerator of Eq(4.21), we can derive the conditional value of $[\gamma_i]^+$ as follows :

$$1 - \sum_k^g z_{ik} \frac{\lambda (\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W})_{ik}}{B_{ik}} = \sum_k^g z_{ik} \frac{\nabla_{z_{ik}} \mathcal{F} + [\gamma_i]^+}{B_{ik}}.$$

From this equality, $[\gamma_i]^+ = \max(\max(-\nabla_{z_{ik}} \mathcal{F} | k = 1, \ldots, g), 0)$ since $[\gamma_i]^+ \geq 0$. From the expression of $[\gamma_i]^+$, the following inequality guarantees B_{ik} to be positive. If $\lambda \sum_j^d w_{jk} + 1 + \log z_{ik} \leq 0$, we have :

$$-\lambda \Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W} \Big)_{ik} + \lambda \sum_j^d w_{jk} + 1 + \log z_{ik} \leq 0$$

$$\nabla_{z_{ik}} \mathcal{F} \leq 0.$$

As a consequence $\lambda \sum_j^d w_{jk} + 1 + \log z_{ik} + \max(\max(-\nabla_{z_{ik}} \mathcal{F} | k = 1, \ldots, g), 0) \geq 0$. From the stationary Eq (4.19), another multiplicative update rule can also be written s.t. $\log z_{ik}$ appears in the numerator such as :

$$z_{ik} \leftarrow z_{ik} \frac{\lambda (\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top} \boldsymbol{W})_{ik} - \log z_{ik} + [\gamma_i]^-}{\lambda \sum_j^d w_{jk} + 1 + [\gamma_i]^+}. \tag{4.22}$$

Deriving $[\gamma_i]^-$ by substituting Eq(4.22) into $\nabla_{\gamma_i}\mathcal{L}$ gives :

$$[\gamma_i]^- = \frac{1 - \sum_k^g z_{ik} \frac{\lambda\left(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\right)_{ik} - \log z_{ik}}{B_{ik}}}{\sum_k^g \frac{z_{ik}}{B_{ik}}}, \tag{4.23}$$

where $B_{ik} = \lambda \sum_j^d w_{jk} + 1 + [\gamma_i]^+$. Deriving the conditional value of $[\gamma_i]^+$ from the numerator also leads to $[\gamma_i]^+ = \max(\max(-\nabla_{z_{ik}}\mathcal{F} | k = 1, \ldots, g), 0)$. The positivity of this expression is straightforward since $-\log z_{ik}$ is positive. In practice, we observed similar performance in terms of clustering from various local minima. However, we found that Eq (4.20) is more subject to numerical overflow, especially on small datasets and therefore prefer to use Eq (4.22). The optimization procedure is given by Algorithm (16). Note that to guarantee the convergence to a stationary point, we use a modified version of the multiplicative updates rules proposed by Chi and Kolda [144] which prevents inadmissible zeros that do not satisfies the KKT conditions. Therefore, zeros entries in $\boldsymbol{Z}$ and $\boldsymbol{W}$ are monitored and replaced by a small constant if their partial derivatives are negative (respectively).

Algorithm 16 `cNMF`$_{\mathtt{H}}$

Input : $\boldsymbol{X}$, g, λ, $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
1. update $\boldsymbol{Z}$ with Eq(4.22) ;
2. update $\boldsymbol{W}$ with Eq(4.14) ;

until convergence

4.1.5 Uncertainty and clustering validity

Thank to the new expression of z_{ik} in `cNMF`$_{\mathtt{H}}$, maximum entropy distributions subject to several values of λ can now be produced without reaching numerical overflow. In the following, we conduct an empirical setting of λ regarding several clustering scores. A small real-word dataset is considered for several illustrations (CSTR : $\boldsymbol{X} \in \mathbb{R}_+^{475\times1000}$, #clusters $g = 4$, % of non-zero observations : 3.40).

Here, and in the following sections, the quality of a clustering is assessed using two measures widely used for quantifying the correspondence between the clustering and the true partition. The Normalized Mutual Information (NMI) [119], which measures the mutual dependency between two random variables, and, the Adjusted Rand Index (ARI) [122], which measures the degree of agreement between two partitions.

Figure 4.1 displays the averages of the min-max normalization of $H(\boldsymbol{Z})$, the NMI and the ARI, for

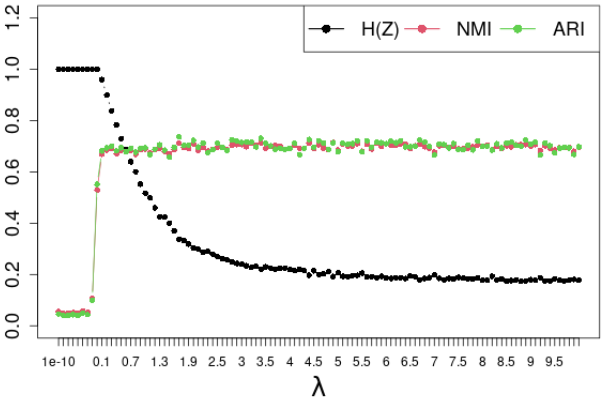
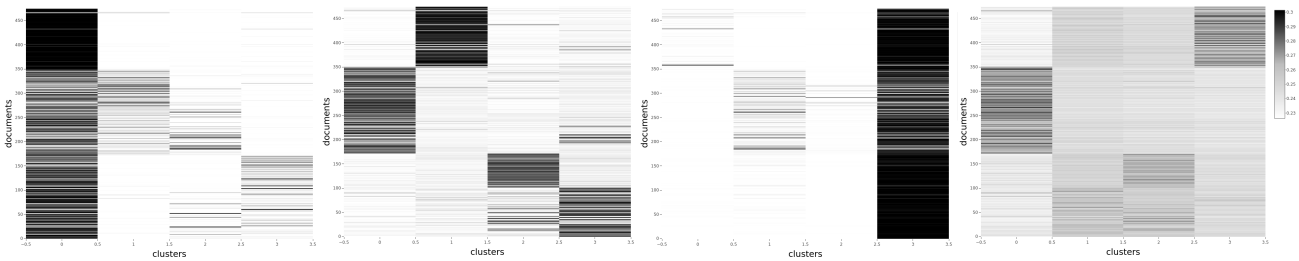


FIGURE 4.1 – $H(\boldsymbol{Z})$ (min-max normalized) variations according to $\lambda \in [10^{-10}, 10]$. Note that $\min(H(\boldsymbol{Z})) := \inf(H(\boldsymbol{Z}))$, $\forall \boldsymbol{Z}^\top \in (\Delta_g)^n$ in the min-max function.

λ going from 10^{-10} to 10. The averages are computed over 30 epochs. As expected, $H(\boldsymbol{Z})$ decreases when λ increases which leads to more clustering validity. However, the NMI and ARI scores remain similar alongside the variations of λ, resulting in no additional performance using local minima with higher clustering validity. Since λ is not a solution of the constraint equation $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) = S$ and is a parameter for all the partial derivatives $\partial\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}$, $\forall k = 1, \ldots, g$, $\forall i = 1, \ldots, n$, this behavior confirms that λ acts simply as a normalization parameter for increasing or decreasing $H(\boldsymbol{Z})$ based on the discrepancy between the partial derivatives. In this case, $D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)$ leads to enough difference such that $H(\boldsymbol{Z})$ varies significantly for $\lambda \in [10^{-10}, 10]$. However, as long as λ varies from values distant from 0, the amount of uncertainty will not disturb the partial derivatives order (before and after discretization) and therefore the underlying hard clustering. Based upon these remarks, λ was set to 1. $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top) = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2$ is an example where λ values with several orders of



(a) I-divergence, $\boldsymbol{X}$ (b) I-divergence, $\hat{\boldsymbol{X}}$ (c) SSQ, $\boldsymbol{X}$ (d) SSQ, $\hat{\boldsymbol{X}}$

FIGURE 4.2 – Heatmap of $\boldsymbol{Z}$ obtained from $\texttt{cNMF}_\texttt{H}$. $\hat{\boldsymbol{X}} \in \mathbb{S}^{d-1}$.

magnitude higher might be required to impact $H(\boldsymbol{Z})$. In the case of the I-divergence or the SSQ, this

behavior can be described using the underlying probability distributions, namely the Poisson and the Normal.

In fact, since $\boldsymbol{Z}^\top \in (\Delta_g)^n$, a surrogate of the respective likelihood obtained from these two distributions can be expressed using these Jensen inequalities defined on the convex negative logarithm and power functions arising in $D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)$ and $\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2$ respectively :

$$-\log\Big(\sum_k^g z_{ik}w_{jk}\Big) \leq -\sum_k^g z_{ik}\log w_{jk}, \tag{4.24}$$

resulting in $D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) \leq \sum_{i,k}^{n,g} z_{ik}D_I(\boldsymbol{x}_i||\boldsymbol{w}_k)$, and

$$\Big(\sum_k^g z_{ik}w_{jk}\Big)^2 \leq \sum_k^g z_{ik}w_{jk}^2, \tag{4.25}$$

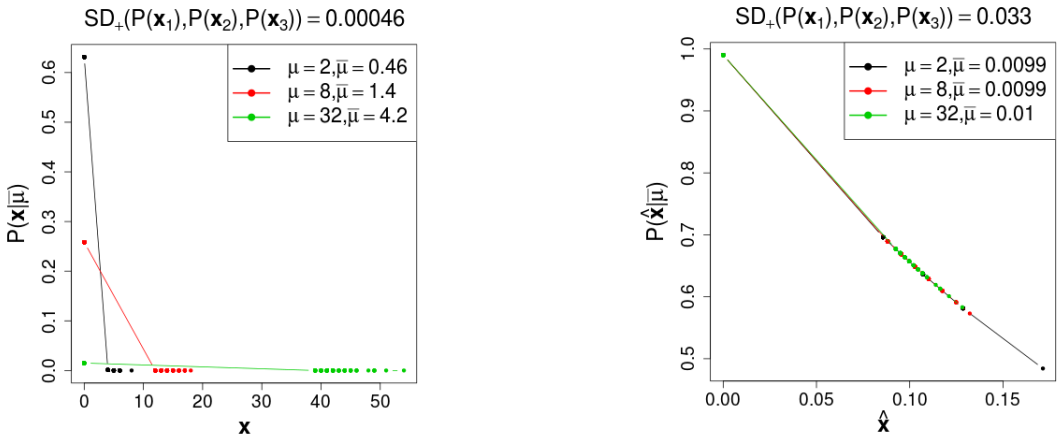
resulting in $\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^\top\|_F^2 \leq \sum_{i,k}^{n,g} z_{ik}\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{w}_k\|_F^2$.

Additionally, the maximum entropy distributions subject to these surrogates will effectively leads to discretization in the g-dimensional space of the Poisson and Normal likelihoods up to the normalizing constant λ. As a consequence, $\boldsymbol{Z}$ obtained from problems (4.4) or (16), is therefore a set of probability distributions defined in the neighborhood of the discretized likelihood s.t. :

$$e^{-\lambda\partial\mathcal{D}(\boldsymbol{X},\boldsymbol{Z}\boldsymbol{W}^\top)/\partial z_{ik}} \equiv e^{-\lambda\mathcal{D}(\boldsymbol{x}_i,\boldsymbol{w}_k)} \propto e^{-\lambda f(\boldsymbol{x}_i;\boldsymbol{w}_k)}, \tag{4.26}$$

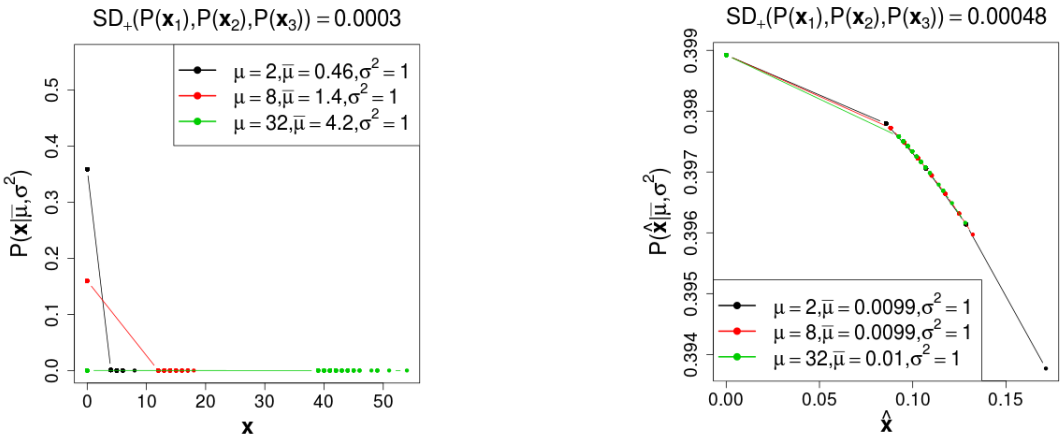
where $f(\boldsymbol{x}_i;\boldsymbol{w}_k)$ is the probability distribution of $\boldsymbol{x}_i$. By way of illustration, we display in Figure 4.3-4.4 the Poisson pmf and the Normal pdf for 3 random Poisson variables in $\mathbb{R}_+^{1000}$, namely $\mathbf{x}_1 \sim \mathcal{P}(\mu = 2), \mathbf{x}_2 \sim \mathcal{P}(\mu = 8)$, and $\mathbf{x}_3 \sim \mathcal{P}(\mu = 32)$, where 900 observations are set to 0. In both Figures, the subfigure (a) displays the probabilities of the original variables while (b) displays the probabilities of their normalization given the L_2-norm. The discrepancy between the probabilities $(P(\mathbf{x}_1), P(\mathbf{x}_2), P(\mathbf{x}_3))$ is measured using the standard deviation (SD). Note that due to the excess of zeros, SD is computed between probabilities of non-zero observations and denoted SD_+.

As shown in Figures 4.3(a) and 4.4(a), the discrepancy between the probabilities of non-zero observations is small. In addition, a substantial gap is observed between those probabilities and the probability of null observations. However, as shown by Figures 4.3(b) and 4.4(b), normalizing the random variables substantially diminishes the impact of the sparsity. For Poisson, more discrepancy is now observed between the probability of non-zero observations, whilst for the Normal distribution,



(a) $\mathbf{x}$, $n_z = 90\%$ (b) $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$, $n_z = 90\%$

FIGURE 4.3 – Poisson probability mass functions, $\bar{\mu}$ designate the mean estimate.



(a) $\mathbf{x}$, $n_z = 90\%$ (b) $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$, $n_z = 90\%$

FIGURE 4.4 – Normal probability density functions, $\bar{\mu}$ designate the mean estimate.

the gap between probabilities of non-zeros elements and zero elements is substantially reduced. Furthermore, we illustrate these behaviors in terms of clustering and display the heatmap of $\boldsymbol{Z}$ obtained from $\texttt{cNMF}_\texttt{H}$ where $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$ is set as the I-divergence or the SSQ, whether $\boldsymbol{X} = [\boldsymbol{x}_1|\ldots|\boldsymbol{x}_n]^\top$ is normalized in the unit-sphere $\mathbb{S}^{d-1} = \{\boldsymbol{x}_i \in \mathbb{R}^d : \|\boldsymbol{x}_i\| = 1\}$ (denoted by $\hat{\boldsymbol{X}}$) or not (denoted by $\boldsymbol{X}$).

We can clearly see that the behavior expressed in Figures 4.3(a) and 4.4(a) leads to solutions which favor one unique cluster (see Figures 4.2(a) and 4.2(c)) ; on the other hand, the behavior described in Figure 4.4(b) leads to solution with maximum entropy despite higher value of λ (see Figure 4.2(d)) ; finally, the behavior described by Figure 4.3(b) leads to the best distinction between the samples and

the best clustering.

Moreover, to emphasize the interest around the maximization of $H(\boldsymbol{Z})$, we consider a generalization of Shannon's entropy and others referred to as the Rényi entropy [238] and stated as :

$$H_\alpha(\boldsymbol{z}) = \frac{1}{1-\alpha}\log\Big(\sum_{k=1}^{g} z_k^\alpha\Big), \tag{4.27}$$

where $\alpha = 0$ gives the Hartley entropy ; $\alpha \to 1$ the Shannon entropy ; $\alpha = 2$ the collision entropy (also know as the Rényi entropy) ; $\alpha \to \infty$ the Min-entropy. We refer to the proposed method as $\texttt{cNMF}_{H_\alpha}$. Derivation of the algorithm w.r.t. the Rényi entropy is only achieved for $\alpha \in \mathbb{R}_+/]0,1[$ and without λ (since $\lambda := 1$). $\texttt{cNMF}_{H_\alpha}$ is defined as the following non-linear optimization problem :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0, \boldsymbol{Z}\mathbf{1}_g = \mathbf{1}_n} \{\mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) = -H_\alpha(\boldsymbol{Z}) + D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)\}, \tag{4.28}$$

where $H_\alpha(\boldsymbol{Z}) = \sum_i^n \frac{1}{1-\alpha}\log\Big(\sum_k^g z_{ik}^\alpha\Big)$. Further deductions will also highlight that setting $\alpha = 0$ collapses this problem to $\texttt{NMF}-\texttt{KL}$ subject to the probability constraint and nonnegativity constraint without maximization of $H(\boldsymbol{Z})$. We refer to this case as $\texttt{cNMF}$ or $\texttt{cNMF}_{H_0}$:

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0, \boldsymbol{Z}\mathbf{1}_g = \mathbf{1}_n} D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top). \tag{4.29}$$

Moreover, for generalization purposes, we might refer to $\texttt{cNMF}_\texttt{H}$ as $\texttt{cNMF}_{H_1}$. The derivation of $\texttt{cNMF}_{H_\alpha}$ is completed in section 4.1.9.

4.1.6 Jensen upper bound

First, we shall notice that the Jensen inequality given by (4.24) provides a surrogate for $D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)$ that can be regarded as a clustering criterion. Furthermore, (4.24) takes equality when $\boldsymbol{Z}$ becomes a classification matrix s.t. $\boldsymbol{Z} \in [0,1]^{n\times g}$ and $\sum_k^g z_{ik} = 1, \forall i = 1,\ldots,n$. In order to improve the clustering ability of our method, we choose to minimize the Jensen upper bound (denoted $\mathcal{Q}(\boldsymbol{Z},\boldsymbol{W})$) of $\mathcal{F}(\boldsymbol{Z},\boldsymbol{W})$ in problem (4.16) w.r.t w_{jk}. $\mathcal{Q}(\boldsymbol{Z},\boldsymbol{W})$ is expressed as :

$$\begin{aligned}\mathcal{Q}(\boldsymbol{Z},\boldsymbol{W}) =& -H(\boldsymbol{Z}) + \lambda\sum_{i,k}^{n,g} z_{ik} D_I(\boldsymbol{x}_i||\boldsymbol{w}_k)\\ =& -H(\boldsymbol{Z}) + \lambda\Bigg(\sum_{i,j}^{n,d}\Big[x_{ij}\log x_{ij} - x_{ij} + [\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}\Big] - \sum_{i,k,j}^{n,g,d} z_{ik}x_{ij}\log w_{jk}\Bigg).\end{aligned} \tag{4.30}$$

Naturally, the logarithm forces $\boldsymbol{W}$ to be positive or null. Differentiation w.r.t. w_{jk} leads therefore to :

$$\nabla_{w_{jk}}\mathcal{L} = -\lambda\Big(\frac{1}{w_{jk}}(\boldsymbol{X}^\top\boldsymbol{Z})_{jk}\Big) + \lambda\sum_i^n z_{ik}. \tag{4.31}$$

Setting this derivative to zero yields the following estimate for w_{jk} :

$$w_{jk} = \frac{(\boldsymbol{X}^\top\boldsymbol{Z})_{jk}}{\sum_i^n z_{ik}}. \tag{4.32}$$

In section 4.1.8, $\boldsymbol{W}$ in Algorithms 16 and 17 is updated using Eq(4.32).

4.1.7 Convergence analysis

To show the convergence of $\mathcal{F}(\boldsymbol{Z},\boldsymbol{W})$ under the update formulas (4.20) and (4.22), we use a similar approach to the one used in [132, 239] and inspired by the Expectation-Maximization (EM) algorithm which involves auxiliary functions. Thereby, the convergence is proved in relying on the following Theorem 4.1.2. Note that the multiplicative updates rules for $\boldsymbol{Z}$ in $\texttt{cNMF}_\texttt{H}$ and $\texttt{cNMF}_{H_\alpha}$ are conditioned on the Lagrange multipliers, so the convergence is analyzed for the Lagrangian functions.

Theorem 4.1.1. *$\mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) + \sum_i^n \gamma_i\Big(\sum_k^g z_{ik} - 1\Big)$ in problem (4.16) is non-increasing under the update formulas (4.20) and (4.14).*

Theorem 4.1.2. *According to α, we have*

1. *Let $\alpha \in]1,\infty[$, $\mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) + \sum_i^n \gamma_i\Big(\sum_k^g z_{ik} - 1\Big)$ in problem (4.28) is non-increasing under Eq(4.61) and Eq(4.14).*
2. *Let $\alpha = 0$, $D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)$ in problem (4.29) is non-increasing under Eq(4.61) and Eq(4.14)*

Let $\Delta_g = \{\forall \boldsymbol{z} \in \mathbb{R}_+^g : \sum_k^g z_k = 1\}$ be a probability simplex, $\boldsymbol{e_k} \in \Delta_g$ be a binary vector that only the kth component is equal to 1, and $G = (1/g,\ldots,1/g)^\top \in \Delta_g$ denote the uniform vector in which all component are equal to $\frac{1}{g}$.

Definition 4.1.1. Let $(\boldsymbol{z},\boldsymbol{z}') \subseteq \Delta_g \times \Delta_g$, $\mathcal{H}(\boldsymbol{z},\boldsymbol{z}')$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$ if the following conditions are satisfied :

$$\forall \boldsymbol{z} \neq \boldsymbol{z}', \mathcal{H}(\boldsymbol{z},\boldsymbol{z}') \geq \mathcal{L}(\boldsymbol{z}) \quad \text{and} \quad \mathcal{H}(\boldsymbol{z},\boldsymbol{z}) = \mathcal{L}(\boldsymbol{z}).$$

A key point to the auxiliary function is the following lemma :

Lemma 4.1.3. *If $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$, $\mathcal{L}(\boldsymbol{z})$ is non-increasing under the update*

$$\boldsymbol{z}^{(t+1)} = \arg\min_{\boldsymbol{z}} \mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$$

Proof. $\mathcal{L}(\boldsymbol{z}^{(t+1)}) \leq \mathcal{H}(\boldsymbol{z}^{(t+1)}, \boldsymbol{z}^{(t)}) \leq \mathcal{H}(\boldsymbol{z}^{(t)}, \boldsymbol{z}^{(t)}) = \mathcal{L}(\boldsymbol{z}^{(t)})$. □

In the following, we rewrite $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta})$ in a vector coordinates format.

Let

$$\begin{cases} \mathcal{L}_1(\boldsymbol{z}) \stackrel{def}{=} \sum_{j}^{d} \left(x_j \log \frac{x_j}{\sum_k^g z_k w_{jk}} - x_j\right) + \sum_{j,k}^{d,g} z_k w_{jk} + [\gamma]^+ \Big(\sum_k^g z_k - 1\Big), & (4.33a) \\ \mathcal{L}_2(\boldsymbol{z}) \stackrel{def}{=} -[\gamma]^- \Big(\sum_k^g z_k - 1\Big). & (4.33b) \end{cases}$$

and

$$\begin{cases} \mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \stackrel{def}{=} \sum_{j}^{d} \left(x_j \log x_j - x_j\right) + \sum_{j,k}^{d,g} z_k w_{jk} - \sum_{j,k}^{d,g} x_j \frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}} \Bigg[\log(z_k w_{jk}) \\ \qquad\qquad - \log\left(\frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}}\right)\Bigg] + [\gamma]^+ \Big(\sum_k^g z_k - 1\Big), & (4.34a) \\ \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \stackrel{def}{=} \mathcal{L}_2(\boldsymbol{z}^{(t)}) + \nabla_{\boldsymbol{z}^{(t)}} \mathcal{L}_2 \Big(\sum_k^g z_k^{(t)} \log z_k - \sum_k^g z_k^{(t)} \log z_k^{(t)}\Big). & (4.34b) \end{cases}$$

Lemma 4.1.4. $\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_1(\boldsymbol{z})$

Proof. From $-\log\left(\sum_k^g z_k w_{jk}\right) = -\log\left(\sum_k^g \mu_k \frac{z_k w_{jk}}{\mu_k}\right)$, we use the convexity of the negative logarithm to derive the following inequality :

$$-\log\Big(\sum_k^g \mu_k \frac{z_k w_{jk}}{\mu_k}\Big) \stackrel{Jensen}{\leq} -\sum_k^g \mu_k \log\Big(\frac{z_k w_{jk}}{\mu_k}\Big), \tag{4.35}$$

where $\mu_k = \frac{z_k^{(t)} w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}}$ and $\sum_k^g \mu_k = 1$. From the inequality (4.35), it follows that $\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_1(\boldsymbol{z})$. □

Lemma 4.1.5. $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_2(\boldsymbol{z})$

Proof. Since $\boldsymbol{z}$ is a probability distribution, we derive the following inequality :

$$\sum_k^g z_k^{(t)} \log z_k^{(t)} \stackrel{Gibbs}{\geq} \sum_k^g z_k^{(t)} \log z_k. \tag{4.36}$$

Consequently, $\sum_k^g z_k^{(t)} \log z_k - \sum_k^g z_k^{(t)} \log z_k^{(t)} \leq 0$. Since $[\gamma]^- \geq 0$,

$$-[\gamma]^-\Big(\sum_k^g z_k^{(t)} \log z_k - \sum_k^g z_k^{(t)} \log z_k^{(t)}\Big) \geq 0. \tag{4.37}$$

Since $\mathcal{L}_2(\boldsymbol{z}^{(t)}) = \mathcal{L}_2(\boldsymbol{z})$, using expression (4.37) leads to :

$$\mathcal{L}_2(\boldsymbol{z}^{(t)}) - [\gamma]^-\Big(\sum_k^g z_k^{(t)} \log z_k - \sum_k^g z_k^{(t)} \log z_k^{(t)}\Big) \geq \mathcal{L}_2(\boldsymbol{z}), \tag{4.38}$$

and therefore : $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_2(\boldsymbol{z})$. □

4.1.7.1 Convergence for $\alpha = 1$ ($\texttt{cNMF}_{H_1}$)

Let $\mathcal{L}(\boldsymbol{z}) = \mathcal{L}_1(\boldsymbol{z}) + \mathcal{L}_2(\boldsymbol{z}) - \mathcal{L}_3(\boldsymbol{z})$ where $\mathcal{L}_1(\boldsymbol{z})$ and $\mathcal{L}_2(\boldsymbol{z})$ remain unchanged and

$$\mathcal{L}_3(\boldsymbol{z}) \stackrel{def}{=} -\sum_k^g z_k \log z_k = H(\boldsymbol{z}). \tag{4.39}$$

Proposition 4.1.1. $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) - \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$ where

$$\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = -\sum_k^g \Big[\log z_k (z_k^{(t)} \log z_k^{(t)}) + z_k\Big], \tag{4.40}$$

and $\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)})$, $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ remain unchanged.

Lemma 4.1.6. $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq -\mathcal{L}_3(\boldsymbol{z})$

Proof. Since $z_k^{(t)}$ and z_k are in $[0, 1]$, $\log z_k^{(t)} \leq 0 \leq z_k$ and

$$\begin{aligned}
z_k^{(t)} \log z_k^{(t)} &\leq z_k \\
\log z_k (z_k^{(t)} \log z_k^{(t)}) &\geq z_k \log z_k \\
\sum_k^g \log z_k (z_k^{(t)} \log z_k^{(t)}) &\geq \sum_k^g z_k \log z_k \\
\sum_k^g \Big[\log z_k (z_k^{(t)} \log z_k^{(t)}) + z_k\Big] &\geq \sum_k^g z_k \log z_k \\
-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) &\geq -\mathcal{L}_3(\boldsymbol{z}).
\end{aligned}$$

□

Proof of proposition 4.1.1. From Lemmas 4.1.4, 4.1.5 and 4.1.6, $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$. ■

Proof of Theorem 4.1.1. To satisfy Lemma 4.1.3, we compute the gradient of $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ w.r.t. z_k :

$$\nabla_{z_k}\mathcal{H} = -\frac{1}{z_k} z_k^{(t)} \Big[\sum_j^d x_j \frac{w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}} - \log z_k^{(t)} + [\gamma]^- \Big] + \sum_j^d w_{jk} + [\gamma]^+ + 1. \tag{4.41}$$

Setting this gradient to zero leads to :

$$z_k^{(t+1)} = z_k^{(t)} \frac{\sum_j^d \frac{x_j}{\sum_\ell^g z_\ell^{(t)} w_{jl}} w_{jk} - \log z_k^{(t)} + [\gamma]^-}{\sum_j^d w_{jk} + 1 + [\gamma]^+}. \tag{4.42}$$

Since $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$, $\mathcal{L}(\boldsymbol{z})$ is non-increasing under this update. Rewritten in a matrix coordinates format, Eq(4.42) is similar to the update given by Eq(4.22).

■

Proposition 4.1.2. $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) - \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$ where

$$\begin{aligned} \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) =& \mathcal{L}_3(\boldsymbol{z}^{(t)}) + \nabla_{\boldsymbol{z}^{(t)}}^\top \mathcal{L}_3(\boldsymbol{z} - \boldsymbol{z}^{(t)}) \\ =& 2\mathcal{L}_3(\boldsymbol{z}^{(t)}) + \sum_k^g z_k \log z_k^{(t)}, \end{aligned} \tag{4.43}$$

and $\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)})$, $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ remain unchanged.

Lemma 4.1.7. $\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_3(\boldsymbol{z})$ *if* $\mathcal{L}_3(\boldsymbol{z}^{(t)}) \geq \mathcal{L}_3(\boldsymbol{z})$.

Proof. $\mathcal{L}_3(\boldsymbol{z}) = H(\boldsymbol{z})$ is Schur-concave and $\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is the first-order Taylor approximation of $\mathcal{L}_3(\boldsymbol{z})$. Therefore, using the property that the tangent to any point is an upper bound of a concave function, $\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_3(\boldsymbol{z})$. □

$\mathcal{L}(\boldsymbol{z})$ is convex in $\boldsymbol{z}$, however, the problem is not jointly convex in $(\boldsymbol{z}, \boldsymbol{W})$. As a consequence, the convergence of $\mathcal{L}(\boldsymbol{z}^{(t)})$ s.t. $\mathcal{L}(\boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z}^{(t+1)}) \geq \mathcal{L}(\boldsymbol{z}^{(t+2)}) \geq \ldots \geq \mathcal{L}(\boldsymbol{z}^{(\infty)})$ does not yields the same requirements for $\mathcal{L}_3(\boldsymbol{z}^{(t)})$. No strict order is required for the series $\mathcal{L}_3(\boldsymbol{z}^{(t)})$, $\mathcal{L}_3(\boldsymbol{z}^{(t+1)})$, $\mathcal{L}_3(\boldsymbol{z}^{(t+2)})$, $\ldots$, $\mathcal{L}_3(\boldsymbol{z}^{(\infty)})$ even though $\mathcal{L}_3(\boldsymbol{z})$ is maximized overall. For instance, assuming that $\boldsymbol{z}^{(0)} := G$ is set such that $H(\boldsymbol{z}) = \sup(H(.))$. Since λ is set manually to 1 and therefore not a solution of the active constraint equation $\mathcal{D}(\boldsymbol{x}, \boldsymbol{z}\boldsymbol{W}^\top) = S$. Since no constraint is defined on $H(\boldsymbol{z})$, minimizing $-H(\boldsymbol{z}) + \lambda\mathcal{D}(\boldsymbol{x}, \boldsymbol{z}\boldsymbol{W}^\top)$ w.r.t. z_k cannot increase $H(\boldsymbol{z})$, due to the Schur-concavity. Therefore, $H(\boldsymbol{z}^{(0)}) \geq H(\boldsymbol{z}^{(1)})$.

The following corollary is a one case scenario that will partially ensure that $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$ when $H(\boldsymbol{z})$ increases or $\mathcal{L}_3(\boldsymbol{z}) \geq \mathcal{L}_3(\boldsymbol{z}^{(t)})$.

Corollary 4.1.7.1. *Let* $(\boldsymbol{z}, \boldsymbol{z}^{'}) \in \Delta_g \times \Delta_g$*, we have :*

$$-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{'}) + \mathcal{L}_3(\boldsymbol{z}) \geq 0 \text{ if } \mathcal{L}_3(\boldsymbol{z}^{(t)}) \leq \mathcal{L}_3(\boldsymbol{z}).$$

Proof. We have

$$\begin{aligned} -2\mathcal{L}_3(\boldsymbol{z}^{(t)}) &\geq -2\mathcal{L}_3(\boldsymbol{z}) \\ -2\mathcal{L}_3(\boldsymbol{z}^{(t)}) + 2\mathcal{L}_3(\boldsymbol{z}) &\geq 0. \end{aligned} \tag{4.44}$$

Using the Gibbs inequality, we have :

$$\mathcal{L}_3(\boldsymbol{z}) = -\sum_k^g z_k \log z_k \leq -\sum_k^g z_k \log z_k^{(t)}. \tag{4.45}$$

Therefore :

$$\begin{aligned} \mathcal{L}_3(\boldsymbol{z}) - 2\mathcal{L}_3(\boldsymbol{z}^{(t)}) - \sum_k^g z_k \log z_k^{(t)} &\geq 2\mathcal{L}_3(\boldsymbol{z}) - 2\mathcal{L}_3(\boldsymbol{z}^{(t)}), \\ \mathcal{L}_3(\boldsymbol{z}) - \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) &\geq 2\mathcal{L}_3(\boldsymbol{z}) - 2\mathcal{L}_3(\boldsymbol{z}^{(t)}), \end{aligned} \tag{4.46}$$

From Eq (4.44), we have $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{L}_3(\boldsymbol{z}) \geq 0$ and therefore $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$. □

The overall convergence is guaranteed by the following lemma. For generalization with the following proofs, the demonstration is achieved for $H_\alpha(.)$ since $H(.)$ is a case of $H_\alpha(.)$ when $\alpha to 1$ and both functions have the same extrema.

Lemma 4.1.8. $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$ *when* $\mathcal{L}_3(\boldsymbol{z}^{(t)}) \geq \mathcal{L}_3(\boldsymbol{z})$.

Proof. Since $\sum_k^g z_k = 1$, we have that $\mathcal{L}_2(\boldsymbol{z}) = 0$. A sufficient remaining condition for $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$ is therefore :

$$\left(\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) - \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})\right) - \left(\mathcal{L}_1(\boldsymbol{z}) - \mathcal{L}_3(\boldsymbol{z})\right) \geq 0. \tag{4.47}$$

Based on the Schur-concavity of $H_\alpha(.)$, we show that the condition is respected when $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \inf(-H_\alpha(.))$ is the infimum for $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ while fixing $-\mathcal{L}_3(\boldsymbol{z})$ at its supremum s.t. $-\mathcal{L}_3(\boldsymbol{z}) = \sup(-H_\alpha(.)) = 0$. Therefore, the condition will holds for any values of $\boldsymbol{z} \in \Delta_g$ s.t. $-\mathcal{L}_3(\boldsymbol{z}) \leq$

$\sup(-H_\alpha(.))$ as well as any value of $\boldsymbol{z}^{(t)} \in \Delta_g$ s.t. $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \inf(-H_\alpha(.))$. From Eq(4.34a), we have :

$$\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \sum_j^d x_j \sum_k^g \hat{w}_{jk}[z_k^{(t)} \log z_k^{(t)} - z_k^{(t)} \log z_k] + \hat{\mathcal{L}}_1(\boldsymbol{z}^{(t)}), \tag{4.48}$$

where $\hat{w}_{jk} = \frac{w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}}$ and

$$\hat{\mathcal{L}}_1(\boldsymbol{z}^{(t)}) = \sum_j^d (x_j \log x_j - x_j) - \sum_j^d x_j \log\Big(\sum_\ell^g z_\ell^{(t)} w_{jl}\Big) + \sum_{j,k}^{d,g} z_k w_{jk} + [\gamma]^+\Big(\sum_k^g z_k - 1\Big).$$

Let $\boldsymbol{z} = \boldsymbol{e_k}$ s.t. $-\mathcal{L}_3(\boldsymbol{z}) = \sup(-H(.)) = 0$ and $\boldsymbol{z}^{(t)} = G$ s.t. $-\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = -\log(g)$. The difference between $\mathcal{H}_1 - \mathcal{H}_3 + \mathcal{H}_2$ and $\mathcal{L}_1 - \mathcal{L}_3$ boils down to the following quantity :

$$\begin{aligned}\Omega = &\sum_j^d x_j \sum_k^g \hat{w}_{jk}[z_k^{(t)} \log z_k^{(t)} - z_k^{(t)} \log z_k] + \log(g) \sum_j^d x_j \\ &- \sum_j^d x_j \log\left(\frac{w_{jk}|\{z_k = 1\}}{\sum_\ell^g w_{jl}}\right) - \log(g) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}),\end{aligned} \tag{4.49}$$

where $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq 0$, $\hat{w}_{jk} \geq 0, \forall k = 1, \ldots, g$ and $\frac{w_{jk}}{\sum_\ell^g w_{jl}} \in [0, 1]$. Therefore, using the Gibbs inequality and the negativity of the logarithm for value in $[0, 1]$, we obtain the following inequalities :

$$\sum_j^d x_j \sum_k^g \hat{w}_{jk}[z_k^{(t)} \log z_k^{(t)} - z_k^{(t)} \log z_k] \geq 0, \tag{4.50}$$

$$-\sum_j^d x_j \log\left(\frac{w_{jk}|\{z_k = 1\}}{\sum_\ell^g w_{jl}}\right) \geq 0. \tag{4.51}$$

Consequently, $-\log(g)$ is the only negative term in Ω and, is cancelled out whenever $\sum_j^d x_j \geq 1$. In the context of document clustering, this condition holds for any raw document-term matrix or normalized with the L_2-norm.

□

Proof of Proposition 4.1.2. From Lemmas 4.1.4, 4.1.5, 4.1.8, and corollary 4.1.7.1, we have $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$. ■

Proof of Theorem 4.1.1. To satisfy Lemma 4.1.3, we compute the gradient of $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ w.r.t. z_k :

$$\nabla_{z_k}\mathcal{H} = -\frac{1}{z_k} z_k^{(t)}\Big[\sum_j^d x_j \frac{w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}} + [\gamma]^-\Big] + \sum_j^d w_{jk} + [\gamma]^+ + 1 + \log z_k^{(t)}. \tag{4.52}$$

Setting this gradient to zero leads to :

$$z_k^{(t+1)} = z_k^{(t)} \frac{\sum_j^d \frac{x_j}{\sum_\ell^g z_\ell^{(t)} w_{jl}} w_{jk} + [\gamma]^-}{\sum_j^d w_{jk} + 1 + \log z_k^{(t)} + [\gamma]^+}. \tag{4.53}$$

Since $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$, $\mathcal{L}(\boldsymbol{z})$ is non-increasing under this update. Rewritten in a matrix coordinates format, Eq(4.53) is similar to the update given by Eq(4.20).

■

4.1.7.2 Convergence for $\alpha \in]1, \infty[$

Let $\mathcal{L}(\boldsymbol{z}) = \mathcal{L}_1(\boldsymbol{z}) + \mathcal{L}_2(\boldsymbol{z}) - \mathcal{L}_3(\boldsymbol{z})$ where $\mathcal{L}_1(\boldsymbol{z})$ and $\mathcal{L}_2(\boldsymbol{z})$ remain unchanged and

$$\mathcal{L}_3(\boldsymbol{z}) \stackrel{def}{=} \frac{1}{1-\alpha} \log\Big(\sum_k^g z_k^\alpha\Big) = H_\alpha(\boldsymbol{z}). \tag{4.54}$$

Proposition 4.1.3. $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)}) - \mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$ where

$$\begin{aligned}\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \stackrel{def}{=} & \mathcal{L}_3(\boldsymbol{z}^{(t)}) + \nabla_{\boldsymbol{z}^{(t)}}^\top \mathcal{L}_3(\boldsymbol{z} - \boldsymbol{z}^{(t)}) \\ = & \mathcal{L}_3(\boldsymbol{z}^{(t)}) + \sum_k^g \frac{\alpha {z_k^{(t)}}^{\alpha-1}}{(1-\alpha)\sum_{k'}^g {z_{k'}^{(t)}}^\alpha}(z_k - z_k^{(t)}),\end{aligned} \tag{4.55}$$

and $\mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)})$, $\mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ remain unchanged.

Proof of Proposition 4.1.3. As for the Shannon entropy, $H_\alpha(\boldsymbol{z}) = \mathcal{L}_3(\boldsymbol{z})$ is Schur-concave. $\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is the first-order Taylor approximation of $\mathcal{L}_3(\boldsymbol{z})$. Therefore, using the property that the tangent to any point is an upper bound of a concave function, $\mathcal{H}_3(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}_3(\boldsymbol{z})$. Consequently, as previously, $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) \geq \mathcal{L}(\boldsymbol{z})$ results from Lemmas 4.1.4, 4.1.5 and 4.1.8. ■

Proof of Theorem 4.1.2. To satisfy Lemma 4.1.3, we compute the gradient of $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ w.r.t. z_k :

$$\nabla_{z_k}\mathcal{H} = -\frac{1}{z_k} z_k^{(t)} \sum_j^d x_j \frac{w_{jk}}{\sum_\ell^g z_\ell^{(t)} w_{jl}} + \sum_j^d w_{jk} + [\gamma]^+ - \frac{\alpha {z_k^{(t)}}^{\alpha-1}}{(1-\alpha)\sum_{k'}^g {z_{k'}^{(t)}}^\alpha} - \frac{1}{z_k} z_k^{(t)} [\gamma]^-. \tag{4.56}$$

Setting this gradient to zero leads to :

$$z_k^{(t+1)} = z_k^{(t)} \frac{\sum_j^d \frac{x_j}{\sum_\ell^g z_\ell^{(t)} w_{jl}} w_{jk} + [\gamma]^-}{\sum_j^d w_{jk} - \alpha {z_k^{(t)}}^{\alpha-1} [(1-\alpha)\sum_{k'}^g {z_{k'}^{(t)}}^\alpha]^{-1} + [\gamma_i]^+}. \tag{4.57}$$

Since $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$, $\mathcal{L}(\boldsymbol{z})$ is non-increasing under this update. Rewritten in a matrix coordinates format, Eq(4.57) is similar to the update given by Eq(4.61). ■

4.1.7.3 Convergence for $\alpha = 0$ ($\texttt{cNMF}_{H_0}$)

Proof of Theorem 4.1.2. As $\alpha = 0$, $H_\alpha(.)$ boils down to a constant s.t. $\mathcal{L}_3(\boldsymbol{z}) = \log(g)$. From the prior Lemmas 4.1.4 and 4.1.5, the convergence of $\texttt{cNMF}_{H_0}$ is demonstrated by setting $\mathcal{L}(\boldsymbol{z}) = \mathcal{L}_1(\boldsymbol{z}) + \mathcal{L}_2(\boldsymbol{z})$ and $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)}) = \mathcal{H}_1(\boldsymbol{z}, \boldsymbol{z}^{(t)}) + \mathcal{H}_2(\boldsymbol{z}, \boldsymbol{z}^{(t)})$.

■

4.1.7.4 Complexity analysis

The following propositions define the computational complexities scales linearly with the number of entries $n \times d$ in $\boldsymbol{X}$. Futhermore, multiplicative update rules are parallelizable. Matrix transpositions are assumed to be in-place with complexity $\mathcal{O}(1)$.

Proposition 4.1.4. Let t be the number of iterations. The computational complexities of $\texttt{cNMF}_{H_1}$ and $\texttt{cNMF}_{H_\alpha}$ (for $\{0\} \cup \alpha \in]1, \infty[$) remain similar to $\texttt{NMF} - \texttt{KL}$'s, which is $\mathcal{O}(t \cdot (gnd))$.

Proof. $\texttt{cNMF}_{H_1}$ and $\texttt{cNMF}_{H_\alpha}$ have partially the same update rules than $\texttt{NMF} - \texttt{KL}$. The number of operations including multiplications, additions and divisions of Eq (4.61) is $nd \cdot (2g+1) + dg + 5ng + 2n + C$ in $\texttt{cNMF}_{H_\alpha}$ and $nd \cdot (2dg+1) + dg + 2ng + C$ for Eq (4.20) and (4.22) in $\texttt{cNMF}_{H_1}$ where $C = 5ng$ is the number of operations required outside those involving the computations of the gradients. Their complexities are both equal to $\mathcal{O}(gnd)$. The overall complexity of $\texttt{cNMF}_{H_\alpha}$ and $\texttt{cNMF}_{H_1}$ is then $\mathcal{O}(t \cdot (gnd))$. ■

4.1.8 Application on real-world text datasets

4.1.8.1 Datasets

We apply $\texttt{cNMF}_{H_\alpha}$ for $\alpha \in \{0, 1, 2\}$ on 8 bench-marking document-term matrices for which the detailed characteristics are available in Table 4.1. Th term nz indicates the percentage of non-zero scalar and the *balance* coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class. These datasets highlight several varieties of challenging situations such as the amount of clusters, the dimensions, the clusters balance, the degree of overlapping of clusters and the sparsity. We normalized each data matrix with TF-IDF and their respective documents-vectors to unit L_2-norm to remove the bias introduced by their length.

TABLE 4.1 – Datasets description : # denotes the cardinality.

	Characteristics				
Datasets	#Documents	#Words	#Clusters	$nz(\%)$	Balance
NG5	4905	10167	5	0.92	0.943
CLASSIC3	3891	4303	3	1.05	0.707
NG20	18846	14390	20	0.59	0.628
OHSCAL	11162	11465	10	0.53	0.437
CLASSIC4	7095	5896	4	0.59	0.323
LA12	6279	31472	6	0.48	0.282
RCV1	6387	16921	4	0.25	0.080
SPORTS	8580	14870	7	0.86	0.036

4.1.8.2 Empirical results on benchmark datasets

We compare our algorithm against several NMF models acknowledged for improving document clustering with NMF. The list includes : the original NMF with the Frobenius norm (`NMF`) and the I-divergence (`NMF` $-$ `KL`), Orthogonal NMF (`ONMF`) [158], Projective NMF (`PNMF`) [161] and, Graph Regularized NMF (`GNMF`) [163]. A Deep-Learning algorithm namely Deep Clustering Network (`DCN`) [203] is retained. It showed significant improvements for document clustering against several clustering (K-means, Spectral Clustering), NMF based method such as (`LCCF`) [204] and Deep Learning algorithms (e.g. `SAE` [205]). All algorithms with parameters were launched accordingly to the respective settings advocated by their authors.

Each of the algorithms was launched 30 times on every dataset. Among those 30 trials, only the 10 best solutions (ranked according to the criterion) were kept. Table 4.2 displays the NMI and ARI of those subsets of solutions in terms of average and standard deviation (SD). From the results, it is clear that $\texttt{cNMF}_{H_\alpha}$ outperforms by a substantial margin the stat-of-the-arts algorithms. Primarily, we point that $\texttt{cNMF}_{H_0}$ improves slightly over `NMF` $-$ `KL`, with noticeable improvements seen on 5 datasets (NG5, NG20, OHSCAL, CLASSIC4, RCV1). The performance shown by $\texttt{cNMF}_{H_2}$ is less prominent. Using the Collision entropy ($\alpha = 2$) seems to vary the subset of best solutions (much higher standard deviation). However those are not necessarily better at retrieving the original partition for some datasets. Finally, the main advancements come down to $\texttt{cNMF}_{H_1}$. On some datasets, the performance is up by 10 to 20% compare to `NMF` $-$ `KL` or e.g. `DCN` (see NG20, LA12) with an average improvement rate of 7.2%. Furthermore, the lower standard deviations for the subset of best solutions suggest that $H(.)$ improves clustering validity (lack of uncertainty), by shrinking the convergences toward the minima delivering

TABLE 4.2 – NMI and ARI means and standard deviations (SD) over different datasets (Mean±SD).

Datasets	Metrics	NMF	ONMF	PNMF	GNMF	DCN	NMF − KL	cNMF$_{H_0}$	cNMF$_{H_2}$	cNMF$_{H_1}$
NG5	NMI	56±0.0	65±4.0	65±5.0	63±7.0	62±2.8	80±3.3	82±1.4	80±2.5	**87**±.01
	ARI	33±0.0	48±8.0	47±9.0	62±9.0	47±2.7	82±4.3	84±1.4	83±2.8	**90**±.02
CLASSIC3	NMI	49±0.0	58±0.0	71±22	63±6.8	92±4.6	95±0.1	95±.03	95±0.2	**96**±.01
	ARI	44±0.0	55±0.0	70±26	57±9.2	94±4.5	97±0.1	97±.02	97±0.1	**98**±0.0
NG20	NMI	42±0.8	44±2.0	45±2.0	50±1.0	43±1.0	48±2.2	50±0.9	50±1.1	**66**±1.2
	ARI	23±0.8	22±2.0	24±2.0	35±5.0	17±1.5	34±2.2	36±1.4	35±1.6	**56**±2.4
OHSCAL	NMI	38±0.6	37±1.8	39±1.2	38±1.3	35±1.0	35±1.2	36±1.2	34±1.2	**40**±0.3
	ARI	**29**±0.9	28±1.8	**29**±2.0	28±1.6	25±1.9	24±1.5	25±1.4	23±1.4	**29**±0.5
CLASSIC4	NMI	53±0.4	55±9.0	59±5.0	65±4.0	57±1.4	70±2.5	**76**±0.6	54±8.6	**76**±.01
	ARI	45±0.3	39±9.0	44±1.0	49±5.0	42±1.3	64±5.9	66±1.8	41±9.4	**68**±0.0
LA12	NMI	42±1.6	44±2.2	43±3.0	47±2.0	52±3.5	48±3.9	43±3.3	44±3.6	**57**±0.4
	ARI	36±2.8	40±4.1	37±6.0	43±3.0	44±5.6	45±4.4	38±4.3	40±5.0	**54**±0.4
RCV1	NMI	35±0.0	49±2.0	46±4.5	48±4.0	34±0.6	47±2.4	48±0.7	39±6.4	**51**±.09
	ARI	13±0.0	39±4.0	37±5.3	39±3.0	12±0.8	42±2.2	43±1.0	36±7.0	**46**±.05
SPORTS	NMI	55±0.0	55±2.0	56±0.1	55±0.1	59±1.5	55±2.6	54±1.2	55±2.3	**61**±1.5
	ARI	28±0.0	28±1.0	28±0.1	28±0.1	37±3.4	39±2.2	40±2.6	41±3.7	**45**±2.5

the best "hard" clustering partition.

Figure 4.5 displays the Shannon entropy during the respective convergence of cNMF$_{H_0}$ and cNMF$_{H_1}$, whereas the Rényi entropy is displayed for cNMF$_{H_2}$. The same starting values $\boldsymbol{Z}^{(0)}$, and $\boldsymbol{W}^{(0)}$ were used for each method. The results shows that $H(.)$ offers a completely different behavior with cNMF$_{H_1}$ compared to cNMF$_{H_0}$ or $H_\alpha(.)$ in cNMF$_{H_2}$. cNMF$_{H_1}$ seems to escape bad entropy maximum by leveraging the entropy before settling down (see corollary 4.1.7.1 for variations of $H(.)$ during the convergence of cNMF$_{H_1}$) whilst the others tend to directly reduce $H(.)$. cNMF$_{H_0}$ which does not maximize $H(.)$ has a much lower entropy. This behavior could be problematic as using multiplicative update leads to noninterchangeable solution once $z_{ik} = \nabla_{z_{ik}}\mathcal{F} = 0$. Maximizing entropy does reduce cluster validity in $\boldsymbol{Z}$, but eventually also reduces the chance to meet noninterchangeable solutions. In addition, since those problems are non jointly convex, the convergence rate and the set of solutions are drastically limited in case of bad starting points. Therefore, the reshuffling ability of cNMF$_{H_1}$ before converging toward a local minima looks appealing (even if we were to consider setting a higher value for λ). In fact, the set of 30 trials for cNMF$_{H_1}$ was pretty much condensed in terms of solutions. Added up to the

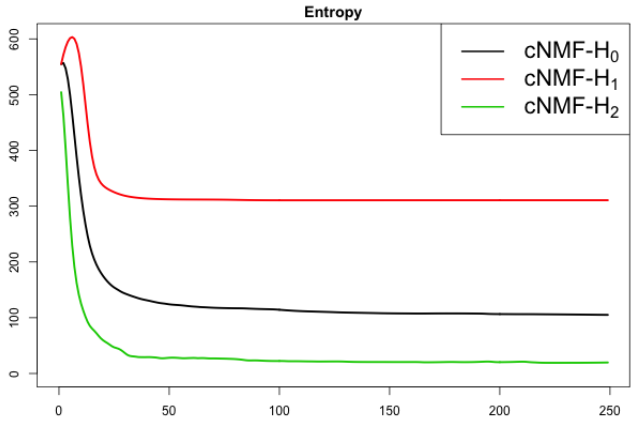


FIGURE 4.5 – Variations of $H(\boldsymbol{Z})$ for $\texttt{cNMF}_{H_0}$ and $\texttt{cNMF}_{H_1}$; and $H_2(\boldsymbol{Z})$ for $\texttt{cNMF}_{H_2}$.

observed behavior, this translates that $\texttt{cNMF}_{H_1}$ does not require extensive initialization.

4.1.9 $\texttt{cNMF}_{H_\alpha}$ algorithm where $\alpha \in \mathbb{R}_+/]0,1[$

The Lagrangian function associated with problem (4.28) is :

$$\mathcal{L}(\boldsymbol{Z},\boldsymbol{W},\boldsymbol{\gamma},\boldsymbol{\epsilon},\boldsymbol{\beta}) = \mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) + \sum_i^n \gamma_i \Big(\sum_k^g z_{ik} - 1 \Big) - \mathrm{Tr}(\boldsymbol{\epsilon}\boldsymbol{Z}^\top) - \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \tag{4.58}$$

where $\boldsymbol{\gamma} \in \mathbb{R}_+^n$, $\boldsymbol{\epsilon} \in \mathbb{R}_+^{n\times g}$, and $\boldsymbol{\beta} \in \mathbb{R}_+^{d\times g}$ are the Lagrange multipliers. In the following, we define the Lagrangian multipliers γ_i in terms of their respective positive and negative orthants as follows $\gamma_i = [\gamma_i]^+ - [\gamma_i]^-$ where $[\gamma_i]^+ \geq 0$ and $[\gamma_i]^- \geq 0$. Differentiation w.r.t. w_{jk} leads to the same update as in problem (4.4). The gradient w.r.t. z_{ik} is given by :

$$\nabla_{z_{ik}}\mathcal{L} = -\Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W} \Big)_{ik} + \sum_j^d w_{jk} - \frac{\alpha z_{ik}^{\alpha-1}}{(1-\alpha)\sum_{k'}^g z_{ik'}^\alpha} + [\gamma_i]^+ - [\gamma_i]^- - \epsilon_{ik}. \tag{4.59}$$

Setting this gradient to zero and making use of the KKT conditions $\boldsymbol{\epsilon} \odot \boldsymbol{Z} = 0$ lead to the following stationary equation :

$$z_{ik}\left[-\Big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W} \Big)_{ik} + \sum_j^d w_{jk} - \frac{\alpha z_{ik}^{\alpha-1}}{(1-\alpha)\sum_{k'}^g z_{ik'}^\alpha} + [\gamma_i]^+ - [\gamma_i]^- \right] = 0. \tag{4.60}$$

From Eq(4.60), we obtain the following multiplicative update rule :

$$z_{ik} \leftarrow z_{ik} \frac{\big(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\big)_{ik} + [\gamma_i]^-}{\sum_j^d w_{jk} - \alpha z_{ik}^{\alpha-1}[(1-\alpha)\sum_{k'}^g z_{ik'}^\alpha]^{-1} + [\gamma_i]^+}, \tag{4.61}$$

Let $B_{ik} = \sum_j^d w_{jk} - \alpha z_{ik}^{\alpha-1}[(1-\alpha)\sum_{k'}^g z_{ik'}^{\alpha}]^{-1} + [\gamma_i]^+$. Substituting Eq(4.61) into the constraint equation gives :

$$[\gamma_i]^- = \frac{1 - \sum_k^g z_{ik} \frac{\left(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\right)_{ik}}{B_{ik}}}{\sum_k^g \frac{z_{ik}}{B_{ik}}}. \tag{4.62}$$

From Eq(4.62), $[\gamma_i]^-$ depends on $[\gamma_i]^+$. Using the numerator of Eq(4.62), we derive the conditional value of $[\gamma_i]^+$:

$$\begin{aligned} 1 - \sum_k^g z_{ik} \frac{\left(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\right)_{ik}}{B_{ik}} &= \sum_k^g z_{ik} \frac{B_{ik}}{B_{ik}} - \sum_k^g z_{ik} \frac{\left(\frac{\boldsymbol{X}}{\boldsymbol{Z}\boldsymbol{W}^\top}\boldsymbol{W}\right)_{ik}}{B_{ik}} \\ &= \sum_k^g z_{ik} \frac{\nabla_{z_{ik}}\mathcal{F} + [\gamma_i]^+}{B_{ik}}. \end{aligned}$$

From this equality, $[\gamma_i]^+ = \max(\max(-\nabla_{z_{ik}}\mathcal{F} | k = 1, \ldots, g), 0)$ since $[\gamma_i]^+ \geq 0$.

Note that for $\alpha \in]0, 1[$, update (4.61) is not guaranteed to be positive which violates the first-order conditions for an optimal local minimum. The gradient $\nabla_{z_{ik}}\mathcal{L}$ involves $z_{ik}^{\alpha-1}$ which leads to negative exponents for $\alpha \in]0, 1[$. For $\alpha = 0$, $\nabla_{z_{ik}}\mathcal{L} = 0$.

Algorithm 17 $\texttt{cNMF}_{H_\alpha}$

Input : $\boldsymbol{X}$, g, $\boldsymbol{Z}^{(0)}$, $\boldsymbol{W}^{(0)}$, $\alpha \in \mathbb{R}_+/]0, 1[$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
repeat
1. update $\boldsymbol{Z}$ with eq(4.61) if $\alpha \in \{0\} \cup]1, \infty[$;
2. update $\boldsymbol{W}$ with eq(4.14) ;

until convergence

4.2 An unified framework for Nonnegative Matrix Factorization and Finite Mixture Models in the unit-sphere

4.2.1 Motivations

Depending on the cost function $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$, NMF can be seen as the minimization of the negative log-likelihood of some continuous or discrete distributions using the GEM algorithm while assuming, for instance, Gaussian, Poisson or Erlang distributions of the x_{ij}'s such that

$$P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Z}\boldsymbol{W}^\top\|_F^2}, \tag{4.63}$$

$$P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}) \propto e^{-D_{KL}(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)} \propto \prod_{i,j}^{n,d} [\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}^{x_{ij}} e^{-[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}}, \tag{4.64}$$

or

$$P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}) \propto e^{-D_{IS}(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top)} \propto \prod_{i,j}^{n,d} \frac{e^{-\frac{x_{ij}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}}}}{[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij}}, \tag{4.65}$$

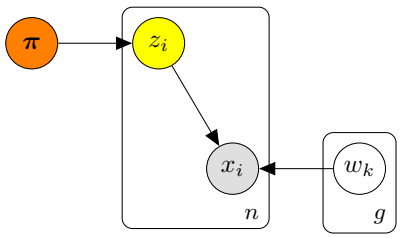
where the NMF factors (from which we deduce a clustering) are set as parameters of the mixture where $\boldsymbol{\Theta} = \{\boldsymbol{Z}, \boldsymbol{W}\}$.

Considering the additivity of Gaussian or Poisson random variables, NMF can also be cast as a statistical composite model [240] which highlights the presence of a third tensor latent variable $\boldsymbol{C} = (c_{ijk}) \in \mathbb{R}_+^{n\times d\times g}$ s.t. $\sum_k^g c_{ijk} = x_{ij}$. By sticking with the ability to deduce a clustering from $\boldsymbol{Z}$, we apply a set of probabilistic constraints and use the convex part of most common cost functions. This result in showing that NMF is equivalent to maximizing a bound of the log-likelihood of a Finite Mixture Model (FMM) where the latent variable $\boldsymbol{Z}$ is shifted into the set of parameters $\boldsymbol{\Theta}$.

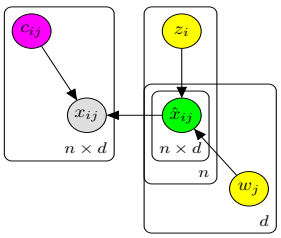
For example, by setting $\boldsymbol{Z} = [\boldsymbol{z}_1|\dots|\boldsymbol{z}_n]^\top$ as a set of probability distributions s.t. $\boldsymbol{Z}^\top \in (\Delta_g)^n$ where $\Delta_g = \{\forall \boldsymbol{z}_i \in \mathbb{R}_+^g : \sum_k^g z_{ik} = 1\}$ is a probability simplex, the I-divergence between $\boldsymbol{X}$ and $\boldsymbol{Z}\boldsymbol{W}^\top$ can be rewritten in terms of expectation s.t. $[\boldsymbol{Z}\boldsymbol{W}^\top]_{ij} = \sum_k^g z_{ik} w_{jk} = \mathbb{E}_{\boldsymbol{z}_i}\boldsymbol{w}_j$. Therefore,

$$D_I(\boldsymbol{X}||\boldsymbol{Z}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} x_{ij} \log \frac{x_{ij}}{\mathbb{E}_{\boldsymbol{z}_i}\boldsymbol{w}_j} - x_{ij} + \mathbb{E}_{\boldsymbol{z}_i}\boldsymbol{w}_j \tag{4.66}$$

$$\leq \mathbb{E}_{\boldsymbol{Z}}\left[\log \prod_j^d \frac{e^{\boldsymbol{w}_j}\boldsymbol{w}_j^{-x_{ij}}}{x_{ij}^{-x_{ij}} e^{x_{ij}}}\right] \equiv \mathcal{Q}(\boldsymbol{\Theta}', \boldsymbol{\Theta}), \tag{4.67}$$



(a) Finite mixture as a graphical model.



(b) NMF as a graphical model. $\hat{x}_{ij} = \boldsymbol{z}_i^\top \boldsymbol{w}_j$.

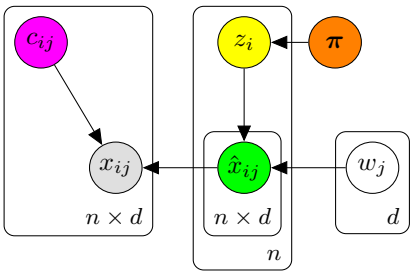


FIGURE 4.7 – $\texttt{cNMF}_{H_0}$ as a graphical model. $\hat{x}_{ij} = \boldsymbol{z}_i^\top \boldsymbol{w}_j$.

where $\mathcal{Q}(\boldsymbol{\Theta}', \boldsymbol{\Theta})$ is the well known $\mathcal{Q}-function$ maximized by the Expectation-Maximization (EM) algorithm [241] for a Poisson finite mixture. By estimating the matching mixture model from the cost function, further deduction will show that NMF intuitive shifting of the latent variable $\boldsymbol{Z}$ into the set of parameters contributes to avoid a well known problem arising with balanced/parsimonious mixture model which is maximum entropy. This problem is usually leveraged by specifying more parameters to the model such as mixture weights or decomposition of the dispersion parameter (e.g. in Gaussian mixture). Hence, more estimates are required which increases the computational time.

4.2.2 Related Works

In this contribution, we refocus on a prior work on entropy maximization subject to NMF constraints referred to as $\texttt{cNMF}_{\texttt{H}}$. This problem is expressed as follows :

$$\min_{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0, \boldsymbol{Z}\mathbf{1}_g = \mathbf{1}_n} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = -H(\boldsymbol{Z}) + \lambda \mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)\}. \tag{4.68}$$

where $H(\boldsymbol{Z}) = \sum_{i,k}^{n,g} z_{ik} \log z_{ik}$ is the Shannon entropy functional [110].

A graphical model of $\texttt{cNMF}_{H_0}$ is given in Figure 4.7 ; it displays a generalizing model including mixing weights for the latent variable $\boldsymbol{Z}$ which will be discussed in Section 4.2.5. The generalized Kullback-Leibler divergence is undoubtedly the most relevant cost function for achieving NMF for the task of document clustering. An application on document-term matrices comparing several algorithms can be

found in [223]. The main bottleneck with the I-divergence comes down to the high computational cost of its gradient. Therefore, by taking advantages of the probability factor of $\texttt{cNMF}_{H_\alpha}$ and the convexity of the negative logarithm, we aim at accelerating the convergence by deriving a surrogate function (bounded by the Jensen inequality) which provides a less expensive gradient to compute. Eventually, we notice that this transformation describes a larger relation between $\texttt{cNMF}_{H_1}$ using the class of Bregman divergence and Finite mixture models. Our work goes back to the relation of Bregman divergence and exponential families (Distributions) formulated by Banerjee [41] and the work of Hathaway [242] and its alternative EM [241] objective function based upon the log-likelihood and the Shannon entropy. In addition, a common practice in text analysis and NMF document clustering is to normalize the observed data $\boldsymbol{X} = [\boldsymbol{x}_1|\dots|\boldsymbol{x}_n]^\top$ into the unit-sphere $\mathbb{S}^{d-1} = \{\boldsymbol{x}_i \in \mathbb{R}^d : \|\boldsymbol{x}_i\| = 1\}$ so that the bias introduced by their length vanishes and helps to improve the clustering. To our knowledge, no real attention has been given to the impact of this normalization on the Poisson probability distribution compared to a dedicated directional distribution (e.g. von Mises-Fisher). In this contribution, we will define a version of NMF for directional data based upon the $(1-\cos)$ dissimilarity and show that this normalization can benefit substantially more to discrete FMMs than to continuous FMMs or NMF as the normalization in the unit-sphere circumvents the maximum entropy encountered with the discrete Poisson distribution.

The paper is organized as follows. In Section 4.2.3, we describe the transition from NMF to mixture models and quantify the difference between both methods. Section 4.2.4 presents a comparison between NMF and the mixture models as well as against other state-of-the-art algorithms on several benchmarking datasets. In Section 4.2.5, several properties of $\texttt{cNMF}_{\texttt{H}}$ and the perspectives offered by this model are discussed.

4.2.3 From cNMF to finite mixture models

In order to distinguish the notion of classification log-likelihood based on hard indicator and its expectation based on the conditional probabilities, we will adopt the following notation for more convenience such that $\boldsymbol{Z} \in \{0,1\}^{n\times g}$ will now denote a hard classification matrix, while $\widetilde{\boldsymbol{Z}} \in [0,1]^{n\times g}$ will be set as a matrix of conditional probabilities.

This section highlights the transition from cNMF to finite mixture models of the class of exponential Families when $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is set as a *Bregman* divergence. The *Bregman* divergence is a measure of

the distance between two points defined in terms of a strictly convex function.

Definition 4.2.1. (**Bregman divergence** [243]). Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, and $\phi : \mathcal{S} \to \mathbb{R}$, $\mathcal{S} = \text{dom}(\phi)$ be a strictly convex function defined on a convex set $\mathcal{S} \subset \mathbb{R}^d$ such that ϕ is differential on $\text{ri}(\mathcal{S})$, the Bregman divergence denoted $d_\phi : \mathcal{S} \times \text{ri}(\mathcal{S}) \to [0, +\infty)$ is given as follows :

$$d_\phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla_{\boldsymbol{y}} \phi \rangle. \tag{4.69}$$

Let $\texttt{cNMF}_\texttt{H}$ be the problem of $\texttt{cNMF}_{H_1}$ where the entropic regularization is achieved using the Shannon entropy $H(.)$ such as :

$$\min_{\widetilde{\boldsymbol{Z}} \geq 0, \boldsymbol{W} \geq 0, \widetilde{\boldsymbol{Z}} \mathbf{1}_g = \mathbf{1}_n} \{\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top) - H(\widetilde{\boldsymbol{Z}})\}, \tag{4.70}$$

where $H(\widetilde{\boldsymbol{Z}}) = -\sum_{i,k}^{n,g} \tilde{z}_{ik} \log \tilde{z}_{ik}$. It can be shown that $\texttt{cNMF}_\texttt{H}$ is equivalent to maximizing a surrogate of a log-likelihood function of a FMM. Let $\delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top)$ be the auxiliary function of $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top)$ obtained through a set of Jensen inequalities for one or several real convex or concave functions $\Phi = [\phi_1, \ldots, \phi_m]$ such that :

$$\delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top) \overset{def}{=} \sum_{i,k}^{n,g} \tilde{z}_{ik} \mathcal{D}(\boldsymbol{x}_i, \boldsymbol{w}_k). \tag{4.71}$$

Proposition 4.2.1. If $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top)$ is a Bregman divergence, the minimization of the following objective results in the maximization of the fuzzy criterion $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta})$ of a finite mixture model such as :

$$\begin{aligned} \min_{\widetilde{\boldsymbol{Z}} \geq 0, \boldsymbol{W} \geq 0, \widetilde{\boldsymbol{Z}} \mathbf{1}_g = \mathbf{1}_n} \{\delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top) - H(\widetilde{\boldsymbol{Z}})\} &\equiv \max \Big\{ \log \prod_{i,k}^{n,g} \left[\pi_k f(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\right]^{\tilde{z}_{ik}} + H(\widetilde{\boldsymbol{Z}}) \Big\} \\ &\equiv \max \tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta}), \end{aligned} \tag{4.72}$$

where f is a probability density (or mass) function of an exponential family, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ the proportions (assumed to be equal) and $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\theta}\}$ the set of parameters with $\boldsymbol{\theta} = [\boldsymbol{w}_1 | \ldots | \boldsymbol{w}_g]$.

Lemma 4.2.1. *Maximizing $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta})$ is equivalent to maximizing the expectation of the conditional classification log-likelihood and therefore the likelihood $\mathcal{L}(\boldsymbol{\Theta}) = \prod_i^n \sum_k^g \pi_k f(\boldsymbol{x}_i, \boldsymbol{\theta}_k)$.*

Proof. $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta})$ is a sum of fuzzy complete data log-likelihood $\mathcal{L}_c(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta})$ and entropy. Its maximization has been shown by Hathaway to be equivalent to the maximization of the likelihood by the EM algorithm (see [242]). □

Lemma 4.2.2. *If $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is a Bregman divergence, $\delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is equal to the logarithm of the negative Bregman Soft clustering criterion [41] up to a normalizing constant $\sum_i^n \log b_\phi(\boldsymbol{x}_i)$.*

Proof. Using proposition 4.2.1 leads to $\delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \sum_{i,k}^{n,g} \tilde{z}_{ik} d_\phi(\boldsymbol{x}_i, \boldsymbol{w}_k)$. This interpretation can be generalized by the relation between Bregman divergence and Exponential Families/distributions [41] stating that for a Bregman divergence d_ϕ derived from ϕ :

$$p_{(\psi,\boldsymbol{\theta})}(\boldsymbol{x}_i) = \exp(-d_\phi(\boldsymbol{x}_i, \boldsymbol{w}_k)) b_\phi(\boldsymbol{x}_i), \quad \forall \boldsymbol{x}_i \in \mathrm{dom}(\phi) \tag{4.73}$$

where $p_{(\psi,\boldsymbol{\Theta})}$ is a probability density (or mass) function of a regular exponential family/distribution, ψ is the cumulant closed convex function with natural parameter space $\mathrm{dom}(\psi)$, $\boldsymbol{\theta}$ the natural parameter, $\boldsymbol{w}_k$ the expectation parameter, ϕ a convex conjugate function of ψ so that $(\int(\mathrm{dom}(\phi)), \phi)$ is the Legendre dual of $(\mathrm{dom}(\psi), \psi)$ and $b_\phi(\boldsymbol{x}_i)$ is a real value function. Consequently :

$$\begin{aligned} -\log \prod_{i,k}^{n,g} [p_{(\psi,\boldsymbol{\theta})}(\boldsymbol{x}_i)]^{\tilde{z}_{ik}} &= -\sum_{i,k}^{n,g} \tilde{z}_{ik} \log \left[\exp(-d_\phi(\boldsymbol{x}_i, \boldsymbol{w}_k)) b_\phi(\boldsymbol{x}_i) \right] \\ &= \sum_{i,k}^{n,g} \tilde{z}_{ik} d_\phi(\boldsymbol{x}_i, \boldsymbol{w}_k)) - \sum_{i}^{n} \log b_\phi(\boldsymbol{x}_i) \\ &= \delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) - \sum_{i}^{n} \log b_\phi(\boldsymbol{x}_i). \end{aligned} \tag{4.74}$$

□

Proof of Proposition 4.2.1 when D is the I-divergence. Let denote the problem of $\mathtt{cNMF}_{\mathtt{H}}$ with the I-divergence as follows :

$$\min_{\widetilde{\boldsymbol{Z}} \geq 0, \boldsymbol{W} \geq 0, \widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n} \{\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \sum_{i,j}^{n,g} \Big[x_{ij} \log \frac{x_{ij}}{[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij}} - x_{ij} + [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij} \Big] - H(\widetilde{\boldsymbol{Z}})\}, \tag{4.75}$$

where $[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij} = \sum_k^g \tilde{z}_{ik} w_{jk}$, $\forall i = 1, \ldots, n$, $\forall j = 1, \ldots, d$.

For any convex function $\phi : [0, +\infty) \to \mathbb{R}$, given a random variable $Y = (y_1, \ldots, y_n)$, the Jensen inequality states that :

$$E[\phi(Y)] \geq \phi(E[Y]),$$

with equality when $\phi(Y)$ reaches its canonical form s.t. $\phi(Y) = Y$. Considering the finite form of this inequality defined as : $\phi\Big(\sum_i^n a_i y_i\Big) \leq \sum_i^n a_i \phi(y_i)$, where the elements $(a_1, \ldots, a_n)$ are weights. Since $\widetilde{\boldsymbol{Z}}$

denotes a matrix of probabilities s.t. $\sum_k^g \tilde{z}_{ik} = 1$ and because of the convexity of the function $-\log(x)$, we can derive the following inequality :

$$-\log \sum_k^g \tilde{z}_{ik} w_{jk} \leq -\sum_k^g \tilde{z}_{ik} \log w_{jk}. \tag{4.76}$$

From this result, we can define an upper bound $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ for $\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ such that :

$$\begin{aligned}
\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) =& \sum_{i,j}^{n,d} \Big[x_{ij} \log x_{ij} - \sum_k^g x_{ij} \tilde{z}_{ik} \log w_{jk} - x_{ij} + [\widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top]_{ij} \Big] - H(\widetilde{\boldsymbol{Z}}) \\
=& \sum_{i,k}^{n,g} \tilde{z}_{ik} \sum_j^d \Big[x_{ij} \log x_{ij} - x_{ij} \log w_{jk} - x_{ij} + \sum_j^d w_{jk} \Big] - H(\widetilde{\boldsymbol{Z}}) \\
=& \delta_\Phi(\boldsymbol{X}, \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top) - H(\widetilde{\boldsymbol{Z}}).
\end{aligned}$$

From Lemma 4.2.2, we have :

$$\begin{aligned}
\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) =& \sum_{i,k}^{n,g} \tilde{z}_{ik} d_\phi(\boldsymbol{x}_i, \boldsymbol{w}_k) - H(\widetilde{\boldsymbol{Z}}) \\
=& -\log \prod_{i,k}^{n,g} [p_{(\psi,\boldsymbol{\theta})}(\boldsymbol{x}_i)]^{\tilde{z}_{ik}} + \sum_i^n \log b_\phi(\boldsymbol{x}_i) - H(\widetilde{\boldsymbol{Z}}) \\
=& -\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) + \sum_i^n \log b_\phi(\boldsymbol{x}_i).
\end{aligned} \tag{4.77}$$

where $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ is the fuzzy criterion assuming that the proportions are equal, and $f(x_{ij}; w_{jk}) = \mathcal{P}(w_{jk}) = \frac{w_{jk}^{x_{ij}} e^{-w_{jk}}}{x_{ij}!}$ is the Poisson pmf's, such that :

$$\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \log \prod_{i,k,j}^{n,g,d} [f(x_{ij}; w_{jk})]^{\tilde{z}_{ik}} + H(\widetilde{\boldsymbol{Z}}). \tag{4.78}$$

The optimization of $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ given in Section 4.2.6 shows that minimizing $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ is equivalent to maximizing (4.78) with respect to $\tilde{z}_{ik}$ and w_{jk}. In this particular case, we have that $\sum_i^n \log b_\phi(\boldsymbol{x}_i) = \sum_{i,j}^{n,d} \log \frac{x_{ij}^{x_{ij}}}{x_{ij}!}$. In the Following, we refer to this model as PMM (Poisson Mixture Model).

Remarks.

— Since maximizing (4.78) is equivalent to maximizing the likelihood of Poisson mixture models (where the proportions are assumed equal), the EM algorithm can be used.

— NMF with I-divergence can be viewed as an EM algorithm. Then a comparison between both algorithms should be interesting.

— We can insert the marginals in the model and propose variants of cNMF.

In the case of the I-divergence, Φ has an unique element $\phi = -\log x$. Consequently, the difference between $\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ in $\texttt{cNMF}_\texttt{H}$ and the fuzzy criterion results in :

$$E[\phi(Y)] - \phi(E[Y]) = E[d_\phi(Y, E[Y])] = I_\phi(Y),$$

where d_ϕ is a Bregman divergence. This difference is known as the Bregman information [244, 41]. In the case of $\texttt{cNMF}_\texttt{H}$ with the I-divergence, $I_\phi(\boldsymbol{W})$ results in $E_{\widetilde{\boldsymbol{Z}}}[d_\phi(\boldsymbol{W}, E_{\widetilde{\boldsymbol{Z}}}[\boldsymbol{W}])]$ where d_ϕ is the Itakura-Saito divergence since $\phi(x) = -\log(x)$.

■

However, whilst the I-divergence involves only one convex functions to shift from $\texttt{cNMF}_\texttt{H}$ to a FMM, the transition with other divergences such as the Logictic loss toward the Bernouilli FMM or the Itakura-Saito divergence toward the Erlang FMM may involve several functions. In this case, the difference between the objective function of $\texttt{cNMF}_\texttt{H}$ and the fuzzy criterion will be given as a sum of the Bregman information criterion defined on each convex or concave functions. Recalling $\Phi = [\phi_1, \ldots, \phi_m]$, we define the sum of Bregman information as follows :

$$\begin{aligned}\sum_m^n I_{\phi_m}(\boldsymbol{W}) =& E_{\widetilde{\boldsymbol{Z}}}[d_{\phi_1}(\boldsymbol{W}, E_{\widetilde{\boldsymbol{Z}}}[\boldsymbol{W}])] + \ldots + E_{\widetilde{\boldsymbol{Z}}}[d_{\phi_m}(\boldsymbol{W}, E_{\widetilde{\boldsymbol{Z}}}[\boldsymbol{W}])] \\ =& E_{\widetilde{\boldsymbol{Z}}}[d_{\phi_1}(\boldsymbol{W}, E_{\widetilde{\boldsymbol{Z}}}[\boldsymbol{W}]) + \ldots + d_{\phi_m}(\boldsymbol{W}, E_{\widetilde{\boldsymbol{Z}}}[\boldsymbol{W}])] = I_\Phi(\boldsymbol{W}). \end{aligned} \tag{4.79}$$

Several other examples are available in Appendix D.1 for other common Bregman divergences used in NMF such as the Frobenius norm (relative to the Gaussian distribution) or the Itakura-Saito divergence (relative to the Erlang or Exponential distributions). An example with the $(1 - \cos)$ dissimilarity obtained through the Frobenius norm (relative to the Von Mises-Fisher distribution) is also given. Table 4.3 summaries the relations between cNMF and the underlying mixture models according to the distance function.

4.2.4 Numerical experiments

After establishing the relation between $\texttt{cNMF}_\texttt{H}$ with the I-divergence and the Poisson mixture model, we tackle several numerical experiments aiming at comparing their clustering performances on several document-term matrices. As mentioned earlier, normalizing the samples in the unit-sphere is a default

TABLE 4.3 – Examples of $\texttt{cNMF}_\texttt{H}$ with Bregman divergences and the corresponding finite mixture models.

Distance	$\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$	NMF constraints	Mixture
Frobenius norm	$\frac{1}{2}\|\boldsymbol{X} - \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\|_F^2$	$\widetilde{\boldsymbol{Z}} \geq 0$, $\boldsymbol{W} \geq 0$, $\widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n$	Gaussian
$(1 - \cos)$ dissimilarity	$\sum_i^n \left(1 - \langle \boldsymbol{x}_i, [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i \rangle\right)$	$\widetilde{\boldsymbol{Z}} \geq 0, \boldsymbol{W} \geq 0, \widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n,$ $\|[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i\| = 1$	von Mises-Fisher
I-divergence	$D_I(\boldsymbol{X} \| \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$	$\widetilde{\boldsymbol{Z}} \geq 0$, $\boldsymbol{W} \geq 0$, $\widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n$	Poisson
Itakura-Saito div	$D_{IS}(\boldsymbol{X} \| \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$	$\widetilde{\boldsymbol{Z}} \geq 0$, $\boldsymbol{W} \geq 0$, $\widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n$	Erlang/Exponential

practice in NMF. However, to our knowledge, nobody has so far considered to achieve NMF using a directional measure such as the $(1 - \cos)$ dissimilarity and assess the impact of this normalization across several methods. Therefore, for our comparative study, we introduced `SpNMF` (Spherical NMF) along side the like of NMF with the Frobenius norm and the I-divergence. `SpNMF` is defined as the problem of NMF where $\|\boldsymbol{x}_i\| = \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\| = 1$ and $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$ is the $(1 - \cos)$ dissimilarity derived from the Frobenius norm such as :

$$\min_{\substack{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0 \\ \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|=1, \forall i=1,\ldots,n}} \{\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) = \sum_i^n \left(1 - \langle \boldsymbol{x}_i, [\boldsymbol{Z}\boldsymbol{W}^\top]_i \rangle\right)\}, \tag{4.80}$$

The derivation of `SpNMF` is available in Section 4.2.7. Note that, in regards to the notation established earlier, $\boldsymbol{Z} \in \mathbb{R}_+^{n \times g}$ for `SpNMF`.

As mentioned in the last section, estimating the conditional probabilities using EM for the Gaussian and von Mises-Fisher mixture models lead to poor partitioning with maximum entropy. Therefore, only the hard classifications of CEM were considered. Since the proportions and variance (or concentration) parameters are assumed to be equal, these algorithms are equivalent to K-means and Spherical K-means respectively and denoted subsequently in the following. For the opposite reason (also explained in the last section), K-means with the Poisson log-divergence (equivalent to a CEM) was also omitted. Naturally a Spherical version of `cNMF` could also be introduced. However, as shown in Section 4.2.7.3, the optimization of this method is equivalent to a fuzzy Spherical K-means algorithm and therefore would be affected by the maximum entropy since the criterion is also equal to the log-likelihood of a von Mises-Fisher FMM with a missing normalization. Finally `cNMF` with the Frobenius norm was not included in this section. Note that thanks to the low computational cost of its gradient, the Frobenius norm has been widely employed in regularized NNF models assigned to document clustering. However, in this domain, this distance remains less relevant compared to the I-divergence. Each algorithm was

launched 30 times on every dataset. Among those 30 trials, only the 10 best solutions (ranked according to the criterion) were kept.

4.2.4.1 Datasets

We draw our comparison on 8 bench-marking document-term matrices for which the detailed characteristics are available in Table 4.4. nz indicates the percentage of non-zero scalar and the *balance* coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class. These datasets highlight several varieties of challenging situations such as the amount of clusters, the dimensions, the clusters balance, the degree of overlapping of clusters and the sparsity. We normalized each data matrix with TF-IDF and their respective documents-vectors to unit L_2-norm.

TABLE 4.4 – Datasets description : # denotes the cardinality.

	Characteristics				
Datasets	#Documents	#Words	#Clusters	$nz(\%)$	Balance
NG5	4905	10167	5	0.92	0.943
CLASSIC3	3891	4303	3	1.05	0.707
NG20	18846	14390	20	0.59	0.628
OHSCAL	11162	11465	10	0.53	0.437
CLASSIC4	7095	5896	4	0.59	0.323
LA12	6279	31472	6	0.48	0.282
RCV1	6387	16921	4	0.25	0.080
SPORTS	8580	14870	7	0.86	0.036

4.2.4.2 Empirical results on benchmark datasets

Several NMF models acknowledged for improving document clustering with NMF were added in the parallel. The list includes : the original NMF with the Frobenius norm (`NMF`) and the I-divergence (`NMF − KL`), Orthogonal NMF (`ONMF`) [158], Projective NMF (`PNMF`) [161] and, Graph Regularized NMF (`GNMF`) [163]. A Deep-Learning algorithm namely Deep Clustering Network (`DCN`) [203] was also included. The `DCN` algorithm showed significant improvements for document clustering against several clustering (K-means, Spectral Clustering), NMF (`LCCF`) [204] and Deep Learning algorithms (`SAE`)[205]. All algorithms requiring parameters were launched accordingly to the respective settings advocated by their authors. The quality of the clustering was assess using two measures widely used for quantifying the correspondence between the clustering and the true labels. The Normalized Mutual

Information (NMI) [119], which measures the mutual dependency between two random variables, and, the Adjusted Rand Index (ARI) [122], which measures the degree of agreement between two partitions.

TABLE 4.5 – NMI and ARI means and standard deviations (SD) over different datasets (Mean±SD).

Datasets	Metrics	NMF	K-ms	SpNMF	S-Kms	NMF − KL	cNMF	$\texttt{cNMF}_\texttt{H}$	PMMU
NG5	NMI	56±0.0	53±3.5	77±3.1	72±1.8	80±3.3	82±1.4	87±.01	**90**±10
	ARI	33±0.0	31±2.7	72±6.8	60±1.0	82±4.3	84±1.4	90±.02	**92**±10
CLASSIC3	NMI	49±0.0	91±0.0	94±0.0	95±0	95±0.1	95±.03	**96**±.01	**96**±0.1
	ARI	44±0.0	94±0.0	97±0.0	97±0.0	97±0.1	97±.02	**98**±0.0	**98**±0.0
NG20	NMI	42±0.8	40±1.9	46±1.8	49±2.1	48±2.2	50±0.9	66±1.2	**70**±0.8
	ARI	23±0.8	14±2.4	32±2.0	30±2.4	34±2.2	36±1.4	56±2.4	**57**±1.4
OHSCAL	NMI	38±0.6	36±1.3	41±0.1	**43**±0.2	35±1.2	36±1.2	40±0.3	41±0.5
	ARI	29±0.9	21±1.6	32±0.1	**33**±0.3	24±1.5	25±1.4	29±0.5	29±0.9
CLASSIC4	NMI	53±0.4	55±0.3	58±0.2	60±0.1	70±2.5	**76**±0.6	**76**±.01	**76**±0.0
	ARI	45±0.3	37±0.3	47±0.1	47±0.1	64±5.9	66±1.8	**68**±0.0	63±0.0
LA12	NMI	42±1.6	45±6.6	47±0.6	**58**±3.0	48±3.9	43±3.3	57±0.4	50±3.5
	ARI	36±2.8	31±8.5	44±0.7	53±2.2	45±4.4	38±4.3	**54**±0.4	45±4.9
RCV1	NMI	35±0.0	45±9.5	48±7.5	38±.02	47±2.4	48±0.7	**51**±.09	50±1.8
	ARI	13±0.0	28±14.4	38±13	18±.03	42±2.2	43±1.0	**46**±.05	44±1.5
SPORTS	NMI	55±0.0	45±5.4	57±2.3	**62**±2.3	55±2.6	54±1.2	61±1.5	61±1.4
	ARI	28±0.0	17±6.5	39±1.6	43±4.1	39±2.2	40±2.6	45±2.5	**46**±1.3

From the results in Table 4.5, the best performances are shared between $\texttt{cNMF}_\texttt{H}$ and the Poisson mixture model in the unit-sphere (PMMU). Spherical K-means has the best scores on only one dataset (OHSCAL) overall and its other best performances on LA12 and SPORTS can be matched by $\texttt{cNMF}_\texttt{H}$ and PMMU respectively. In addition, SpNMF achieved better performance than Spherical K-means on two datasets and remains close overall. Since the I-divergence is scale invariant, cNMF has similar results to NMF − KL. The difference in performance between $\texttt{cNMF}_\texttt{H}$ and PMMU seems to be related to the balanced of the partitions. When the datasets are very balanced, PMMU gives better results than $\texttt{cNMF}_\texttt{H}$ (see NG5, NG20). For the oppositve, $\texttt{cNMF}_\texttt{H}$ seems to achieve better performance (see CLASSIC4, LA12, RCV1). Overall, both algorithm deliver close results and the choice of the user might come back to its needs. For a scalability, PMMU should undeniably be preferred. However, if the user can assess the balanced of the partition (e.g. through visualisation), $\texttt{cNMF}_\texttt{H}$ could be chosen. In the context of this contribution, PMMU fulfills our need for less computationally expensive gradients and therefore will

be selected. In Table 4.6, we compare PMMU against several state-of the arts algorithm. As it was for $\texttt{cNMF}_{\texttt{H}}$, it is clear that PMMU outperforms and by a substantial margin.

TABLE 4.6 – NMI and ARI means and standard deviations (SD) over different datasets (Mean±SD).

Datasets	Metrics	ONMF	PNMF	GNMF	DCN	PMMU
NG5	NMI	65±4.0	65±5.0	63±7.0	62±2.8	**90**±10
	ARI	48±8.0	47±9.0	62±9.0	47±2.7	**92**±10
CLASSIC3	NMI	58±0.0	71±22	63±6.8	92±4.6	**96**±0.1
	ARI	55±0.0	70±26	57±9.2	94±4.5	**98**±0.0
NG20	NMI	44±2.0	45±2.0	50±1.0	43±1.0	**70**±0.8
	ARI	22±2.0	24±2.0	35±5.0	17±1.5	**57**±1.4
OHSCAL	NMI	37±1.8	39±1.2	38±1.3	35±1.0	**41**±0.5
	ARI	28±1.8	**29**±2.0	28±1.6	25±1.9	**29**±0.9
CLASSIC4	NMI	55±9.0	59±5.0	65±4.0	57±1.4	**76**±0.0
	ARI	39±9.0	44±1.0	49±5.0	42±1.3	**63**±0.0
LA12	NMI	44±2.2	43±3.0	47±2.0	52±3.5	**50**±3.5
	ARI	40±4.1	37±6.0	43±3.0	44±5.6	**45**±4.9
RCV1	NMI	49±2.0	46±4.5	48±4.0	34±0.6	**50**±1.8
	ARI	39±4.0	37±5.3	39±3.0	12±0.8	**44**±1.5
SPORTS	NMI	55±2.0	56±0.1	55±0.1	59±1.5	**61**±1.4
	ARI	28±1.0	28±0.1	28±0.1	37±3.4	**46**±1.3

4.2.5 Discussion

4.2.5.1 Additional regularizations for cNMF

From the link established between cNMF and FMMs in the previous section, several regularizations of cNMF's objective can be provided. By adding the proportions term, we denote the following optimization problem referred to as $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$:

$$\min_{\substack{\widetilde{\boldsymbol{Z}}\geq 0,\boldsymbol{W}\geq 0,\\ \widetilde{\boldsymbol{Z}}\mathbf{1}_g=\mathbf{1}_n,\boldsymbol{\pi}^\top\mathbf{1}_g=\mathbf{1}}} \{\mathcal{F}(\widetilde{\boldsymbol{Z}},\boldsymbol{W})=\mathcal{D}(\boldsymbol{X},\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)+\sum_{i}^{n} D_{KL}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})\}. \tag{4.81}$$

$\texttt{cNMF}_{\texttt{H}}$ is therefore a special case of $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ where the proportions are assumed to be equal. Another objective regularization could also be suggested using only the proportions. We refer to this optimization

problem as $\texttt{cNMF}_{\boldsymbol{\pi}}$:

$$\min_{\substack{\widetilde{\boldsymbol{Z}}\geq 0, \boldsymbol{W}\geq 0, \\ \widetilde{\boldsymbol{Z}}\mathbf{1}_g=\mathbf{1}_n, \boldsymbol{\pi}^\top\mathbf{1}_g=\mathbf{1}}} \{\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) - \sum_{i,k}^{n,g} \tilde{z}_{ik} \log \pi_k\}. \tag{4.82}$$

4.2.5.2 Scale change invariance

cNMF was initially introduced with the I-divergence which by definition is invariant to scale. However, when $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is scale sensitive (e.g. the Frobenius norm), $\widetilde{\boldsymbol{Z}}$ defined as a set of probability distribution drastically reduces the space of possible solutions and leads to a sensible decrease of the performance.

Considering two discretized random variables $\boldsymbol{p} \in \Delta_d$ and $\boldsymbol{q} \in \Delta_d$, a divergence D_{inv} is said to be invariant by changing of scale w.r.t. $\boldsymbol{q}$ if :

$$D_{inv}(\boldsymbol{p}||\boldsymbol{q}) = D_{inv}(\boldsymbol{p}||a\boldsymbol{q}),\ \forall a \in \mathbb{R}^+. \tag{4.83}$$

The I-divergence is naturally invariant to scale. Consequently, the linear approximation $\boldsymbol{x}_i \approx [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i$ w.r.t. the sum marginal of $\boldsymbol{x}_i$ gives :

$$D_I(\boldsymbol{x}_i||a[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i) = D_I(\boldsymbol{x}_i||[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i). \tag{4.84}$$

Therefore, the change of scale produced by the following constraint : $\tilde{\boldsymbol{z}}_i \in \Delta_g, \forall i = 1, \ldots, n$, does not affect cNMF. For solving this issue with the class of scale sensitive divergence, we illustrate the approach of Eguchi and Kato [245] for building invariant divergences by introducing a positive *invariance factor* $\kappa(\boldsymbol{p}, \boldsymbol{q})$ such that $D(\boldsymbol{p}||\kappa\boldsymbol{q}) = D_{inv}(\boldsymbol{p}||\boldsymbol{q})$. As emphasized by Lantéri [246], a divergence D can be made invariant according to several invariance factor w.r.t. the fundamental properties of invariant divergences based on the gradient of $D(\boldsymbol{p}||\kappa\boldsymbol{q})$ w.r.t. $\boldsymbol{q}$ and the differential equation stated as follows :

$$\sum_j^d q_j \frac{\partial D(\boldsymbol{p}||\kappa\boldsymbol{q})}{\partial q_j} = 0, \tag{4.85}$$

$$\kappa + \sum_j^d q_j \frac{\partial \kappa}{\partial q_j} = 0. \tag{4.86}$$

In this contribution, we draw our interest in the use of a invariance factor defined directly from D [245]. This factor is referred to as the *nominal invariance factor* $\kappa_0(\boldsymbol{p}, \boldsymbol{q})$ such that :

$$\kappa_0(\boldsymbol{p}, \boldsymbol{q}) = \arg\min_{\kappa>0} D(\boldsymbol{p}||\kappa\boldsymbol{q}). \tag{4.87}$$

Note that : $D(\boldsymbol{p}||\kappa_0\boldsymbol{q}) \leq D(\boldsymbol{p}||\kappa_1\boldsymbol{q})$, for any non-nominal invariance factor $\kappa_1(p,q)$. Therefore, in the following, we construct an invariant Frobenius norm and introduce a modified version of the multiplicative updates algorithm for practicing cNMF, $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ or $\texttt{cNMF}_{\texttt{H}}$ without scale perturbation. More precisely, we consider a set of n nominal invariance factors $\kappa_0(\boldsymbol{x}_i, [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i), \forall i = 1,\ldots,n$, one for each linear transformation regarding each approximation $\boldsymbol{x}_i \approx [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i$.

4.2.5.2.1 Invariant Frobenius norm for cNMF Let $D_F(\boldsymbol{X}||\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ be the squared Frobenius norm between $\boldsymbol{X}$ and $\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top$, we denote by

$$D_{inv-F}(\boldsymbol{X}||\operatorname{diag}(\boldsymbol{\kappa})\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$$

its invariant transformation w.r.t. a set of invariance factor $\boldsymbol{\kappa} = \{\kappa_1,\ldots,\kappa_n\}$ as follows :

$$D_{inv-F}(\boldsymbol{X}||\operatorname{diag}(\boldsymbol{\kappa})\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d}\Big(x_{ij} - \kappa_i\sum_k^g \tilde{z}_{ik}w_{jk}\Big)^2. \tag{4.88}$$

The partial derivative of $D_{inv-F}(\boldsymbol{X}||\operatorname{diag}(\boldsymbol{\kappa})\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ w.r.t. κ_i is therefore :

$$\frac{\partial D_{inv-F}}{\partial \kappa_i} = \sum_j^d\Big[2\kappa_i\Big(\sum_k^g \tilde{z}_{ik}w_{jk}\Big)^2 - 2x_{ij}\sum_k^g \tilde{z}_{ik}w_{jk}\Big]. \tag{4.89}$$

Setting this derivative to zero, we obtain the following expression for a nominal invariance factor :

$$\kappa_i = \frac{\sum_j^d x_{ij}\sum_k^g \tilde{z}_{ik}w_{jk}}{\sum_j^d(\sum_k^g \tilde{z}_{ik}w_{jk})^2}. \tag{4.90}$$

4.2.5.2.2 $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$, $\texttt{cNMF}_{\texttt{H}}$ and cNMF with the invariant Frobenius norm $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ where $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is set as the invariant Frobenius norm is described as the following optimization problem :

$$\min_{\substack{\widetilde{\boldsymbol{Z}}\geq 0, \boldsymbol{W}\geq 0\\ \widetilde{\boldsymbol{Z}}\mathbf{1}_g=\mathbf{1}_n, \boldsymbol{\pi}^\top\mathbf{1}_g=\mathbf{1}}} \{\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa}) = \frac{1}{2}\|\boldsymbol{X} - \operatorname{diag}(\boldsymbol{\kappa})\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\|_F^2 + \sum_i^n D_{KL}^{\xi,\iota}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})\}, \tag{4.91}$$

where $D_{KL}^{\xi,\iota}(\boldsymbol{p}||\boldsymbol{q}) \overset{def}{=} \xi\sum_i^n p_i\log p_i - \iota\sum_i^n p_i\log q_i$, with $\xi \in \{0,1\}$ and $\iota \in \{0,1\}$ s.t. $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ collapses to $\texttt{cNMF}_{\texttt{H}}$ for $\iota = 0$ and cNMF for $\xi = \iota = 0$. The associated Lagrangian function is expressed as follows :

$$\mathcal{L}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa}, \boldsymbol{\gamma}, \tau, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa}) + \sum_i^n \gamma_i\Big(\sum_k^g \tilde{z}_{ik} - 1\Big) + \tau\Big(\sum_k^g \pi_k - 1\Big) + \operatorname{Tr}(\boldsymbol{\epsilon}\widetilde{\boldsymbol{Z}}^\top) + \operatorname{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \tag{4.92}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\tau \in \mathbb{R}$, $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times g}$, and $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$ are the Lagrange multipliers. In the following, we define the Lagrange multipliers γ_i in terms of their respective positive and negative orthants such as $\gamma_i = [\gamma_i]^+ - [\gamma_i]^-$ where $[\gamma_i]^+ \geq 0$ and $[\gamma_i]^- \geq 0$. Let $\boldsymbol{K} = \text{diag}(\boldsymbol{\kappa})$ The partial derivative w.r.t. $\tilde{z}_{ik}$ and w_{jk} are denoted as follows :

$$\nabla_{\tilde{z}_{ik}} \mathcal{L} = -(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} + (\boldsymbol{K}^2 \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top \boldsymbol{W})_{ik} + \xi(1 + \log \tilde{z}_{ik}) - \iota \log \pi_k + \epsilon_{ik} + [\gamma_i]^+ - [\gamma_i]^-, \tag{4.93}$$

$$\nabla_{w_{jk}} \mathcal{L} = -(\boldsymbol{X}^\top \boldsymbol{K} \widetilde{\boldsymbol{Z}})_{ik} + (\boldsymbol{W} \widetilde{\boldsymbol{Z}}^\top \boldsymbol{K}^2 \widetilde{\boldsymbol{Z}})_{jk} + \beta_{jk}. \tag{4.94}$$

Setting these gradients to zero and making use of the Karush-Kuhn-Tucker conditions $\boldsymbol{\epsilon} \odot \widetilde{\boldsymbol{Z}} = 0$ and $\boldsymbol{\beta} \odot \boldsymbol{W} = 0$ lead to the following stationary equations :

$$\tilde{z}_{ik}\big[(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} - (\boldsymbol{K}^2 \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top \boldsymbol{W})_{ik} - \xi(1 + \log \tilde{z}_{ik}) - \iota \log \pi_k - [\gamma_i]^+ + [\gamma_i]^-\big] = 0, \tag{4.95}$$

$$w_{jk}\big[(\boldsymbol{X}^\top \boldsymbol{K} \widetilde{\boldsymbol{Z}})_{jk} - (\boldsymbol{W} \widetilde{\boldsymbol{Z}}^\top \boldsymbol{K}^2 \widetilde{\boldsymbol{Z}})_{jk}\big] = 0. \tag{4.96}$$

From these equations, we obtain the following multiplicative update rules :

$$\tilde{z}_{ik} \leftarrow \tilde{z}_{ik} \frac{(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} - \xi \log \tilde{z}_{ik} + [\gamma_i]^-}{(\boldsymbol{K}^2 \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top \boldsymbol{W})_{ik} + \xi - \iota \log \pi_k + [\gamma_i]^+}, \tag{4.97}$$

$$w_{jk} \leftarrow w_{jk} \frac{(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk}}{(\boldsymbol{W} \widetilde{\boldsymbol{Z}}^\top \boldsymbol{K} \widetilde{\boldsymbol{Z}})_{jk}}. \tag{4.98}$$

Let $B_{ik} = (\boldsymbol{K}^2 \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top \boldsymbol{W})_{ik} + \xi - \iota \log \pi_k + [\gamma_i]^+$. Plugging eq(4.97) into the constraint gives :

$$[\gamma_i]^- = \frac{1 - \sum_k^g \tilde{z}_{ik} \frac{(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} - \xi \log \tilde{z}_{ik} + [\gamma_i]^-}{B_{ik}}}{\sum_k^g \frac{\tilde{z}_{ik}}{B_{ik}}}. \tag{4.99}$$

From eq(4.99), $[\gamma_i]^-$ depends on $[\gamma_i]^+$. Using the first term of eq(4.99), we derive the conditional value of $[\gamma_i]^+$:

$$\begin{aligned} 1 - \sum_k^g \tilde{z}_{ik} \frac{(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} - \xi \log \tilde{z}_{ik} + [\gamma_i]^-}{B_{ik}} &= \sum_k^g \tilde{z}_{ik} \frac{B_{ik}}{B_{ik}} - \sum_k^g \tilde{z}_{ik} \frac{(\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} - \xi \log \tilde{z}_{ik} + [\gamma_i]^-}{B_{ik}} \\ &= \sum_k^g \tilde{z}_{ik} \frac{\nabla_{\tilde{z}_{ik}} \mathcal{F} + [\gamma_i]^+}{B_{ik}}. \end{aligned}$$

From this equality, $[\gamma_i]^+ = \max(\max(-\nabla_{\tilde{z}_{ik}} \mathcal{F} | k = 1, \ldots, g), 0)$ since $[\gamma_i]^+ \geq 0$.

The partial derivative of $\mathcal{L}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa}, \boldsymbol{\gamma}, \tau, \boldsymbol{\epsilon}, \boldsymbol{\beta})$ w.r.t. π_k is :

$$\nabla_{\pi_k}\mathcal{L} = -\xi \sum_i^n \tilde{z}_{ik}\frac{1}{\pi_k} + \tau. \tag{4.100}$$

Setting this derivative to zero leads to : $\pi_k = \frac{\xi \sum_i^n \tilde{z}_{ik}}{\tau}$. Plugging this expression into $\sum_k^g \pi_k = 1$ gives :

$$\pi_k = \frac{\sum_i^n \tilde{z}_{ik}}{n}, \tag{4.101}$$

The optimization procedure is given by Algorithm (18).

Algorithm 18 $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$, $\texttt{cNMF}_{\texttt{H}}$ & $\texttt{cNMF}$ with D_{inv-F}

Input : $\boldsymbol{X}$, g, $\widetilde{\boldsymbol{Z}}^{(0)}$; $\boldsymbol{W}^{(0)}$, $\xi \in \{0,1\}$, $\iota \in \{0,1\}$.
Initialization : $\boldsymbol{\pi}^{(0)}$ using eq(4.101).
Output : $\widetilde{\boldsymbol{Z}}$, $\boldsymbol{W}$, $\boldsymbol{\kappa}$.
repeat
1 (if $\iota = 1$). update $\boldsymbol{\pi}$ using eq(4.101) ;
2. update $\boldsymbol{\kappa}$ using eq(4.90) ;
3. update $\widetilde{\boldsymbol{Z}}$ with eq(4.97) ;
4. update $\boldsymbol{\kappa}$ using eq(4.90)
5. update $\boldsymbol{W}$ with eq(4.98) ;
until convergence

4.2.5.2.3 Convergence analysis The update formula for $\boldsymbol{W}$ given by eq(4.98) is identical to the one obtained with the original NMF. Therefore, from [132], $\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa})$ in problem(4.91) is non-increasing under the update rule (4.98). The convergence analysis for the update of $\tilde{z}_{ik}$ is given subsequently.

Theorem 4.2.3. *$\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\kappa})$ in problem(4.91) is non-increasing under eq(4.97) and eq(4.98).*

Considering the Lagrangian function for problem(4.91) defined as :

$$\mathcal{L}(\tilde{z}_{ik}) = \mathcal{F}(\tilde{z}_{ik}) + \sum_i^n \gamma_i\Big(\sum_k^g \tilde{z}_{ik} - 1\Big) + \tau\Big(\sum_k^g \pi_k - 1\Big), \tag{4.102}$$

and following the definition of an auxiliary function recalled subsequently :

Definition 4.2.2. $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}')$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$ if the following conditions are satisfied :

$$\forall \boldsymbol{z} \neq \boldsymbol{z}', \mathcal{H}(\boldsymbol{z}, \boldsymbol{z}') \geq \mathcal{L}(\boldsymbol{z}) \quad \text{and} \quad \mathcal{H}(\boldsymbol{z}, \boldsymbol{z}) = \mathcal{L}(\boldsymbol{z}).$$

A key point to the auxiliary function is the following lemma :

Lemma 4.2.4. *If $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$ is an auxiliary function for $\mathcal{L}(\boldsymbol{z})$, $\mathcal{L}(\boldsymbol{z})$ is non-increasing under the update*

$$\boldsymbol{z}^{(t+1)} = \arg\min_{\boldsymbol{z}} \mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^{(t)})$$

Proof. $\mathcal{L}(\boldsymbol{z}^{(t+1)}) \leq \mathcal{H}(\boldsymbol{z}^{(t+1)}, \boldsymbol{z}^{(t)}) \leq \mathcal{H}(\boldsymbol{z}^{(t)}, \boldsymbol{z}^{(t)}) = \mathcal{L}(\boldsymbol{z}^{(t)})$. □

We formulate the following proposition.

Proposition 4.2.2.

$$\begin{aligned}\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}^{(t)}) =& \mathcal{L}(\tilde{z}_{ik}^{(t)}) + \mathcal{L}'(\tilde{z}_{ik}^{(t)})(\tilde{z}_{ik} - \tilde{z}_{ik}^{(t)}) \\ &+ \frac{(\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} - \iota\log\pi_k + \xi + [\gamma_i]^+}{2\tilde{z}_{ik}^{(t)}}\Big(\tilde{z}_{ik} - \tilde{z}_{ik}^{(t)}\Big)^2, \end{aligned} \tag{4.103}$$

where $\mathcal{L}'(\tilde{z}_{ik}^{(t)}) = -\iota\log\pi_k - (\boldsymbol{K}\boldsymbol{X}\boldsymbol{W})_{ik} + (\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} + \xi(1+\log\tilde{z}_{ik}) + [\gamma_i]^+ - [\gamma_i]^-$, is an auxiliary function for $\mathcal{L}(\tilde{z}_{ik})$.

Proof. It is straightforward to verify that $\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}) = \mathcal{L}(\tilde{z}_{ik})$. We will demonstrate that $\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}^{(t)}) \geq \mathcal{L}(\tilde{z}_{ik})$ by using the second order Taylor expansion of $\mathcal{L}(\tilde{z}_{ik})$ denoted as follows :

$$\mathcal{L}(\tilde{z}_{ik}) = \mathcal{L}(\tilde{z}_{ik}^{(t)}) + \mathcal{L}'(\tilde{z}_{ik}^{(t)})(\tilde{z}_{ik} - \tilde{z}_{ik}^{(t)}) + \frac{\kappa_i^2(\boldsymbol{W}^\top\boldsymbol{W})_{kk} + \xi[\tilde{z}_{ik}^{(t)}]^{-1}}{2}\Big(\tilde{z}_{ik} - \tilde{z}_{ik}^{(t)}\Big)^2. \tag{4.104}$$

From $(\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} = \kappa_i^2\sum_{k'}^g \tilde{z}_{ik'}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{k'k}$,

$$\begin{aligned}(\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} \geq& \kappa_i^2\tilde{z}_{ik}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{kk} \\ (\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} + \xi - \iota\log\pi_k + [\gamma_i]^+ \geq& \kappa_i^2\tilde{z}_{ik}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{kk} + \xi \\ \frac{(\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} + \xi - \iota\log\pi_k + [\gamma_i]^+}{\tilde{z}_{ik}^{(t)}} \geq& \frac{\kappa_i^2\tilde{z}_{ik}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{kk} + \xi}{\tilde{z}_{ik}^{(t)}} \\ \geq& \kappa_i^2(\boldsymbol{W}^\top\boldsymbol{W})_{kk} + \xi[\tilde{z}_{ik}^{(t)}]^{-1}. \end{aligned} \tag{4.105}$$

From this inequality, we have that $\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}^{(t)}) \geq \mathcal{L}(\tilde{z}_{ik})$. □

Proof of Theorem 4.2.3. In order to satisfy Lemma 4.2.4, we compute the gradient of $\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}^{(t)})$ w.r.t $\tilde{z}_{ik}$:

$$\nabla_{\tilde{z}_{ik}}\mathcal{H} = \mathcal{L}'(\tilde{z}_{ik^{(t)}}) + \frac{(\boldsymbol{K}^2\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\boldsymbol{W})_{ik} + \xi - \iota\log\pi_k + [\gamma_i]^+}{2\tilde{z}_{ik}}\Big(2\tilde{z}_{ik} - 2\tilde{z}_{ik}^{(t)}\Big). \tag{4.106}$$

Setting this gradient to zero leads to :

$$\tilde{z}_{ik}^{(t+1)} = \tilde{z}_{ik}^{(t)} \frac{(\boldsymbol{KXW})_{ik} - \xi \log \tilde{z}_{ik} + [\gamma_i]^-}{(\boldsymbol{K}^2 \widetilde{\boldsymbol{Z}} \boldsymbol{W}^\top \boldsymbol{W})_{ik} + \xi - \iota \log \pi_k + [\gamma_i]^+}, \tag{4.107}$$

Since $\mathcal{H}(\tilde{z}_{ik}, \tilde{z}_{ik}^{(t)})$ is an auxiliary function for $\mathcal{L}(\tilde{z}_{ik})$, $\mathcal{L}(\tilde{z}_{ik})$ is non-increasing under this update. By reversing $\tilde{z}_{ik}$ and w_{jk}, $\mathcal{L}(w_{jk})$ can be shown similarly to be non-increasing under the update rules of w_{jk}.

■

4.2.5.2.4 Evaluation of $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$, $\texttt{cNMF}_{\texttt{H}}$ & cNMF with D_{inv-F} and D_I Solving (4.81) is achieved in our experiment considering the invariant Frobenius norm and the I-divergence. Table 4.7 shows the results of (NMF, cNMF) using the Frobenius norm, and (cNMF, $\texttt{cNMF}_{\texttt{H}}$, $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$) using D_{inv-F}. First, the results illustrate the benefits of using the invariant Frobenius divergence compared to the original Frobenius norm. Second, the results of $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ show that assuming mixed proportions improves the clustering of highly unbalanced text datasets compared to $\texttt{cNMF}_{\texttt{H}}$ (see LA12, RCV1). However, with the I-divergence, $\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$ showed less performance than $\texttt{cNMF}_{\texttt{H}}$. Similarly, we noticed that assuming mixed proportions in PMMU (equivalent mixture model) deteriorates the quality of the clustering, especially on unbalanced datasets which usually benefits from that extra parameterization in FMMs. Further assumptions and comments behind that behavior are discussed in the following (Section 4.2.5.3).

4.2.5.3 $\widetilde{\boldsymbol{Z}}$ low entropy in discrete mixture models

Let $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\theta})$ be the Fuzzy criterion of any random mixture model where the proportions are specified, we have :

$$\begin{aligned}\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta}) &= \log \prod_{i,k}^{n,g} \left[\pi_k f(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\right]^{\tilde{z}_{ik}} + H(\widetilde{\boldsymbol{Z}}) \\ &= \log \prod_{i,k}^{n,g} \left[f(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\right]^{\tilde{z}_{ik}} - \sum_{i}^{n} D_{KL}(\tilde{\boldsymbol{z}}_i || \boldsymbol{\pi}),\end{aligned} \tag{4.108}$$

where $\tilde{z}_{ik} = \frac{\pi_k f(\boldsymbol{x}_i;\boldsymbol{\theta}_k)}{\sum_\ell^g \pi_\ell f(\boldsymbol{x}_i;\boldsymbol{\theta}_\ell)} \in [0,1]$. Hathaway [242] pointed out that $-\sum_i^n D_{KL}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})$ in eq(4.108) acts as a penalization. Consequently, maximizing $\tilde{\mathcal{F}}(\widetilde{\boldsymbol{Z}}, \boldsymbol{\Theta})$ when the proportion are assumed to be equal tends to produce equal conditional probabilities $\tilde{z}_{ik}$. Indeed, since $-D_{KL}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})$ reaches its maximum when $\boldsymbol{\pi} = \tilde{\boldsymbol{z}}_i$, $\forall i = 1, \ldots, n$ as shown with the Gibbs inequality : $-\sum_k^g \tilde{z}_{ik} \log \tilde{z}_{ik} \leq -\sum_k^g \tilde{z}_{ik} \log \pi_k$.

TABLE 4.7 – Comparison of `cNMF` using the original and invariant form of the Frobenius norm. Mean ± SD (standard deviation) of NMI and ARI scores are given over different datasets.

		$\|.\|_F^2$		$D_{inv-F}(.,.)$		
Datasets	Metrics	`NMF`	`cNMF`	`cNMF`	$\texttt{cNMF}_{\texttt{H}}$	$\texttt{cNMF}_{\boldsymbol{\pi},\texttt{H}}$
NG5	NMI	56±0.0	56±4.8	71±2.3	68±7.0	68±3.6
	ARI	33±0.0	35±6.8	63±1.8	66±10.0	63±6.1
CLASSIC3	NMI	49±0.0	68±0.1	93±0.2	89±1.4	93±0.8
	ARI	44±0.0	65±0.1	96±0.1	93±1.3	96±0.5
NG20	NMI	42±0.8	40±1.7	45±1.1	46±1.8	45±1.0
	ARI	23±0.8	21±1.4	31±1.3	31±1.6	30±0.9
OHSCAL	NMI	38±0.6	33±1.8	40±2.2	32±2.4	32±2.3
	ARI	29±0.9	23±2.3	32±2.5	23±3.3	22±3.2
CLASSIC4	NMI	53±0.4	45±1.4	57±1.3	63±8.2	57±9.0
	ARI	45±0.3	27±1.9	46±1.2	59±9.6	49±11.0
LA12	NMI	42±1.6	37±1.5	43±3.3	50±3.5	57±0.4
	ARI	36±2.8	24±2.4	38±4.3	45±4.9	54±0.4
RCV1	NMI	35±0.0	36±0.0	39±0.4	36±3.1	43±4.1
	ARI	13±0.0	14±0.0	21±0.9	29±3.4	37±4.6
SPORTS	NMI	55±0.0	51±1.6	57±2.6	56±1.6	55±1.2
	ARI	28±0.0	23±0.4	37±1.6	37±1.5	36±1.1

This describes the problem of maximum entropy which occur quite frequently with balance FMMs. It follows that most mixture derived from proposition 4.2.1 in the previous section present this behavior. Note that FMMs with mixing proportions or unfixed variance are special cases which decrease this phenomenon but at the expense of computing more parameters. Another solution can be enhanced through the use of algorithms such as Classification EM (CEM) or K-means which by setting $\tilde{z}_{ik} \in \{0, 1\}$, essentially contributes to force $\widetilde{\boldsymbol{Z}}$ to have minimum entropy ($H(\widetilde{\boldsymbol{Z}}) = 0$) at each iteration and therefore bypass $-\sum_i^n D_{KL}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})$.

However, depending on the characteristics of the observed variable $\boldsymbol{x}_i$ and the nature of the distribution f assumed for the model (continuous or discrete), the entropy of $f(\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ will not always be maximum when the proportions are not specified. For instance, it is well known that the Gaussian distribution with fixed mean and variance or the Gamma distribution with fixed mean have maximum entropy. In the case of a directional distribution such as the von Mises-Fisher where $\boldsymbol{x}_i \in \mathbb{S}^{d-1}$ by

definition, the entropy is maximum for a fixed mean and concentration [247]. For the Poisson and Binomial distributions, the maximum entropy is observed on suitably defined sets [248, 249]. In our situation where the observations $\boldsymbol{x}_i$ are highly sparse (resulting in low means around 0) and normalized s.t. $\boldsymbol{x}_i \in \mathbb{S}^{d-1}$, the Poisson probability mass function $f(\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ has a low entropy. This behavior was illustrated in the previous section of this chapter where we displayed the values of the Poisson pmf and the Gaussian pdf for a sparse and normalized discrete Poisson random variable $\hat{\mathbf{x}} \in \mathbb{R}_+^{1000}$. Therefore, the conditional probability $\tilde{z}_{ik}$ will naturally tend toward 0 or 1, bypassing the effect of $-\sum_i^n D_{KL}(\tilde{\boldsymbol{z}}_i||\boldsymbol{\pi})$ in eq(4.108), assuming that the proportions are equal. Moreover, in these conditions, a CEM will instantaneously be trapped at the first iteration.

In this sense, it is appealing to study the performance of a parsimonious Poisson FMM for sparse random variables in the unit-sphere, and the effect of the unit-norm in NMF to know which method benefits more from that practice.

4.2.6 Optimization of $\texttt{cNMF}_\texttt{H}$ with $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ obtained from the I-divergence

To verify Lemma 4.2.1, we minimize $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ w.r.t. the constraints formulated in problem(4.75). This optimization requires the definition of the following Lagrangian function :

$$\mathcal{L}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) + \sum_i^n \gamma_i \Big(\sum_k^g \tilde{z}_{ik} - 1 \Big) + \mathrm{Tr}(\boldsymbol{\epsilon} \widetilde{\boldsymbol{Z}}^\top) + \mathrm{Tr}(\boldsymbol{\beta} \boldsymbol{W}^\top), \tag{4.109}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times g}$, and $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$ are the Lagrange multipliers. Its gradient w.r.t each factor are denoted as follows :

$$\nabla_{\tilde{z}_{ik}} \mathcal{L} = -(\boldsymbol{X} \log \boldsymbol{W})_{ik} + \sum_j^d w_{jk} + 1 + \log \tilde{z}_{ik} + \gamma_i + \epsilon_{ik}, \tag{4.110}$$

$$\nabla_{w_{jk}} \mathcal{L} = -\frac{(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk}}{w_{jk}} + \sum_i^n \tilde{z}_{ik} + \beta_{jk}. \tag{4.111}$$

Setting these gradients to zero and making use of the Karush-Kuhn-Tucker conditions $\epsilon \odot \widetilde{\boldsymbol{Z}} = 0$, $\beta \odot \boldsymbol{W} = 0$ lead to the following stationary equations :

$$\tilde{z}_{ik} (\boldsymbol{X} \log \boldsymbol{W})_{ik} - \tilde{z}_{ik} \Big(\sum_j^d w_{jk} + 1 + \log \tilde{z}_{ik} + \gamma_i \Big) = 0, \tag{4.112}$$

$$(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk} - w_{jk} \sum_i^n \tilde{z}_{ik} = 0. \tag{4.113}$$

Since the gradient of $\mathcal{Q}$ w.r.t $\tilde{z}_{ik}$ will now leads to the term $(\boldsymbol{X}\log\boldsymbol{W})_{ik} \in \mathbb{R}$ due to the logarithm, deriving a multiplicative update w.r.t the positivity of $\widetilde{\boldsymbol{Z}}$ using the KTT conditions becomes difficult. In addition, $\log\tilde{z}_{ik}$ is also negative. ϵ can be cancelled out and a closed-form expression for $\widetilde{\boldsymbol{Z}}$ is obtained from the gradient of H namely $1+\log\tilde{z}_{ik}$. Consequently, we obtain the following update rules forming an Expectation-Minimization procedure :

$$\tilde{z}_{ik} \leftarrow \frac{e^{(\boldsymbol{X}\log\boldsymbol{W})_{ik}-\sum_j^d w_{jk}}}{e^{1+\gamma_i}}, \tag{4.114}$$

$$w_{jk} \leftarrow \frac{(\boldsymbol{X}^\top\widetilde{\boldsymbol{Z}})_{jk}}{\sum_i^n \tilde{z}_{ik}}, \tag{4.115}$$

where $e^{1+\gamma_i} = \sum_k^g [e^{(\boldsymbol{X}\log\boldsymbol{W})_{ik}-\sum_j^d w_{jk}}]_{ik}$. This is equivalent to an EM algorithm derived from the negative fuzzy criterion (4.78).

4.2.7 Spherical NMF (SpNMF)

In the likes of Spherical PCA, several attempts in linear dimensional reduction working toward a transformation of the data in the unit-sphere are denoted [250, 251]. While their transformations lie in $\mathbb{R}$, we propose an approach which keeps the reconstruction data in the nonnegative real space $\mathbb{R}^+$ faithful to the original data space.

Recalling that $\boldsymbol{Z} \in \mathbb{R}_+^{n\times d}$, we define the problem of NMF where $\|\boldsymbol{x}_i\| = \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\| = 1$ and $\mathcal{D}(\boldsymbol{X}, \boldsymbol{Z}\boldsymbol{W}^\top)$ is $(1-\cos)$ dissimilarity as follows :

$$\min_{\substack{\boldsymbol{Z}\geq 0, \boldsymbol{W}\geq 0 \\ \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|=1, \forall i=1,\dots,n}} \{\mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) = \sum_i^n \frac{1}{2}\|\boldsymbol{x}_i - [\boldsymbol{Z}\boldsymbol{W}^\top]_i\|_F^2\}, \tag{4.116}$$

where

$$\mathcal{F}(\boldsymbol{Z},\boldsymbol{W}) = \sum_i^n \frac{1}{2}\Big(\|\boldsymbol{x}_i\|^2 + \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^2 - 2\sum_j^d x_{ij}\sum_k^g z_{ik}w_{jk}\Big) \tag{4.117}$$

collapses to $\sum_i^n(1-\sum_j^d x_{ij}\sum_k^g z_{ik}w_{jk})$. Note that the optimization of problem(4.116) requires that the reconstruction matrix $\boldsymbol{Z}\boldsymbol{W}^\top$ lies in the unit-sphere s.t. $\forall i = 1,\dots,n$, $\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^2 = \sum_j^d \big(\sum_k^g z_{ik}w_{jk}\big)^2 = \sum_{j,k,k'} z_{ik}w_{jk}z_{ik'}w_{jk'} = [\boldsymbol{Z}(\boldsymbol{W}^\top\boldsymbol{W})\boldsymbol{Z}^\top]_{ii} = 1$.

4.2.7.1 Optimization

Minimizing $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ w.r.t the constraints formulated in problem(4.116) requires the definition of the Lagrangian function

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{F}(\boldsymbol{Z}, \boldsymbol{W}) + \sum_{i}^{n} \gamma_i (\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\| - 1) + \mathrm{Tr}(\boldsymbol{\epsilon}\boldsymbol{Z}^\top) + \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \tag{4.118}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times g}$, and $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$ are the Lagrange multipliers. The gradients of $\mathcal{L}$ w.r.t z_{ik} and w_{jk} are stated as follows :

$$\nabla_{z_{ik}} \mathcal{L} = -(\boldsymbol{X}\boldsymbol{W})_{ik} + \gamma_i \frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}]_i\|} + \epsilon_{ik}, \tag{4.119}$$

$$\nabla_{w_{jk}} \mathcal{L} = -(\boldsymbol{X}^\top\boldsymbol{Z})_{jk} + \sum_{i}^{n} \gamma_i \frac{(\boldsymbol{W}\boldsymbol{Z}^\top)_{ji} z_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} + \beta_{jk}. \tag{4.120}$$

Setting these gradients to zero and making use of the Karush-Kuhn-Tucker conditions $\epsilon \odot \boldsymbol{Z} = 0$, $\beta \odot \boldsymbol{W} = 0$ lead to the following stationary equations :

$$z_{ik}(\boldsymbol{X}\boldsymbol{W})_{ik} - z_{ik}\gamma_i \frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}]_i\|} = 0, \tag{4.121}$$

$$w_{jk}(\boldsymbol{X}^\top\boldsymbol{Z})_{jk} - w_{jk} \sum_{i}^{n} \gamma_i \frac{(\boldsymbol{W}\boldsymbol{Z}^\top)_{ji} z_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} = 0. \tag{4.122}$$

The update rule of z_{ik} and w_{jk} takes the following form :

$$z_{ik} \leftarrow z_{ik} \frac{(\boldsymbol{X}\boldsymbol{W})_{ik} \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|}{\gamma_i (\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}, \tag{4.123}$$

$$w_{jk} \leftarrow w_{jk} \frac{(\boldsymbol{X}^\top\boldsymbol{Z})_{jk}}{\sum_i^n \gamma_i \frac{(\boldsymbol{W}\boldsymbol{Z}^\top)_{ji} z_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|}}. \tag{4.124}$$

Let $\hat{z}_{ik} = z_{ik} \frac{(\boldsymbol{X}\boldsymbol{W}^\top)_{ik}}{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}$, replacing z_{ik} with eq(4.123) into the constraint gives :

$$\begin{aligned} \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\| &= 1 \\ \sqrt{\sum_j \left(\frac{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|)}{\gamma_i} \hat{z}_{ik} w_{jk} \right)^2} &= 1 \implies \gamma_i = \|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\| \sqrt{\sum_j^d (\hat{z}_{ik} w_{jk})^2}. \end{aligned}$$

Plugging γ_i into eq(4.123) and eq(4.124) gives :

$$z_{ik} \leftarrow z_{ik} \frac{(\boldsymbol{XW})_{ik}}{(\boldsymbol{ZW}^\top \boldsymbol{W})_{ik} \sqrt{\sum_j^d (\hat{z}_{ik} w_{jk})^2}}, \quad (4.125) \qquad w_{jk} \leftarrow w_{jk} \frac{(\boldsymbol{X}^\top \boldsymbol{Z})_{jk}}{\sum_i^n (\boldsymbol{WZ}^\top)_{ji} z_{ik} \sqrt{\sum_j^d (\hat{z}_{ik} w_{jk})^2}}. \quad (4.126)$$

Remark. The normalization of $[\boldsymbol{ZW}^\top]$ in the unit-sphere is subject to a row multiplier γ_i only. Consequently, its estimation from one derivative is a sufficient condition since it acts as a constant for every column vector.

The optimization procedure is given by Algorithm (19).

Algorithm 19 `SpNMF`

Input : $\boldsymbol{X}$, g, $\boldsymbol{Z}^{(0)}$; $\boldsymbol{W}^{(0)}$.
Output : $\boldsymbol{Z}$ and $\boldsymbol{W}$.
Normalize $\boldsymbol{X}$ s.t. $\boldsymbol{x}_i = \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}$, $\forall i = 1, \ldots, n$
repeat
1. update $\boldsymbol{Z}$ with eq(4.125) ;
2. update $\boldsymbol{W}$ with eq(4.126) ;
until convergence

The convergence analysis is given in Section 4.2.7.2.

4.2.7.2 Convergence analysis for `SpNMF`

Theorem 4.2.5. *$\mathcal{F}(\boldsymbol{Z}, \boldsymbol{W})$ in problem(4.116) is non-increasing under eq(4.125) and eq(4.126).*

Considering the Lagrangian function for problem(4.116) defined as :

$$\mathcal{L}(z_{ik}) = \sum_i^n \Big(1 - \sum_j^d x_{ij} \sum_k^g z_{ik} w_{jk}\Big) + \sum_i^n \gamma_i \left(\sqrt{\sum_j^d \Big(\sum_k^g z_{ik} w_{jk} \Big)^2} - 1 \right), \quad (4.127)$$

and following Definition (4.2.2), we formulate the following proposition.

Proposition 4.2.3.

$$\mathcal{H}(z_{ik}, z_{ik}^{(t)}) = \mathcal{L}(z_{ik}^{(t)}) + \mathcal{L}'(z_{ik}^{(t)})(z_{ik} - z_{ik}^{(t)}) + \frac{\gamma_i (\boldsymbol{ZW}^\top \boldsymbol{W})_{ik}}{2\|[\boldsymbol{ZW}^\top]_i\|} \Big(z_{ik} - z_{ik}^{(t)} \Big)^2, \quad (4.128)$$

where $\mathcal{L}'(z_{ik}^{(t)}) = -(\boldsymbol{XW})_{ik} + \gamma_i \frac{(\boldsymbol{ZW}^\top \boldsymbol{W})_{ik}}{\|[\boldsymbol{ZW}]_i\|}$ is an auxiliary function for $\mathcal{L}(z_{ik})$.

Proof. It is straightforward to verify that $\mathcal{H}(z_{ik}, z_{ik}) = \mathcal{L}(z_{ik})$. We will demonstrate that $\mathcal{H}(z_{ik}, z_{ik}^{(t)}) \geq \mathcal{L}(z_{ik})$ by using the second order Taylor expansion of $\mathcal{L}(z_{ik})$ denoted as follows :

$$\mathcal{L}(z_{ik}) = \mathcal{L}(z_{ik}^{(t)}) + \mathcal{L}'(z_{ik}^{(t)})(z_{ik} - z_{ik}^{(t)}) + \frac{\gamma_i}{2}\left[\frac{(\boldsymbol{W}^\top\boldsymbol{W})_{kk}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} - \frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}^2}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^3}\right]\left(z_{ik} - z_{ik}^{(t)}\right)^2. \tag{4.129}$$

From $(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik} = \sum_{k'}^{g} z_{ik'}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{k'k}$,

$$\begin{aligned}\frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} \geq& \frac{z_{ik}^{(t)}(\boldsymbol{W}^\top\boldsymbol{W})_{kk}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} - \frac{z_{ik}^{(t)}(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}^2}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^3} \\ \frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}{z_{ik}^{(t)}\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} \geq& \frac{(\boldsymbol{W}^\top\boldsymbol{W})_{kk}}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|} - \frac{(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}^2}{\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^3}.\end{aligned} \tag{4.130}$$

From this inequality, we have that $\mathcal{H}(z_{ik}, z_{ik}^{(t)}) \geq \mathcal{L}(z_{ik})$. □

Proof of Theorem 4.2.5. In order to satisfy Lemma 4.2.4, we compute the gradient of $\mathcal{H}(z_{ik}, z_{ik}^{(t)})$ w.r.t z_{ik} :

$$\nabla_{z_{ik}}\mathcal{H} = \mathcal{L}'(z_{ik^{(t)}}) + \frac{\gamma_i(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}}{2z_{ik}^{(t)}\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|}\left(2z_{ik} - 2z_{ik}^{(t)}\right). \tag{4.131}$$

Setting this gradient to zero leads to :

$$z_{ik}^{(t+1)} = z_{ik}^{(t)}\frac{(\boldsymbol{X}\boldsymbol{W})_{ik}}{\gamma_i(\boldsymbol{Z}\boldsymbol{W}^\top\boldsymbol{W})_{ik}\|[\boldsymbol{Z}\boldsymbol{W}^\top]_i\|^{-1}}. \tag{4.132}$$

Since $\mathcal{H}(z_{ik}, z_{ik}^{(t)})$ is an auxiliary function for $\mathcal{L}(z_{ik})$, $\mathcal{L}(z_{ik})$ is non-increasing under this update. By reversing z_{ik} and w_{jk}, $\mathcal{L}(w_{jk})$ can be shown similarly to be non-increasing under the update rules of w_{jk}.

■

4.2.7.3 Spherical `cNMF`

In the case of `cNMF` with the $(1 - \cos)$ dissimilarity, since the normalization constraint vanishes the convex part of the Euclidean distance, the probability constraint on $\widetilde{\boldsymbol{Z}}$ is a sufficient condition for producing the equivalence of a fuzzy Spherical K-means.

Proposition 4.2.4. Solving the following minimization problem :

$$\min_{\substack{\widetilde{\boldsymbol{Z}}\geq 0, \boldsymbol{W}\geq 0, \widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n \\ \|[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i\| = 1, \forall i = 1,\ldots,n}} \{\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \sum_{i}^{n}\frac{1}{2}\|\boldsymbol{x}_i - [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i\|_F^2\}, \tag{4.133}$$

is directly equivalent to minimizing the Spherical K-means algorithm.

Proof. Since $\widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n$, $\|[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i\| = 1 \implies \|\boldsymbol{w}_k\| = 1$

$$\mathcal{F}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \sum_i^n \Big(1 - \sum_j^d x_{ij} \sum_k^g \tilde{z}_{ik} w_{jk}\Big) = \sum_{i,k}^{n,g} \tilde{z}_{ik} \Big(1 - \sum_j^d x_{ij} w_{jk}\Big) \tag{4.134}$$

which is the Spherical K-means criterion. □

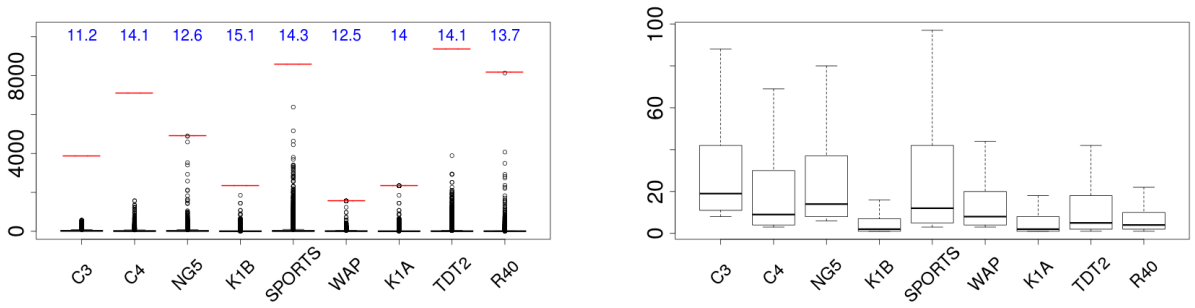
4.3 Conclusion

We proposed a regularized NMF algorithm which brings major improvements for the task of document clustering. More specifically, we have seen that embedding the clustering into a probabilistic factor and applying an entropic regularization increase clustering validity (lack of uncertainty). Furthermore, we have shown that the algorithm, whose convergence is guaranteed, is less sensitive to initialization than the classical $\texttt{NMF} - \texttt{KL}$ algorithm. In addition, using the properties of convex functions and Bregman divergences, we have established the connection between $\texttt{cNMF}_\texttt{H}$ and FMMs of Exponential distributions, and proposed a straightforward comparison between both methods. We highlighted that the underlying Poisson mixture model with data in the unit-sphere could match and particularly overcome $\texttt{cNMF}_\texttt{H}$ on balanced datasets. Moreover, the cheaper computation of its gradient allows for larger scalability. In addition, thanks to this property, we have shown that the Poisson FMM requires as much or less parameters than a FMM using a dedicated directional distribution such as the von Mises-Fisher to give good partitions. Following the great interest shown by this distribution in FMMs, in the next chapter, we will study its performance in the context of finite block mixture modeling through the Latent Block Model.

Chapitre 5

Gamma-Poisson Latent Block Model for noisy text data

In text mining, where data are usually high-dimensional, co-clustering has shown its ability to handle and simplify the interpretation of large, complex sparse structures such as document-term datasets. Due to its flexibility, the *Latent Block Model* is undoubtedly a good way of dealing with this kind of data, allowing a parameterization that is appropriate to the underlying structure of the particular data in question. The Sparse Poisson Latent Block Model can identify a diagonal mixture of blocks that favors co-clustering. However, in text analysis, noisy terms features often arise as the amount of data grows making the learning process less efficient. Good pre-processing techniques tend to be time-consuming, and as a consequence are not always used. In this chapter, by exploiting the flexibility of LBM, we propose a new model-based co-clustering for analyzing document-term matrices and tackling the automatic recognition of noisy term features during the learning phase. Furthermore, we propose a suitable Bayesian version of the model in order to address the overfitting mixture issue encountered with finite mixture models. As a novelty, we investigate the impact of the prior on the model clustering performance and takes advantages of the parsimony of our model to introduce additional hyperparameters for the component parameters priors.



(a) (b)

FIGURE 5.1 – (a) Boxplots of $\boldsymbol{m}$ for each dataset (with outliers). The percentage of outliers is indicated in blue. The red line indicates the number of documents. (b) Boxplot of $\boldsymbol{m}$ per datasets (without outliers).

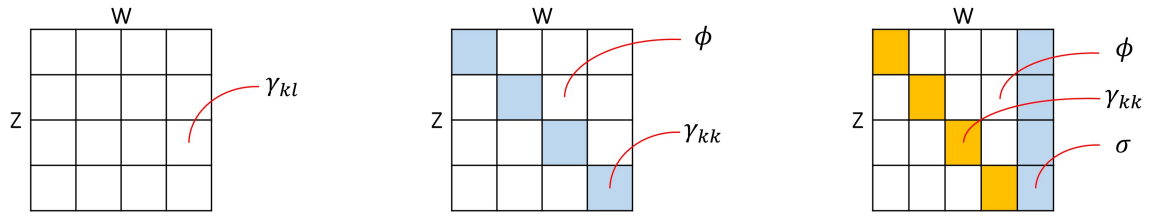
5.1 Capturing noisy features in diagonal document-term co-clustering

5.1.1 Motivations

In a case of text analysis where the data arise in the form of highly dimensional and sparse document-term matrix, co-clusters will often lack of heterogeneity because of the substantial prevalence of zeros in each block, reflecting the overall sparsity and resulting in no meaningful lower interpretations/summary. To overcome this deficiency, an effective method known as diagonal co-clustering has been proposed [252]. This method attempts to identify an optimal block diagonal structure of samples and features. Diagonal co-clustering does not eliminate words but instead puts more focus on the most discriminating words so as to reach a good separability in terms of document clustering. This is natural since documents are grouped together because they share similar words, which induces a block diagonal structure, as it has been demonstrated by several recent works [253, 254]. The present contribution addresses an additional but major issue relating to text analysis, namely the presence of noisy terms, which can impact the learning and the clustering quality negatively (see [255]). More precisely, we refer to the concept of noise based upon to the prevalence of a term within the entire collection of documents, independently of the document clustering. In practice, several pre-processing steps can be applied to filter and remove the most common and irrelevant terms from the set (e.g. stopwords). However, these processes can be overshadowed. For instance, due to the substantial amount of documents, generic lexical stopwords are frequently used instead of domain-specific stopwords. Furthermore, normalization steps (e.g. such as *stemming* or *lemmatization*), which replace specific terms

with their roots naturally increase the prevalence in the remaining set of terms. Indeed, a root will be more common than its descendants but the later were perhaps not as common or irrelevant). Also, a threshold or interval of citations could be employed to label terms as noisy. However, with no a priori knowledge on the document partitions, this could lead to removing cluster-specific terms, sensibly useful for learning accurate partitions. Overall, there are different methods for cleaning textual data, which may be selected according to users' requirements. Careful cleaning can have a substantial time cost when processing large amounts of data, and will sometimes be avoided by users who wish to obtain results as quickly as possible. In the following, we illustrate the presence of noise in a document-term matrix. Let $\boldsymbol{M} = (m_{ij})$ be the presence-absence (binary) matrix derived from $\boldsymbol{X}$. Let $m_{\cdot j} = \sum_{i=1}^{n} m_{ij}$, we denote the vector of column marginals $\boldsymbol{m} = (m_{\cdot 1}, \ldots, m_{\cdot d})$ which indicates for each term j, the number of documents in which it appears.

Figure 5.1(a) and 5.1(b) display the boxplots of $\boldsymbol{m}$ for several *benchmarking* and *pre-processed* document-term matrices which will be described and utilized later in the manuscript. Figure 5.1(a) also includes the outliers and their proportions (in percentage) for each dataset. Using the upper whisker as statistical threshold, Figure 5.1(a) shows that there is a small percentage of terms which appear in a large majority of documents, sometimes in every documents. On the other hand, Figure 5.1(b) highlights the main distributions of these marginals with the quartiles suggesting that overall, 50% of the terms are cited in less than 20 documents, and 75% in less than 40 documents. These observations questions how to define what is an excessive amount of citations and stresses out the suggestion that this factor should be learned having no information on the optimal partition. Therefore, in this contribution, we provide an approach that can overcome a weak or bad pre-processing by estimating the probability of a term to be noisy. To this end, we harness the flexibility of the Latent Block Model (LBM) in introducing a new parameterization pattern for identifying a column structure of noisy terms as well as a diagonal structure into dense blocks (see Figure 5.2(c)). The derived model, called *Gamma Poisson LBM* and referred to as GPLBM, is particularly suitable for co-clustering of sparse data with or without noise while introducing inference for the noisy term-cluster to decide automatically which terms will be affected.



(a) PLBM (b) SPLBM (c) GPLBM

FIGURE 5.2 – Different variants of LBM : SPLBM taking into account sparsity, and GPLBM taking into account sparsity and noise.

5.1.2 Related Works

In the field of unsupervised learning, several contributions using *feature selection* are proposed [256, 255]. In contrast, the approach introduced in this paper does not explicitly select and remove noisy features from the learning set but consists in identifying a subset of noisy features to facilitate the learning of parameters. The key materials of our contribution can be traced backed to the work on co-clustering carried out in [80, 257] . The authors introduced a model-based approach called the Latent Block Model (LBM) which aims at identifying a couple of partitions $(\boldsymbol{Z}, \boldsymbol{W})$ from $\boldsymbol{X}$. The partition $\boldsymbol{Z} = [\boldsymbol{z}_1|\ldots|\boldsymbol{z}_n]^\top$ is a latent variable indicating the cluster membership of each element $i \in I$ among g clusters using a binary vector $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ig})^\top \in \{0,1\}^g | \sum_k^g z_{ik} = 1, \forall i = 1, \ldots, n$; $\boldsymbol{W} = [\boldsymbol{w}_1|\ldots|\boldsymbol{w}_d]^\top$ is a latent variable indicating the cluster membership of each $j \in J$ among c clusters where $\boldsymbol{w}_j = (w_{j1}, \ldots, w_{jc})^\top \in \{0,1\}^c | \sum_\ell^c w_{j\ell} = 1, \forall j = 1, \ldots, d$. $(\boldsymbol{Z},\boldsymbol{W})$ can be referred to as the classification matrices. For convenience, they will sometimes be expressed as categorical vectors denoted $\mathbf{z} = (\mathrm{z}_\mathbf{1}, \ldots, \mathrm{z}_\mathbf{n}) \in \{\mathbf{1}, \ldots, \mathbf{g}\}^\mathbf{n}$ and $\mathbf{w} = (\mathrm{w}_\mathbf{1}, \ldots, \mathrm{w}_\mathbf{d}) \in \{\mathbf{1}, \ldots, \mathbf{c}\}^\mathbf{d}$ which indicate the group label for each element $i \in I$ and $j \in J$ respectively.

The marginal density is denoted as follows :

$$p(\boldsymbol{X}; \boldsymbol{\Theta}) = \sum_{\boldsymbol{Z} \in \mathcal{Z}, \boldsymbol{W} \in \mathcal{W}} p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}), \tag{5.1}$$

where $\mathcal{Z}$ and $\mathcal{W}$ denote the set of all possible partitions I and J respectively. Learning the double latent structure of LBM is quite challenging as $p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})$ is intractable. To this end, we restrict $(\boldsymbol{Z}, \boldsymbol{W})$ to be independent. $p(\boldsymbol{Z}; \boldsymbol{\Theta})$ and $p(\boldsymbol{W}; \boldsymbol{\Theta})$ are set as Multinomial distributions with parameters $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c)$ s.t. $\pi_k = p(z_{ik} = 1), \forall i \in I$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c), \rho_\ell = p(w_{j\ell} = 1), \forall j \in J$. The observation x_{ij} are assumed to be drawn independently. $p(\boldsymbol{X}; \boldsymbol{\Theta})$ from Eq(5.1) is now

equal to

$$\sum_{(\boldsymbol{Z},\boldsymbol{W})\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k}^{n,g} \pi_k^{z_{ik}} \prod_{j,\ell}^{d,c} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell}^{n,d,g,c} f(x_{ij};\boldsymbol{\theta}_{k\ell})^{z_{ik}w_{j\ell}},$$

where $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\theta})$ and $\boldsymbol{\theta}$ is the set of parameters for the density f. In this paper, we get interest in an application of this model for contingency tables, called the Poisson Latent Block Model (PLBM, see Figure 5.2(a)), which additionally, takes into account the independence structure of these tables. In PLBM, λ_{ij} is assumed to be drawn by block s.t. $\lambda_{ij} = x_{i.}x_{.j}\gamma_{\mathrm{z}_i\mathrm{w}_j}$. $f(x_{ij};\boldsymbol{\theta}_{\mathrm{z}_i\mathrm{w}_j})$ is therefore set as the Poisson probability mass function s.t. $\mathcal{P}(x_{ij};\lambda_{ij}) = \frac{\lambda_{ij}^{x_{ij}} e^{-\lambda_{ij}}}{x_{ij}!}$ and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\gamma})$ where $\boldsymbol{\gamma} = (\gamma_{k\ell}) \in \mathbb{R}_+^{g\times c}$ are the block parameters and $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are set as the marginals of $\boldsymbol{X}$ s.t. $\boldsymbol{\mu} = (x_{1.}, \ldots, x_{n.})$ and $\boldsymbol{\nu} = (x_{.1}, \ldots, x_{.d})$ given that $x_{i.} = \sum_j^d x_{ij}$ and $x_{.j} = \sum_i^n x_{ij}$. The identifiability of the model is discussed in [257].

More recently, Ailem et al. [254] introduced the Sparse Poisson Latent Block Model (SPLBM, see Figure 5.2(b)) to identify a diagonal structure with dense co-clusters. This leads to posit the that blocks inside the diagonal get their respective parameters while those outside share the same parameter.

In PLBM and SPLBM, the authors considered the Variational Maximum Likelihood Estimation (MLE) to estimate the parameters of the model. However, LBM as most mixture models is subject to mixture overfitting. To address this issue, we will adopt a similar approach to [258] with the Multinomial LBM and use Variational Bayesian inference. Additionally, Markov Chain Monte Carlo (MCMC) techniques will be exploited to simulate estimates from the exact posterior since Variational Bayesian inference only provides a locally optimal approximation of the joint posterior probability.

5.1.3 GPlbm model

As described in Figure 5.2(c), GPLBM dedicates the rightmost column of blocks for the noisy observations and a diagonal structure into dense blocks. An observation x_{ij} follows a conditional distribution depending on the value of $\boldsymbol{z}_i$ and $\boldsymbol{w}_j$. To achieve Bayesian inference, a non-informative prior distribution is provided for each parameter of the model present in $\boldsymbol{\Theta}$ respectively, to learn the posterior distribution. Therefore, we denote the Gamma distribution as a conjugate prior for the Poisson distribution, and the Dirichlet distribution for the Multinomial s.t. the sampling process of

GPLBM $\forall i = 1, \ldots, n, \forall j = 1, \ldots, d$ and $\forall k = 1, \ldots, g, \forall \ell = 1, \ldots, g$ is denoted as :

$$\begin{aligned}
\boldsymbol{z}_i|\boldsymbol{\pi} &\sim \mathcal{M}(1, \boldsymbol{\pi}),\ \boldsymbol{\pi} \sim \mathcal{D}(\boldsymbol{a}),\\
\boldsymbol{w}_j|\boldsymbol{\rho} &\sim \mathcal{M}(1, \boldsymbol{\rho}),\ \boldsymbol{\rho} \sim \mathcal{D}(\boldsymbol{b}),\\
x_{ij}|z_{ik}w_{jc} = 1 &\sim \mathcal{P}(x_{i.}x_{.j}\sigma),\ \sigma \sim \mathcal{G}(\zeta, \eta),\\
x_{ij}|z_{ik}w_{jk} = 1 &\sim \mathcal{P}(x_{i.}x_{.j}\epsilon_k),\ \epsilon_k \sim \mathcal{G}(\alpha_k, \beta_k),\\
x_{ij}|z_{ik}w_{j\ell} = 1 &\sim \mathcal{P}(x_{i.}x_{.j}\phi),\ \forall \ell \neq k,\ \phi \sim \mathcal{G}(\tau, \upsilon),
\end{aligned}$$

where $c = g + 1$, $\boldsymbol{a} = (a_1, \ldots, a_g)$, $\boldsymbol{b} = (b_1, \ldots, b_c)$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_g)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_g)$, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_g)$ is the set of diagonal block parameters, ϕ is the parameter for the blocks outside the diagonal and σ is the parameter for the rightmost column of blocks. The conditional probability of $\boldsymbol{X}$ relative to Eq(5.1) is expressed as follows :

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\theta}) = \prod_{i,j,k}^{n,d,g} \mathcal{P}(x_{ij}; x_{i.}x_{.j}\sigma)^{z_{ik}w_{jc}} \times \prod_{i,j,k}^{n,d,g} \mathcal{P}(x_{ij}; x_{i.}x_{.j}\epsilon_k)^{z_{ik}w_{jk}} \times \prod_{i,j,k,\ell\neq k}^{n,d,g,c-1} \mathcal{P}(x_{ij}; x_{i.}x_{.j}\phi)^{z_{ik}w_{j\ell}}, \tag{5.2}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\nu}, \sigma, \boldsymbol{\epsilon}, \phi)$. Figure 5.3 reports the graphical model of GPLBM.

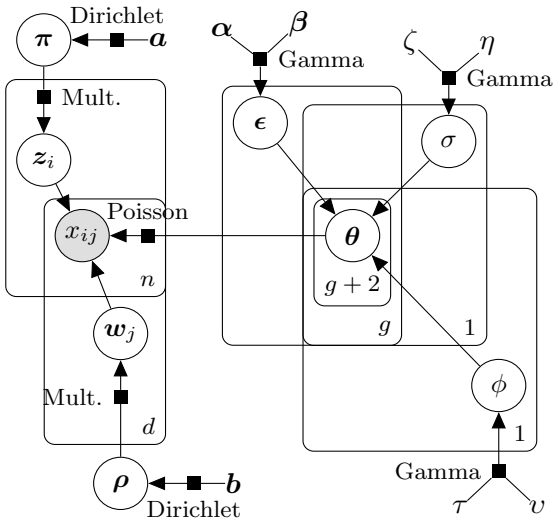


FIGURE 5.3 – GPLBM as a graphical model.

5.1.3.1 Inference and algorithms

To simplify the notation and avoid the computation of $\boldsymbol{Z}$ and $\boldsymbol{W}$ w.r.t. $(\sigma, \epsilon_k|\forall k = 1, \ldots, g, \phi)$ independently, we use the following settings. Let $\boldsymbol{\gamma} = (\gamma_{k\ell}) \in \mathbb{R}_+^{g\times c}$ be the matrix of block parameters,

$\forall k = 1, \ldots, g$, $\forall \ell = 1, \ldots, c$,

$$\gamma_{k\ell} = \sigma, \forall k = 1, \ldots, g, \forall \ell = c, \tag{5.3}$$

$$\text{diag}(\boldsymbol{\gamma}) = \boldsymbol{\epsilon}, \tag{5.4}$$

$$\gamma_{k\ell} = \phi, \forall k = 1, \ldots, g, \forall \ell \neq k | \ell \in \{1, \ldots, c-1\}. \tag{5.5}$$

As mentioned earlier, a Variational Bayes Expectation-Maximization (EM) algorithm was introduced in [259] to estimate $\boldsymbol{\Theta}$ for the Multinomial LBM. However, as pointed by the authors, this algorithm was subject to prominent overfitting. In this section we propose another variational Bayes EM (`VBEM`) algorithm which will be showed to be a much better alternative against mixture overfitting later in section 5.1.4. In the variational approximation, the maximization of the log-likelihood $\log p(\boldsymbol{X}; \boldsymbol{\Theta})$ is replaced by the maximization of the Evidence Lower bound (ELBO).

From the notion of complete-data, LBM includes a set of latent variables $(\boldsymbol{Z}, \boldsymbol{W})$ for which we want to estimate the probability given the observed data $\boldsymbol{X}$ s.t.

$$p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta}) = \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})}{p(\boldsymbol{X}; \boldsymbol{\Theta})}. \tag{5.6}$$

Since $p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta})$ is intractable, we introduced an arbitrary density denoted $q(\boldsymbol{Z}, \boldsymbol{W})$ to estimate $p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta})$. Using Eq(5.6), we have :

$$\begin{aligned}
\log p(\boldsymbol{X}; \boldsymbol{\Theta}) =& \log p(\boldsymbol{X}; \boldsymbol{\Theta}) \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \\
=& \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log p(\boldsymbol{X}; \boldsymbol{\Theta}) \\
=& \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})}{p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta})} \\
=& \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) q(\boldsymbol{Z}, \boldsymbol{W})}{p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta}) q(\boldsymbol{Z}, \boldsymbol{W})} \\
=& \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{q(\boldsymbol{Z}, \boldsymbol{W})}{p(\boldsymbol{Z}, \boldsymbol{W} | \boldsymbol{X}; \boldsymbol{\Theta})} \\
&+ \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})}{q(\boldsymbol{Z}, \boldsymbol{W})} \\
=& D_{KL}(q || p) + \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})}{q(\boldsymbol{Z}, \boldsymbol{W})}.
\end{aligned} \tag{5.7}$$

Since $D_{KL}(q||p) \geq 0$ with equality when $q = p$, it follows that :

$$\log p(\boldsymbol{X}; \boldsymbol{\Theta}) \geq \sum_{\boldsymbol{Z}, \boldsymbol{W}} q(\boldsymbol{Z}, \boldsymbol{W}) \log \frac{p(\boldsymbol{X} | \boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta}) p(\boldsymbol{Z}, \boldsymbol{W}; \boldsymbol{\Theta})}{q(\boldsymbol{Z}, \boldsymbol{W})} = \mathcal{L}(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta}), \tag{5.8}$$

where $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ is known as the evidence lower bound (ELBO) of the model marginal log-likelihood $\log p(\boldsymbol{X};\boldsymbol{\Theta})$. Therefore, maximizing $\log p(\boldsymbol{X};\boldsymbol{\Theta})$ by maximizing $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ can be considered whether the arbitrary density q manages to minimize $D_{KL}(q||p)$ properly. Tractability is achieved by restricting q to a class of "manageable" densities. Finding the latter for which $\boldsymbol{Z}$ and $\boldsymbol{W}$ can be dependent is also highly intractable. Therefore, we use the mean-field approximation so that $q(\boldsymbol{Z},\boldsymbol{W})$ is restricted and factorises independently s.t. $q(\boldsymbol{Z},\boldsymbol{W}) = q_{\mathrm{z}}(\boldsymbol{Z}) \times q_{\mathrm{w}}(\boldsymbol{W})$. Consequently, $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ can now be decoupled into several forms w.r.t. $q_{\mathrm{z}}(\boldsymbol{Z})$ or $q_{\mathrm{w}}(\boldsymbol{W})$, reminiscent of Hinton and Neal interpretation of EM [260]. The ELBO can now be expressed as follows :

$$\begin{aligned}\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta}) &= \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log\left(\frac{p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta})}{q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W})}\right) \\ &= \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta}) + H(\boldsymbol{Z}) + H(\boldsymbol{W}), \end{aligned} \tag{5.9}$$

where $H(\boldsymbol{Z}) = -\sum_{\boldsymbol{Z}} q_{\mathrm{z}}(\boldsymbol{Z}) \log q_{\mathrm{z}}(\boldsymbol{Z})$ and $H(\boldsymbol{W}) = -\sum_{\boldsymbol{W}} q_{\mathrm{w}}(\boldsymbol{W}) \log q_{\mathrm{w}}(\boldsymbol{W})$ are entropy functionals. In addition, $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ can also be expressed in terms of an expectation of $p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta})$ knowing one of the latent variable s.t.

$$\begin{aligned}\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta}) =& \sum_{\boldsymbol{Z}} q_{\mathrm{z}}(\boldsymbol{Z}) \log\left(\frac{\tilde{p}(\boldsymbol{X},\boldsymbol{Z};\boldsymbol{\Theta})}{q_{\mathrm{z}}(\boldsymbol{Z})}\right) + H(\boldsymbol{W}) \\ =& \mathcal{L}(q(\boldsymbol{Z}|\boldsymbol{W});\boldsymbol{\Theta}) + H(\boldsymbol{W}) + c_1, \end{aligned} \tag{5.10}$$

where c_1 is a normalizing constant added since $\tilde{p}(\boldsymbol{X},\boldsymbol{Z};\boldsymbol{\Theta}) = \exp\left(\sum_{\boldsymbol{W}} q_{\mathrm{w}}(\boldsymbol{W}) \log p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta})\right)$ is not a true density. Similarly, $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ can be expressed w.r.t. $\mathcal{L}(q(\boldsymbol{W}|\boldsymbol{Z});\boldsymbol{\Theta})$ as follows :

$$\begin{aligned}\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta}) =& \sum_{\boldsymbol{W}} q_{\mathrm{w}}(\boldsymbol{W}) \log\left(\frac{\tilde{p}(\boldsymbol{X},\boldsymbol{W};\boldsymbol{\Theta})}{q_{\mathrm{w}}(\boldsymbol{W})}\right) + H(\boldsymbol{Z}) \\ =& \mathcal{L}(q(\boldsymbol{W}|\boldsymbol{Z});\boldsymbol{\Theta}) + H(\boldsymbol{Z}) + c_2. \end{aligned} \tag{5.11}$$

where c_2 is a normalizing constant and $\tilde{p}(\boldsymbol{X},\boldsymbol{W};\boldsymbol{\Theta}) = \exp\left(\sum_{\boldsymbol{Z}} q_{\mathrm{z}}(\boldsymbol{Z}) \log p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta})\right)$. Therefore, it follows that maximizing $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ can be achieved by respectively maximizing $\mathcal{L}(q(\boldsymbol{Z}|\boldsymbol{W});\boldsymbol{\Theta})$ and $\mathcal{L}(q(\boldsymbol{W}|\boldsymbol{Z});\boldsymbol{\Theta})$ using two inner variational EM algorithms successively.

Development of $\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta})$ leads to :

$$\begin{aligned}
\mathcal{L}(q(\boldsymbol{Z},\boldsymbol{W});\boldsymbol{\Theta}) &= \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{W};\boldsymbol{\Theta}) + H(\boldsymbol{Z}) + H(\boldsymbol{W}) \\
&= \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log \left(\prod_{i,k}^{n,g} \pi_k^{z_{ik}} \prod_{j,\ell}^{d,c} \rho_\ell^{w_{j\ell}} \prod_{i,j,k,\ell}^{n,d,g,c} \mathcal{P}(x_{ij}; x_{i.} x_{.j} \gamma_{k\ell})^{z_{ik} w_{j\ell}} \right) \\
&\quad + H(\boldsymbol{Z}) + H(\boldsymbol{W}) \\
&= \sum_{i,k}^{n,g} \sum_{z_{ik}\in\{0,1\}} q_{\mathrm{z}}(z_{ik}) z_{ik} \log(\pi_k) + \sum_{j,\ell}^{d,c} \sum_{w_{j\ell}\in\{0,1\}} q_{\mathrm{w}}(w_{j\ell}) w_{j\ell} \log(\rho_\ell) \\
&\quad + \sum_{i,j,k,\ell}^{n,d,g,c} \sum_{z_{ik}\in\{0,1\}} q_{\mathrm{z}}(z_{ik}) z_{ik} \sum_{w_{j\ell}\in\{0,1\}} q_{\mathrm{w}}(w_{j\ell}) w_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.} x_{.j} \gamma_{k\ell}) \\
&\quad + H(\boldsymbol{Z}) + H(\boldsymbol{W}).
\end{aligned} \tag{5.12}$$

Since z_{ik} and $w_{j\ell}$ are binary indicators, their expectations lead to probabilities, i.e. $\sum_{z_{ik}\in\{0,1\}} q_{\mathrm{z}}(z_{ik}) z_{ik} = q(z_{ik}=1)$ and $\sum_{w_{j\ell}\in\{0,1\}} q_{\mathrm{w}}(w_{j\ell}) w_{j\ell} = q(w_{j\ell}=1)$. Moreover, the entropy can be easily re-written in terms of each integration as follows :

$$\begin{aligned}
H(\boldsymbol{Z}) + H(\boldsymbol{W}) &= -\sum_{i,k}^{n,g} \sum_{z_{ik}\in\{0,1\}} q_{\mathrm{z}}(z_{ik}) \log(q_{\mathrm{z}}(z_{ik})) - \sum_{j,\ell}^{d,c} \sum_{w_{j\ell}\in\{0,1\}} q_{\mathrm{w}}(w_{j\ell}) \log(q_{\mathrm{w}}(w_{j\ell})) \\
&= -\sum_{i,k}^{n,g} q_{\mathrm{z}}(z_{ik}=0) \log(q_{\mathrm{z}}(z_{ik}=0)) - \sum_{i,k}^{n,g} q_{\mathrm{z}}(z_{ik}=1) \log(q_{\mathrm{z}}(z_{ik}=1)) \\
&\quad - \sum_{j,\ell}^{d,c} q_{\mathrm{w}}(w_{j\ell}=0) \log(q_{\mathrm{w}}(w_{j\ell}=0)) - \sum_{j,\ell}^{d,c} q_{\mathrm{w}}(w_{j\ell}=1) \log(q_{\mathrm{w}}(w_{j\ell}=1)) \\
&= H_0(\boldsymbol{Z}) + H_1(\boldsymbol{Z}) + H_0(\boldsymbol{W}) + H_1(\boldsymbol{W}).
\end{aligned} \tag{5.13}$$

Let $\tilde{z}_{ik} = q(z_{ik}=1)$ and $\tilde{w}_{j\ell} = q(w_{j\ell}=1)$ denote the conditional probabilities s.t. $\sum_k^g \tilde{z}_{ik} = 1, \forall i = 1,\ldots,n$, $\sum_\ell^c \tilde{w}_{j\ell} = 1, \forall j = 1,\ldots,d$ where $\widetilde{\boldsymbol{Z}} = (\tilde{z}_{ik}) \in [0,1]^{n\times g}$ and $\widetilde{\boldsymbol{W}} = (\tilde{w}_{j\ell}) \in [0,1]^{d\times c}$; we denote the fuzzy ELBO as $\mathcal{L}(q(\widetilde{\boldsymbol{Z}},\widetilde{\boldsymbol{W}});\boldsymbol{\Theta})$ s.t.

$$\begin{aligned}
\mathcal{L}(q(\widetilde{\boldsymbol{Z}},\widetilde{\boldsymbol{W}});\boldsymbol{\Theta}) &= \sum_{i,k}^{n,g} \tilde{z}_{ik} \log(\pi_k) + \sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log(\rho_\ell) + \sum_{i,j,k,\ell}^{n,d,g,c} \tilde{z}_{ik} \tilde{w}_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.} x_{.j} \gamma_{k\ell}) \\
&\quad + H(\widetilde{\boldsymbol{Z}}) + H(\widetilde{\boldsymbol{W}}) + \Phi,
\end{aligned} \tag{5.14}$$

where $H(\widetilde{\boldsymbol{Z}}) + H(\widetilde{\boldsymbol{W}})$ effectively equals $H_1(\boldsymbol{Z}) + H_1(\boldsymbol{W})$ and $\Phi = H_0(\boldsymbol{Z}) + H_0(\boldsymbol{W})$. Since the expectation only depends on $q(z_{ik}=1)$ and $q(w_{j\ell}=1)$, Φ may be omit in various expressions of $\mathcal{L}(q(\widetilde{\boldsymbol{Z}},\widetilde{\boldsymbol{W}});\boldsymbol{\Theta})$ and for the following optimization problem. Maximizing the resulting $\mathcal{L}(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}});\boldsymbol{\Theta})$

and $\mathcal{L}(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}});\boldsymbol{\Theta})$ so that $\mathcal{L}(q(\widetilde{\boldsymbol{Z}},\widetilde{\boldsymbol{W}});\boldsymbol{\Theta})$ is maximized overall leads to solving the following maximization problems :

$$\max_{\widetilde{\boldsymbol{Z}}^\top\in(\Delta_g)^n} \mathcal{L}(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}});\boldsymbol{\Theta}) \tag{5.15}$$

$$\max_{\widetilde{\boldsymbol{W}}^\top\in(\Delta_c)^d} \mathcal{L}(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}});\boldsymbol{\Theta}) \tag{5.16}$$

where Δ_g and Δ_c are probability simplex-es defined respectively as $\Delta_g = \{\forall \tilde{\boldsymbol{z}}_i \in \mathbb{R}_+^g : \sum_k^g \tilde{z}_{ik} = 1\}$ and $\Delta_c = \{\forall \tilde{\boldsymbol{w}}_j \in \mathbb{R}_+^c : \sum_\ell^c \tilde{w}_{j\ell} = 1\}$. The respective associated Lagrangian functions are :

$$L(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}},\boldsymbol{\lambda}) = \mathcal{L}(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}});\boldsymbol{\Theta}) + \sum_i^n \lambda_i \Big(\sum_k^n \tilde{z}_{ik} - 1\Big), \tag{5.17}$$

$$L(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}},\boldsymbol{\omega}) = \mathcal{L}(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}});\boldsymbol{\Theta}) + \sum_j^d \omega_j \Big(\sum_\ell^c \tilde{w}_{j\ell} - 1\Big), \tag{5.18}$$

where $\boldsymbol{\lambda} = (\lambda_i) \in \mathbb{R}_+^n$ and $\boldsymbol{\omega} = (\omega_j) \in \mathbb{R}_+^d$ are the Lagrange multipliers. Differentiation w.r.t. $\tilde{z}_{ik}$ leads to :

$$\tilde{z}_{ik} = \frac{\pi_k \exp\big(\sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.}x_{.j}\gamma_{k\ell})\big)}{\exp(\lambda_i + 1)}. \tag{5.19}$$

Substituting this expression into the constraint yields

$$\exp(\lambda_i + 1) = \sum_k \pi_k \exp\big(\sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.}x_{.j}\gamma_{k\ell})\big),$$

and therefore

$$\tilde{z}_{ik} \propto \pi_k \exp\Big(\sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \mathcal{P}(x_{ij}; x_{i.}x_{.j}\gamma_{k\ell})\Big). \tag{5.20}$$

In the same manner, differentiation w.r.t. $\tilde{w}_{j\ell}$ leads to :

$$\tilde{w}_{ik} \propto \rho_\ell \exp\Big(\sum_{i,k}^{n,g} \tilde{z}_{ik} \log \mathcal{P}(x_{ij}; x_{i.}x_{.j}\gamma_{k\ell})\Big). \tag{5.21}$$

With Bayesian inference, $\boldsymbol{\Theta}$ is estimated by maximizing the posterior joint probability $p(\boldsymbol{\Theta}|\boldsymbol{X})$, which leads to the Maximum A Posteriori (MAP) estimate :

$$\hat{\boldsymbol{\Theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\Theta}} p(\boldsymbol{\Theta}|\boldsymbol{X}).$$

From the Bayes formula :

$$p(\boldsymbol{\Theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})}{p(\boldsymbol{X})},$$

it is straightforward to define an EM algorithm [90, 261]. The likelihood $p(\boldsymbol{X}|\boldsymbol{\Theta})$ is therefore augmented into a proper marginal distribution over the space of plausible parameters $\boldsymbol{\Theta}$. In the same way for the variational case, the augmented ELBO denoted $\mathcal{L}_A(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta})$ has a similar functional form that $\mathcal{L}(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta})$. For GPLBM, we have

$$\begin{aligned}
\mathcal{L}_A(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta}) &= \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})}{q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W})} \\
&= \mathcal{L}(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta}) + \sum_{\boldsymbol{Z},\boldsymbol{W}} q_{\mathrm{z}}(\boldsymbol{Z}) q_{\mathrm{w}}(\boldsymbol{W}) \log p(\boldsymbol{\Theta}) \\
&= \mathcal{L}(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta}),
\end{aligned} \tag{5.22}$$

where $p(\boldsymbol{\Theta})$ takes the following form :

$$\mathcal{D}(\boldsymbol{\pi}; \boldsymbol{a}) \times \mathcal{D}(\boldsymbol{\rho}; \boldsymbol{b}) \times \mathcal{G}(\sigma; \zeta, \eta) \times \mathcal{G}(\phi; \tau, \upsilon) \times \prod_{k=1}^{g} \mathcal{G}(\epsilon_k; \alpha_k, \beta_k).$$

This holds also for $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ and additionally $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}}); \boldsymbol{\Theta}) = \mathcal{L}(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}}); \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta})$, $\mathcal{L}_A(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}}); \boldsymbol{\Theta}) = \mathcal{L}(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}}); \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta})$ (up to a constant).

Development of $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ leads to :

$$\begin{aligned}
\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta}) = &\sum_{i,k}^{n,g} \tilde{z}_{ik} \log \pi_k + \sum_{k}^{g} \Big[(a_k - 1) \log \pi_k + \log \Gamma\Big(\sum_{k}^{g} a_k \Big) \Big] - \sum_{k}^{g} \log \Gamma(a_k) \\
&+ \sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \rho_\ell + \sum_{\ell}^{c} \Big[(b_\ell - 1) \log \rho_\ell + \log \Gamma\Big(\sum_{\ell}^{c} b_\ell \Big) \Big] - \sum_{\ell}^{c} \log \Gamma(b_\ell) \\
&+ \sum_{i,j,k}^{n,d,g} \Big(\big[\tilde{z}_{ik} \tilde{w}_{jc} x_{ij} + (\zeta - 1) \big] \log(\sigma) - \sigma(\tilde{z}_{ik} \tilde{w}_{jc} x_{i.} x_{.j} + \eta) \Big) \\
&+ \sum_{i,j,k}^{n,d,g} \Big(\big[\tilde{z}_{ik} \tilde{w}_{jk} x_{ij} + (\alpha_k - 1) \big] \log(\epsilon_k) - \epsilon_k(\tilde{z}_{ik} \tilde{w}_{jk} x_{i.} x_{.j} + \beta_k) \Big) \\
&+ \sum_{i,j,k,\ell \neq k}^{n,d,g,c-1} \Big(\big[\tilde{z}_{ik} \tilde{w}_{j\ell} x_{ij} + (\tau - 1) \big] \log(\phi) - \phi(\tilde{z}_{ik} \tilde{w}_{j\ell} x_{i.} x_{.j} + \upsilon) \Big) \\
&- \sum_{i,k}^{n,g} \tilde{z}_{ik} \log \tilde{z}_{ik} - \sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \tilde{w}_{j\ell} + c
\end{aligned}$$

where c is equal to $\sum_{i,j}^{n,d} x_{ij}\log(x_{i.}x_{.j}) - \log(x_{ij}!) + n \times d$. Differentiation w.r.t. π_k, ρ_ℓ, σ, ϵ_k and ϕ leads to the following estimates :

$$\pi_k = \frac{(a_k - 1) + \sum_i^n \tilde{z}_{ik}}{\sum_k^g (a_k - 1) + n}, \tag{5.23}$$

$$\rho_\ell = \frac{(b_\ell - 1) + \sum_j^d \tilde{w}_{j\ell}}{\sum_\ell^c (b_\ell - 1) + d}, \tag{5.24}$$

$$\sigma = \frac{\sum_{i,j,k}^{n,d,g} \tilde{z}_{ik}\tilde{w}_{jc}x_{ij} + (\zeta - 1)}{\sum_{i,j,k}^{n,d,g} \tilde{z}_{ik}\tilde{w}_{jc}x_{i.}x_{.j} + \eta}, \tag{5.25}$$

$$\epsilon_k = \frac{\sum_{i,j}^{n,d} \tilde{z}_{ik}\tilde{w}_{jk}x_{ij} + (\alpha_k - 1)}{\sum_{i,j}^{n,d} \tilde{z}_{ik}\tilde{w}_{jk}x_{i.}x_{.j} + \beta_k}, \tag{5.26}$$

$$\phi = \frac{\sum_{i,j,k,\ell\neq k}^{n,d,g,c-1} \tilde{z}_{ik}\tilde{w}_{j\ell}x_{ij} + (\tau - 1)}{\sum_{i,j,k,\ell\neq k}^{n,d,g,c-1} \tilde{z}_{ik}\tilde{w}_{j\ell}x_{i.}x_{.j} + \upsilon}. \tag{5.27}$$

The `VBEM` procedure for GPLBM is reported in Algorithm 20. Note that by setting $a_1 = \ldots = a_g = 1$, $b_1 = \ldots = b_c = 1$, $\zeta = \alpha_1 = \ldots = \alpha_g = \tau = 1$ and $\beta_1 = \ldots = \beta_g = \eta = \upsilon = 0$, `VBEM` collapses to a Variational EM algorithm (`VEM`) maximizing the non-augmented fuzzy ELBO function.

Algorithm 20 `VBEM` algorithm for `GPLBM`

input : $\boldsymbol{X}$, g, $c = g + 1$, $\boldsymbol{a}$, $\boldsymbol{b}$, ζ, η, τ, υ ; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$;
initialization : $\widetilde{\boldsymbol{Z}}$, $\widetilde{\boldsymbol{W}}$, π_k, ρ_ℓ ; σ, ϵ_k, ϕ ;
repeat
  repeat
    step 1 : compute $\tilde{z}_{ik}$ using eq(5.20) ;
    step 2 : compute π_k, ϵ_k, σ and ϕ using eq(5.23), eq(5.26), eq(5.25), eq(5.27) ;
  until convergence of $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$
  repeat
    step 3 : compute $\tilde{w}_{j\ell}$ using eq(5.21) ;
    step 4 : compute ρ_ℓ, ϵ_k, σ and ϕ using eq(5.24), eq(5.26), eq(5.25), eq(5.27) ;
  until convergence of $\mathcal{L}_A(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}}); \boldsymbol{\Theta})$
until convergence of $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$

Considering the posterior probability from the complete-data, the Bayes formula leads to : $p(\boldsymbol{\Theta}|\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\Theta})p(\boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})$. The developments and results of this expression are given in the supplementary materials. From here, we recognize the following posterior distributions for $\boldsymbol{\Theta}$:

$$\boldsymbol{\pi} \sim \mathcal{D}\Big(\sum_i^n z_{i1} + a_1, \ldots, \sum_i^n z_{ig} + a_g\Big), \tag{5.28}$$

$$\boldsymbol{\rho} \sim \mathcal{D}\Big(\sum_j^d w_{j1} + b_1, \ldots, \sum_j^d w_{jc} + b_c \Big), \tag{5.29}$$

$$\sigma \sim \mathcal{G}\Big(\sum_{i,j,k}^{n,d,g} z_{ik} w_{jc} x_{ij} + \zeta, \sum_{i,j,k}^{n,d,g} z_{ik} w_{jc} a_{ij} + \eta \Big), \tag{5.30}$$

$$\epsilon_k \sim \mathcal{G}\Big(\sum_{i,j}^{n,d} z_{ik} w_{jk} x_{ij} + \alpha_k, \sum_{i,j}^{n,d} z_{ik} w_{jk} a_{ij} + \beta_k \Big), \tag{5.31}$$

$$\phi \sim \mathcal{G}\left(\sum_{i,j,k,\ell \neq k}^{n,d,g,c-1} z_{ik} w_{j\ell} x_{ij} + \tau, \sum_{i,j,k,\ell \neq k}^{n,d,g,c-1} z_{ik} w_{j\ell} a_{ij} + \upsilon \right), \tag{5.32}$$

where $a_{ij} = x_{i.} x_{.j}$. From these distributions, we implemented a `Gibbs sampler` described in Algorithm 21.

Algorithm 21 `Gibbs sampler` for `GPLBM`

input : $\boldsymbol{X}$, g, $c = g + 1$, $\boldsymbol{a}$, $\boldsymbol{b}$, ζ, η, τ, υ ; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$;
initialization : $\boldsymbol{Z}$, $\boldsymbol{W}$, π_k, ρ_ℓ, σ, ϵ_k, ϕ ;
for iteration t=1,2,... **do**
  step 1 : compute $\tilde{z}_{ik}$ using eq(5.20) ;
  step 2 : $\boldsymbol{z}_i \sim \mathcal{M}(1, \tilde{z}_{i1}, \ldots, \tilde{z}_{ig})$;
  step 3 : compute $\tilde{w}_{j\ell}$ using eq(5.21) ;
  step 4 : $\boldsymbol{w}_j \sim \mathcal{M}(1, \tilde{w}_{j1}, \ldots, \tilde{w}_{jc})$;
  step 5 : draw $\boldsymbol{\pi}$, $\boldsymbol{\rho}$, σ, $\boldsymbol{\epsilon}$, ϕ using (5.28-5.32) ;
end for

5.1.4 Experiments on Text data

5.1.4.1 Datasets and clustering scores

Nine different real-world text datasets were chosen. They present a number of challenges in relation to the data structure (overlapping clusters), clusters balance (proportion-wise), the number of clusters, and in some case the data dimension. The datasets are the following : CLASSIC3, built from the CLASSIC4 database (a collection of medical documents) ; K1A, K1B, and WAP, created for the WebAce project and containing Yahoo web pages ; Reuters40, extracted from the forty largest groups in the Reuters-21578 database, comprising newspapers published in 1978 ; SPORTS, to be found in the CLUTO toolkit [201] and containing documents relating to seven different sports ; TDT2, the NIST Topic Detection and Tracking (TDT2) corpus consisting of data acquired during the first half of 1998 from

six sources, including two newswires (APW, NYT), two radio programs (VOA, PRI) and two television programs (CNN, ABC) ; NG5, consisting of five groups from NG20. Their characteristics are reported in Table 5.1.

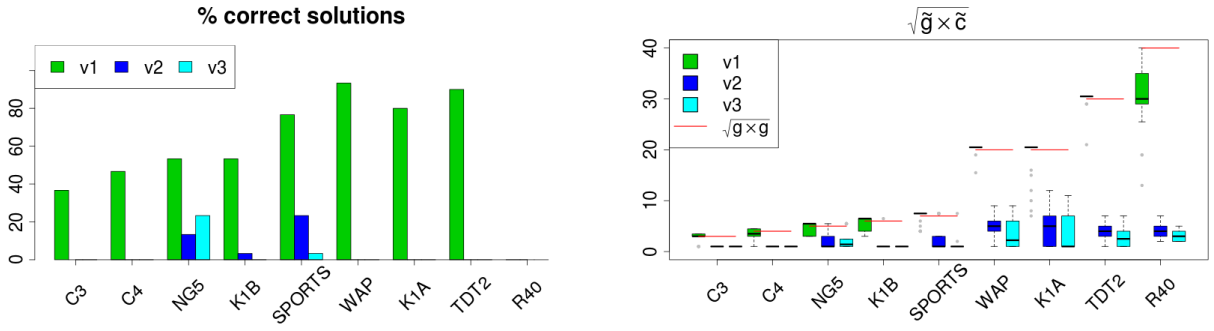
TABLE 5.1 – Data characteristics.

Datasets	Documents	Words	#Clusters	0(%)	Balance
CLASSIC3/C3	3891	4303	3	99.86	0.708
CLASSIC4/C4	7095	5896	4	99.41	0.322
NG5	4905	10167	5	99.08	1
K1B	2340	21819	6	99.41	0.0432
SPORTS/SPS	8580	14870	7	99.14	0.036
WAP	1560	8460	20	98.33	0.0147
K1A	2340	21839	20	99.32	0.0182
TDT2/T2	9394	36771	30	99.65	0.028
Reuters40/R40	8203	18914	40	99.75	0.003

The numbers of clusters in each algorithm (g in ours) is set as the ground truth. The document clustering quality is assessed using two scores widely acknowledged for quantifying the correspondence between the clustering and the true labels. These are, first, Normalized Mutual Information (NMI) [119], which measures the mutual dependency between two random variables, and, second, Adjusted Rand Index (ARI) [122], which measures the degree of agreement between two partitions. The term clustering cannot be assessed with these metrics since the true term labels are unknown.

5.1.4.2 Degree of overtiffing

In this section, we compare the various alternatives found in the literature to maximize $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ or especially $\mathcal{L}(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ since the hyperparameters are set s.t. `VBEM` collapses to `VEM`. This comparison aims at quantifying the level of overfitting (more precisely empty-clusters solutions) arising with MLE given that Bayesian inference is usually portrayed as an alternative to decrease this behavior. The first version (v1) is the one introduced in this contribution which maximizes $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ by alternatively maximizing (until convergence) $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ and $\mathcal{L}_A(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}}); \boldsymbol{\Theta})$. The second version (v2) is inspired by [254, 262] and maximizes $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ by maximizing $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}|\widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ and $\mathcal{L}_A(q(\widetilde{\boldsymbol{W}}|\widetilde{\boldsymbol{Z}}); \boldsymbol{\Theta})$ respectively during one iteration only. The third version (v3) is the one introduced in [258] which attempts to maximize $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$ directly. The normalized variational densities $\tilde{z}_{ik}$



(a) (b)

FIGURE 5.4 – (a) % of correct solutions returned by the VEM versions. (b) Boxplots of $\sqrt{\tilde{g} \times \tilde{c}}$ for each VEM version.

and $\tilde{w}_{j\ell}$ are computed successively and followed by one maximization step ($\arg\max_{\boldsymbol{\Theta}} \mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$). The procedures of v2 and v3 are given in algorithm 22) and 23) respectively. Furthermore, note that in each version, a classification step may also be included after an expectation step so as to produce hard partitions from $\widetilde{\boldsymbol{Z}}$ and $\widetilde{\boldsymbol{W}}$ respectively to maximize $\mathcal{L}_A(q(\boldsymbol{Z}, \boldsymbol{W}); \boldsymbol{\Theta})$ overall. The resulting algorithm would be referred to as Variational Bayes Classification EM (VBCEM) and would be useful for speeding up the convergence [79, 257]. Similarly, a stochastic step that successively draws $\boldsymbol{z}_i^{(t+1)}, \forall i = 1, \ldots, n$ and $\boldsymbol{w}_j^{(t+1)}, \forall j = 1, \ldots, d$ from $P(\widetilde{\boldsymbol{Z}}; \boldsymbol{\Theta}^{(c)})$ and $P(\widetilde{\boldsymbol{W}}; \boldsymbol{\Theta}^{(c)})$ after their respective expectation steps can be introduced. This algorithm would be referred to as Variational Bayes Stochastic EM (VBSEM).

Algorithm 22 VBEM algorithm for GPLBM (v2)

input : $\boldsymbol{X}$, g, $c = g + 1$, $\boldsymbol{a}$, $\boldsymbol{b}$, ζ, η, τ, υ ; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$;
initialization : $\widetilde{\boldsymbol{Z}}$, $\widetilde{\boldsymbol{W}}$, π_k, ρ_ℓ ; σ, ϵ_k, ϕ ;
repeat
  step 1 : compute $\tilde{z}_{ik}$ using eq(5.20) ;
  step 2 : compute π_k, ϵ_k, σ and ϕ using eq(5.23), eq(5.26), eq(5.25), eq(5.27) ;
  step 3 : compute $\tilde{w}_{j\ell}$ using eq(5.21) ;
  step 4 : compute ρ_ℓ, ϵ_k, σ and ϕ using eq(5.24), eq(5.26), eq(5.25), eq(5.27) ;
until convergence of $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$

Figure 5.4(a) displays the percentage of correct solutions (no-empty-clusters) returned by each version of VEM over 30 trials. The results show that v2 and v3 are strongly subject to overfitting. Clearly, both struggle substantially more than v1 which remains fairly consistent on all the datasets (except R40). Figure 5.4(b) shows the boxplots of the square root of the number of non-empty co-clusters $\hat{g} \times \hat{c}$ returned by each version of their 30 trials. As a reference, a red line is drawn at $\sqrt{g \times g}$

Algorithm 23 `VBEM` algorithm for GPLBM (v3)

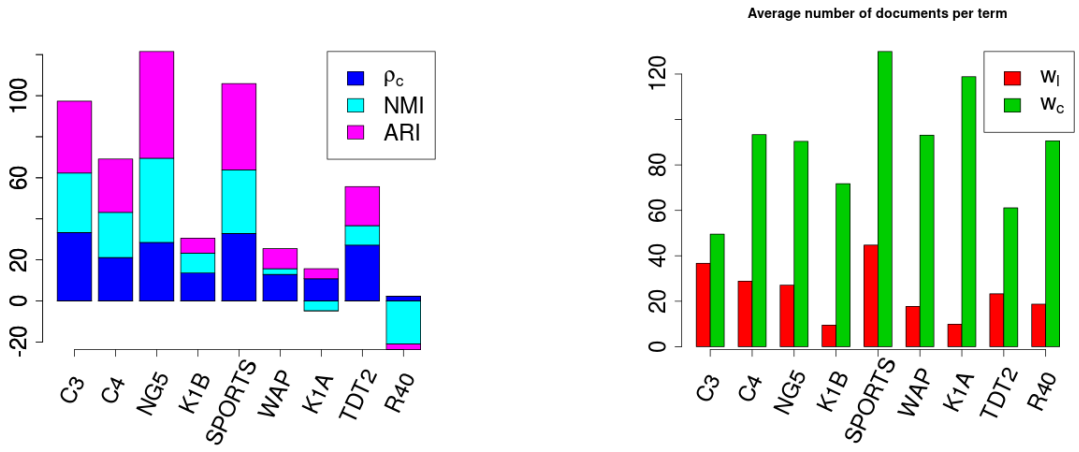
input : $\boldsymbol{X}$, g, $c = g + 1$, $\boldsymbol{a}$, $\boldsymbol{b}$, ζ, η, τ, υ ; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$;
initialization : $\widetilde{\boldsymbol{Z}}$, $\widetilde{\boldsymbol{W}}$, π_k, ρ_ℓ ; σ, ϵ_k, ϕ ;
repeat
  step 1 : compute $\tilde{z}_{ik}$ using eq(5.20) ;
  step 2 : compute $\tilde{w}_{j\ell}$ using eq(5.21) ;
  step 3 : compute π_k, ρ_ℓ, ϵ_k, σ and ϕ using eq(5.23), eq(5.24), eq(5.26), eq(5.25), eq(5.27) ;
until convergence of $\mathcal{L}_A(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$

for each dataset. This figure reveals the error margin (or degree of overfitting) for v1, v2 and v3. From the results, it appears that v2 and v3 overfit undeniably more than v1. This effect is increased on datasets with larger number of clusters (K1A, WAP, TDT2, R40) where the partitions retrieved have substantially less co-clusters than the expected numbers. Overall, while v2 and v3 are definitely faster than v1, they produce the wort procedures in terms of overfitting with optimally local MLE. Therefore they should not be used as baselines when solving this issue since v1 is already a much better alternative.

5.1.4.3 Noise detection and clustering performance with MLE

We evaluate the ability of `GPLBM` to detect a sample of noisy features without any prior knowledge on $\boldsymbol{\Theta}$. Therefore, the hyperparameters are set s.t. `VBEM` collapses to `VEM`. Note that to reduce the sensitivity of `VEM` to starting values, we use the solutions of a stochastic algorithm called `SEM-Gibbs` (which is the `Gibbs sampler` where the simulation of $\boldsymbol{\Theta}$ (step 5) is replaced by the maximum Likelihood closed-form estimates [259]) as initialization for `VEM`. Additionally, to provide a faithful user case, among 30 correct trials, only the first 10 are kept for evaluation (the trials are ranked criterion-wise using $\mathcal{L}(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$). Figure 5.5(a) shows the average value of ρ_c, as well as the percentage of improvement with regards to `SPLBM` in terms of NMI and ARI. The graphic shows that `GPLBM` noisy cluster w_c has an average proportion (ρ_c) of 20% of features across the datasets (with a maximum of 33.3% on C3 and minimum of 2.23% on R40). Undeniably, the best NMI and ARI improvements w.r.t. `SPLBM` are observed when ρ_c are high. When ρ_c is low and the number of expected co-clusters is high (e.g. K1A, R40), GPLBM seems to struggle and can be outperformed by its predecessor (SPLBM). Moreover, from Figures 5.1(a)-5.1(b) presented earlier in the introduction, K1A, R40 are the datasets with the lowest prevalence of documents per term, which intuitively makes intricate the detection of noisy features for those partitions. From the experiments in last section, `VEM` also substantially struggles to provide

correct partitions on R40, which potentially restricts the learning of better solutions.



(a) (b)

FIGURE 5.5 – (a) Average ρ_c using `VEM`; % of improvements in terms of NMI and ARI, w.r.t. `SPLBM`. (b) Average prevalence for terms in $\mathbf{w}_\ell, \forall \ell = 1, \ldots, g$ and terms in $\mathbf{w}_c$.

Additionally, Figure 5.5(b) depicts the average prevalence of documents per term, in the non-noisy clusters $\mathrm{w}_\ell, \forall \ell = 1, \ldots, g$ and in the noisy cluster w_c respectively. Clearly, the features gathered in w_c have a much higher prevalence compared to those in w_ℓ. Furthermore, a comparative study of GPLBM with MLE against several state-of-the-art algorithms was achieved to demonstrate the superiority of GPLBM. The pertinence of ρ_c was assessed by making ρ_c vary manually in $\{0, \ldots, .35\}$ and displaying the NMI and ARI curves for each dataset. The results (given the the following) show robustness for the learning of ρ_c as its estimated value matches the best clustering partitions and avoid overestimation which could deteriorate the classification (e.g. on R40). Moreover, due to the lack of true labels for the set of features, additional experiments using the Wikipedia corpus were achieved to assess to the partitions of terms found by `GPLBM` against other co-clustering algorithms.

5.1.4.3.1 Clustering scores using MLE. To leverage the impact of GPLBM's parameterization, we compared it against PLBM and SPLBM within the framework of MLE. For each model, the empirical results of `VEM` initialized with `SEM-Gibbs` are used and denoted by "`sg+v`". Algorithms such as `ITCC` (Information Theory Co-Clustering) [252] and the popular Latent Dirichlet Allocation (`LDA`) [263] are also inserted for reference. 30 trials are ran for each algorithm on each of the datasets. The results are displayed in Tables 5.2-5.3 and show the means and standard deviations of the NMI and ARI scores

for the 10 best solutions (w.r.t. each dataset and algorithm). The score of *the best solution* (which is ultimately selected by the user) among the set of 10 is given in parenthesis. Note that this score can sometimes be inferior to the average as the best solution does not always provide the best clustering. This problem, often overlooked in unsupervised learning, has been raised and solved in [264] by using an ensemble approach.

TABLE 5.2 – NMI scores averages and standard deviations (Mean±SD) for GPLBM with MLE.

Data	LDA	ITCC	PLBMsg+v	SPLBMsg+v	GPLBMv	GPLBMsg	GPLBMsg+v
C3	0.91±0.0028 (0.91)	0.93±0.002 (0.93)	0.93±0.0017 (0.93)	0.93±0.0015 (0.93)	**0.95**±0.00082 (**0.95**)	**0.95**±0.0034 (**0.95**)	**0.95**±0.0019 (**0.95**)
C4	0.75±0.0023 (0.75)	0.58±0.02 (0.59)	0.63±0.0033 (0.63)	0.68±0.072 (0.78)	**0.78**±0.018 (0.78)	0.74±0.045 (0.77)	0.75±0.048 (**0.79**)
NG5	0.68±0.054 (0.58)	0.68±0.032 (0.71)	0.67±0.04 (0.7)	0.59±0.074 (0.7)	0.71±0.054 (0.78)	**0.75**±0.032 (**0.8**)	**0.75**±0.032 (**0.8**)
K1B	0.58±0.014 (0.58)	0.57±0.027 (0.6)	**0.61**±0.029 (0.6)	0.53±0.057 (**0.64**)	0.55±0.031 (0.61)	0.59±0.044 (**0.64**)	0.58±0.033 (**0.64**)
SPORTS	0.48±0.029 (0.43)	0.55±0.034 (0.58)	0.54±0.033 (0.61)	0.51±0.046 (0.5)	0.62±0.048 (0.67)	**0.66**±0.044 (**0.69**)	**0.66**±0.045 (**0.69**)
WAP	**0.57**±0.011 (0.55)	0.56±0.013 (0.56)	0.56±0.02 (0.58)	0.53±0.024 (0.52)	0.47±0.024 (0.53)	0.54±0.02 (**0.58**)	0.54±0.02 (**0.58**)
K1A	**0.58**±0.012 (0.6)	0.49±0.013 (0.5)	**0.58**±0.021 (**0.63**)	0.53±0.019 (0.56)	0.45±0.029 (0.51)	0.51±0.022 (0.53)	0.5±0.023 (0.53)
TDT2	0.72±0.0066 (0.72)	0.74±0.017 (0.76)	0.74±0.012 (0.76)	0.75±0.014 (0.77)	**0.79**±0.022 (**0.8**)	0.77±0.013 (0.79)	0.77±0.012 (0.79)
R40	0.49±0.0037 (0.49)	0.49±0.0082 (0.49)	0.43±0.015 (0.41)	**0.52**±0.012 (0.51)	0.51±0.021 (**0.55**)	0.42±0.033 (0.47)	0.42±0.054 (0.51)

First, we note that GPLBMsg and GPLBMsg+v are comparable and outperform GPLBMv ; in the sequel we retain GPLBMsg+v. With GPLBMsg+v, we observe an improvement over PLBMsg+v and SPLBMsg+v, and a clear superiority on LDA and ITCC commonly used for the same tasks.

TABLE 5.3 – ARI scores averages and standard deviations (Mean±SD) for GPLBM with MLE.

Data	LDA	ITCC	PLBMsg+v	SPLBMsg+v	GPLBMv	GPLBMsg	GPLBMsg+v
C3	0.94±0.0022	0.96±0.0014	0.96±0.0012	0.95±9e-04	**0.97**±7e-04	**0.97**±0.0025	**0.97**±0.0013
	(0.94)	(0.96)	(0.96)	(0.96)	**(0.97)**	**(0.97)**	**(0.97)**
C4	0.76±0.0035	0.42±0.02	0.5±0.0038	0.59±0.15	**0.78**±0.023	0.68±0.14	0.7±0.15
	(0.76)	(0.44)	(0.5)	(0.79)	(0.79)	(0.78)	**(0.8)**
NG5	0.63±0.062	0.62±0.057	0.58±0.047	0.47±0.097	0.68±0.088	**0.74**±0.071	**0.74**±0.071
	(0.53)	(0.66)	(0.65)	(0.67)	(0.78)	**(0.82)**	**(0.82)**
K1B	0.34±0.021	0.36±0.046	0.4±0.042	0.37±0.087	0.41±0.083	**0.46**±0.1	0.41±0.061
	(0.36)	(0.4)	(0.39)	**(0.54)**	(0.53)	(0.52)	(0.52)
SPORTS	0.4±0.03	0.42±0.035	0.41±0.036	0.39±0.065	0.56±0.094	0.64±0.11	**0.65**±0.1
	(0.34)	(0.43)	(0.47)	(0.33)	(0.65)	**(0.7)**	**(0.7)**
WAP	0.35±0.022	0.37±0.041	0.38±0.049	0.47±0.028	0.33±0.074	**0.53**±0.023	**0.53**±0.023
	(0.32)	(0.38)	(0.37)	(0.45)	(0.45)	(0.54)	**(0.55)**
K1A	0.37±0.02	0.31±0.043	0.4±0.051	0.49±0.036	0.37±0.055	**0.52**±0.027	**0.52**±0.026
	(0.38)	(0.31)	(0.48)	(0.52)	(0.41)	**(0.55)**	**(0.55)**
TDT2	0.47±0.017	0.48±0.037	0.47±0.028	0.71±0.032	0.74±0.049	**0.77**±0.017	**0.77**±0.017
	(0.47)	(0.5)	(0.48)	(0.72)	(0.76)	**(0.79)**	**(0.79)**
R40	0.2±0.017	0.18±0.024	0.33±0.06	**0.48**±0.021	0.47±0.046	0.46±0.06	0.46±0.065
	(0.19)	(0.14)	(0.3)	(0.44)	(0.5)	**(0.58)**	(0.51)

5.1.4.3.2 GPlbm noise estimation pertinence. In this section, we evaluate the ability of GPLBM to estimate an amount of noise matching a good clustering performance. Therefore, we display the behavior of GPLBM (in terms of clustering performance) for different values of ρ_c using MLE (`VBEM` is set to collapse to `VEM` and initialized with the solutions of `SEM-Gibbs`).

Assuming that ρ_c is fixed, the remaining weights of the non-noisy clusters $\rho_\ell\,; \forall \ell = 1, \ldots, c-1$ are obtained by introducing the following constraint : $\sum_{\ell}^{c-1} \rho_\ell = 1 - \rho_c$ into the maximization problem of $\mathcal{L}(q(\widetilde{\boldsymbol{Z}}, \widetilde{\boldsymbol{W}}); \boldsymbol{\Theta})$. Let $\sum_{j,\ell}^{d,c} \tilde{w}_{j\ell} \log \rho_\ell = \sum_{j,\ell}^{d,c-1} \tilde{w}_{j\ell} \log \rho_\ell + \sum_{j}^{d} \tilde{w}_{jc} \log \rho_c$ and λ be a Lagrange multiplier for the constraint, from eq(5.14) where the terms where $\boldsymbol{\rho}$ appears, we can express the Lagrangian

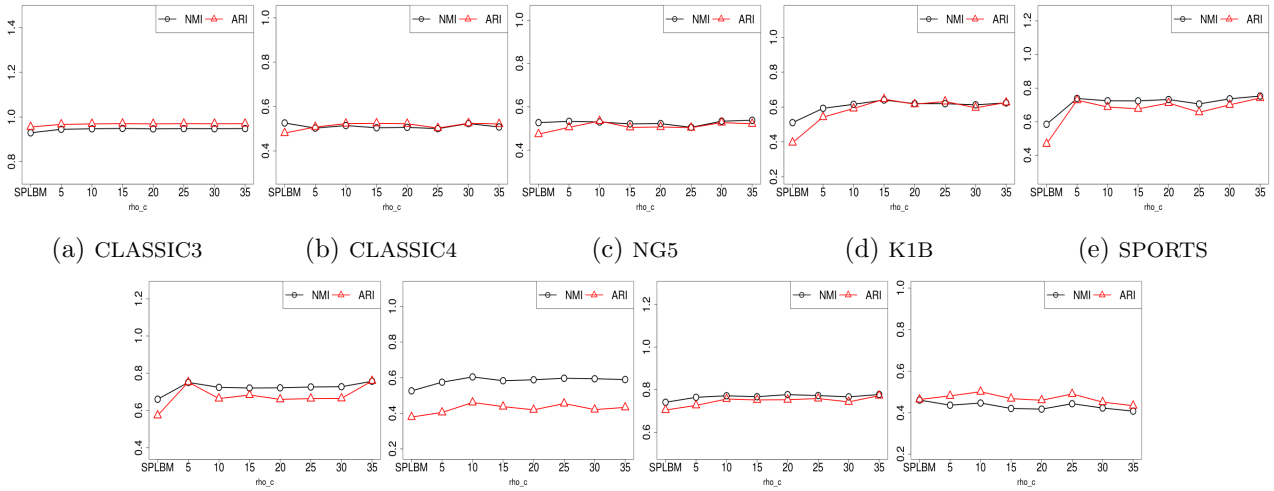
$\mathcal{L}(\boldsymbol{\rho}, \lambda)$ by

$$\sum_{j,\ell}^{d,c-1} \tilde{w}_{j\ell} \log \rho_\ell + \sum_{j}^{d} \tilde{w}_{jc} \log \rho_c + \lambda\Big(1 - \rho_c - \sum_{\ell}^{c-1} \rho_\ell\Big).$$

Setting the gradient of $\mathcal{L}$ w.r.t ρ_ℓ to 0 leads to $\rho_\ell = \frac{\sum_j \tilde{w}_{j\ell}}{\lambda}$. Plugging ρ_ℓ into $\mathcal{L}$ and using the constraint leads to $\lambda = (\sum_{\ell}^{c-1} \sum_{j}^{d} \tilde{w}_{j\ell})/(1 - \rho_c)$. Thereby, we denote

$$\rho_\ell = \frac{(1 - \rho_c) \sum_j^d \tilde{w}_{j\ell}}{\big[\sum_{\ell,j}^{c-1,d} \tilde{w}_{j\ell}\big]}.$$

Figure 5.6 shows how NMI and ARI change as ρ_c varies from 0.05 to 0.35 by increments of 0.05.



(a) CLASSIC3 (b) CLASSIC4 (c) NG5 (d) K1B (e) SPORTS

(f) WAP (g) K1A (h) TDT2 (i) R40

FIGURE 5.6 – NMI and ARI evolution with `GPLBM` according to ρ_c.

The starting value corresponds to $\rho_c = 0$, and therefore to SPLBM. For reference, Table 5.4 recalls the values of ρ_c found by GPLBM on each dataset.

TABLE 5.4 – GPLBM : estimated ρ_c.

	C3	C4	NG5	K1B	SPORTS	WAP	K1A	TDT2	R40
ρ_c	0.33	0.21	0.29	0.14	0.33	0.13	0.11	0.27	0.02

Focusing on (K1B, K1A, WAP, and especially R40) where the estimated values of ρ_c are the lowest, GPLBM gives estimates of ρ_c that look to avoid poor clustering performance. In addition, this behavior seems to be also consistent when there are little scores fluctuations and performance to be gained (see C3, C4, NG5, SPORTS and TDT2).

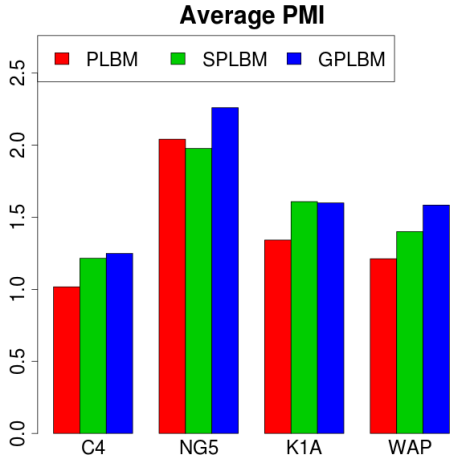


FIGURE 5.7 – PMI of the top 15 terms per cluster in Wikipedia (window size = 10).

5.1.4.3.3 Assessing the term clusters. Successively, we evaluate the quality of the term clusters obtained with GPLBM against PLBM and SPLBM. NG5, C4, K1A and WAP are the only document-term matrices in our set for which the respective list of terms is available. We applied a similar approach to the one used in [265] to build a context matrix. Then, we used a sliding window which consists in searching for pairwise occurrences of terms within an interval of L-words in every article in the entire Wikipedia corpus. We fixed $L = 10$, and for each partition selected the top 15 terms per cluster. Furthermore, we computed the Point-wise Mutual Information (PMI) in order to measure the association between the different top terms. The posterior conditional probability $\tilde{w}_{j\ell}, \forall j = 1, \ldots, d, \forall \ell = 1, \ldots, c$ are used to rank the terms in each cluster and select the top 15 terms. Figure 5.7 displays the average PMI obtained for each model. The top terms for the five best solutions (criterion-wise) were retained. The graphic shows that GPLBM obtains the highest PMI on average.

5.1.4.4 Hyperparameters settings and overfitting

As shown earlier, `GPLBM` tends to leave one or several empty clusters (in $\boldsymbol{Z}$ or $\boldsymbol{W}$) which is one case of finite mixture model overfitting. Another case is when two or more vector-components parameters $\boldsymbol{\theta}_k = (\theta_{k1}, \ldots, \theta_{kd})$ are rather identical. In the literature, Bayesian inference with specific priors $\mathcal{P}(e_1, \ldots, e_g)$ has often been suggested to address this issue in a finite mixture model. Usually, the prior (on the weights) is a Dirichlet and the hyperparameters are exchangeable s.t. $e_k \equiv e_0 | \forall k = 1, \ldots, g$; the popular and often-discussed uniform distribution $\mathcal{D}(1, \ldots, 1)$ is an example. Frühwirth-Schnatter [266]

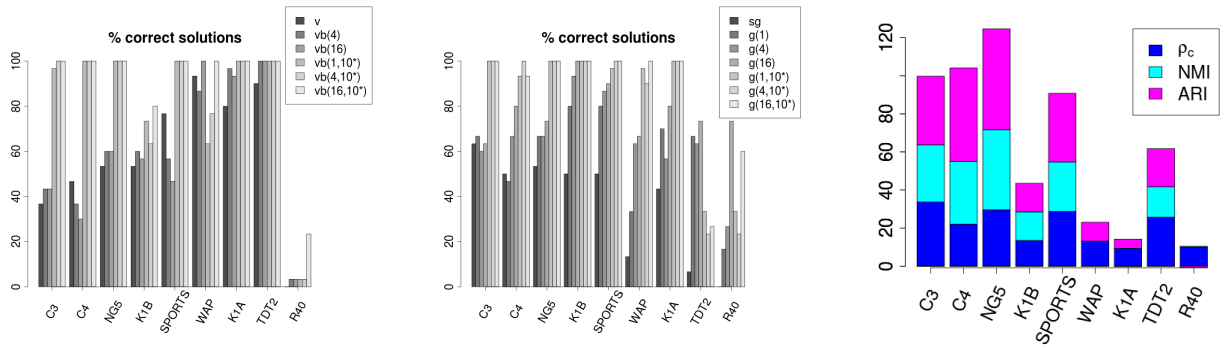


FIGURE 5.8 – (a) % of correct solutions with priors for `VBEM`. (b) % of correct solutions with priors for the `Gibbs sampler`. (c) Average ρ_c using `VBEM`; % of improvements in terms of NMI and ARI, w.r.t. `SPLBM`.

(Section 4.2.2) advocated $e_0 = 4$ to bound the posterior outside regions allowing empty clusters and later recommended to use $e_0 = 16.5$ [267] when $\dim(\boldsymbol{\theta}_k) > 8$, or to set e_0 appropriately as $\dim(\boldsymbol{\theta}_k)$ increases. This recommendation follows the results of the asymptotic analysis of Rousseau and Mergersen [268] which showed that : (a) $\min_{k=1,\ldots,g} e_k > \dim(\boldsymbol{\theta}_k)/2$ concentrates the posterior where at least two components are rather identical; (b) $\max_{k=1,\ldots,g} e_k < \dim(\boldsymbol{\theta}_k)/2$ concentrates the posterior within regions leaving empty groups. With LBM, given the double latent structure and the block parameters $\boldsymbol{\theta} = (\theta_{k\ell}) \in \mathbb{R}^{g\times c}$, the vector-components for the row partition $\boldsymbol{Z}$ are denoted as $\boldsymbol{\theta}_k = (\theta_{k1}, \ldots, \theta_{kc}) \in \mathbb{R}^c$, $\forall k = 1, \ldots, g$. The vector-components for the column partition $\boldsymbol{W}$ are denoted $\boldsymbol{\theta}_\ell^\top = (\theta_{\ell 1,\ldots}, \ell g)^\top \in \mathbb{R}^g$, $\forall \ell = 1, \ldots, c$. Several experiments following Frühwirth-Schnatter recommendations were conducted. The prior hyperparameters for each weight were set equally s.t. $a_1 = \ldots = a_g = e_0$ and $b_1 = \ldots = b_c = e_0$ with $e_0 \in \{1, 4, 16.5\}$ in `VBEM` and the `Gibbs sampler` (in these cases, all the elements in $\{\zeta, \eta, \boldsymbol{\alpha}, \boldsymbol{\beta}, \tau, \upsilon\}$ are set to one). However, compared to `VEM`, no significant improvements were noted with `VBEM` whilst the `Gibbs sampler` showed a bit of refinement against `SEM-Gibbs`. The diagonal restriction in `GPLBM` guarantees that $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_g$, $\forall k = 1, \ldots, g$ are not identical despite having $g-1$ parameters in common (the same remarks apply to the column partition components $\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_{c-1}^\top$, $\forall \ell = 1, \ldots, c-1$). As a consequence, this restriction diminishes the usual impact of the weights priors aiming at producing identical component-vectors. Therefore, to reduce the discrepancy between the diagonal elements whilst keeping distinction between $\boldsymbol{\epsilon}$ and ϕ (for a sparse recovery), we considered two specific priors for the block parameters in addition to the weights priors.

Thereby, the diagonal parameters were given the same prior $\mathcal{G}(10^*, 1)$ (s.t. $\alpha_1 = \ldots = \alpha_g = 10^*$, $\beta_1 = \ldots = \beta_g = 1$) while ϕ was subject to $\mathcal{G}(1, 10^*)$ ($\tau = 1, \upsilon = 10^*$), where $* \in \{2, 3\}$. With a higher value for the shape parameter, $\mathcal{G}(10^*, 1)$ aims at favoring higher and identical estimates for $\boldsymbol{\epsilon}$ whilst $\mathcal{G}(1, 10^*)$ should fit a lower and spreaded distribution for ϕ. $* := 3$ for datasets with $g \leq 8$ and $* := 2$ when $g > 8$. Figures 5.8(a)-5.8(b) displays the percentages of correct solutions returned by each prior setting using `VBEM` (denoted as "vb") and the `Gibbs sampler` ("g") respectively over 30 trials. The percentage of `VEM` ("v") and `SEM-GIBBS` ("sg") are added for reference. The graphics show clearly that our settings allow a substantial decrease of empty clusters solutions by `GPLBM`. In practice, $*$ should be decreased as g increases and leads to lower values for $\boldsymbol{\epsilon}$. This was achieved on R40 where $* := 1.5$ led to small improvements.

5.1.4.5 Clustering performance

Overall, Bayesian inference with our settings (in most situations) and the recommended settings [267] (using the `Gibbs sampler`) undeniably limits empty clusters solutions. However it is key to ensure that the clustering performances are better if not at least similar to those obtained using MLE. Doing so, we achieve a statistical study of the clustering performance regarding to all the settings mentioned previously using `VBEM` and the `Gibbs sampler` compared to the MLE algorithms. We use the Nemenyi non-parametric statistical test [269, 270] which quantifies the performance between several algorithms by measuring the pairwise Critical Difference (CD) between their average ranks. The results (available in the next section) suggest, to guarantee the best clustering partitions, the following hyperparameters : $a_1 = \ldots = a_g = b_1 = \ldots = b_c = 1$; $\alpha_1 = \ldots = \alpha_g = \upsilon = 10^3$; $\beta_1 = \ldots = \beta_c = \tau = 1$ where $g \leq 8$ and $a_1 = \ldots = a_g = b_1 = \ldots = b_c = 16.5$ where $g > 8$. Following these priors, we compared `GPLBM` (using the results obtained from `VBEM` initialized with the solution of `Gibbs sampler`) against several benchmark algorithms acknowledged for their good performance or popularity in document clustering. Algorithms such as : `ITCC` (Information Theory Co-Clustering) [252], Latent Dirichlet Allocation (`LDA`) [263], Spherical K-means (S-Kmeans) [39, 271], the recently introduced Deep Clustering Network (`DCN`) [203], `K-means` and `SPLBM` are used for reference. Note that `DCN` showed significant improvements for document clustering against several clustering (K-means, Spectral Clustering), NMF (`LCCF`) [204] and Deep Learning algorithms (`SAE`)[205]. Algorithms which require parameters settings are launched accordingly to the settings advocated by their authors. As previously, a set of 30 runs is made for each

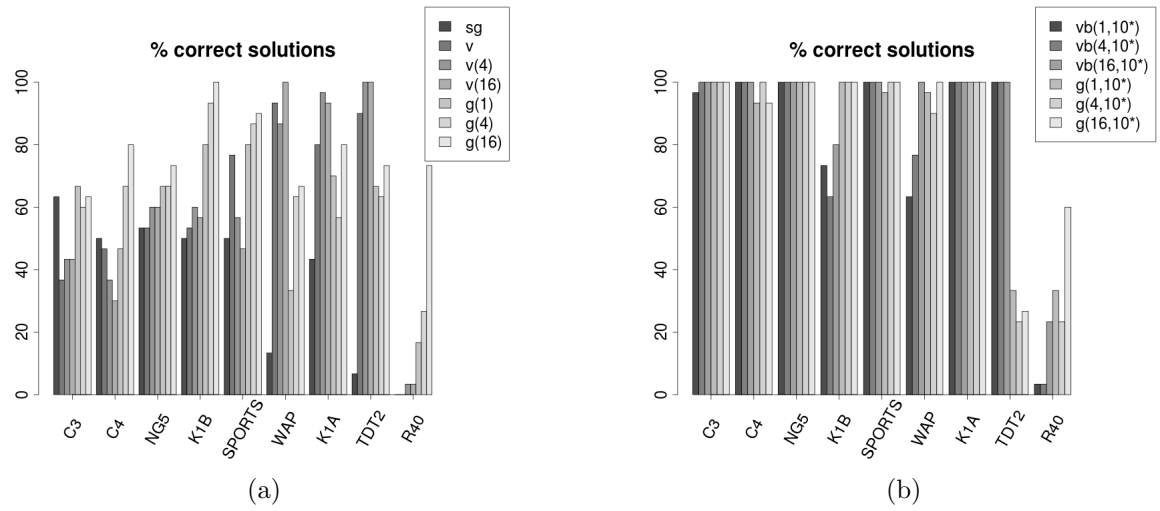
TABLE 5.5 – Clustering scores.

Data	Scores	K-Means	S-Kmeans	LDA	ITCC	DCN	SPLBM	GPLBM
C3	NMI	0.37±0.17	0.92±0.001	0.91±0.003	0.93±0.002	0.92±0.046	0.93±0.001	**0.95**±0.002
		(0.27)	(0.91)	(0.91)	(0.93)	(0.93)	(0.93)	**(0.95)**
	ARI	0.21±0.22	0.95±0.001	0.94±0.002	0.96±0.001	0.94±0.045	0.95±0.001	**0.97**±0.001
		(0.1)	(0.95)	(0.94)	(0.96)	(0.96)	(0.96)	**(0.97)**
C4	NMI	0.29±0.008	0.54±0.001	0.75±0.002	0.58±0.02	0.57±0.014	0.68±0.072	**0.79**±0.013
		(0.29)	(0.54)	(0.75)	(0.59)	(0.58)	(0.78)	**(0.79)**
	ARI	0.16±0.003	0.43±0.001	0.76±0.003	0.42±0.02	0.42±0.013	0.59±0.15	**0.79**±0.018
		(0.16)	(0.43)	(0.76)	(0.44)	(0.42)	(0.79)	**(0.8)**
NG5	NMI	0.035±0.007	0.38±0.021	0.68±0.054	0.68±0.032	0.62±0.028	0.59±0.074	**0.76**±0.035
		(0.035)	(0.37)	(0.58)	(0.71)	(0.59)	(0.7)	**(0.8)**
	ARI	0.003±0.001	0.22±0.024	0.63±0.062	0.62±0.057	0.47±0.027	0.47±0.097	**0.75**±0.066
		(0.003)	(0.21)	(0.53)	(0.66)	(0.46)	(0.67)	**(0.81)**
K1B	NMI	0.47±0.023	0.62±0.02	0.58±0.014	0.57±0.027	**0.66**±0.047	0.53±0.057	0.6±0.028
		(0.5)	(0.61)	(0.58)	(0.6)	**(0.64)**	**(0.64)**	**(0.64)**
	ARI	0.39±0.066	0.41±0.024	0.34±0.021	0.36±0.046	**0.64**±0.093	0.37±0.087	0.46±0.065
		(0.45)	(0.4)	(0.36)	(0.4)	**(0.64)**	(0.54)	(0.52)
SPS	NMI	0.17±0.04	0.46±0.048	0.48±0.029	0.55±0.034	0.59±0.015	0.51±0.046	**0.64**±0.027
		(0.24)	(0.42)	(0.43)	(0.58)	(0.60)	(0.5)	**(0.61)**
	ARI	0.023±0.016	0.26±0.054	0.4±0.03	0.42±0.035	0.37±0.034	0.39±0.065	**0.61**±0.065
		(0.046)	(0.22)	(0.34)	(0.43)	(0.40)	(0.33)	**(0.59)**
WAP	NMI	0.45±0.016	0.56±0.009	0.57±0.011	0.56±0.013	**0.58**±0.016	0.53±0.024	0.53±0.021
		(0.47)	(0.56)	(0.55)	(0.56)	(0.56)	(0.52)	**(0.57)**
	ARI	0.14±0.03	0.28±0.026	0.35±0.022	0.37±0.041	0.32±0.030	0.47±0.028	**0.53**±0.038
		(0.14)	(0.27)	(0.32)	(0.38)	(0.30)	(0.45)	**(0.56)**
K1A	NMI	0.43±0.011	0.57±0.009	0.58±0.012	0.49±0.013	**0.59**±0.008	0.53±0.019	0.53±0.017
		(0.44)	(0.57)	**(0.6)**	(0.5)	(0.58)	(0.56)	(0.55)
	ARI	0.13±0.024	0.29±0.037	0.37±0.02	0.31±0.043	0.34±0.029	0.49±0.036	**0.51**±0.046
		(0.12)	(0.27)	(0.38)	(0.31)	(0.38)	(0.52)	**(0.54)**
T2	NMI	0.41±0.009	0.76±0.007	0.72±0.007	0.74±0.017	0.78±0.01	0.75±0.014	**0.79**±0.013
		(0.42)	(0.77)	(0.72)	(0.76)	(0.79)	(0.77)	**(0.8)**
	ARI	0.042±0.005	0.45±0.026	0.47±0.017	0.48±0.037	0.48±0.028	0.71±0.032	**0.77**±0.017
		(0.044)	(0.48)	(0.47)	(0.5)	(0.51)	(0.72)	**(0.77)**
R40	NMI	0.38±0.023	0.47±0.008	0.49±0.004	0.49±0.008	0.50±0.008	**0.52**±0.012	**0.52**±0.019
		(0.41)	(0.47)	(0.49)	(0.49)	(0.50)	(0.51)	**(0.53)**
	ARI	0.11±0.046	0.11±0.01	0.2±0.017	0.18±0.024	0.13±0.012	**0.48**±0.021	0.47±0.041
		(0.15)	(0.11)	(0.19)	(0.14)	(0.13)	**(0.44)**	**(0.44)**

algorithm from which the 10 best results (criterion-wise) are extracted. The results in Table 5.5 are shown in terms of mean and standard deviation. The scores of the run with the best criterion is shown in parenthesis. They show that `GPLBM` is clearly superior for the task of document-term clustering. Additionally, as with Figure 5.5(a), Figure 5.8(c) depicts the performance gain of `GPLBM` using `VBEM` over `SPLBM`. The graphic shows that `GPLBM` using `VBEM` outperforms `SPLBM` substantially with much better improvements overall on the majority of datasets. Furthermore, the performance deficiency observed on K1A and R40 in Figure 5.5(a) using `VEM` has now been leveraged.

5.1.5 Hyperparameters settings selection

Table 5.6 describes the hyperparameters settings used for the different priors. Figure 5.9(a) displays the percentage of correct solutions returned by `VBEM` and the `Gibbs sampler` over 30 trials and according to the Dirichlet priors settings $(1, 4, 16)$. Figure 5.9(b) displays the percentage of correct solutions using the new priors settings $(1, 10^*), (4, 10^*), (16, 10^*)$.



(a) (b)

FIGURE 5.9 – Percentage of correct solutions for GPLBM with `VBEM` and the `Gibbs sampler` regarding the various settings.

From Figure 5.9(a), it is clears that $g(16)$ consistently outperforms $g(1)$ and $g(4)$ and looks like a better alternative for producing more correct solutions. However, from [266, 267], (4) is the recommended setting when $g \leq 8$ and (16) when $g > 8$. On the other hand, From Figure 5.9(b), all the new settings perform similarly with the `Gibbs sampler`. Therefore, in the light of the above, we required a statistical test to decide which setting should be advocated for each type of partition ($g \leq 8$ and

TABLE 5.6 – Hyperparameters settings, $* \in \{2, 3\}$.

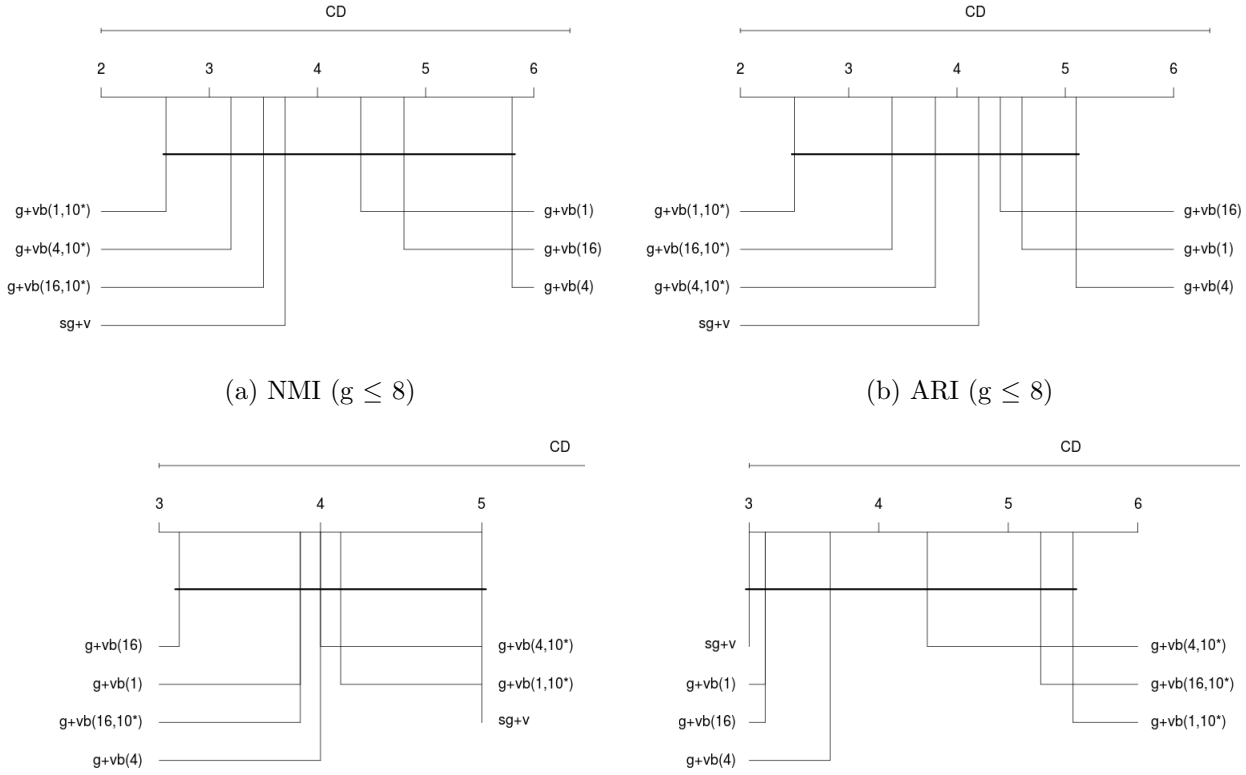
	Dirichlet priors		Gamma priors					
settings	$a_1, \ldots, a_g$	$b_1, \ldots, b_c$	ζ	η	$\alpha_1 \ldots, \alpha_g$	$\beta_1, \ldots, \beta_g$	τ	υ
(1)	1	1	1	1	1	1	1	1
(4)	4	4	1	1	1	1	1	1
(16)	16.5	16.5	1	1	1	1	1	1
(1,10*)	1	1	1	1	10*	1	1	10*
(4,10*)	4	4	1	1	10*	1	1	10*
(16,10*)	16.5	16.5	1	1	10*	1	1	10*

$g > 8$) according to the clustering scores. Several studies [272, 273, 274] have shown the relevance of statistical comparisons in analyzing the behavior of multiple algorithms in an experimental set-up. In ours, the difference between the mean scores of each settings is of interest and quantified using the average ranks (AR) method proposed by Brazdil and Soares in [275] w.r.t. each algorithm, namely `VBEM` initialized with the `Gibbs sampler solutions` (denoted `g+vb`"), `VBEM` (denoted "`vb`") and the `Gibbs sampler` (denoted "`g`"). This method is inspired by Friedman's M statistic and consists in measuring the error rates (here the NMI and ARI mean scores respectively) to assign a rank accordingly. In addition, we also use the Nemenyi non-parametric statistical test [269, 270] (part of the "scmamp" R package [1]) which quantifies the performance between several classifiers by measuring the pairwise Critical Difference (CD) between their average ranks. Tables 5.7 and 5.8 display the empirical results in terms of mean and standard deviation obtained over the best 10 solutions (criterion-wise) from a set of 30 trials. The average score and average rank are also given for each setting w.r.t. each algorithm. `SEM-Gibbs` and `VEM` are respectively denoted "`sg`" and "`v`".

In terms of NMI and ARI, it is clear that partitions where $g \leq 8$ undeniably favor $(1, 10^*)$ indifferently of the algorithm employed. For partitions where $g > 8$, the narrative is more complex. Overall (16) is selected as the best setting when $g > 8$ since (16) obtains the best rank in terms of NMI. However, in terms of ARI, MLE seems to outperform Bayesian Inference as `sg+v` and `sg` obtain better ranks than `g+vb(16)` and `g(16)` respectively. Using `VBEM`, `vb(16)` remains ahead of `VEM`. These results are summarized in Figure 5.10 which shows the diagrams of pairwise Critical Difference (CD) between the settings in terms of NMI and ARI for `g+vb`. Each figure illustrates the critical difference between the various settings according to their average ranks. Therefore, the far-left setting is supposedly the

1. `https://cran.r-project.org/web/packages/scmamp/vignettes/Statistical_assessment_of_the_differences.html`

one with the best average rank, while the far-right will be the worst. In our graphics, there is only one bold line, indicating that the differences between results are not significant.



(a) NMI ($g \leq 8$)

(b) ARI ($g \leq 8$)

(c) NMI ($g > 8$)

(d) ARI ($g > 8$)

FIGURE 5.10 – Critical Difference (CD) between the results of GPLBM according to the various hyperparameters settings.

TABLE 5.7 – NMI scores for the "`Gibbs sampler + VBEM`", the "`Gibbs sampler`" and "`VBEM`". "AS" and "AR" stand for Average Scores and Average Ranks.

	MLE	Gibbs-Sampler+VBEM (old)			Gibbs-Sampler+VBEM (new)			SEM-Gibbs	Gibbs-Sampler (old)			Gibbs-Sampler (new)			VEM	VBEM (old)		VBEM (new)		
Data	sg+v	g+vb(1)	g+vb(4)	g+vb(16)	g+vb(1,10^3)	g+vb(4,10^3)	g+vb(16,10^3)	sg	g(1)	g(4)	g(16)	g(1,10^3)	g(4,10^3)	g(16,10^3)	v	vb(4)	vb(16)	vb(1,10^3)	vb(4,10^3)	vb(16,10^3)
C3	0.95±0.0019	0.95±0.0017	0.95±0.002	0.95±0.0014	0.95±0.0016	0.95±0.0017	0.95±0.0016	0.95±0.0034	0.95±0.0025	0.95±0.0017	0.95±0.0017	0.95±0.0022	0.95±0.0027	0.95±0.0013	0.95±0.00082	0.95±0.0023	0.95±0.0021	0.95±0.0014	0.95±0.0011	0.95±0.0029
C4	0.75±0.048	0.78±0.019	0.73±0.052	0.71±0.049	**0.79**±0.013	0.76±0.034	0.78±0.037	0.74±0.045	0.76±0.019	0.72±0.048	0.7±0.048	**0.77**±0.012	0.75±0.031	0.76±0.044	**0.78**±0.018	0.77±0.021	**0.78**±0.019	0.77±0.023	0.75±0.0091	0.76±0.032
NG5	0.75±0.032	0.73±0.058	0.72±0.047	0.74±0.047	0.76±0.035	**0.77**±0.026	0.72±0.047	0.75±0.032	0.73±0.057	0.72±0.046	0.74±0.047	**0.76**±0.036	**0.76**±0.027	0.71±0.048	0.71±0.054	0.69±0.053	0.71±0.045	**0.72**±0.051	0.7±0.012	0.69±0.046
K1B	0.58±0.033	0.58±0.039	0.58±0.031	0.58±0.05	0.6±0.028	**0.61**±0.042	0.6±0.026	0.59±0.044	0.58±0.039	0.58±0.031	0.58±0.05	0.6±0.027	**0.62**±0.034	0.61±0.027	0.55±0.031	0.55±0.032	0.55±0.033	**0.59**±0.036	0.57±0.035	0.56±0.03
SPS	**0.66**±0.045	0.62±0.05	0.6±0.047	0.64±0.043	0.64±0.027	0.61±0.037	0.65±0.026	**0.66**±0.044	0.62±0.05	0.6±0.048	0.64±0.043	0.64±0.026	0.61±0.038	0.65±0.025	0.62±0.048	0.61±0.035	0.63±0.038	**0.64**±0.043	0.61±0.025	0.62±0.029
AS	0.74	0.73	0.72	0.72	**0.75**	0.74	0.74	**0.74**	0.73	0.71	0.72	**0.74**	**0.74**	**0.74**	0.72	0.71	0.72	**0.73**	0.72	0.72
AR	3.70	4.40	5.80	4.80	**2.60**	3.20	3.50	3.40	4.50	5.80	4.90	**2.60**	3.30	3.50	3.20	4.60	2.90	**2.00**	4.20	4.10
WAP	0.54±0.02	0.54±0.022	0.53±0.027	0.53±0.021	0.52±0.011	0.52±0.015	0.53±0.022	**0.54**±0.02	**0.54**±0.022	0.53±0.027	0.53±0.02	0.52±0.011	0.52±0.015	0.53±0.022	0.47±0.024	0.47±0.018	**0.48**±0.029	0.46±0.014	0.47±0.027	0.45±0.016
K1A	0.5±0.023	**0.53**±0.025	0.51±0.019	**0.53**±0.017	0.5±0.017	0.5±0.015	0.49±0.023	0.51±0.022	**0.53**±0.026	0.51±0.019	**0.53**±0.015	0.5±0.017	0.5±0.016	0.5±0.022	**0.45**±0.029	**0.45**±0.013	**0.45**±0.027	0.43±0.013	0.42±0.014	0.43±0.017
T2	0.77±0.012	0.77±0.014	0.78±0.011	0.79±0.013	**0.8**±0.018	0.79±0.012	0.79±0.013	0.77±0.013	0.77±0.014	0.78±0.011	0.79±0.013	**0.8**±0.018	0.79±0.012	0.79±0.013	**0.79**±0.022	0.78±0.019	**0.79**±0.018	**0.79**±0.013	**0.79**±0.012	**0.79**±0.0099
R40	0.42±0.054	0.51±0.038	0.52±0.03	0.52±0.019	0.52±0.017	**0.53**±0.0094	**0.53**±0.0095	0.42±0.033	0.51±0.039	0.52±0.03	0.52±0.019	**0.53**±0.015	**0.53**±0.0097	**0.53**±0.0093	0.51±0.021	0.51±0.026	**0.53**±0.018	0.49±0.0067	0.24±0.077	0.45±0.1
AS	0.56	**0.59**	0.58	**0.59**	0.58	0.58	0.58	0.56	**0.59**	0.58	**0.59**	**0.59**	0.58	**0.59**	**0.56**	0.55	**0.56**	0.54	0.48	0.53
AR	5.00	3.88	4.00	**3.12**	4.12	4.00	3.88	4.62	3.88	4.25	**3.25**	3.88	4.38	3.75	2.62	3.38	**1.75**	4.12	4.50	4.62

TABLE 5.8 – ARI scores for the "`Gibbs sampler + VBEM`", the "`Gibbs sampler`" and "`VBEM`". "AS" and "AR" stand for Average Scores and Average Ranks.

	MLE	Gibbs-Sampler+VBEM (old)			Gibbs-Sampler+VBEM (new)			SEM-Gibbs	Gibbs-Sampler (old)			Gibbs-Sampler (new)			VEM	VBEM (old)		VBEM (new)		
Data	sg+v	g+vb(1)	g+vb(4)	g+vb(16)	g+vb(1,10^3)	g+vb(4,10^3)	g+vb(16,10^3)	sg	g(1)	g(4)	g(16)	g(1,10^3)	g(4,10^3)	g(16,10^3)	v	vb(4)	vb(16)	vb(1,10^3)	vb(4,10^3)	vb(16,10^3)
C3	0.97±0.0013	0.97±0.0013	0.97±0.0013	0.97±9e-04	0.97±0.0011	0.97±0.0014	0.97±0.0013	0.97±0.0025	0.97±0.0019	0.97±0.0013	0.97±0.0013	0.97±0.0017	0.97±0.002	0.97±0.0012	0.97±7e-04	0.97±0.0016	0.97±0.0017	0.97±0.0013	0.97±0.00069	0.97±0.0019
C4	0.7±0.15	**0.78**±0.023	0.63±0.16	0.62±0.13	0.79±0.018	0.74±0.096	0.76±0.1	0.68±0.14	0.76±0.023	0.62±0.15	0.6±0.13	**0.77**±0.014	0.72±0.091	0.72±0.13	0.78±0.023	0.77±0.028	**0.78**±0.025	0.76±0.032	0.75±0.02	0.74±0.093
NG5	0.74±0.071	0.7±0.095	0.67±0.088	0.71±0.096	0.75±0.066	**0.76**±0.07	0.67±0.091	0.74±0.071	0.69±0.094	0.68±0.082	0.71±0.095	0.75±0.067	**0.76**±0.069	0.67±0.091	**0.68**±0.088	0.62±0.09	0.67±0.084	0.67±0.093	0.66±0.029	0.64±0.079
K1B	0.41±0.061	0.42±0.061	0.44±0.063	0.43±0.1	**0.46**±0.065	0.45±0.082	**0.46**±0.075	0.46±0.1	0.42±0.061	0.44±0.063	0.44±0.095	0.47±0.054	**0.48**±0.077	**0.48**±0.082	0.41±0.083	0.4±0.08	0.42±0.056	**0.52**±0.073	0.48±0.059	0.47±0.057
SPS	**0.65**±0.1	0.59±0.078	0.6±0.063	**0.65**±0.046	0.61±0.065	0.57±0.072	**0.65**±0.031	0.64±0.11	0.59±0.078	0.6±0.064	**0.65**±0.046	0.61±0.065	0.57±0.073	**0.65**±0.029	0.56±0.094	0.57±0.054	0.58±0.1	**0.61**±0.11	0.57±0.046	0.6±0.036
AS	0.69	0.69	0.66	0.68	**0.72**	0.70	0.70	0.70	0.69	0.66	0.67	**0.71**	0.70	0.70	0.68	0.67	0.68	**0.71**	0.69	0.68
AR	4.20	4.60	5.10	4.40	**2.50**	3.80	3.40	3.80	4.80	5.30	4.40	**2.80**	3.40	3.50	3.40	4.60	2.90	**2.40**	3.80	3.90
WAP	**0.53**±0.023	0.5±0.053	0.51±0.057	**0.53**±0.038	0.45±0.042	0.49±0.032	0.43±0.073	**0.53**±0.023	0.5±0.053	0.51±0.057	0.52±0.041	0.45±0.042	0.49±0.032	0.43±0.073	0.33±0.074	0.33±0.063	**0.38**±0.061	0.32±0.044	0.33±0.057	0.33±0.034
K1A	0.52±0.026	**0.53**±0.034	0.52±0.044	0.51±0.046	0.48±0.047	0.52±0.054	0.48±0.059	0.52±0.027	**0.53**±0.033	0.52±0.044	0.5±0.061	0.48±0.047	0.51±0.052	0.48±0.057	0.37±0.055	**0.38**±0.07	0.34±0.046	0.34±0.074	0.33±0.073	0.35±0.064
T2	**0.77**±0.017	0.75±0.027	0.75±0.026	**0.77**±0.017	**0.77**±0.045	**0.77**±0.035	**0.77**±0.032	**0.77**±0.017	0.75±0.027	0.75±0.026	**0.77**±0.017	**0.77**±0.044	**0.77**±0.035	**0.77**±0.032	**0.74**±0.049	0.72±0.048	**0.74**±0.053	0.72±0.039	0.72±0.04	0.72±0.038
R40	0.46±0.065	**0.5**±0.072	0.49±0.063	0.47±0.041	0.44±0.023	0.44±0.0093	0.46±0.026	0.46±0.06	0.48±0.077	**0.49**±0.063	0.47±0.041	0.45±0.023	0.44±0.013	0.46±0.026	0.47±0.046	0.46±0.037	**0.49**±0.054	0.43±0.01	0.22±0.073	0.43±0.1
AS	**0.57**	0.57	0.57	0.57	0.54	0.56	0.54	**0.57**	0.56	**0.57**	0.56	0.54	0.55	0.54	0.48	0.47	**0.49**	0.45	0.40	0.46
AR	**3.00**	3.12	3.62	3.12	5.50	4.38	5.25	**2.75**	3.38	3.25	3.25	5.38	4.75	5.25	2.25	3.00	**2.00**	4.88	5.00	3.88

5.2 Conclusion

The objective of co-clustering can be achieved by different approaches. In our proposal, we chose the approach based on the *Poisson Latent Block Model* [257] for its flexibility. By handling the noisy data directly in the inference, we have proposed a variant of this model which ensures that the clustering remains unbiased in regards to irrelevant information leaked despite the pre-processing. The diagonal and column parameterization looks well suited to detect the presence of noisy features while improving the learning of latent variables. In addition, the model provides an estimate of the precise amount of noise contained within a document-term matrix. We also showed, in various noisy situations, that our model guaranties better results compared to competitive methods devoted to the same task.

The Bayesian inference introduced to reduce the number of empty cluster solutions proves effective and leads overall to better clustering performances. In the future, it be would interesting to modulate the heterogeneity factor, which appears to play an important role in the clustering performance of LBM. Diagonal parameterization appeared beneficial as regards block heterogeneity in sparse data analysis, but even better performance might be obtainable with more homogeneous parameterization. Different ways of achieving this might be explored, including, for instance, Bayesian inference on Plbm. Consideration of other conjugate priors, such as the newly weighted Lindley distribution [276] for the block parameters, could also be useful in handling heterogeneity.

Conclusion and Perspectives

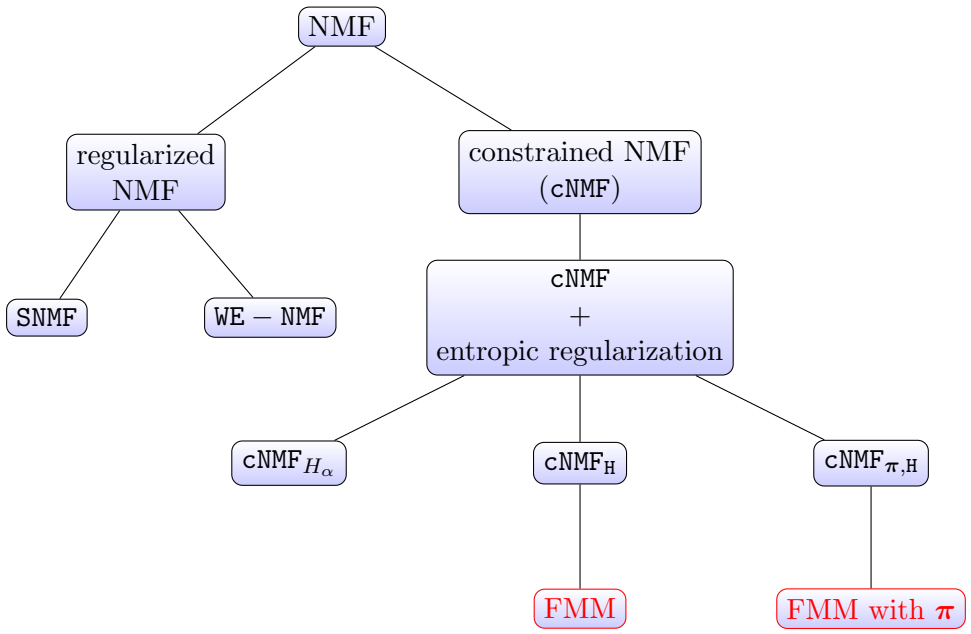


FIGURE 5.11 – Summary of the algorithms and relations explored in this thesis

Two different clustering approaches for the analysis of text data were studied in this thesis. The first through dimensionality reduction using Nonnegative Matrix Factorization, the second throughout the use of Finite mixture models and Latent Block Models. In the light of the above, we have presented several paths for improving document clustering of highly dimensional datasets.

First, regarding NMF, we reveal that the ultimate best solutions do no necessarily carry the best clustering after a study of the optimal local minima obtained by some usual cost functions. As a consequence, we have suggested a consensus approach for handling this behavior and extracting the best clustering partition. This approach uses a set of optimal best solutions and proved effective as the consensus partition successfully outperforms the clustering retrieved from the best local minimum. (Chapter 2)

Secondly, we tackled the problem of NMF from an optimization point of view and undertook to improve clustering using several objective regularizations. In the context of text analysis, we suggested two semantic regularizations. The first with a Neural Embedding method, supplying a word-context matrix (SNMF). The regularization is achieved by performing the joint decomposition of the data matrix and the word-context matrix simultaneously into a shared factor. The Frobenius norm is set as the cost function. The second regularization is achieved using the embedding of the Kantorovich–Rubinstein

distance, which proved effectiveness in capturing non linear relations between histograms in general as much as term features ($\texttt{WE} - \texttt{NMF}$). In this case, the cost function is the I-divergence. For both approaches, a set of multiplicative update rules is derived. In both cases, the regularization shows success in leveraging hidden semantic relationships which ultimately led to an improvement of the clustering partitions. In addition, the consensus approach introduced for the original NMF was confirmed to be feasible for the class of regularized NMF problems. Moreover, the solutions obtained with the Frobenius norm were shown to be less prominent than the ones derived from the I-divergence, for the task document-term clustering. (Chapter 3)

Thirdly, we proposed a clustering characterization of NMF called `cNMF` which introduced an additional probability constraint in the optimization problem of NMF. Thanks to this new representation, Information theoretic measures from the class of Rényi entropies are integrated to `cNMF`'s objective and maximized in order to increase cluster validity ($\texttt{cNMF}_{H_\alpha}$). This new method proves effectiveness in handling the search of the best clustering partitions while diminishing NMF's sensitivity to starting points. Extended experiments on several benchmark datasets shows the superiority of $\texttt{cNMF}_{H_1}$ compared to the current state-of-the-art algorithms. In an attempt to derive an more efficient gradient for accelerating the convergence, the connection of this new method with the Poisson Finite mixture model is characterized. Furthermore, using the properties of convex function and Bregman divergences, this connection is generalized to FMMs of exponential Famillies. Finally, a comparative study between directional and count data methods across the unit-sphere highlights the minimum entropy of the Poisson distribution for sparse random variables and its advantage in this case. (Chapter 4)

Finally, using Finite mixture of Poisson distributions and taking advantages of the *Latent Block Model* flexibility, we are able to tackle the recurrent problem of noisy features encountered in text analysis. Thanks to the block parameterization of LBM, a dedicated column cluster of noisy features is implemented, and inference for learning its unique parameter is easily integrated to the VEM algorithm. Besides, Bayesian Inference is later employed to resolve the overfitting issue leading to empty cluster solutions. A study of the advocated priors usually applied in this situation is achieved in terms of clustering, and highlighted performance losses. Due to the parsimony of our model (Diagonal parameterization), to remedy, we allow specific priors on the block parameters which results in improvements in terms of diminished overfitting as well as clustering performance. (Chapter 5)

In Figure 5.11, we illustrate a summary of the different algorithms proposed in this thesis and their

relations

Following the results obtained with NMF, several perspectives can be determined. As mentioned previously, the context matrix $\boldsymbol{M}$ in `SNMF` can be built using larger external corpora. $\boldsymbol{M}$ could also be domain-specific, e.g. in sentiments analysis where $\boldsymbol{M}$ could describe relationships between words in terms of their positivity, negativity or neutrality. Regarding the clustering characterization, for instance, `cNMF` could be extended to Nonnegative Matrix Tri-Factorization (NMTF) as described by the following problem :

$$\min_{\substack{\boldsymbol{Z}\in\mathbb{R}_+^{n\times g},\boldsymbol{S}\in\mathbb{R}_+^{g\times c},\boldsymbol{W}\in\mathbb{R}_+^{d\times c},\\ \boldsymbol{Z}\mathbf{1}_g=\mathbf{1}_n,\boldsymbol{W}\mathbf{1}_c=\mathbf{1}_d}} \{\mathcal{F}(\boldsymbol{Z},\boldsymbol{S},\boldsymbol{W})=\mathcal{D}(\boldsymbol{X},\boldsymbol{Z}\boldsymbol{S}\boldsymbol{W}^\top)\}. \tag{5.33}$$

This optimization remains manageable. However, setting a double entropic regularization becomes more challenging and would be a key achievement to establish the relation with the Latent block model. Naturally, a comparison of both algorithms would be interesting.

As reviewed in 2, several algorithms for solving the problem of NMF are denoted. Therefore deriving them for our extensions in order to achieve a comparative study of their solutions in terms of clustering would be key.

Similarly, as for GPLBM, integrate the noise parameterization into a matrix approximation problem could be effective for gaining better low dimensional spaces.

Furthermore, in the case of GPLBM, it be would interesting to adapt GPLBM to the class of infinite mixture models by deriving a non-parametric version of the model using the Dirichlet process so that the number of co-clusters can also be estimated.

Bibliographie

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer et R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, n°. 6, p. 391–407, 1990.

[2] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, n°. 11, p. 559–572, 1901.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, n°. 6, p. 417, 1933.

[4] P. Paatero et U. Tapper, "Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, n°. 2, p. 111–126, 1994.

[5] D. D. Lee et H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, n°. 6755, p. 788–791, 1999.

[6] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," dans *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, n°. 14. Oakland, CA, USA, 1967, p. 281–297.

[7] S. C. Madeira et A. L. Oliveira, "Biclustering algorithms for biological data analysis : a survey," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 1, n°. 1, p. 24–45, 2004.

[8] G. Salton, E. A. Fox et H. Wu, "Extended boolean information retrieval," *Communications of the ACM*, vol. 26, n°. 11, p. 1022–1036, 1983.

[9] G. Salton, A. Wong et C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, n°. 11, p. 613–620, 1975.

[10] G. Sahon et M. McGill, "Introduction to modem information retrieval," *New York : McGraw Hill*, 1983.

[11] J. D. Anderson *et al.*, *Guidelines for indexes and related information retrieval devices.* Niso Press Bethesda, MD, 1997.

[12] G. Salton et C.-S. Yang, "On the specification of term values in automatic indexing," *Journal of documentation*, 1973.

[13] G. Salton et C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, n°. 5, p. 513–523, 1988.

[14] T. Korenius, J. Laurikkala, K. Järvelin et M. Juhola, "Stemming and lemmatization in the clustering of finnish text documents," dans *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 2004, p. 625–633.

[15] A. G. Jivani *et al.*, "A comparative study of stemming algorithms," *Int. J. Comp. Tech. Appl*, vol. 2, n°. 6, p. 1930–1938, 2011.

[16] G. W. Milligan et M. C. Cooper, "Methodology review : Clustering methods," *Applied psychological measurement*, vol. 11, n°. 4, p. 329–354, 1987.

[17] R. Sibson, "Slink : an optimally efficient algorithm for the single-link cluster method," *The computer journal*, vol. 16, n°. 1, p. 30–34, 1973.

[18] T. J. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons.* Munksgaard Copenhagen, 1948, vol. 5.

[19] R. R. Sokal, "A statistical method for evaluating systematic relationships." *Univ. Kansas, Sci. Bull.*, vol. 38, p. 1409–1438, 1958.

[20] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, n°. 301, p. 236–244, 1963.

[21] G. N. Lance et W. T. Williams, "A general theory of classificatory sorting strategies : 1. hierarchical systems," *The computer journal*, vol. 9, n°. 4, p. 373–380, 1967.

[22] R. M. Cormack, "A review of classification," *Journal of the Royal Statistical Society : Series A (General)*, vol. 134, n°. 3, p. 321–353, 1971.

[23] B. Everitt, S. Landau, M. Leese et D. Stahl, "Cluster analysis," 2011.

[24] M. Lorr, *Cluster analysis for social scientists*. Jossey-Bass Incorporated Pub, 1983.

[25] G. W. Milligan, "Ultrametric hierarchical clustering algorithms," *Psychometrika*, vol. 44, n°. 3, p. 343–346, 1979.

[26] A. D. Gordon, "A review of hierarchical classification," *Journal of the Royal Statistical Society : Series A (General)*, vol. 150, n°. 2, p. 119–137, 1987.

[27] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, n°. 4, p. 364–366, 1977.

[28] L. Kaufman et P. J. Rousseeuw, *Finding groups in data : an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[29] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, n°. 1, p. 27–64, 2007.

[30] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, n°. 4, p. 395–416, 2007.

[31] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." dans *Kdd*, vol. 96, n°. 34, 1996, p. 226–231.

[32] H.-P. Kriegel, P. Kröger, J. Sander et A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, vol. 1, n°. 3, p. 231–240, 2011.

[33] R. Xu et D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, n°. 3, p. 645–678, 2005.

[34] A. K. Jain et R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[35] H.-H. Bock, "Clustering methods : a history of k-means algorithms," *Selected contributions in data analysis and classification*, p. 161–172, 2007.

[36] L. Kaufman et P. J. Rousseeuw, "Partitioning around medoids (program pam)," *Finding groups in data : an introduction to cluster analysis*, vol. 344, p. 68–125, 1990.

[37] W. Kaplan, *Maxima and minima with applications : practical optimization and duality*. John Wiley & Sons, 1998, vol. 51.

[38] E. Diday, "Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques," *Revue de statistique appliquée*, vol. 19, n°. 2, p. 19–33, 1971.

[39] I. S. Dhillon et D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine learning*, vol. 42, n°. 1, p. 143–175, 2001.

[40] P. S. Bradley, O. L. Mangasarian et W. N. Street, "Clustering via concave minimization," *Advances in neural information processing systems*, p. 368–374, 1997.

[41] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh et J. Lafferty, "Clustering with bregman divergences." *Journal of machine learning research*, vol. 6, n°. 10, 2005.

[42] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the american statistical association*, vol. 67, n°. 337, p. 123–129, 1972.

[43] G. Govaert *et al.*, "Algorithme de classification d'un tableau de contingence." 1977.

[44] G. Govaert, "Classification croisée," Thèse de doctorat, Thèse d'état, Université Paris 6, France, 1983.

[45] ——, "Simultaneous clustering of rows and columns," *Control and Cybernetics*, vol. 24, p. 437–458, 1995.

[46] H. Bock, "Simultaneous clustering of objects and variables," 1980.

[47] F. Marcotorchino, "Block seriation problems : A unified approach. reply to the problem of h. garcia and jm proth (applied stochastic models and data analysis, 1,(1), 25–34 (1985))," *Applied Stochastic Models and Data Analysis*, vol. 3, n°. 2, p. 73–91, 1987.

[48] P. Arabie et L. J. Hubert, "The bond energy algorithm revisited," *IEEE transactions on systems, man, and cybernetics*, vol. 20, n°. 1, p. 268–274, 1990.

[49] J. Trejos et W. Castillo, "Simulated annealing optimization for two-mode partitioning," dans *Classification and Information Processing at the Turn of the Millennium.* Springer, 2000, p. 135–142.

[50] W. Castillo et J. Trejos, "Two-mode partitioning : review of methods and application of tabu search," *Classification, clustering, and data analysis*, p. 43–51, 2002.

[51] D. E. Duffy *et al.*, "A permutation-based algorithm for block clustering," *Journal of Classification*, vol. 8, n°. 1, p. 65–91, 1991.

[52] I. Van Mechelen et J. Schepers, "A unifying model for biclustering," dans *Compstat 2006-Proceedings in Computational Statistics*. Springer, 2006, p. 81–88.

[53] R. Rocci et M. Vichi, "Two-mode multi-partitioning," *Computational Statistics & Data Analysis*, vol. 52, n°. 4, p. 1984–2003, 2008.

[54] L. Labiod et M. Nadif, "A unified framework for data visualization and coclustering," *IEEE transactions on neural networks and learning systems*, vol. 26, n°. 9, p. 2194–2199, 2014.

[55] M. Ailem, F. Role et M. Nadif, "Co-clustering document-term matrices by direct maximization of graph modularity," dans *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, p. 1807–1810.

[56] Y. Cheng et G. M. Church, "Biclustering of expression data." dans *Ismb*, vol. 8, n°. 2000, 2000, p. 93–103.

[57] H. Cho, I. S. Dhillon, Y. Guan et S. Sra, "Minimum sum-squared residue co-clustering of gene expression data," dans *Proceedings of the 2004 SIAM international conference on data mining*. SIAM, 2004, p. 114–125.

[58] H. Cho et I. S. Dhillon, "Coclustering of human cancer microarrays using minimum sum-squared residue coclustering," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, n°. 3, p. 385–400, 2008.

[59] N. Gupta et S. Aggarwal, "Mib : Using mutual information for biclustering gene expression data," *Pattern Recognition*, vol. 43, n°. 8, p. 2692–2697, 2010.

[60] A. Tanay, R. Sharan et R. Shamir, "Biclustering algorithms : A survey," *Handbook of computational molecular biology*, vol. 9, n°. 1-20, p. 122–124, 2005.

[61] S. Busygin, O. Prokopyev et P. M. Pardalos, "Biclustering in data mining," *Computers & Operations Research*, vol. 35, n°. 9, p. 2964–2987, 2008.

[62] R. Santamaría, L. Quintales et R. Therón, "Methods to bicluster validation and comparison in microarray data," dans *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2007, p. 780–789.

[63] L. Li, Y. Guo, W. Wu, Y. Shi, J. Cheng et S. Tao, "A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data," *BioData mining*, vol. 5, n°. 1, p. 1–10, 2012.

[64] B. Hanczar et M. Nadif, "Using the bagging approach for biclustering of gene expression data," *Neurocomputing*, vol. 74, n°. 10, p. 1595–1605, 2011.

[65] ——, "Ensemble methods for biclustering tasks," *Pattern Recognition*, vol. 45, n°. 11, p. 3938–3949, 2012.

[66] ——, "Precision-recall space to correct external indices for biclustering," dans *International Conference on Machine Learning*. PMLR, 2013, p. 136–144.

[67] J. Tantrum, A. Murua et W. Stuetzle, "Hierarchical model-based clustering of large datasets through fractionation and refractionation," *Information Systems*, vol. 29, n°. 4, p. 315–326, 2004.

[68] I. S. Dhillon et S. Sra, "Modeling data using directional distributions," Citeseer, Rapport technique, 2003.

[69] A. Banerjee et J. Ghosh, "Frequency sensitive competitive learning for clustering on high-dimensional hyperspheres," dans *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, vol. 2. IEEE, 2002, p. 1590–1595.

[70] ——, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *IEEE Transactions on Neural Networks*, vol. 15, n°. 3, p. 702–719, 2004.

[71] M. Li et L. Zhang, "Multinomial mixture model with feature selection for text clustering," *Knowledge-Based Systems*, vol. 21, n°. 7, p. 704–708, 2008.

[72] L. Rigouste, O. Cappé et F. Yvon, "Inference and evaluation of the multinomial mixture model for text clustering," *Information processing & management*, vol. 43, n°. 5, p. 1260–1280, 2007.

[73] J. Yin et J. Wang, "A dirichlet multinomial mixture model-based approach for short text clustering," dans *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, p. 233–242.

[74] J. Qiang, Y. Li, Y. Yuan et X. Wu, "Short text clustering based on pitman-yor process mixture model," *Applied Intelligence*, vol. 48, n°. 7, p. 1802–1812, 2018.

[75] A. Rau, G. Celeux, M.-L. Martin-Magniette et C. Maugis-Rabusseau, "Clustering high-throughput sequencing data with poisson mixture models," Thèse de doctorat, Inria, 2011.

[76] K. P. Nigam, "Using unlabeled data to improve text classification," CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, Rapport technique, 2001.

[77] K. Nigam, A. K. McCallum, S. Thrun et T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, n°. 2, p. 103–134, 2000.

[78] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," dans *AAAI-98 workshop on learning for text categorization*, vol. 752, n°. 1. Citeseer, 1998, p. 41–48.

[79] G. Celeux et G. Govaert, "A classification em algorithm for clustering and two stochastic versions," *Computational statistics & Data analysis*, vol. 14, n°. 3, p. 315–332, 1992.

[80] G. Govaert et M. Nadif, "Clustering with block mixture models," *Pattern Recognition*, vol. 36, n°. 2, p. 463–473, 2003.

[81] M. Nadif et G. Govaert, "Block clustering of contingency table and mixture model," dans *International Symposium on Intelligent Data Analysis*. Springer, 2005, p. 249–259.

[82] G. Govaert et M. Nadif, "Clustering of contingency table and mixture model," *European Journal of Operational Research*, vol. 183, n°. 3, p. 1055–1066, 2007.

[83] ——, "Latent block model for contingency table," *Communications in Statistics—Theory and Methods*, vol. 39, n°. 3, p. 416–425, 2010.

[84] ——, *Co-clustering : models, algorithms and applications*. John Wiley & Sons, 2013.

[85] C. H. Papadimitriou, P. Raghavan, H. Tamaki et S. Vempala, "Latent semantic indexing : A probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, n°. 2, p. 217–235, 2000.

[86] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, n°. 1, p. 177–196, 2001.

[87] ——, "Probabilistic latent semantic analysis," *arXiv preprint arXiv :1301.6705*, 2013.

[88] E. Gaussier et C. Goutte, "Relation between plsa and nmf and implications," dans *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, p. 601–602.

[89] C. Ding, T. Li et W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," *Computational Statistics & Data Analysis*, vol. 52, n°. 8, p. 3913–3927, 2008.

[90] A. P. Dempster, N. M. Laird et D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 39, n°. 1, p. 1–22, 1977.

[91] A. J. Scott et M. J. Symons, "Clustering methods based on likelihood ratio criteria," *Biometrics*, p. 387–397, 1971.

[92] M. J. Symons, "Clustering criteria and multivariate normal mixtures," *Biometrics*, p. 35–43, 1981.

[93] J. T. Ormerod et M. P. Wand, "Explaining variational approximations," *The American Statistician*, vol. 64, n°. 2, p. 140–153, 2010.

[94] D. Titterington *et al.*, "Bayesian methods for neural networks and related models," *Statistical Science*, vol. 19, n°. 1, p. 128–139, 2004.

[95] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola et L. K. Saul, "An introduction to variational methods for graphical models," dans *Learning in graphical models*. Springer, 1998, p. 105–161.

[96] ——, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, n°. 2, p. 183–233, 1999.

[97] G. Parisi, *Statistical field theory*. Addison-Wesley, 1988.

[98] J. Zhang, "The mean field theory in em procedures for markov random fields," *IEEE Transactions on signal processing*, vol. 40, n°. 10, p. 2570–2583, 1992.

[99] ——, "The mean field theory in em procedures for blind markov random field image restoration," *IEEE Transactions on Image Processing*, vol. 2, n°. 1, p. 27–40, 1993.

[100] N. Metropolis et S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, n°. 247, p. 335–341, 1949.

[101] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[102] A. E. Gelfand et A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, n°. 410, p. 398–409, 1990.

[103] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods.* Department of Computer Science, University of Toronto Toronto, Ontario, Canada, 1993.

[104] S. M. Lynch, *Introduction to applied Bayesian statistics and estimation for social scientists.* Springer Science & Business Media, 2007.

[105] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari et D. B. Rubin, *Bayesian data analysis.* CRC press, 2013.

[106] S. Geman et D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, n°. 6, p. 721–741, 1984.

[107] A. E. Raftery et S. Lewis, "How many iterations in the gibbs sampler ?" WASHINGTON UNIV SEATTLE DEPT OF STATISTICS, Rapport technique, 1991.

[108] W. R. Gilks, S. Richardson et D. J. Spiegelhalter, "Introducing markov chain monte carlo," *Markov chain Monte Carlo in practice*, p. 1, 1995.

[109] L. Tierney, "Introduction to general state-space markov chain theory," *Markov chain Monte Carlo in practice*, p. 59–74, 1996.

[110] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, n°. 3, p. 379–423, 1948.

[111] T. M. Cover, *Elements of information theory.* John Wiley & Sons, 1999.

[112] D. L. Davies et D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, n°. 2, p. 224–227, 1979.

[113] T. Caliński et J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, n°. 1, p. 1–27, 1974.

[114] A. E. Raftery, "A note on bayes factors for log-linear contingency table models with vague prior information," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 48, n°. 2, p. 249–250, 1986.

[115] P. J. Rousseeuw, "Silhouettes : a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, p. 53–65, 1987.

[116] E. Rendón, I. M. Abundez, C. Gutierrez, S. D. Zagal, A. Arizmendi, E. M. Quiroz et H. E. Arzate, "A comparison of internal and external cluster validation indexes," dans *Proceedings of the 2011 American Conference, San Francisco, CA, USA*, vol. 29, 2011, p. 1–10.

[117] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote sensing of environment*, vol. 37, n°. 1, p. 35–46, 1991.

[118] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote sensing of Environment*, vol. 62, n°. 1, p. 77–89, 1997.

[119] A. Strehl et J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, n°. Dec, p. 583–617, 2002.

[120] N. D. Cahill, "Normalized measures of mutual information with general definitions of entropy for multimodal image registration," dans *International workshop on biomedical image registration*. Springer, 2010, p. 258–268.

[121] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, n°. 336, p. 846–850, 1971.

[122] L. Hubert et P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, n°. 1, p. 193–218, 1985.

[123] G. H. Golub et C. F. Van Loan, *Matrix computations*. JHU press, 2013, vol. 4.

[124] L. Eldén, *Matrix methods in data mining and pattern recognition*. SIAM, 2007.

[125] A. Cauchy, "Sur l'équation dé l'aide de laquelle on détermine les inégalités séculaires des mouvements des planetes, vol. 9," *Oeuvres Completes (IIeme Série)*, 1829.

[126] C. Jordan, "Mémoire sur les formes bilinéaires." *Journal de mathématiques pures et appliquées*, vol. 19, p. 35–54, 1874.

[127] H. Abdi et L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews : computational statistics*, vol. 2, n°. 4, p. 433–459, 2010.

[128] C. Eckart et G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, n°. 3, p. 211–218, 1936.

[129] I. S. Dhillon et J. A. Tropp, "Matrix nearness problems with bregman divergences," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, n°. 4, p. 1120–1146, 2008.

[130] N.-D. Ho, "Nonnegative matrix factorization algorithms and applications," Thèse de doctorat, PhD thesis, Université catholique de Louvain, 2008.

[131] I. Markovsky et K. Usevich, *Low rank approximation*. Springer, 2012, vol. 139.

[132] D. D. Lee et H. S. Seung, "Algorithms for non-negative matrix factorization," dans *Advances in neural information processing systems*, 2001, p. 556–562.

[133] V. P. Pauca, F. Shahnaz, M. W. Berry et R. J. Plemmons, "Text mining using non-negative matrix factorizations," dans *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 2004, p. 452–456.

[134] E. Hosseini-Asl et J. M. Zurada, "Nonnegative matrix factorization for document clustering : A survey," dans *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2014, p. 726–737.

[135] T. Li et C.-c. Ding, "Nonnegative matrix factorizations for clustering : A survey," dans *Data Clustering*. Chapman and Hall/CRC, 2018, p. 149–176.

[136] C. Févotte, N. Bertin et J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence : With application to music analysis," *Neural computation*, vol. 21, n°. 3, p. 793–830, 2009.

[137] X. Fu, K. Huang, N. D. Sidiropoulos et W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics : Identifiability, algorithms, and applications." *IEEE Signal Process. Mag.*, vol. 36, n°. 2, p. 59–80, 2019.

[138] D. Guillamet, B. Schiele et J. Vitria, "Analyzing non-negative matrix factorization for image classification," dans *Object recognition supported by user interaction for service robots*, vol. 2. IEEE, 2002, p. 116–119.

[139] V. P. Pauca, J. Piper et R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear algebra and its applications*, vol. 416, n°. 1, p. 29–47, 2006.

[140] J. M. Ortega et W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[141] M. Chu, F. Diele, R. Plemmons et S. Ragni, "Optimality, computation, and interpretation of nonnegative matrix factorizations," dans *SIAM Journal on Matrix Analysis*. Citeseer, 2004.

[142] E. F. Gonzalez et Y. Zhang, "Accelerating the lee-seung algorithm for nonnegative matrix factorization," Rapport technique, 2005.

[143] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, n°. 6, p. 1589–1596, 2007.

[144] E. C. Chi et T. G. Kolda, "On tensors, sparsity, and nonnegative factorizations," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, n°. 4, p. 1272–1299, 2012.

[145] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, n°. 10, p. 2756–2779, 2007.

[146] J. Nocedal et S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[147] N. Gillis, "The why and how of nonnegative matrix factorization," 2014.

[148] N.-D. Ho, P. Van Dooren et V. D. Blondel, "Descent methods for nonnegative matrix factorization," dans *Numerical Linear Algebra in Signals, Systems and Control*. Springer, 2011, p. 251–293.

[149] A. N. Langville, C. D. Meyer, R. Albright, J. Cox et D. Duling, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," *arXiv preprint arXiv :1407.7299*, 2014.

[150] S. Wild, W. S. Wild, J. Curry, A. Dougherty et M. Betterton, "Seeding non-negative matrix factorizations with the spherical k-means clustering," Thèse de doctorat, University of Colorado, 2003.

[151] S. Wild, J. Curry et A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern recognition*, vol. 37, n°. 11, p. 2217–2232, 2004.

[152] C. Boutsidis et E. Gallopoulos, "Svd based initialization : A head start for nonnegative matrix factorization," *Pattern recognition*, vol. 41, n°. 4, p. 1350–1362, 2008.

[153] S. M. Atif, S. Qazi et N. Gillis, "Improved svd-based initialization for nonnegative matrix factorization using low-rank correction," *Pattern Recognition Letters*, vol. 122, p. 53–59, 2019.

[154] H. Qiao, "New SVD based initialization strategy for non-negative matrix factorization," *Pattern Recognition Letters*, vol. 63, p. 71–77, 2015.

[155] N. Gillis *et al.*, "Nonnegative matrix factorization : Complexity, algorithms and applications," *Unpublished doctoral dissertation, Université catholique de Louvain. Louvain-La-Neuve : CORE*, 2011.

[156] D. Kuang, C. Ding et H. Park, "Symmetric nonnegative matrix factorization for graph clustering," dans *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, p. 106–117.

[157] C. Ding, T. Li, W. Peng et H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," dans *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, p. 126–135.

[158] J. Yoo et S. Choi, "Orthogonal nonnegative matrix factorization : Multiplicative updates on stiefel manifolds," dans *International conference on intelligent data engineering and automated learning*. Springer, 2008, p. 140–147.

[159] S. Choi, "Algorithms for orthogonal nonnegative matrix factorization," dans *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*. IEEE, 2008, p. 1828–1832.

[160] C. H. Ding, T. Li et M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, n°. 1, p. 45–55, 2008.

[161] Z. Yuan et E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction," dans *Scandinavian Conference on Image Analysis*. Springer, 2005, p. 333–342.

[162] Z. Yuan, Z. Yang et E. Oja, "Projective nonnegative matrix factorization : Sparseness, orthogonality, and clustering," *Neural Process. Lett*, p. 11–13, 2009.

[163] D. Cai, X. He, J. Han et T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, n°. 8, p. 1548–1560, 2010.

[164] F. Shang, L. Jiao et F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering," *Pattern Recognition*, vol. 45, n°. 6, p. 2237–2250, 2012.

[165] J. J.-Y. Wang, H. Bensmail et X. Gao, "Multiple graph regularized nonnegative matrix factorization," *Pattern Recognition*, vol. 46, n°. 10, p. 2840–2847, 2013.

[166] J. Huang, F. Nie, H. Huang et C. Ding, "Robust manifold nonnegative matrix factorization," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, n°. 3, p. 1–21, 2014.

[167] H. Gao, F. Nie et H. Huang, "Local centroids structured non-negative matrix factorization," dans *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, n°. 1, 2017.

[168] Y.-X. Wang et Y.-J. Zhang, "Nonnegative matrix factorization : A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, n°. 6, p. 1336–1353, 2012.

[169] A. Cichocki, R. Zdunek, A. H. Phan et S.-i. Amari, *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[170] C. Ding, X. He et H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," dans *Proceedings of the 2005 SIAM international conference on data mining.* SIAM, 2005, p. 606–610.

[171] T. Li et C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," dans *ICDM*, 2006, p. 362–371.

[172] B. Long, Z. Zhang et P. S. Yu, "Co-clustering by block value decomposition," dans *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, p. 635–640.

[173] J. Yoo et S. Choi, "Orthogonal nonnegative matrix tri-factorization for co-clustering : Multiplicative updates on stiefel manifolds," *Information processing & management*, vol. 46, n°. 5, p. 559–570, 2010.

[174] Q. Gu et J. Zhou, "Co-clustering on manifolds," dans *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, p. 359–368.

[175] H. Wang, F. Nie, H. Huang et F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," dans *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[176] M. Febrissy et M. Nadif, "Classification croisée par tri-factorisation matricielle non-négative," dans *2018 25th SFC.* Société Francophone de la Classification, 2018, p. 99–102.

[177] H. Wang, F. Nie, H. Huang et C. Ding, "Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation," dans *2011 IEEE 11th international conference on data mining.* IEEE, 2011, p. 774–783.

[178] A. J. Sharkey, "Multi-net systems," dans *Combining artificial neural nets.* Springer, 1999, p. 1–30.

[179] J. Ghosh, "Multiclassifier systems : Back to the future," dans *International Workshop on Multiple Classifier Systems.* Springer, 2002, p. 1–15.

[180] A. J. Sharkey, "On combining artificial neural nets," *Connection Science*, vol. 8, n°. 3-4, p. 299–314, 1996.

[181] P. S. Bradley et U. M. Fayyad, "Refining initial points for k-means clustering." dans *ICML*, vol. 98. Citeseer, 1998, p. 91–99.

[182] A. Topchy, A. K. Jain et W. Punch, "A mixture model for clustering ensembles," dans *SDM.* SIAM, 2004, p. 379–390.

[183] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, n°. 2, p. 461–464, 1978.

[184] K. Allab, L. Labiod et M. Nadif, "A semi-nmf-pca unified framework for data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, n°. 1, p. 2–16, 2016.

[185] ——, "Simultaneous spectral data embedding and clustering," *IEEE transactions on neural networks and learning systems*, vol. 29, n°. 12, p. 6396–6401, 2018.

[186] M. Ailem, A. Salah et M. Nadif, "Non-negative matrix factorization meets word embedding," dans *SIGIR*, 2017, p. 1081–1084.

[187] A. Salah, M. Ailem et M. Nadif, "A way to boost SEMI-NMF for document clustering," dans *CIKM*, 2017, p. 2275–2278.

[188] ——, "Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering," dans *AAAI*, 2018, p. 3992–3999.

[189] Z. S. Harris, "Distributional structure," *Word*, vol. 10, n°. 2-3, p. 146–162, 1954.

[190] O. Levy et Y. Goldberg, "Neural word embedding as implicit matrix factorization," *Advances in neural information processing systems*, vol. 27, p. 2177–2185, 2014.

[191] K. Lund et C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior research methods, instruments, & computers*, vol. 28, n°. 2, p. 203–208, 1996.

[192] Y. Bengio, R. Ducharme, P. Vincent et C. Janvin, "A neural probabilistic language model," *The journal of machine learning research*, vol. 3, p. 1137–1155, 2003.

[193] A. Mnih et G. E. Hinton, "A scalable hierarchical distributed language model," *Advances in neural information processing systems*, vol. 21, p. 1081–1088, 2008.

[194] T. Mikolov, I. Sutskever, K. Chen, G. Corrado et J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv :1310.4546*, 2013.

[195] T. Mikolov, W.-t. Yih et G. Zweig, "Linguistic regularities in continuous space word representations," dans *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics : Human language technologies*, 2013, p. 746–751.

[196] M. Ailem, A. Salah et M. Nadif, "Non-negative matrix factorization meets word embedding," dans *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, p. 1081–1084.

[197] J. A. Bullinaria et J. P. Levy, "Extracting semantic representations from word co-occurrence statistics : A computational study," *Behavior research methods*, vol. 39, n°. 3, p. 510–526, 2007.

[198] J. Pennington, R. Socher et C. D. Manning, "Glove : Global vectors for word representation," dans *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, p. 1532–1543.

[199] R. Lebret et R. Collobert, "Word emdeddings through hellinger pca," *arXiv preprint arXiv :1312.5542*, 2013.

[200] T. Li, "A general model for clustering binary data," dans *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, p. 188–197.

[201] G. Karypis, "Cluto- a clustering toolkit," Minnesota Univ. Minneaplis Dept. of Computer Science, Rapport technique, 2002.

[202] W. Xu, X. Liu et Y. Gong, "Document clustering based on non-negative matrix factorization," dans *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, p. 267–273.

[203] B. Yang, X. Fu, N. D. Sidiropoulos et M. Hong, "Towards k-means-friendly spaces : Simultaneous deep learning and clustering," dans *international conference on machine learning*. PMLR, 2017, p. 3861–3870.

[204] D. Cai, X. He et J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, n°. 6, p. 902–913, 2010.

[205] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol et L. Bottou, "Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, n°. 12, 2010.

[206] D. Newman, J. H. Lau, K. Grieser et T. Baldwin, "Automatic evaluation of topic coherence," dans *Human language technologies : The 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, p. 100–108.

[207] D. Newman, S. Karimi et L. Cavedon, "External evaluation of topic models," dans *in Australasian Doc. Comp. Symp., 2009.* Citeseer, 2009.

[208] F. Role et M. Nadif, "Handling the impact of low frequency events on co-occurrence based measures of word similarity-a case study of pointwise mutual information." dans *KDIR*, 2011, p. 226–231.

[209] A. Mnih et R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, p. 1257–1264, 2007.

[210] R. Sandler et M. Lindenbaum, "Nonnegative matrix factorization with earth mover's distance metric," dans *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2009, p. 1873–1880.

[211] A. Rolet, M. Cuturi et G. Peyré, "Fast dictionary learning with a smoothed wasserstein loss," dans *Artificial Intelligence and Statistics*, 2016, p. 630–638.

[212] M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngole, D. Coeurjolly, M. Cuturi, G. Peyré et J.-L. Starck, "Wasserstein dictionary learning : Optimal transport-based unsupervised nonlinear dictionary learning," *SIAM Journal on Imaging Sciences*, vol. 11, n°. 1, p. 643–678, 2018.

[213] C. Villani, *Optimal transport : old and new.* Springer Science & Business Media, 2008, vol. 338.

[214] H. Ling et K. Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, n°. 5, p. 840–853, 2007.

[215] O. Pele et M. Werman, "Fast and robust earth mover's distances," dans *2009 IEEE 12th International Conference on Computer Vision.* IEEE, 2009, p. 460–467.

[216] M. Cuturi, "Sinkhorn distances : Lightspeed computation of optimal transport," dans *Advances in neural information processing systems*, 2013, p. 2292–2300.

[217] M. Cuturi et A. Doucet, "Fast computation of wasserstein barycenters," dans *International Conference on Machine Learning*, 2014, p. 685–693.

[218] S. Shirdhonkar et D. W. Jacobs, "Approximate earth mover's distance in linear time," dans *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, p. 1–8.

[219] L. Mi, W. Zhang, X. Gu et Y. Wang, "Variational wasserstein clustering," dans *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, p. 322–337.

[220] B. Muzellec, R. Nock, G. Patrini et F. Nielsen, "Tsallis regularized optimal transport and ecological inference," dans *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[221] A. Genevay, M. Cuturi, G. Peyré et F. Bach, "Stochastic optimization for large-scale optimal transport," dans *Advances in neural information processing systems*, 2016, p. 3440–3448.

[222] L. McInnes, J. Healy et J. Melville, "Umap : Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv :1802.03426*, 2018.

[223] Z. Yang, H. Zhang, Z. Yuan et E. Oja, "Kullback-leibler divergence for nonnegative matrix factorization," dans *International Conference on Artificial Neural Networks*. Springer, 2011, p. 250–257.

[224] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, n°. 4, p. 620, 1957.

[225] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[226] E. T. Jaynes, "Information theory and statistical mechanics. ii," *Physical review*, vol. 108, n°. 2, p. 171, 1957.

[227] E. J. Candes et T. Tao, "Decoding by linear programming," *IEEE transactions on information theory*, vol. 51, n°. 12, p. 4203–4215, 2005.

[228] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, n°. 4, p. 1289–1306, 2006.

[229] E. Rietsch *et al.*, "The maximum entropy approach to inverse problems-spectral analysis of short data records and density structure of the earth," *Journal of Geophysics*, vol. 42, n°. 1, p. 489–506, 1976.

[230] S. F. Gull et G. J. Daniell, "Image reconstruction from incomplete and noisy data," *Nature*, vol. 272, n°. 5655, p. 686–690, 1978.

[231] D. L. Donoho, I. M. Johnstone, J. C. Hoch et A. S. Stern, "Maximum entropy and the nearly black object," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 54, n°. 1, p. 41–67, 1992.

[232] V. Koltchinskii, "Sparse recovery in convex hulls via entropy penalization," *The Annals of Statistics*, vol. 37, n°. 3, p. 1332–1359, 2009.

[233] B. R. Frieden, "Restoring with maximum likelihood and maximum entropy," *JOSA*, vol. 62, n°. 4, p. 511–518, 1972.

[234] R. Lyon, J. Hollis et J. Dorband, "A maximum entropy method with a priori maximum likelihood constraints," *The Astrophysical Journal*, vol. 478, n°. 2, p. 658, 1997.

[235] A. B. TEMPLEMAN et L. Xingsi, "A maximum entropy approach to constrained non-linear programming," *Engineering Optimization+ A35*, vol. 12, n°. 3, p. 191–205, 1987.

[236] A. Berger, S. A. Della Pietra et V. J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, n°. 1, p. 39–71, 1996.

[237] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," 1996.

[238] A. Rényi *et al.*, "On measures of entropy and information," dans *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1 : Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[239] C. Févotte et J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, n°. 9, p. 2421–2456, 2011.

[240] C. Févotte et A. T. Cemgil, "Nonnegative matrix factorizations as probabilistic inference in composite models," dans *2009 17th European Signal Processing Conference*. IEEE, 2009, p. 1913–1917.

[241] A. P. Dempster, N. M. Laird et D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 39, n°. 1, p. 1–22, 1977.

[242] R. J. Hathaway, "Another interpretation of the em algorithm for mixture distributions," *Statistics & probability letters*, vol. 4, n°. 2, p. 53–56, 1986.

[243] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, n°. 3, p. 200–217, 1967.

[244] A. Banerjee, X. Guo et H. Wang, "Optimal bregman prediction and jensen's equality," dans *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* IEEE, 2004, p. 169.

[245] S. Eguchi et S. Kato, "Entropy and divergence associated with power function and the statistical application," *Entropy*, vol. 12, n°. 2, p. 262–274, 2010.

[246] H. Lantéri, "Divergences. scale invariant divergences. applications to linear inverse problems. nmf blind deconvolution," *arXiv preprint arXiv :2003.01411*, 2020.

[247] K. V. Mardia, "Characterizations of directional distributions," *A Modern Course on Statistical Distributions in Scientific Work*, p. 365–385, 1975.

[248] P. Harremoës, "Binomial and poisson distributions as maximum entropy distributions," *IEEE Transactions on Information Theory*, vol. 47, n°. 5, p. 2039–2041, 2001.

[249] O. Johnson, "Log-concavity and the maximum entropy property of the poisson distribution," *Stochastic Processes and their Applications*, vol. 117, n°. 6, p. 791–802, 2007.

[250] J. Fujiki et S. Akaho, "Spherical pca with euclideanization," dans *Proc. of Workshop on ACCV*, vol. 7, 2007.

[251] K. Liu, Q. Li, H. Wang et G. Tang, "Spherical principal component analysis," dans *Proceedings of the 2019 SIAM International Conference on Data Mining.* SIAM, 2019, p. 387–395.

[252] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," dans *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, p. 269–274.

[253] M. Ailem, F. Role et M. Nadif, "Graph modularity maximization as an effective method for co-clustering text data," *Knowledge-Based Systems*, vol. 109, p. 160–173, 2016.

[254] ——, "Model-based co-clustering for the effective handling of sparse data," *Pattern Recognition*, vol. 72, p. 108–122, 2017.

[255] M. H. Law, M. A. Figueiredo et A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, n°. 9, p. 1154–1166, 2004.

[256] S. Vaithyanathan et B. Dom, "Generalized model selection for unsupervised learning in high dimensions." dans *NIPS*. Citeseer, 1999, p. 970–976.

[257] G. Govaert et M. Nadif, "Mutual information, phi-squared and model-based co-clustering for contingency tables," *Advances in Data Analysis and Classification*, vol. 12, n°. 3, p. 455–488, 2018.

[258] C. Keribin, V. Brault, G. Celeux et G. Govaert, "Estimation and selection for the latent block model on categorical data," *Statistics and Computing*, vol. 25, n°. 6, p. 1201–1216, 2015.

[259] C. Keribin, V. Brault, G. Celeux, G. Govaert *et al.*, "Model selection for the binary latent block model," dans *Proceedings of COMPSTAT*, vol. 2012, 2012.

[260] R. M. Neal et G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," dans *Learning in graphical models*. Springer, 1998, p. 355–368.

[261] G. McLachlan et T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[262] M. Ailem, F. Role et M. Nadif, "Sparse poisson latent block model for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, n°. 7, p. 1563–1576, 2017.

[263] D. M. Blei, A. Y. Ng et M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, n°. Jan, p. 993–1022, 2003.

[264] M. Febrissy et M. Nadif, "A consensus approach to improve nmf document clustering," dans *International Symposium on Intelligent Data Analysis*. Springer, 2020, p. 171–183.

[265] T. Mikolov, K. Chen, G. Corrado et J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv :1301.3781*, 2013.

[266] S. Frühwirth-Schnatter, *Finite mixture and Markov switching models*. Springer Science & Business Media, 2006.

[267] ——, "Label switching under model uncertainty," *Mixtures : Estimation and Application*, p. 213–239, 2011.

[268] J. Rousseau et K. Mengersen, "Asymptotic behaviour of the posterior distribution in overfitted mixture models," *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, vol. 73, n°. 5, p. 689–710, 2011.

[269] P. Nemenyi, "Distribution-free mulitple comparisons phd thesis princeton university princeton," 1963.

[270] M. Hollander, D. A. Wolfe et E. Chicken, *Nonparametric statistical methods*. John Wiley & Sons, 2013, vol. 751.

[271] C. Buchta, M. Kober, I. Feinerer et K. Hornik, "Spherical k-means clustering," *Journal of Statistical Software*, vol. 50, n°. 10, p. 1–22, 2012.

[272] S. García, A. Fernández, J. Luengo et F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining : Experimental analysis of power," *Information sciences*, vol. 180, n°. 10, p. 2044–2064, 2010.

[273] S. Garcia et F. Herrera, "An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons," *Journal of machine learning research*, vol. 9, n°. Dec, p. 2677–2694, 2008.

[274] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, n°. Jan, p. 1–30, 2006.

[275] P. B. Brazdil et C. Soares, "A comparison of ranking methods for classification algorithm selection," dans *European conference on machine learning*. Springer, 2000, p. 63–75.

[276] M. Ghitany, F. Alqallaf, D. K. Al-Mutairi et H. Husain, "A two-parameter weighted lindley distribution and its applications to survival data," *Mathematics and Computers in simulation*, vol. 81, n°. 6, p. 1190–1201, 2011.

[277] R. A. Horn et C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[278] P. Jaccard, "The distribution of the flora in the alpine zone. 1," *New phytologist*, vol. 11, n°. 2, p. 37–50, 1912.

[279] B. Jørgensen, "Exponential dispersion models," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 49, n°. 2, p. 127–145, 1987.

[280] ——, "Exponential dispersion models and extensions : A review," *International Statistical Review/Revue Internationale de Statistique*, p. 5–20, 1992.

[281] B. Jorgensen, *The theory of dispersion models.* CRC Press, 1997.

[282] M. C. Tweedie, "An index which distinguishes between some important exponential families," dans *Statistics : Applications and new directions : Proc. Indian statistical institute golden Jubilee International conference*, vol. 579, 1984, p. 579–604.

[283] S. K. Bar-Lev, P. Enis *et al.*, "Reproducibility and natural exponential families with power variance functions," *The Annals of Statistics*, vol. 14, n°. 4, p. 1507–1522, 1986.

[284] P. Hougaard, "A class of multivanate failure time distributions," *Biometrika*, vol. 73, n°. 3, p. 671–678, 1986.

[285] ——, "Survival models for heterogeneous populations derived from stable distributions," *Biometrika*, vol. 73, n°. 2, p. 387–396, 1986.

[286] J. Forster et M. K. Warmuth, "Relative expected instantaneous loss bounds," *Journal of Computer and System Sciences*, vol. 64, n°. 1, p. 76–102, 2002.

[287] A. Banerjee, I. Dhillon, J. Ghosh et S. Merugu, "An information theoretic analysis of maximum likelihood mixture estimation for exponential families," dans *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 8.

[288] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming.* Springer, 1984, vol. 2.

Annexe A

Matrix theory and Vector Spaces

Vectors which were initially introduced in the field of geometry and physics (mechanics) are the basis to the construction of vector spaces. In this thesis, we use Caley's representation of a vector space denoted as a matrix. A $n \times d$ matrix is a $n - row$ and $d - column$ table containing scalars. In this thesis, matrices in several fields are considered :

— The set of all $n \times d$ real matrices is denoted by $\mathbb{R}^{n \times d}$.

— The set of all $n \times d$ nonnegative real matrices is denoted by $\mathbb{R}_+^{n \times d}$.

— The set of all $n \times d$ binary matrices is denoted by $\{0,1\}^{n \times d}$.

We use bold uppercase letter to denote a matrix (e.g. $\boldsymbol{X}$). The matrix scalars at the (i,j)-th positions are denoted in lower case (e.g. x_{ij}). At each time, the position indexes are given when defining a matrix following these two manners : $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n \times d}$ or $\boldsymbol{X} = (x_{ij})_{n \times d}$ (for an arbitrary matrix) making the sets of row and column indexes intuitive (in this case, $i = 1, \ldots, n$ and $j = 1, \ldots, d$). Therefore the i-th row of a matrix is denoted $\boldsymbol{x}_i$ and the j-th column is given by $\boldsymbol{x}_j$.

A column vector is a matrix with one column (e.g. $(1,2,0)^\top$). A row vector is a matrix with one row (e.g. $(1,2,0)$). Unless explicit statement, a vector is always a column vector. We use bold lower case to denote a vector (e.g. $\boldsymbol{x}$).

— The set of all $n - size$ real vector is denoted by $\mathbb{R}^n$.

— The set of all $n - size$ nonnegative real vector is denoted by $\mathbb{R}_+^n$.

— The set of all $n - size$ binary vector is denoted by $\{0,1\}^n$.

Remark. Let $\boldsymbol{X} = (x_{ij})_{n \times d}$, a vector accessed by a row index in a matrix (e.g. $\boldsymbol{x}_i$) is a row vector. A vector accessed by a column index (e.g. $\boldsymbol{x}_j$) is a column vector.

A matrix is said to be square if the number of rows equals the number or columns. A squared matrix $\boldsymbol{X} = (x_{i,i'})_{n\times n}$ is said to be symmetric if $x_{i,i'} = x_{i',i}$. A diagonal matrix is a square matrix with non-zero elements on the diagonal and zero elsewhere. A matrix $\boldsymbol{X}$ is said to be *positive semidefinite* if it can be obtained by the product of a matrix by its transpose, e.g. :

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{A}^\top. \tag{A.1}$$

This implies that a *positive semidefinite* matrix is always symmetric.

Elementary vector operations such as addition, scalar-vector multiplication, inner product (or dot product), and elementary matrix operations such as transposition $(.)^\top$, addition, scalar-matrix multiplication, matrix-matrix multiplication (dot product), point-wise operation such as point-wise multiplication $\odot$ (Hadamard product), point-wise power (Hadamard power), point-wise division (Hadamard division), are assumed to be part of the reader knowledge.

A.1 Basic linear algebra

This section reviews some basic properties of linear algebra necessary in the understanding of the following sections.

Subspace. A subspace of $\mathbb{R}^n$ is a subset that is also a vector space, similarly for $\mathbb{R}^{n\times d}$.

Linear Independence. Considering the set of vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\} \in \mathbb{R}^n$, the smallest set of all linear combinations of these vectors is a subspace called a span of $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$ denoted as :

$$\text{span}(\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}) = \Big\{ \sum_j^d \alpha_j \boldsymbol{v}_j,\ \alpha_j \in \mathbb{R} \Big\}. \tag{A.2}$$

$\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$ is said to be *linearly independent* if and only if :

$$\sum_j^d \alpha_j \boldsymbol{v}_j = \boldsymbol{0} \iff \alpha_j = 0, \forall j = 1, \ldots, d, \tag{A.3}$$

where $\boldsymbol{0} = (0, \ldots, 0) \in \mathbb{R}^n$. In this case, $\text{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ is a set with an unique linear combination of the vectors $\boldsymbol{v}_j$. Otherwise, if there is a nontrivial combination of the $\boldsymbol{v}_j$ equals to $\boldsymbol{0}$, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$ is said to be *linearly dependent* and $\text{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ has multiple elements.

Basis. A set of *linearly independent* vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$ in $\mathbb{R}^n$ is called a basis if no vector $\boldsymbol{v}_j$ can be removed from the set without changing $\text{span}(\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\})$, e.g. $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\}$ and $\text{span}(\boldsymbol{V}) = \boldsymbol{V}$.

Dimension. Let $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\} \in \mathbb{R}^{n \times d}$ be a vector space, all possible bases for $\boldsymbol{V}$ have the same exact number of vectors and this number is called the *dimension* and denoted $\dim(\boldsymbol{V})$.

Range. Let $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, the range of a matrix denoted $\text{ran}(\boldsymbol{V})$ is the equivalent to the span of the set of column-vectors (or columns space) $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$. It is therefore the set of all possible linear combinations $\boldsymbol{u} \in \mathbb{R}^n$ of the column-vectors such that :

$$\text{ran}(\boldsymbol{V}) = \{\boldsymbol{u} \in \mathbb{R}^n : \boldsymbol{u} = \boldsymbol{V}\boldsymbol{x}, \text{ for some } \boldsymbol{x} \in \mathbb{R}^d\}. \tag{A.4}$$

Rank. Let $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, the rank of a matrix denoted $\text{rank}(\boldsymbol{V})$ is the maximum number of linearly independent columns or rows (the columns rank and the row rank are always equal) and can be stated as follows :

$$\text{rank}(\boldsymbol{V}) = \dim(\text{ran}(\boldsymbol{V})). \tag{A.5}$$

A matrix $\boldsymbol{V}$ is said to be *full rank* if $\text{rank}(\boldsymbol{V}) = \min(n, d)$ and *rank deficient* if $\text{rank}(\boldsymbol{V}) < \min(n, d)$.

Orthogonality. A set of vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\} \in \mathbb{R}^n$ is said to be *orthogonal* if $\forall j \neq j', \langle \boldsymbol{v}_j, \boldsymbol{v}_{j'} \rangle = \boldsymbol{v}_j^\top \boldsymbol{v}_{j'} = 0$ and *orthonormal* if $\langle \boldsymbol{v}_j, \boldsymbol{v}_{j'} \rangle = \delta_{jj'}$ where $\delta_{jj'} = 1$ if $i = i'$ and 0 otherwise.

A squared matrix $\boldsymbol{Q}^{n \times n}$ is said to be *orthogonal* if $\boldsymbol{Q}^\top \boldsymbol{Q} = \mathbf{I}$.

Determinant. The determinant is a scalar value function defined over any square matrix. It allows to characterize the linear map of the matrix. Given a finite dimensional square vector space $\Omega^{n \times n}$, we denote $\det : \Omega^{n \times n} \to \Omega$. It can be computed using the recursive Laplace approximation. Let $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$:

$$\det(\boldsymbol{A}) = \sum_{j}^{d} (-1)^{j+1} a_{ij} \det(\boldsymbol{B}), \tag{A.6}$$

where $\boldsymbol{B} = \boldsymbol{A}/\{\boldsymbol{A}_1, \boldsymbol{A}_j\}$ is the submatrix obtained by removing the first row and the j-th column of $\boldsymbol{A}$. Note that $\det(\boldsymbol{A}) = a$ if $\boldsymbol{A} = (a) \in \mathbb{R}^{1 \times 1}$. If $\det(A) = 0$, $\boldsymbol{A}$ is a singular matrix (or linearly dependent).

A.1.1 Eigenvalues and Eigenvectors

The *eigenvalues* of a data matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ are the zeros for the *characteristic polynomial* :

$$p(x) = \det(\boldsymbol{A} - x\boldsymbol{I}). \tag{A.7}$$

In other words, they are the values x for which the matrix $\boldsymbol{C} = \boldsymbol{A} - x\boldsymbol{I}$ becomes singular. Therefore, every $n \times n$ matrix denotes n eigenvalues. The set of eigenvalues of a matrix $\boldsymbol{A}$ is denoted as follows :

$$\lambda(\boldsymbol{A}) = \{x : \det(A - x\boldsymbol{I}) = 0\}, \tag{A.8}$$

where $\lambda(\boldsymbol{A}) = \{\lambda_n(\boldsymbol{A}), \ldots, \lambda_1(\boldsymbol{A})\}$ is an ordered set ranking the eigenvalues from the largest (denoted by $\lambda_n(\boldsymbol{A})$) to the lowest ($\lambda_1(\boldsymbol{A})$).

If λ is an eigenvalue of $\boldsymbol{A}$ ($\lambda \in \lambda(\boldsymbol{A})$), it exists an *eigenvector* $\boldsymbol{x}$ such as $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. If $\boldsymbol{A}$ has n independant eigenvectors $\boldsymbol{x}_j$ such that $\boldsymbol{A}\boldsymbol{x}_j = \lambda_j\boldsymbol{x}_j$, $\boldsymbol{A}$ is said to be *diagonalizable* such as :

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \tag{A.9}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is the eigenvectors matrix and $\boldsymbol{X}^{-1}$ its inverse. This factorization is also referred to as the *Eigen decomposition*. Note that not all square matrices are diagonalizable but, if $\boldsymbol{A}$ is *positive semidefinite*, its *Eigen decomposition* always exists.

Remark. In data analysis, the eigenvectors are also referred to as the principal axes or principal directions.

A.2 Norms and distances in vector spaces

Definition A.2.1. A distance function d given a random set M is a function $d : M \times M \to \mathbb{R}^+$ such as $\forall x, y, z \in M$:

— d is nonnegative, $d(x, y) \geq 0$ and $d(x, y) = 0 \iff x = y$

— d is symmetric, $d(x, y) = d(y, z)$

— d satisfies the triangular inequality, $d(x, z) \leq d(x, y) + d(y, z)$

In the context of clustering where the data usually lies in a vector space, we will consider the following functionals for measuring the magnitude of a vector in Ω^d or a finite vector space in $\Omega^{n\times d}$.

Definition A.2.2. (Vector and Matrix norms properties). Let Ω^n (or $\Omega^{n\times d}$) be a finite dimensional vector space over an arbitrary field ($\mathbb{R}$, $\mathbb{Q}$ or $\mathbb{C}$), a norm $\|.\| : \Omega^n$ (or $\Omega^{n\times d}$) $\to \mathbb{R}^+$ is a real value functional such that :

— $\|\omega\| \geq 0$, $\forall \omega \in \Omega^n$ (or $\Omega^{n\times d}$) where $\|\omega\| = 0 \iff \omega = 0$,

— $\forall k \in \Omega$, $\|k\omega\| = |k|\|\omega\|$, $\forall \omega \in \Omega^n$ (or $\Omega^{n\times d}$),

— $\|\omega + \psi\| \leq \|\omega\| + \|\psi\|$, $\forall \omega, \psi \in \Omega^n$ (or $\Omega^{n \times d}$).

A vector space Ω on which a norm is defined is called a normed vector space. A normed vector space associates a metric d such as $d(\omega, \psi) = \|\omega - \psi\|$, $\forall \omega, \psi \in \Omega^n$ (or $\Omega^{n \times d}$).

Remark. If d is complete, then Ω is a *Banach* space.

A.2.1 Vector norms characteristics

The *Euclidean* vector space ($\Omega = \mathbb{R}$) is one of the most fundamental space of geometry. Let $\boldsymbol{x} \in \mathbb{R}^n$, its norm is referred to as the *Euclidean* norm and given as : $\|\boldsymbol{x}\| = \sqrt{\sum_i^n x_i^2}$. In data analysis, ones usually refers to the *Euclidean* norm, however, in some situations, it might not be relevant and others may be used. The p-norm (or $\mathcal{L}_p$-norm) for vectors is an example which generalizes several popular alternatives. It is defined as :

$$\|\boldsymbol{x}\|_p = \Big(\sum_i^n |x_i|^p \Big)^{\frac{1}{p}}, \quad p \in \mathbb{N}^*, \tag{A.10}$$

where the most common values for p are $\{1, 2, \infty\}$. The 2-norm is the *Euclidean* norm. A unit-vector $\boldsymbol{x}$ w.r.t the norm $\|.\|$ is a vector that satisfies $\|\boldsymbol{x}\| = 1$.

An analogical metric to the p-norm in the Euclidean vector space is the *Minkowski* distance of order p denoted as follows :

$$D(\boldsymbol{x}, \boldsymbol{y}) = \Big(\sum_i^n |x_i - y_i|^p \Big)^{\frac{1}{p}}, \quad p \in \mathbb{N}^*, \boldsymbol{y} \in \mathbb{R}^n, \tag{A.11}$$

for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$. For $p = 1$, we denote the *Manhattan* distance, for $p = 2$, the *Euclidean* distance and for $p \to \infty$, the *Chebyshev* distance. As for the p-norm.

A.2.2 Matrix norms characteristics

Let $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, the most commonly used matrix norms is the *Frobenius* norm defined as : $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j}^{n,d} a_{ij}^2}$ and the p-norms for matrix :

$$\|\boldsymbol{A}\|_p = \sup_{\boldsymbol{x} \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}, \quad \boldsymbol{x} \in \mathbb{R}^n. \tag{A.12}$$

Note that the matrix p-norms are defined in terms of vector p-norm. For more insights, the readers can refer to the books of Horn and Johnson [277] or Golub and Van Loan [123].

A.3 Other distances and dissimilarities

Other distances such as the *Manhalobis* distance are also widely used for cluster analysis. In the context of text analysis where the data matrix might be binarized, one may consider the *Jaccard* distance (firstly denoted as "coefficient de communauté" [278]) suitable for discontinuous variables defined as follows :

$$D_J(\boldsymbol{x}, \boldsymbol{y}) = 1 - J(\boldsymbol{x}, \boldsymbol{y}), \tag{A.13}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are two finite binary sample sets, $J(\boldsymbol{x}, \boldsymbol{y}) = \frac{|\boldsymbol{x} \cap \boldsymbol{y}|}{|\boldsymbol{x} \cup \boldsymbol{y}|}$ is the *Jaccard* index measures the similarity between $\boldsymbol{x}$ and $\boldsymbol{y}$. By definition, we have $0 \leq J(\boldsymbol{x}, \boldsymbol{y}) \leq 1$. For text analysis with nonnegative data or more generally application with count data, one may also consider the chi-squared (χ^2) distance ([43]). Another relevant measuring function which gained a substantial popularity in text analysis is the $(1 - \cos)$ dissimilarity (also referred to as the "Cosine distance") defined as follows :

$$CD(\boldsymbol{x}, \boldsymbol{y}) = 1 - \cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}. \tag{A.14}$$

Remark. The $(1 - \cos)$ dissimilarity is not a proper distance since it does not have the triangular inequality property. Usually, $\boldsymbol{x}$ and $\boldsymbol{y}$ are unit-vectors and the distance results in computing the inner product.

Annexe B

Distributions for Mixture Models

In this section, we review the set of several parametric exponential families/distributions (also referred to as the set of *Koopman–Darmois family*) of interest for the model-based approach introduced in the following chapters and the underlying relations highlighted with the class of *Bregman* divergences used in NMF.

B.1 Exponential Families

Exponential Families form the basis for generalized linear model (GLM). Such distributions are also key in Bayesian Inference since they provide conjugate priors. (Note : to not be confused, an exponential family is an exponential distribution varying according according to a parameter, e.g. the Normal distribution is an exponential family). We denote several sub class family depending on the number of parameter : (i) single (one scalar parameter), (ii) or multiple (vector of parameters) ; and the random variable (scalar of vector).

B.1.1 Single-parameter exponential families

Let x be a random variable in $\mathbb{R}$, a single-parameter (also referred to as one-parameter) exponential family is a set of probability distributions for which the probability density function (for continuous support) or probability mass function (for discrete support) can be written in the following form :

$$p(x|\theta) = h(x) \exp\left(\eta(\theta)S(x) - B(\theta)\right), \tag{B.1}$$

where h is nonnegative and η, B are real value functions. $B(\theta)$ act as a normalization function for $p(x)$ defined once h, η have been set. If $\eta(\theta) = \theta$, the exponential family is said to be in its canonical form. By defining a transformer s.t. $\eta = \eta(\theta)$, we denote the following probability function :

$$q(x|\eta) = h(x) \exp\big(\eta S(x) - A(\eta)\big), \tag{B.2}$$

where $A(\eta)$ is the finite cumulant function acting as a normalization for q(x) s.t. $A(\eta) = \ln \int h(x) exp(\eta S(x)) dx$ and $S(x)$ is a sufficient statistic. In this case, η is referred to as the *natural parameter* whereas $\eta(\theta)$ is referred to as the *link function*. Let Ω be the set of all η s.t. $A(\eta)$ is finite, Ω is referred to as the *space of natural parameter*. Ω is a convex and $A(\eta)$ are convex functions.

If $\eta(\theta) = \theta$ and $S(x) = x$, the exponential family is a special case referred to as *natural exponential family* (NEF) s.t.

$$p(x|\theta) = h(x) \exp\big(\theta x - B(\theta)\big). \tag{B.3}$$

Examples of single-parameter discrete exponential families are the Bernoulli distribution, the Poisson distribution. Example of continuous family is the Exponential distribution. See details in Table B.1.

TABLE B.1 – Single-parameter exponential families.

Family	$B(\theta)$	$n(\theta)$	$A(\eta)$	$S(x)$
Bernoulli	$\ln(1-\theta)$	$\ln\big(\frac{\theta}{(1-\theta)}\big)$	$-\ln(1+exp(\eta))$	x
Poisson	$-\theta$	$\ln(\eta)$	exp(n)	x
Exponential	$\log\theta$	θ	$-\ln(-\eta)$	x

B.1.2 Vector-parameter exponential families

Vector-parameter exponential families (also denoted as k-parameter families) are indexed by several parameter forming the vector $\boldsymbol{\theta}$ s.t.

$$p(x|\boldsymbol{\theta}) = h(x) \exp\left(\sum_{r}^{s} \eta_r(\boldsymbol{\theta}) S_r(x) - B(\boldsymbol{\theta})\right), \tag{B.4}$$

for a random variable $x \in \boldsymbol{x}$, where $\boldsymbol{\eta}(\boldsymbol{\theta}) = [\eta_1(\boldsymbol{\theta}), \ldots, \eta_r(\boldsymbol{\theta})]^\top$ and $\boldsymbol{S}(x) = [S_1(x), \ldots, S_r(x)]^\top$. The canonical form when $\eta_r(\boldsymbol{\theta}) = \theta_r$ gives $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_r]^\top$ s.t. :

$$q(x|\boldsymbol{\eta}) = h(x) \exp\big(\boldsymbol{S}(x)^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\big). \tag{B.5}$$

A vector exponential family is said to be curved when the dimension of the set of parameter $\boldsymbol{\theta}$ is inferior to the one of $\boldsymbol{\eta}(\boldsymbol{\theta})$.

An example of continuous vector exponential family is the Gaussian distribution where $\boldsymbol{\theta} = \{\mu, \sigma^2\}$. More examples are given in Table B.2.

TABLE B.2 – Vector-parameter/vector-variable exponential families.

Family	$B(\boldsymbol{\theta})$	$\boldsymbol{\eta}(\boldsymbol{\theta})$	$A(\boldsymbol{\eta})$	$S(x)$
Gaussian	$-\frac{1}{2}\Big(\frac{\mu^2}{\sigma_2} + \ln(2\pi\sigma^2)\Big)$	$\{\frac{\mu}{\sigma_2}, -\frac{1}{2\sigma^2}\}$	$-\frac{1}{2}\Big(\frac{\eta_1^2}{2\eta_2} + \ln\big(\frac{\pi}{\eta^2}\big)\Big)$	$\{x, x^2\}$
Gamma	$-\ln\frac{\beta^\alpha}{\Gamma(\alpha)}$	$\{(\alpha-1), -\beta\}$	$\log\frac{\Gamma(\eta_1+1)}{\log(-\eta_2)^{(\eta+1)}}$	$\{\ln x, x\}$

B.1.3 Vector-parameter and vector-variable exponential family

The vector-parameter exponential family for one scalar variable x can easily be extended to a vector random variable $\boldsymbol{x} \in \mathbb{R}^d$ such that :

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x}) exp\Big(\sum_r^s \eta_r(\boldsymbol{\theta}) S_r(\boldsymbol{x}) - B(\boldsymbol{\theta})\Big), \tag{B.6}$$

where $\boldsymbol{S}(\boldsymbol{x}) = [S_1(\boldsymbol{x}), \ldots, S_r(\boldsymbol{x})]^\top$ with $S_r(\boldsymbol{x}) \in \mathbb{R}^d$, $\forall t = 1, \ldots, s$.

An example of a discrete vector-parameter family over a vector random variable is the Multinomial distribution with parameters $(\pi_1, \ldots, \pi_d)$. A continuous one would be the von Mises-Fisher. See details in Table B.3.

TABLE B.3 – Vector-parameter/vector-variable exponential families.

Family	$B(\boldsymbol{\theta})$	$\eta(\boldsymbol{\theta})$	$A(\boldsymbol{\eta})$	$S(\boldsymbol{x})$
Multinomial	$\ln\big(1 - \sum_{j'}^{d-1} \pi_{j'}\big)$	$-\ln\Big(\frac{\theta_j}{1-\sum_{j'}^{d-1}\theta_{j'}}\Big)$	$\ln\big(1 + \sum_{j'}^{d-1} \exp(\eta_{j'})\big)$	$x_1, \ldots, x_d$
von Mises-Fisher	0	$\big\{\{\mu_1, \ldots, \mu_d\}, \kappa\big\}$	0	$x_1, \ldots, x_d$

More generally, the set of exponential families includes these most common distributions :

Discrete	Continuous
Bernoulli, Poisson, categorical, geometric	normal, log-normal, inverse-gaussian, exponential, gamma, chi-squared, beta, Dirichlet, von Mises, von Mises-Fisher, Wishart, inverse-Wishart

Note some families are exponential families under certain conditions. For instance if their parameters are fixed (e.g. fixed numbers of trials for the Multinomial and Binomial families).

B.1.4 Conjugate priors

As mentioned previously, a key property of exponential families is that they denote a natural conjugate prior family. Let $p(x|\boldsymbol{\theta})$ be the probability distribution of a particular exponential family and $p(\boldsymbol{\theta})$ a prior family, the posterior distribution results in a functional with the same form as the prior family. Considering a k-parameter family with pdf or pmf $p(x|\boldsymbol{\theta})$, the conjugate distribution on $\boldsymbol{\theta}$ is given by the exponential family of $k+1$ parameters such that :

$$p(\boldsymbol{\theta}) = \exp\Big(\sum_{r}^{s} \eta_r(\boldsymbol{\theta})\lambda_r - \lambda_{s+1}B(\boldsymbol{\theta}) - f(\boldsymbol{\lambda})\Big), \tag{B.7}$$

where $\boldsymbol{\lambda}$ is the natural parameter vector of size $k+1$ and f a real value function. The sufficient Statistic is given by the set $\{\eta_r(\boldsymbol{\theta}), -B(\boldsymbol{\theta})\}$. The canonical form making appearance for the (k+1)th parameter ν is given by :

$$q(\boldsymbol{\lambda}, \nu) = \exp\Big(\sum_{r}^{s} \eta_r\lambda_r - \nu A(\boldsymbol{\eta}) - f(\boldsymbol{\lambda}, \nu)\Big). \tag{B.8}$$

The Gamma prior

The Gamma family is conjugate for the Poisson. Its pdf and log-pdf (considering the shape and rate parameter parameterization) are given as below :

$$p(\theta|\alpha, \beta) = \frac{\beta^\alpha \theta^{\alpha-1} e^{-\beta\theta}}{\Gamma(\alpha)}, \tag{B.9}$$

and

$$\log p(\theta|\alpha, \beta) = \log\frac{\beta^\alpha}{\Gamma(\alpha)} + (\alpha-1)\log\theta - \beta\log(\exp(\theta)). \tag{B.10}$$

The parameters are then expressed as follows : $\{(\lambda-1), \beta\}$, $f(\lambda, \nu) = -\log\frac{\beta^\alpha}{\Gamma(\alpha)}$, $\eta(\theta) = \log\theta$, $B(\theta) = \log(\exp(\theta))$ and $A(\eta) = \exp(\eta)$.

Example with the Conjugate Gamma-Poisson Model

Let $\boldsymbol{x} = (x_1, \ldots, x_d)^\top \in \mathbb{R}_+^d$, the Gamma-Poisson model where $x_i \overset{iid}{\sim} \mathcal{P}(\lambda)$ is described as follows :

$$\boldsymbol{x} \sim \prod_i^d \mathcal{P}(x_i; \lambda) = \prod_i^d \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}, \qquad \lambda \sim \mathcal{G}(\lambda; \alpha, \beta) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)},$$

where $\mathcal{P}(x_i; \lambda)$ is the Poisson distribution with parameter λ and $\mathcal{G}(\lambda; \alpha, \beta)$ the Gamma distribution with parameter α and β. Computing the joint probability gives :

$$p(\boldsymbol{x}|\lambda)p(\lambda) = \prod_i^n \frac{\lambda^{x_i} e^{-\lambda} \beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{x_i!\Gamma(\alpha)} = \lambda^{\sum_i^n x_i + \alpha - 1} e^{-\lambda(n+\beta)} \beta^\alpha \prod_i^n \frac{1}{x_i!}, \tag{B.11}$$

TABLE B.4 – EDM equivalent distributions. (1) Normal (2), Gamma, (3) Inverse Gaussian, (4) Poisson, (5) Binomial, (6) Negative Binomial.

	Distribution	θ	λ	$\kappa(\theta)$	$a(\lambda, x)$	$V(\mu)$	Ω
1.	$\mathcal{N}(\mu, \phi)$	$\frac{\mu}{\phi}$	ϕ	$\frac{1}{2}\theta^2$	$\left(\frac{1}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{x^2}{2\lambda}}$	μ^0	$(-\infty, \infty)$
2.	$\mathcal{G}(\mu, \phi)$	$-\frac{1}{\phi}$	μ	$-\log(-\theta)$	$\Gamma(\lambda)^{-1} x^{\lambda-1}$	μ^2	$(0, \infty)$
3.	$\mathcal{IG}(\mu, \phi)$	$\frac{-\phi}{2\mu^2}$	$\sqrt{\phi}$	$-\sqrt{-2\theta}$	$\frac{\lambda}{\sqrt{2\pi x^3}} e^{-\frac{\lambda^2}{2x}}$	μ^3	$(0, \infty)$
4.	$\mathcal{P}(\lambda\mu)$	$\log \mu$	1	e^θ	$\frac{\lambda}{x!}$	μ^1	$(0, \infty)$
5.	$\mathcal{B}(\lambda, \mu)$	$\log \frac{\mu}{1-\mu}$	λ	$\log(1 + e^\theta)$	$\binom{\lambda}{x}$	$\mu(1-\mu)$	$(0, 1)$
6.	$\mathcal{NB}(\lambda, \mu)$	$\log \mu$	λ	$-\log(1 - e^\theta)$	$\binom{\lambda+x-1}{x}$	$\mu(1+\mu)$	$(0, \infty)$

By making use of the conjugacy, and reintegrating the missing normalization, we denote the posterior Gamma distribution $\mathcal{G}(\lambda; \sum_i^n x_i + \alpha, n + \beta) = \frac{(n+\beta)^{\sum_i^n x_i+\alpha} \lambda^{\sum_i^n x_i+\alpha-1} e^{-(n+\beta)\lambda}}{\Gamma\left(\sum_i^n x_i+\alpha\right)}$.

Another form a generalization for a subset of exponential family is known as the Exponential Dispersion Model introduced by Jorgensen [279, 280]. An exponential dispersion model can be defined by the following probability density function :

$$f(x; \theta, \lambda) = a(\lambda, x) e^{\theta x - \lambda\kappa(\theta)}, \tag{B.12}$$

where $(\theta, \lambda) \in (\mathbb{R}, \mathbb{R}^+)$. This expression originally highlights exponential dispersion model for discrete data. Nevertheless, both continuous and discrete exponential dispersion models can be reviewed through this unique expression [279, 281]. We denote $X \sim ED(\theta, \lambda)$ with expectation $E(X) = \lambda\mu$ and variance $V(X) = \lambda V(\mu)$ where $\mu = \kappa'(\theta)$ and the variance function $V(\mu) = \kappa''(\theta)$ for κ being the cumulant generative function, θ the natural parameter, and λ the dispersion parameter. A couple of prior studies of Tweedie [282], Bar-Lev and Enis [283] and Hougaard [284, 285] have focused on the link between exponential dispersion models and power variance functions (since Tweedie models are EDMs), denoting $V(\mu) = \mu^p$ for an initial $p > 0$. Jorgensen [279] showed later that EDMs exist for all $p \notin]0, 1[$. Table B.4 shows the different forms of the suitable function $a(\lambda, \theta)$ and other parameters allowing the correspondence between 6 common continuous and discrete families.

In the following chapters, we will focus mainly on the Normal, Gamma, Erlang, Dirichlet, von Mises-Fisher as a set of continuous distributions and, the Poisson, Binomial and Negative Binomial

as a set of discrete distributions.

B.2 Exponential Families and Bregman divergences

The *Bregman* divergence is a generalizing measure of the distance between two points defined in terms of a strictly convex function. It is nonnegative and equals zero when both arguments are equal.

Definition B.2.1. (**Bregman divergence** [243]). Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, and $\phi : \mathcal{S} \to \mathbb{R}$, $\mathcal{S} = \mathrm{dom}(\phi)$ be a strictly convex function defined on a convex set $\mathcal{S} \subset \mathbb{R}^d$ such that ϕ is differential on $\mathrm{ri}(\mathcal{S})$, the Bregman divergence denoted $d_\phi : \mathcal{S} \times \mathrm{ri}(\mathcal{S}) \to [0, +\infty)$ is given as follows :

$$d_\phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla_{\boldsymbol{y}} \phi \rangle. \tag{B.13}$$

Several studies have emphasized the close relation between the Bregman divergence and Exponential Families. More precisely, Forster and Warmuth [286] pointed that the negative log-likelihood $f(\boldsymbol{x}, \boldsymbol{\theta})$ of several exponential families could be express as the sum of a negative uniquely determined Bregman divergence and a normalizing function independent of the parameters of the distribution such as :

$$\log f(\boldsymbol{x}, \boldsymbol{\theta}) = -d_\phi(\boldsymbol{x}, \mu(\boldsymbol{\theta})) + \log(b_\phi(\boldsymbol{x})), \tag{B.14}$$

where $\mu = \mu(\boldsymbol{\theta})$ is the expectation parameter corresponding to $\boldsymbol{\theta}$ and $b_\phi(\boldsymbol{x})$ a real value function. Later, Banerjee et al showed in [41, 287] that the results holds for all instance $\boldsymbol{x}$ regarding to any Exponential Family by establishing a bijection between regular Exponential families and *regular Bregman divergence* using the *Legendre* duality.

Annexe C

Optimization

In this thesis, NMF will be essentially formulated as a constrained nonlinear programming problem. In this section, we will first review the definition of unconstrained nonlinear problems and the conditions that must hold at a solution point of this problem. Then, we will extend the theory to problems under inequality or mixed constraints (inequality and equality).

C.1 Convex set and convex function

Definition C.1.1. (Convex sets). A set Ω is said to be convex if $\forall x, y \in \Omega$ and $\forall \alpha \in [0, 1]$, we have :

$$\alpha x + (1 - \alpha)y \in \Omega. \tag{C.1}$$

Geometrically, this definition states that given any two points in Ω, if every point on the line segment joining those two points is in Ω, then Ω is a convex set.

Definition C.1.2. (Convex function). A function f defined on a convex set Ω is said to be convex if $\forall x, y \in \Omega$ and $\forall \alpha \in [0, 1]$

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \tag{C.2}$$

f is said to be strictly convex if $f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y)$.

Theorem C.1.1. *(Duality theorem)*

C.2 Unconstrained Optimization

Let $\mathbb{R}^n$ be the *Euclidean* space of n-dimensional vectors. Considering the optimization problem of the form :

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}) \\ &\text{subject to } \ \boldsymbol{x} \in \Omega \end{aligned} \tag{C.3}$$

where f is a real-valued function and Ω the feasible set of solutions, subset of $\mathbb{R}^n$ ($\Omega \subset \mathbb{R}^n$). From the theorem of *Weierstras*, a solution exists if f is continuous and Ω is compact (or closed), however, several kinds of solution points arise : local minima and global minima.

Definition C.2.1. (Local minimum). A point $\boldsymbol{x}* \in \Omega$ is said to be a local minimum point of f over Ω if there is a neighborhood $N_\epsilon(\boldsymbol{x}^*)$ of x^* with magnitude $\boldsymbol{\epsilon}$ such that $|\boldsymbol{x} - \boldsymbol{x}^*| < \epsilon$, and $\forall \boldsymbol{x} \in N_\epsilon(\boldsymbol{x}^*) \cap \Omega$, $f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$. If $f(\boldsymbol{x}) > f(\boldsymbol{x}^*), \forall \boldsymbol{x} \in N_\epsilon(\boldsymbol{x}^*) \cap \Omega$ and $\boldsymbol{x} \neq \boldsymbol{x}^*$, $\boldsymbol{x}^*$ is said to be a strict local minimum.

Definition C.2.2. (Global minimum). A point $\boldsymbol{x}* \in \Omega$ is said to be a global minimum point of f over Ω if $\forall \boldsymbol{x} \in \Omega$, $f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$. If $f(\boldsymbol{x}) > f(\boldsymbol{x}^*), \forall \boldsymbol{x} \in \Omega$ and $\boldsymbol{x} \neq \boldsymbol{x}^*$, $\boldsymbol{x}^*$ is said to be a strict global minimum.

Solving problem (C.3) leads to a global minimum expectation. However unless f has some convexity properties which guarantee that any local minimum is a global minimum, in practice, this expectation is rarely achievable. Consequently, solving problem (C.3) usually refer to obtain a local minimum point which may be global if some further appropriates conditions holds.

We distinguish two conditions which must hold at a local solution point $\boldsymbol{x}^*$ called the first- and second-order conditions. These are simple extensions to $\mathbb{R}^n$ to the derivative conditions that holds at a minimum or maximum point x^* for a function of one variable in $\mathbb{R}$. The idea is to consider movement away from the point in a feasible direction $\boldsymbol{d}$.

Definition C.2.3. (Feasible direction). Given $\boldsymbol{x} \in \Omega$, a vector $\boldsymbol{d} \in \mathbb{R}^n$ is a feasible direction at $\boldsymbol{x}$ if there is an $\bar{\alpha} > 0$ such that $\boldsymbol{x} + \alpha \boldsymbol{d} \in \Omega$, $\forall \alpha \in [0, \bar{\alpha}]$. From this definition, we define the first-order conditions for a local minimum.

Proposition C.2.1. (First-order necessary conditions). Let $\Omega \subset \mathbb{R}^n$ and f be a continuously differentiable function on Ω. If $\boldsymbol{x}^*$ is a local minimum of f over Ω, for every feasible direction $\boldsymbol{d} \in \mathbb{R}^n$ at $\boldsymbol{x}^*$:

$$\nabla f(\boldsymbol{x}^*)^\top \boldsymbol{d} \geq 0. \tag{C.4}$$

Proof. The proof using the first order approximation of f in the neighbourhood of the local minimum point is given in [288] (chapter 7, section 7.1). □

Note that, when $\boldsymbol{x}$ is an interior point of Ω ($\boldsymbol{x} \in \text{int}(\Omega)$), every direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$ is a feasible direction. Therefore, $\nabla f(\boldsymbol{x}^*)^\top \boldsymbol{d} \geq 0$, $\forall \boldsymbol{d} \in \mathbb{R}^n$ which implies that $\nabla f(\boldsymbol{x}^*) = 0$. The necessary conditions lead to n equations (one for each derivative in $\nabla f(\boldsymbol{x})$) with n unknowns (one for each value in $\boldsymbol{X}^*$). Note that the optimization problem is solved directly without attempting to solve the equation arising from the first-order necessary conditions. Moreover, any local minimum point $\boldsymbol{x}^*$ that satisfies eq(C.4) is called a *stationary point.*

The second-order conditions is defined in terms of the Hessian matrix ($n \times n$ matrix of second partial derivatives of f) given by $\nabla^2 f(\boldsymbol{x}^*)$.

Proposition C.2.2. (Second-order necessary conditions). Let $\Omega \subset \mathbb{R}^n$ and f be a twice continuously differentiable function on Ω. If $\boldsymbol{x}^*$ is a local minimum of f over Ω, for every feasible direction $\boldsymbol{d} \in \mathbb{R}^n$ at $\boldsymbol{x}^*$:

$$1)\ \nabla f(\boldsymbol{x}^*)^\top \boldsymbol{d} \geq 0, \tag{C.5}$$

$$2)\ \text{if } \nabla f(\boldsymbol{x}^*)^\top \boldsymbol{d} = 0,\ \text{then } \boldsymbol{d}\nabla^2 f(\boldsymbol{x}^*)^\top \boldsymbol{d} \geq 0. \tag{C.6}$$

Proof. 1) is just proposition C.2.1. For 2), the proof using the second order approximation of f in the neighbourhood of the local minimum point is given in [288] (chapter 7, section 7.3). □

Similarly as for the first-order conditions, when $\boldsymbol{x}$ is an interior point of Ω ($\boldsymbol{x} \in \int(\Omega)$), every direction $\boldsymbol{d}$ at $\boldsymbol{x}^*$ is a feasible direction. Therefore, $\boldsymbol{d}\nabla^2 f(\boldsymbol{x}^*)^\top \boldsymbol{d} \geq 0$, $\forall \boldsymbol{d} \in \mathbb{R}^n$ and $\nabla f(\boldsymbol{x}^*) = 0$. This implies therefore than the Hessian is positive semidefinite. Note that this matrix plays a key role the convergence analysis of iterative algorithms for solving unconstrained problems and our following NMF constrained minimization problems that will defined in the incoming chapters (More specifically for the convergence of SNMF). By strengthening those propositions, a sufficient condition for $\boldsymbol{x}^*$ to be a local minimum in the unconstrained case can also be derived. For more results, refer to [288] (Chapter 7, section 7.3). In the following, we denote the Hessian matrix by $\boldsymbol{F}(\boldsymbol{x}^*)$.

C.3 Optimization with equality constraint

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a n-dimensional vector of unknowns, the general mathematical nonlinear programming constrained problem can be stated as :

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}) \\ &\text{subject to } h_i(\boldsymbol{x}) = 0,\ i = 1, \ldots, m \iff \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0} \\ &\boldsymbol{x} \in \Omega \in \mathbb{R}^n \end{aligned}$$

where f is twice continuously differentiable, $\boldsymbol{h} = [h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})]$ referred to the functional constraints on $\boldsymbol{x}$ also continuously twice differentiable, and $\boldsymbol{X} \in \Omega$ is a set constraint. The first-order necessary conditions requires that the local minima point is a regular point whose definition is given subsequently.

Definition C.3.1. (Regular point). A point $\boldsymbol{x}^*$ which satisfies the constraints $\boldsymbol{h}(\boldsymbol{x}^*) = \mathbf{0}$ is said to be a *regular point* of the constraints ($\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$) if the gradient vectors $\boldsymbol{\nabla} h_1(\boldsymbol{x}^*), \ldots, \boldsymbol{\nabla} h_m(\boldsymbol{x}^*)$ are linearly independent.

Lemma C.3.1. *(First-order necessary conditions for equality constraints). If $\boldsymbol{x}^*$ is a regular point of $\boldsymbol{h}(\boldsymbol{x})$ and a local extremum point (minimum or maximum) of f subject to $\boldsymbol{h}(\boldsymbol{x}) = 0$, then all vectors $\boldsymbol{y} \in \mathbb{R}^n$ which satisfy $\boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top \boldsymbol{y} = \mathbf{0}$ must also satisfy $\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top \boldsymbol{y} = 0$.*

Proof. The proof is given in chapter 7 of [288] □

This Lemma naturally implies that $f(\boldsymbol{x}^*)$ is a linear combination of $\boldsymbol{\nabla h}(\boldsymbol{x}^*)$. This relation subsequently introduced the relation with the Lagrange multiplier $\boldsymbol{\lambda}$.

Theorem C.3.2. *(Lagrange multiplier duality). Let $\boldsymbol{x}^*$ be a local extremum point (minimum or maximum) of f subject to $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ and also a regular point of theses constraints, there exists a $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that :*

$$\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda} \boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top = \mathbf{0}. \tag{C.7}$$

Proof. From lemma C.7, we have that maximizing $\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top \boldsymbol{y}$ subject to $\boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top \boldsymbol{y} = \mathbf{0}$ results in $\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top \boldsymbol{y} = \mathbf{0}$, by duality of linear optimization, $\exists \boldsymbol{\lambda} \in \mathbb{R}^n$ such that : $\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda} \boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top = \mathbf{0}$. □

The Lagrangian associated with the constrained problem used commonly for solving non linear constrained optimization problem is defined as follows :

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{x})^\top \tag{C.8}$$

Supposing that f and $\boldsymbol{h}$ are twice continuously differentiable, the second-order conditions is partially expressed according to the tangent plane defined by the constraints $\boldsymbol{h}(\boldsymbol{x}) = 0$ at a regular point $\boldsymbol{x}^*$. Let M be the tangent plane defined as $M = \{\boldsymbol{y} : \boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top \boldsymbol{y} = \boldsymbol{0}\}$ at a regular point $\boldsymbol{x}^*$

Proposition C.3.1. (Second-order necessary conditions). If $\boldsymbol{x}^*$ is a local minimum of f subject to $\boldsymbol{h}(\boldsymbol{x}^*) = \boldsymbol{0}$ and a regular of point of these constraints, $\exists \boldsymbol{\lambda} \in \mathbb{R}^m$ such that :

$$f(\boldsymbol{x}^*) + \boldsymbol{\lambda}\boldsymbol{\nabla h}(\boldsymbol{x}^*)^\top = \boldsymbol{0} \tag{C.9}$$

$$\boldsymbol{L}(\boldsymbol{x}^*) = \boldsymbol{F}(\boldsymbol{x}^*) + \boldsymbol{\lambda}\boldsymbol{H}(\boldsymbol{x}^*)^\top, \tag{C.10}$$

where $\boldsymbol{H}(\boldsymbol{x}^*)^\top = \boldsymbol{\nabla h}^2(\boldsymbol{x}^*)$ and $\boldsymbol{L}(\boldsymbol{x}^*)$ is semidefinite on M such as $\boldsymbol{y}\boldsymbol{L}(\boldsymbol{x}^*)^\top \boldsymbol{y} \geq \boldsymbol{0}, \forall \boldsymbol{y} \in M$.

Proof. □

$\boldsymbol{L}$ is the matrix of second partial derivatives w.r.t $\boldsymbol{x}$ from the Lagrangian.

C.4 Optimization with inequality constraints

In the following chapters, we present several new NMF problems which arise equality and inequality constraints formulated in the form of functional. In this section we review the theory behind their optimization and present some results which also benefits to the analysis of their convergences. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a n-dimensional vector of unknowns, the general mathematical nonlinear programming constrained problem can be stated as :

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}) \\ &\text{subject to } h_i(\boldsymbol{x}) = 0,\ i = 1, \ldots, m \iff \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \\ &\qquad\qquad g_j(\boldsymbol{x}) \leq 0,\ j = 1, \ldots, p \iff \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \\ &\qquad\qquad \boldsymbol{x} \in \Omega \in \mathbb{R}^n \end{aligned}$$

where f, $\boldsymbol{h} = [h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})]$, $\boldsymbol{g} = [g_1(\boldsymbol{x}), \ldots, g_p(\boldsymbol{x})]$ are real vector-value functional on $\boldsymbol{x}$. An important concept that simplify the theory is of an *active* constraint, moreover, as it allows the generalization of mixed constrained optimization. At a feasible point $\boldsymbol{x}$, an inequality constraint $g_i(\boldsymbol{x}) \leq 0$ is said to be *active* if $g_i(\boldsymbol{x}) = 0$ and *inactive* if $g_i(\boldsymbol{x}) < 0$. By convention, an equality constraint $h_i(\boldsymbol{x}) = 0$ is *active* at any feasible point. Active constraints at a feasible point $\boldsymbol{x}$ restricts the neighborhood of $\boldsymbol{x}$ while the inactive constraints have no influence in it. Therefore, the conditions of a local minimum are studied only for *active* constraints. Consequently, by generalization of the previous definition (C.3.1), we are looking for a regular point $\boldsymbol{x}^*$ of the constraints $\boldsymbol{h}(\boldsymbol{x}) = 0$ and $\boldsymbol{g}(\boldsymbol{x}) = 0$ such that the gradient vector $\boldsymbol{\nabla} h_1(\boldsymbol{x}^*), \ldots, \boldsymbol{\nabla} h_m(\boldsymbol{x}^*)$ and $\boldsymbol{\nabla} g_1(\boldsymbol{x}^*), \ldots, \boldsymbol{\nabla} g_p(\boldsymbol{x}^*)$ are linearly independent respectively. In these case, the first-order necessary conditions are stated as follows :

Proposition C.4.1. (**Karush-Kuhn-Tucker (KTT) conditions** for inequality constraints). If $\boldsymbol{x}^*$ is a local minimum of f and supposedly a regular point of the constraints $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$. Then, there exists a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ and a vector $\boldsymbol{\mu} \in \mathbb{R}^p_+$ such that :

$$\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda} \boldsymbol{\nabla} \boldsymbol{h}(\boldsymbol{x}^*)^\top + \boldsymbol{\mu} \boldsymbol{\nabla} \boldsymbol{g}(\boldsymbol{x}^*)^\top = \boldsymbol{0} \tag{C.11}$$

$$\boldsymbol{g}(\boldsymbol{x}^*)^\top \boldsymbol{\mu} = 0 \tag{C.12}$$

The Lagrangian associated with the constrained problem whose expression is used in the following chapters for solving non linear constrained optimization problem is defined as follows :

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda} \boldsymbol{h}(\boldsymbol{x})^\top + \boldsymbol{\mu} \boldsymbol{g}(\boldsymbol{x})^\top \tag{C.13}$$

Assuming that f, $\boldsymbol{h}$ and $\boldsymbol{g}$ are twice continuously differentiable, the second-order conditions are defined afterwards.

Proposition C.4.2. (**Second-order necessary conditions** for inequality constraints). If $\boldsymbol{x}^*$ is a local minimum of f and supposedly a regular point of the constraints $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$. Then, there exists a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ and a vector $\boldsymbol{\mu} \in \mathbb{R}^p_+$ such that :

$$\boldsymbol{\nabla} f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda} \boldsymbol{\nabla} \boldsymbol{h}(\boldsymbol{x}^*)^\top + \boldsymbol{\mu} \boldsymbol{\nabla} \boldsymbol{g}(\boldsymbol{x}^*)^\top = \boldsymbol{0} \tag{C.14}$$

$$\boldsymbol{g}(\boldsymbol{x}^*)^\top \boldsymbol{\mu} = 0 \tag{C.15}$$

$$\boldsymbol{L}(\boldsymbol{x}^*) = \boldsymbol{F}(\boldsymbol{x}^*) + \boldsymbol{\lambda} \boldsymbol{H}(\boldsymbol{x}^*)^\top + \boldsymbol{\mu} \boldsymbol{G}(\boldsymbol{x}^*)^\top, \tag{C.16}$$

where $\boldsymbol{L}(\boldsymbol{x}^*)$ is semidefinite on the tangent subspace of the active constraints at $\boldsymbol{x}^*$.

The proof (available in [288]), chapter 7) states an interesting fact that is $\boldsymbol{x}^*$ is also a minimum point for the subset that defines the active constraint $g_j(\boldsymbol{x}^*) \geq 0$ to zero. Therefore the first KTT condition holds if $\mu_j = 0$ and $g_j(\boldsymbol{x}^*) \neq 0$. This implication lead the so-called stationary equations which are key in the derivation of the Multiplicative Update algorithm for NMF.

Annexe D

An unified framework for Nonnegative Matrix Factorization and Finite Mixture Models in the unit-sphere

D.1 NMF to FMM transition examples with other divergences

D.1.1 From NMF to Gaussian mixtures (Euclidean distance)

Lemma D.1.1. *Let* $\sum_k^g \tilde{z}_{ik} = 1$, $\forall i = 1, \ldots, g$ *and* $\delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ *be the auxiliary of* $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \frac{1}{2}\|\boldsymbol{X} - \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\|_F^2$ *given by the Jensen inequality,*

$$\delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) \propto -\log \prod_{i,k,j}^{n,g,d} \mathcal{N}(x_{ij}; w_{jk}, \sigma^2)^{\tilde{z}_{ik}}, \tag{D.1}$$

where $\mathcal{N}(x_{ij}; w_{jk}, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x_{ij} - w_{jk})^2}$ *is the Gaussian probability density function (pdf) and* $\sigma^2 = 1$.

Proof. From

$$\frac{1}{2}\|\boldsymbol{X} - \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top\|_F^2 = \sum_{i,j}^{n,d} x_{ij}^2 - 2x_{ij} \sum_k^g \tilde{z}_{ik} w_{jk} + \Big(\sum_k^g \tilde{z}_{ik} w_{jk} \Big)^2, \tag{D.2}$$

we use the convexity of the power function and define $\delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ as follows :

$$\begin{aligned} \delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) &= \sum_{i,j}^{n,d} x_{ij}^2 - 2x_{ij} \sum_k^g \tilde{z}_{ik} w_{jk} + \sum_k^g \tilde{z}_{ik} w_{jk}^2 \\ &= \sum_{i,j,k}^{n,d,g} \tilde{z}_{ik} (x_{ij} - w_{jk})^2. \end{aligned} \tag{D.3}$$

Rewritting (D.3) with the exponential function, we obtain :

$$\delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = -\log \prod_{i,j,k}^{n,d,g} \left[e^{-\frac{1}{2}(x_{ij}-w_{jk})^2} \right]^{\tilde{z}_{ik}} \propto -\log \prod_{i,k,j}^{n,g,d} \mathcal{N}(x_{ij}; w_{jk}, \sigma^2)^{\tilde{z}_{ik}}. \tag{D.4}$$

□

Remark. Miniminzing eq(D.3) w.r.t $\widetilde{\boldsymbol{Z}} \in \{0,1\}$ is equivalent to the K-means algorithm.

The optimization to guaranty Lemma 4.2.1 when $\delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is given by eq(D.3) is available in Appendix D.2.1.

D.1.2 From NMF to Von Mises-Fisher mixtures ($(1-\cos)$ dissimilarity)

Lemma D.1.2. *Let* $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \frac{1}{2}\sum_i^n \|\boldsymbol{x}_i - [\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i\|_F^2$ *be the* $(1-\cos)$ *dissimilarity function for NMF where* $\boldsymbol{x}_i \in \mathbb{S}^{d-1}$, $[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_i \in \mathbb{S}^{d-1}$ *and* $\sum_k^g \tilde{z}_{ik} = 1$, $\forall i = 1, \ldots, n$. *Let* $\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ *be the auxiliary function of* $\mathcal{D}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ *given by the Jensen inequality,*

$$\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) \propto -\log \prod_{i,k}^{n,g} f_p(\boldsymbol{x}_i; \boldsymbol{w}_k, \kappa)^{\tilde{z}_{ik}}, \tag{D.5}$$

where $f_p(\boldsymbol{x}_i; \boldsymbol{w}_k, \kappa) = C_p(\kappa) e^{(-\kappa \boldsymbol{x}_i \boldsymbol{w}_k)}$ *is the von Mises-Fisher pdf,* $C_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2}\mathbf{I}_{p/2-1}(\kappa)}$, $\mathbf{I}_{p/2-1}(\kappa)$ *is the Bessel function of the first kind and* $\kappa = 1$.

Proof. Recalling the convexity of the power function, we define $\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ as follows :

$$\begin{aligned}
\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) =& \frac{1}{2}\sum_i^n \left[\sum_j^d x_{ij}^2 + \sum_{j,k} \tilde{z}_{ik} w_{jk}^2 - 2\sum_{j,k} x_{ij}\tilde{z}_{ik} w_{jk} \right] \\
=& \sum_{i,k}^{n,g} \tilde{z}_{ik}(1 - \sum_j^d x_{ij} w_{jk}) \\
=& -\log \prod_{i,k}^{n,g} \left[e^{-(1-\sum_j^d x_{ij} w_{jk})} \right]^{\tilde{z}_{ik}} \propto -\log \prod_{i,k}^{n,g} f_p(\boldsymbol{x}_i; \boldsymbol{w}_k, \kappa)^{\tilde{z}_{ik}}.
\end{aligned} \tag{D.6}$$

Since $\|\boldsymbol{x}_i\|^2 = \|\boldsymbol{w}_k\|^2 = 1$, the vMF distribution is obtained from the Gaussian distribution after remarginalizing the pdf with $\boldsymbol{X}$ over the unit-sphere. □

The optimization to guaranty Lemma 4.2.1 when $\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is given by eq(D.6) is available in Appendix D.2.2.

Remark. Miniminzing eq(D.6) w.r.t $\widetilde{\boldsymbol{Z}} \in \{0,1\}$ is equivalent to the Spherical K-means algorithm.

D.1.3 Erlang mixture from the Itakura-Saito NMF

The Itakura-Saito divergence with support on $[0,+\infty)$ is defined as follows :

$$D_{IS}(\boldsymbol{X}||\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} \left[\frac{x_{ij}}{[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij}} - \log \frac{x_{ij}}{[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij}} - 1 \right]. \tag{D.7}$$

Proposition D.1.1. Let $\sum_k^g \tilde{z}_{ik} = 1, \forall i = 1,\dots,g$ and $\delta_{IS}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ be the auxiliary of $D_{IS}(\boldsymbol{X}||\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ given by the Jensen inequality,

$$\delta_{IS}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) \propto -\log \prod_{i,k,j}^{n,g,d} E(x_{ij};\alpha,w_{jk})^{\tilde{z}_{ik}}, \tag{D.8}$$

where $E(x_{ij};\alpha,w_{jk}) = \frac{x_{ij}^{\alpha-1} e^{\frac{x_{ij}}{w_{jk}}}}{w_{jk}^{\alpha}(\alpha-1)!}$ is the Erlang pdf with $\alpha = 1$ (note that the Erlang distribution is generalized by the Gamma distribution).

Proof. After rewritting eq(D.7) as follows :

$$D_{IS}(\boldsymbol{X}||\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) = \sum_{i,j}^{n,d} \Big[x_{ij}[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij}^{-1} - \log x_{ij} + \log[\widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top]_{ij} - 1 \Big], \tag{D.9}$$

The convexity of the multiplicative inverse function on $[0,+\infty[$ and the concavity of the logarithm lead to define $\delta_{IS}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ as follows :

$$\begin{aligned}
\delta_{IS}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) &= \sum_{i,j}^{n,d} \Big[x_{ij} \sum_k^g \tilde{z}_{ik} w_{jk}^{-1} - \log x_{ij} + \sum_k^g \tilde{z}_{ik} \log w_{jk} - 1 \Big] \\
&= -\sum_{i,j}^{n,d} \Big[\log x_{ij} + 1 \Big] + \sum_{i,j,k}^{n,d,g} \tilde{z}_{ik} [x_{ij} w_{jk}^{-1} + \log w_{jk}] \\
&= -\sum_{i,j}^{n,d} \Big[\log x_{ij} + 1 \Big] + \sum_{i,j,k}^{n,d,g} \tilde{z}_{ik} \log \left[e^{\frac{x_{ij}}{w_{jk}}} w_{jk} \right] \\
&= -\sum_{i,j}^{n,d} \Big[\log x_{ij} + 1 \Big] - \log \prod_{i,j,k}^{n,d,g} \left[\frac{e^{-\frac{x_{ij}}{w_{jk}}}}{w_{jk}} \right]^{\tilde{z}_{ik}} \\
&\propto -\log \prod_{i,k,j}^{n,g,d} E(x_{ij};\alpha,w_{jk})^{\tilde{z}_{ik}}.
\end{aligned} \tag{D.10}$$

□

As expressed in eq(D.10), the Itakura-Saito translates an underlying Erlang distribution with a normalizing parameter w_{jk} where $\alpha = 1$.

Remarks.

— Let $w_{jk} := \frac{1}{w_{jk}}$ in δ_{IS}. This leads to $\delta_{IS}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) \propto -\log \prod_{i,k,j}^{n,g,d} \mathcal{E}(x_{ij}; w_{jk})^{\tilde{z}_{ik}}$ where $\mathcal{E}(x_{ij}; w_{jk}) = 1 - w_{jk}e^{-w_{jk}x_{ij}}$ is the Exponential density function.

— Minimizing eq(D.10) w.r.t $\tilde{z}_{ik} \in \{0, 1\}$ is equivalent to a K-means with the Erlang log-divergence.

D.2 Optimization of $\texttt{cNMF}_\texttt{H}$ with $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$

D.2.1 Optimization from the Frobenius norm

Reformulating $\texttt{cNMF}_\texttt{H}$ with respect to $\mathcal{D} = \delta_{\mathcal{F}}$ gives the following minimization problem :

$$\min_{\widetilde{\boldsymbol{Z}} \geq 0, \boldsymbol{W} \geq 0, \widetilde{\boldsymbol{Z}}\mathbf{1}_g = \mathbf{1}_n} \{\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) = \delta_F(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top) - H(\widetilde{\boldsymbol{Z}})\}. \tag{D.11}$$

Minimizing $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ w.r.t the constraints formulated in problem(D.11) requires the definition of the Lagrangian function

$$\mathcal{L}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) + \sum_i^n \gamma_i \Big(\sum_k^g \tilde{z}_{ik} - 1 \Big) + \mathrm{Tr}(\boldsymbol{\epsilon}\widetilde{\boldsymbol{Z}}^\top) + \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \tag{D.12}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times g}$, and $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$ are the Lagrange multipliers. Its gradient w.r.t each factor are denoted as follows :

$$\nabla_{\tilde{z}_{ik}} \mathcal{L} = \frac{1}{2} \sum_j^d (x_{ij}^2 + w_{jk}^2) - (\boldsymbol{X}\boldsymbol{W})_{ik} + 1 + \log \tilde{z}_{ik} + \gamma_i + \epsilon_{ik}, \tag{D.13}$$

$$\nabla_{w_{jk}} \mathcal{L} = -(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk} + \sum_i^n \tilde{z}_{ik} + \beta_{jk}. \tag{D.14}$$

Setting these gradients to zero and making use of the Karush-Kuhn-Tucker conditions $\epsilon \odot \widetilde{\boldsymbol{Z}} = 0$, $\beta \odot \boldsymbol{W} = 0$ lead to the following stationary equations :

$$\tilde{z}_{ik}(\boldsymbol{X}\boldsymbol{W})_{ik} - \tilde{z}_{ik}\left(\frac{1}{2} \sum_j^d (x_{ij}^2 + w_{jk}^2) + 1 + \log \tilde{z}_{ik} + \gamma_i \right) = 0, \tag{D.15}$$

$$w_{jk}(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk} - w_{jk}^2 \sum_i^n \tilde{z}_{ik} = 0. \tag{D.16}$$

Canceling out ϵ and using the gradient $\nabla_{\tilde{z}_{ik}} H$ to ensure the positivity of $\tilde{z}_{ik}$, we obtain the following update rules forming an Expectation-Minimization procedure which is equivalent to an Expectation-Maximization (EM) algorithm derived from the negative fuzzy criterion (4.78) :

$$\tilde{z}_{ik} \leftarrow \frac{e^{(\boldsymbol{XW})_{ik} - \frac{1}{2}\sum_j^d (x_{ij}^2 + w_{jk}^2)}}{e^{1+\gamma_i}}, \qquad \text{(D.17)} \qquad\qquad w_{jk} \leftarrow \frac{(\boldsymbol{X}^\top \widetilde{\boldsymbol{Z}})_{jk}}{\sum_i^n \tilde{z}_{ik}}, \qquad \text{(D.18)}$$

where $e^{1+\gamma_i} = \sum_k^g e^{(\boldsymbol{XW})_{ik} - \frac{1}{2}\sum_j^d (x_{ij}^2 + w_{jk}^2)}$ and the conditional probabilities $\tilde{z}_{ik}$ are missing the normalization constant $\frac{1}{\sqrt{2\pi}}$.

D.2.2 Optimization from the $(1 - \cos)$ dissimilarity

Minimizing $\mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W})$ w.r.t the constraints formulated in problem(D.11) (where $\delta_{CD}(\boldsymbol{X}, \widetilde{\boldsymbol{Z}}\boldsymbol{W}^\top)$ is now given by equation (D.6)) requires the definition of the Lagrangian function

$$\mathcal{L}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\epsilon}, \boldsymbol{\beta}) = \mathcal{Q}(\widetilde{\boldsymbol{Z}}, \boldsymbol{W}) + \sum_i^n \alpha_i \Big(\sum_k^g \tilde{z}_{ik} - 1 \Big) + \sum_k^g \gamma_k (\|\boldsymbol{w}_k\| - 1) + \mathrm{Tr}(\boldsymbol{\epsilon}\widetilde{\boldsymbol{Z}}^\top) + \mathrm{Tr}(\boldsymbol{\beta}\boldsymbol{W}^\top), \quad \text{(D.19)}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{\gamma} \in \mathbb{R}^g$, $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times g}$, and $\boldsymbol{\beta} \in \mathbb{R}^{d \times g}$ are the Lagrange multipliers. $\nabla_{w_{jk}} \mathcal{L}$ obtained from eq(D.19) is similar to the gradient of the Lagrangian of Spherical NMF given by eq(4.120). Therefore we obtain the same expression for w_{jk} which is given by eq(4.126). Consequently, the gradient of eq(D.19) w.r.t $\tilde{z}_{ik}$ is :

$$\nabla_{\tilde{z}_{ik}} \mathcal{L} = -(\boldsymbol{XW})_{ik} + 1 + \log \tilde{z}_{ik} + \alpha_i + \epsilon_{ik}. \qquad \text{(D.20)}$$

Setting this gradient to zero and making use of the Karush-Kuhn-Tucker conditions $\epsilon \odot \widetilde{\boldsymbol{Z}} = 0$ gives

$$\tilde{z}_{ik}(\boldsymbol{XW})_{ik} - \tilde{z}_{ik}(1 + \log \tilde{z}_{ik} + \gamma_i) = 0. \qquad \text{(D.21)}$$

Canceling out ϵ and using $\nabla_{\tilde{z}_{ik}} H$ to ensure the positivity of $\tilde{z}_{ik}$, we obtain the following update rules forming an Expectation-Minimization procedure which is equivalent to an Expectation-Maximization (EM) algorithm derived from the negative fuzzy criterion (4.78) :

$$\tilde{z}_{ik} \leftarrow \frac{e^{(\boldsymbol{XW})_{ik}}}{e^{1+\gamma_i}}. \qquad \text{(D.22)}$$

Replacing $\tilde{z}_{ik}$ with eq(D.22) into the constraint gives $e^{1+\gamma_i} = \sum_k^g e^{(\boldsymbol{XW})_{ik}}$.

le cnam

Mickaël FEBRISSY

Modèles probabilistes et factorisation matricielle non-négative : cadre unifié pour les données textuelles

HESAM UNIVERSITÉ

Résumé : Depuis l'avenènement du Big data, les techniques de réduction de la dimension sont devenues essentielles pour l'exploration et l'analyse de données hautement dimensionnelles issues de nombreux domaines scientifiques. En créant un espace à faible dimension intrinsèque à l'espace de données original, ces techniques offrent une meilleure compréhension dans de nombreuses applications de la science des données. Dans le contexte de l'analyse de textes où les données recueillies sont principalement non négatives, les techniques couramment utilisées produisant des transformations dans l'espace des nombres réels (par exemple, l'analyse en composantes principales, l'analyse sémantique latente) sont devenues moins intuitives car elles ne pouvaient pas fournir une interprétation directe. De telles applications montrent la nécessité de techniques de réduction de la dimensionnalité comme la factorisation matricielle non négative (NMF), utile pour intégrer par exemple, des documents ou des mots dans l'espace de dimension réduite. Par définition, la NMF vise à approximer une matrice non négative par le produit de deux matrices non négatives de plus faible dimension, ce qui aboutit à la résolution d'un problème d'optimisation non linéaire. Notons cependant que cet objectif peut être exploité dans le domaine du regroupement de documents/mots, même si ce n'est pas l'objectif de la NMF. En s'appuyant sur la NMF, cette thèse se concentre sur l'amélioration de la qualité du clustering de grandes données textuelles se présentant sous la forme de matrices document-terme très creuses. Cet objectif est d'abord atteint en proposant plusieurs types de régularisations de la fonction objectif originale de la NMF. En plaçant cet objectif dans un contexte probabiliste, un nouveau modèle NMF est introduit, apportant des bases théoriques pour établir la connexion entre la NMF et les modèles de mélange finis de familles exponentielles, ce qui permet d'offrir des régularisations intéressantes. Cela permet d'inscrire, entre autres, la NMF dans un véritable esprit de clustering. Enfin, un modèle bayésien de blocs latents de Poisson est proposé pour améliorer le regroupement de documents et de mots simultanément en capturant des caractéristiques de termes bruyants. Ce modèle peut être relié à la NMTF (Nonnegative Matrix Tri-Factorization) consacrée au co-clustering. Des expériences sur des jeux de données réelles ont été menées pour soutenir les propositions de la thèse.

Mots clés : classification croisée, factorisation, modèles des blocs latents, modèles de mélanges, text mining.

Abstract : Since the exponential growth of available Data (Big data), dimensional reduction techniques became essential for the exploration and analysis of high-dimensional data arising from many scientific areas. By creating a low-dimensional space intrinsic to the original data space, theses techniques offer better understandings across many data Science applications. In the context of text analysis where the data gathered are mainly nonnegative, recognized techniques producing transformations in the space of real numbers (e.g. *Principal component analysis*, *Latent semantic analysis*) became less intuitive as they could not provide a straightforward interpretation. Such applications show the need of dimensional reduction techniques like Nonnegative Matrix factorization (NMF) useful to embed, for instance, documents or words in the space of reduced dimension. By definition, NMF aims at approximating a nonnegative matrix by the product of two lower dimensional nonnegative matrices, which results in the solving of a nonlinear optimization problem. Note however that this objective can be harnessed to document/word clustering domain even it is not the objective of NMF. In relying on NMF, this thesis focuses on improving clustering of large text data arising in the form of highly sparse document-term matrices. This objective is first achieved, by proposing several types of regularizations of the original NMF objective function. Setting this objective in a probabilistic context, a new NMF model is introduced bringing theoretical foundations for establishing the connection between NMF and Finite Mixture Models of exponential families leading, therefore, to offer interesting regularizations. This allows to set NMF in a real clustering spirit. Finally, a Bayesian Poisson Latent Block model is proposed to improve document and word clustering simultaneously by capturing noisy term features. This can be connected to NMTF (Nonnegative Matrix factorization Tri-factorization) devoted to co-clustering. Experiments on real datasets have been carried out to support the proposals of the thesis.

Keywords : co-clustering, factorization, latent block models, mixture models, text mining.